

Hackerangriffe abwehren

So erhöhen Sie die Sicherheit Ihrer Applikationen ab S. 16



INFO-
Programm
gemäß
§ 14
JuSchG

Ausgabe 3/16

Deutschland
14,95 EUR

CH: 29,90 CHF
A, B, NL, L:
16,45 EUR

Auf der Heft-CD

**Eine Sammlung interessanter JavaScript-Frameworks
sowie eine Auswahl leistungsfähiger Tools für Entwickler S. 50**





Upgrades für Ihr Entwickler-Know-How

Ab sofort in unseren Shops erhältlich!



<https://shop.dotnetpro.de>
<https://shop.webundmobile.de>



Sicherheit 2016

Die Sicherheit von Webanwendungen und mobilen Applikationen wird auch 2016 Entwickler wieder herausfordern.

Web- und Mobile-Entwickler müssen im Hinblick auf die Sicherheit immer auf der Hut sein. Während Großunternehmen mit milliardenschweren IT-Budgets sich auch von einer gravierenden IT-Sicherheitslücke leicht erholen können, kann ein vergleichbarer Vorfall einem kleinen Entwicklerteam oder einer kleinen Agentur einen irreparablen Schaden zufügen. Im Zeitalter täglicher Sicherheitspannen am Anbruch der IoT-Ära sind sorgsam orchestrierte Hackerangriffe zur traurigen Normalität geworden. In einem umfangreichen Special zeigen Filipe Pereira Martins und Anna Kobylinska ab Seite 16, welche Herausforderungen auf Entwickler zukommen.

»Sorgsam orchestrierte Hackerangriffe sind zur traurigen Normalität geworden.«

Foundation – von seinen Entwicklern als fortschrittlichstes Frontend-Framework bezeichnet – ist in Version 6 grundlegend überarbeitet worden und bringt vielversprechende Neuerungen mit. Welche dies sind und wie sie beim Webdesign am besten eingesetzt werden, das zeigt Dr. Florence Maurice in ihrem Artikel ab Seite 52.

Bisher war Apple TV für App-Entwickler eher uninteressant. Nur wenigen Entwicklern war es bisher erlaubt, Apps für Apple TV zu entwickeln und auf dem Gerät zu publizieren. Mit der vierten Generation wird zum ersten Mal ein App Store auf dieser Plattform zur Verfügung gestellt, in dem analog zu iPad, iPhone, iPod und Apple Watch Apps von jedermann zum Download angeboten werden können. Eine Einführung für interessierte Entwickler liefert der Artikel von Christian Bleske ab Seite 66.

Ihr Max Bold
chefredakteur@maxbold.de



Dr. Florence Maurice

stellt die neuen Features von Foundation 6 vor (S. 52)



Christian Bleske

gibt eine Einführung in die Programmierung unter tvOS (S. 66)



Markus Schraudolph

präsentiert das Amazon Product Advertising API (S. 118)



INHALT

Aktuell

Aquaris X5 Cyanogen

Verbesserte Datenschutzoptionen

6

Feature

Sicherheit 2016

Die Sicherheit von Webanwendungen und mobilen Apps ist auch in diesem Jahr ein großes Thema

16

Bedrohungsszenarien und ihre Bewältigung

Mangelndes Sicherheitsbewusstsein als Problemquelle

22

HTML, CSS & JavaScript

Formulargenerierung mit JavaScript

Komfortable Formulargenerierung mit Alpaca Forms

34

Ext JS Version 6

Mit der Ext-JS-Bibliothek durchgängig für Desktops, Tablets und Smartphones Anwendungen entwickeln

38

Foundation 6

Foundation 6 mit nützlichen Tools an Bord

52

Das Navigation Timing API

Per JavaScript verschiedene Aspekte bezüglich der Performance einer Webseite messen

58

Formularvalidierung mit Angular 2

Das komfortable Handling von Formularen ist eine der wichtigsten Aufgaben moderner Webframeworks

62

Mobile Development

Apps für tvOS/iOS

Die neue Generation des Apple TV hat ein neues Betriebssystem und einen eigenen App Store bekommen

66

Must-have Libraries für Android-Entwickler

Mit einigen wenigen Open-Source-Libraries lassen sich schnell Android-Apps mit einem guten Nutzererlebnis entwickeln

72

Material Design und Android

Material Design auf Android 4.x lauffähig machen

80

Diagramme für Android

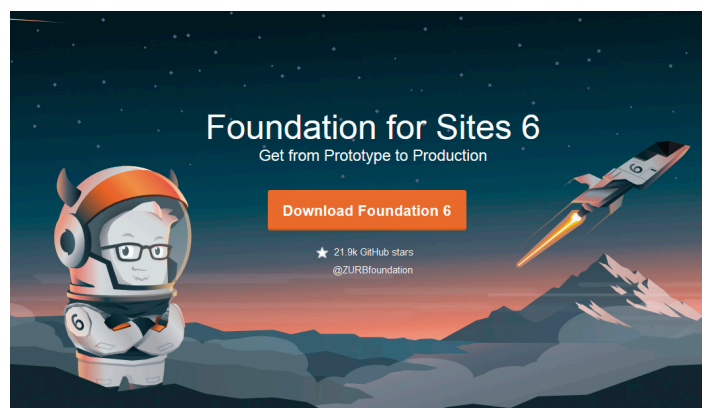
Wer Nutzern große Datenmengen präsentieren möchte, braucht eine Diagrammbibliothek

84



Neue Angriffsvektoren bedrohen die Sicherheit von Webanwendungen, Backend-Systemen und mobilen Apps **S. 16**

Foto: Mathias Vietmeier



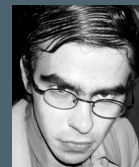
Foundation 6 bietet Komponenten wie Off-Canvas-Funktionalität, Tooltips, Slider und Flex Video **S. 52**

Experten in dieser Ausgabe



Philip Ackermann zeigt, wie man mit dem Framework Alpaca Forms einfach Webformulare generiert.

34



Tam Hanna erläutert, wie man mit passenden Tools Diagramme in Android-Applikationen einbaut.

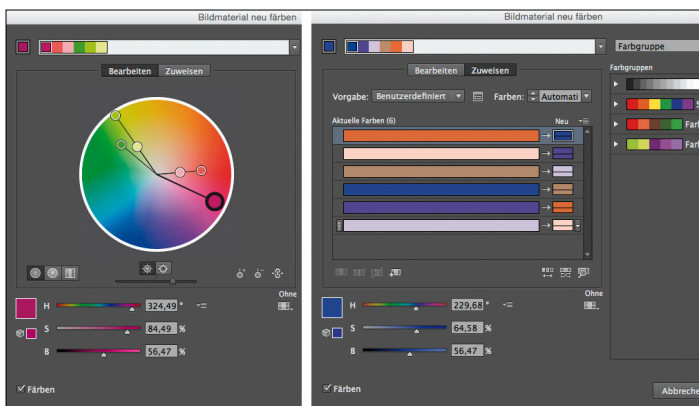
84



Mit Open-Source-Libraries lassen sich Bedienoberflächen ansprechend gestalten **S. 72**

Der weltgrößte Online-Händler

Amazon betreibt eine riesige Produktdatenbank, die sich via Amazon Product Advertising API mit PHP nutzen lässt **S. 118**



Die Creative Cloud und deren Bibliotheken erleichtern eine programmübergreifende Arbeit **S. 124**

Jetzt abonnieren

Sichern Sie sich jetzt die **web & mobile developer** im Jahresabo und profitieren Sie von exklusiven Services und Angeboten für Abonnenten.
<https://shop.webundmobile.de>

Domain Specific Language für C++ / Qt (Teil 4)

Open-Source-DSL (Domain-Specific Language) zur Erstellung von mobilen Business-Anwendungen **94**

Framework: The-M-Project

Framework vereinfacht die Cross-Plattform-Entwicklung **98**

Backend

Amazon Web Services mit PHP

Mit dem AWS PHP SDK kann man PHP-Anwendungen in AWS-Services einfacher integrieren **100**

Entwicklung von Modulen für Magento 2

Service Contracts regeln die Kommunikation **108**

Collaboration braucht Vertrauen in die Teams

Optimale Zusammenarbeit im Team **110**

Amazon Product Advertising API mit PHP nutzen

Wenn Sie Ihre Website durch Werbung monetarisieren, dann ist das Amazon-Partnerprogramm ein möglicher Geldbringer **118**

Beyond Dev

Grafik für Entwickler: Photoshop, Illustrator und InDesign

Photoshop, Illustrator und InDesign ergänzen sich optimal **124**

Search Engine Blocking

Techniken, mit denen Webentwickler Websites vor neugierigen Suchmaschinen schützen können **130**

Accelerated Mobile Pages

Das Open-Source-Artikelformat Accelerated Mobile Pages ist Googles Antwort auf Instant Articles von Facebook **134**

Standards

Editorial **3**

Heft-CD **50**

Impressum **129**

Online-Recht **140**

Arbeitsmarkt **142**

Dienstleisterverzeichnis **145**

Vorschau **146**

NEWS & TRENDS

AKTUELLE NEWS AUS DER ENTWICKLERSZENE

News

Kaspersky-Studie

Gehen vertrauliche Daten verloren, sind in 42 Prozent der Fälle Mitarbeiter dafür verantwortlich. Der durchschnittliche Schaden be-



Holger Suhl, General Manager DACH bei Kaspersky Lab

trägt bis zu 80.000 US-Dollar. Das ist das Ergebnis einer Studie von Kaspersky Lab. Daneben offenbart die Studie, dass 21 Prozent der Unternehmen, die von einem internen Sicherheitsvorfall betroffen waren, auch den Verlust von Daten zu beklagen hatten.

»Sicherheitslösungen allein können keinen hundertprozentigen Schutz für Unternehmensdaten liefern«, sagt Holger Suhl, General Manager DACH bei Kaspersky Lab. »Gefragt sind mehrschichtige Cybersicherheitsansätze, unterstützt von Security Intelligence Services und spezifischen Lösungen in Kombination mit Sicherheitsrichtlinien und -prozessen.«

www.kaspersky.com/de

Aquaris X5 Cyanogen

Verbesserte Datenschutzoptionen

BQ hat das neue Smartphone Aquaris X5 Cyanogen Edition vorgestellt. Das fünf Zoll große Display des Aquaris X5 Cyanogen Edition mit Quantum-Color+-Technologie ist umrahmt von einem leichten und dünnen Aluminium-Gehäuse in Metallic-Optik.

Das LTE-Smartphone arbeitet mit einem Qualcomm-SnapdragonTM 412-Prozessor, der von 2 GByte RAM unterstützt wird. Der 16 GByte große interne Speicher kann durch eine microSD-Karte erweitert werden. Die eigens von BQ entwickelte Kamera-App ermöglicht es, mit nur einem Daumen auf alle Einstellungen der Kamera zuzugreifen.

Mit der 13-Megapixel-Rückkamera können Videos sowohl in Slow Motion als auch in Time Lapse aufgenommen werden. Die 5-Megapixel-Frontkamera ist mit einem Selfieblitz ausgestattet. Mit beiden können Videos in Full HD aufgenommen werden. Ein 2900-mAh-Akku sorgt für die Stromversorgung.

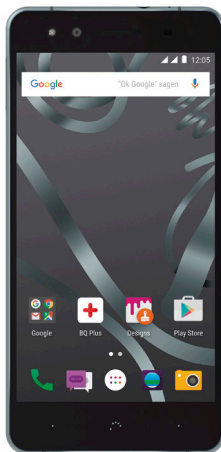
Das Cyanogen OS erlaubt Nutzern eine Vielzahl von Personalisierungen: Hintergründe, Design und Boot-Animationen können individuell gestaltet und Kurzwahlta-

ten den eigenen Wünschen und Bedürfnissen angepasst werden. Eine weitere Verbesserung der Nutzererfahrung zeigt sich in der Truecaller-App – damit können Nutzer Spam-Anrufe frühzeitig identifizieren und blockieren. Die Boxer-Anwendung vereint zudem Termine, Events und E-Mails an einem Ort, sodass das Zusammentragen von Daten und Kontakten überflüssig wird. Auch der Sound kann individuell angepasst werden: So erlaubt die Audio-FX-Funktion, zwischen voreingestellten Profilen der einzelnen Musikrichtungen auszuwählen.

Das Aquaris X5 Cyanogen Edition ermöglicht den Nutzern umfangreiche Datenschutzoptionen. Mit dem Privacy Guard haben sie die Kontrolle über Daten, die mit Drittanbietern geteilt werden. Nutzer sichern ihre Geräte zwar durch eine PIN-Eingabe vor unbefugtem Zugriff, dennoch kann die PIN-Kombination leicht ausspioniert werden.

Mit der PIN Scramble-Funktion werden die bisher feststehenden Zifferntasten zur Eingabe bei jedem Einschalten neu durcheinandergewirbelt. Besonders sensible Daten und Apps können im passwortgeschützten Ordner *Protected Apps* abgelegt werden.

www.bq.com/de/cyanogen



Das Aquaris X5 Cyanogen bietet den Nutzern verbesserte Datenschutzoptionen

Pirobase CMS 9.2

Add-ons als nahtlose Erweiterungen

Von dem Content-Management-System Pirobase gibt es eine neue Version 9.2. Für das neue Major Release wurden zahlreiche Add-ons entwickelt, die das Content Management in



Pirobase bietet eine Kombination von Content Management (CMS) und Produktdatenmanagement (PIM)

Pirobase um nützliche Funktionen erweitern. Die Add-ons lassen sich einfach installieren und an individuelle Bedürfnisse anpassen.

Mit dem Add-on Social Media Posting lassen sich beispielsweise Facebook und Twitter einfach bespielen. Autoren sind so nicht nur näher am Content, sondern nutzen ein zentrales, einheitliches Tool und einen Account für all ihre Ausgabe-Kanäle. Dieses Add-on wird kostenlos zur Verfügung gestellt.

Um den Erfolg der Social Media-Kanäle contentbezogen in Pirobase auszuwerten, erweitert Social Media Analytics Pirobase zusätzlich um alle wichtigen Kennzahlen wie Posts, Kanäle und Statistiken zentral in der Redaktionsoberfläche.

www.pirobase.de/content-management

Zahl des Monats

Durch die Digitalisierung sollen in den nächsten fünf Jahren **5.000.000** Jobs wegfallen. Das besagt eine noch unveröffentlichte Untersuchung des Weltwirtschaftsforums in Davos. Grundlage der Studie ist eine Umfrage unter Top-Managern der 350 größten Konzerne der Welt.

Quelle: Weltwirtschaftsforum

Intel

Bug in Skylake-Prozessoren gefunden

Friert der Rechner während seiner Arbeit komplett ein, muss nicht unbedingt die Software schuld sein, es könnte auch am Prozessor liegen.

Intel hat nach einem Bericht von Ars Technica inzwischen eingeräumt, dass seine Skylake-Prozessoren einen Bug aufweisen, der dazu führen kann, dass Rechner während sehr komplexer Rechenoperationen einfrieren. Entdeckt wurde der Fehler bei der Suche nach Primzahlen mit Hilfe des Programms Prime95.

Dieses nutzt die Zeit, in der der Prozessor eines Rechners nicht von seinem Eigentümer



Intels Skylake-Prozessoren weisen einen Bug auf

gebraucht wird, um nach bislang unbekannten Primzahlen zu suchen. Dabei kam es vor, dass sich der Rechner aufhängte oder unvorhersagbares Systemverhalten aufwies. Herausgefunden wurde, dass der Fehler auftrat, wenn extrem große Zahlen multipliziert wurden. Der Fehler kommt möglicherweise in allen Intel-Prozessoren der sechsten Generation vor, tritt aber nur unter den genannten außergewöhnlichen Bedingungen auf.

www.intel.com

DevOps

Reality Check in deutschen Unternehmen

Positive Erfahrungsberichte aus der Praxis machen den Einsatz von DevOps für immer mehr Unternehmen in Deutschland interessant.

Eine aktuelle Studie des Marktanalyse- und Beratungsunternehmens PAC, in der die Zusammenarbeit zwischen IT-Betrieb und Entwicklungsabteilung untersucht wurde, fand heraus, dass 43 Prozent der Befragten häufig nicht klar ist, wer verantwortlich ist. Für 39 Prozent dauert es zu lange, bis das Zielsystem für eine neue Entwicklung verfügbar und konfiguriert ist, und 29 Prozent gaben an, dass die Testumgebung oft nicht dem späteren Zielsystem entspricht.

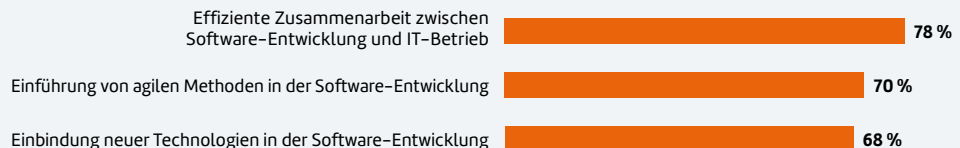
Damit am Ende das gewünschte Ergebnis in kürzerer Zeit herauskommt, ist von Prozessbeginn an eine bessere Abstimmung zwischen Entwicklung und IT-Betrieb unbedingt erforderlich.

Ein Ansatz, der genau dieses Ziel verfolgt, nennt sich DevOps, ein integriertes Konzept für Entwicklung und Betrieb von Software (Development & Operations). Für 70 Prozent der Befragten in der PAC-Studie liefert DevOps die richtige Antwort auf die Schnelllebigkeit in der IT.

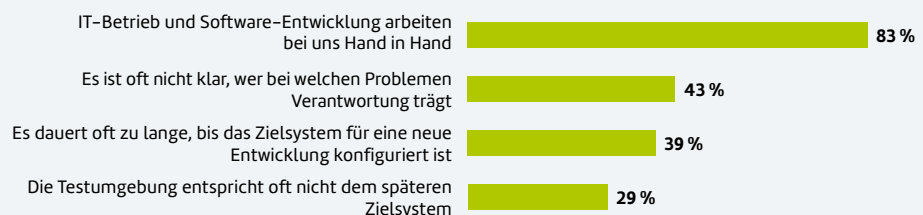
DevOps ist derzeit bei 8 Prozent der befragten Unternehmen bereits fester Bestandteil der Entwicklungsprojekte und kommt bei 24 Prozent der Unternehmen in einzelnen Projekten zum Einsatz. In 9 Prozent der Unternehmen befindet sich DevOps in der Pilotphase. 37 Prozent der Befragten gaben an, dass der Ansatz voraussichtlich in einigen Jahren bei ihnen zum Einsatz kommt, und bei 23 Prozent befindet er sich bereits in der Umsetzungsdiskussion.

www.actifio.com/de

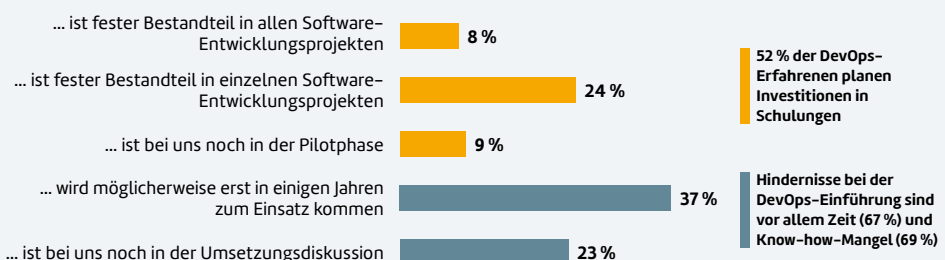
DevOps: Zusammenarbeit klappt meistens gut, ist aber nicht frei von Spannungen und Problemen



DevOps wird bei einigen IT-Herausforderungen als Problemlöser gesehen



DevOps steht in den meisten Unternehmen noch am Anfang



web & mobile developer 3/2016

Quelle: PAC

News

**SoftMaker Office 2016:
Schnelles Office-Paket
für Linux**

Nach einem öffentlichen Beta-Test mit rund zehntausend Anwendern steht das neue SoftMaker Office 2016 für Linux jetzt offiziell zur Verfügung.

Es bietet mehr als 400 neue Funktionen und nochmals gesteigerte Performance: Dank optimierter Software-Architektur, OpenGL-Beschleunigung für die Bildschirmdarstellung und Einsatz mehrerer CPU-Kerne für Grafikfunktionen ist es das mit Abstand schnellste Office-Paket für Linux.

Durch den integrierten Duden-Korrektor eignet sich SoftMaker Office perfekt für alle Anwender, die auf orthografisch und grammatikalisch korrekte Schriftstücke großen Wert legen.

SoftMaker Office 2016 für Linux ist in einer Standard- und einer Professio-

**SoftMaker Office 2016
für Linux** gibt es in zwei Versionen

nal-Version erhältlich. Beide Produktvarianten enthalten die Textverarbeitung TextMaker, die Tabellenkalkulation PlanMaker und das Präsentationsprogramm Presentations.

www.softmaker.de

Internet of Things (IoT)**IBM macht München
zur Watson-IoT-
Hauptstadt**

München wird zur weltweiten Zentrale des neuen IBM Geschäftsbereichs Watson IoT und gleichzeitig der neue Standort für das erste europäische Watson Innovation Center. Rund 1000 IBM-Entwickler, -Berater, -Forscher und -Designer sollen



In den Münchnern HighLight Towers eröffnet IBM die Zentrale des Geschäftsbereichs Watson IoT

zukünftig in den Münchner HighLight Towers tätig sein. Gemeinsam mit Kunden und Partnern arbeiten sie dort an einer neuen Generation vernetzter Lösungen an der Schnittstelle von Cognitive Computing und IoT (Internet of Things). Ziel ist es, über kognitive Fähigkeiten vernetzte Geräte, Systeme und Sensoren intelligent zu machen sowie neue Marktchancen zu erschließen. Die neuen Lösungen werden über die IBM Watson IoT Cloud als globale Plattform verfügbar sein. Das neue Watson-IoT-Hauptquartier repräsentiert die größte Investition der IBM in Europa seit mehr als 20 Jahren.

»Die Ausgangslage der deutschen Wirtschaft in Sachen Digitalisierung und Industrie 4.0 ist hervorragend – ihr großer Vorteil ist der hohe Industrialisierungsgrad«, sagt Martina Koederitz, Vorsitzende der Geschäftsführung der IBM Deutschland und General Ma-

nager Österreich und Schweiz. »Er bildet einen idealen Resonanzboden für datengetriebene, kognitive Geschäftsmodelle. Mit den nun leicht zugänglichen neuen Lösungen werden die Grenzen des Machbaren nochmals weiter verschoben.«

Im Zuge der Eröffnung des Watson IoT Centers stellt IBM auch vier neue kognitive Programmierschnittstellen (APIs) vor. Mit diesen Schnittstellen kann das System die Daten aus unterschiedlichsten Quellen, Systemen, Sensoren oder smarten Produkten noch besser verstehen und bewerten. Die neuen Services sind über die cloudbasierte IBM Watson IoT-Plattform verfügbar.

www.ibm.com/iot

**SanDisk Connect Wireless
Stick****200 GByte Speicher
für Smartphones und
Tablets**

Mit dem SanDisk Connect Wireless Stick können Anwender ihre mobilen Endgeräte um weitere 200 GByte Speicher erweitern.

Die SanDisk Corporation hat ihr Portfolio mobiler Speicherslösungen um den SanDisk Connect Wireless Stick mit einer Speicherkapazität von 200 GByte erweitert.

Der SanDisk Connect Wireless Stick beinhaltet die App SanDisk Connect, die eine mühelose Verwaltung von Inhalten zwischen Geräten ermöglicht, einschließlich iPhone, iPad, iPod, Android-Geräten, Fire-Tablets, Mac und PCs – und dies über eine passwortgeschützte WLAN-Verbindung. Die App wurde zudem vor Kurzem aktualisiert, um das Teilen von In-

**Wireless Stick:
Mini-NAS mit 200
GByte Speicher**

halten auf großen Bildschirmen mit AirPlay auf Apple TV, Google Chromecast und Amazon Fire TV zu ermöglichen. Der SanDisk Connect Wireless Stick mit 200 GByte ist ab sofort zu einem Preis von 119,99 Euro erhältlich.

www.sandisk.com

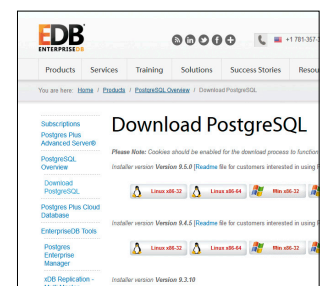
PostgreSQL 9.5**Neue Version soll 96
Prozent schneller sein**

EnterpriseDB (EDB) hat die Version 9.5 der Datenbank PostgreSQL angekündigt. Sie soll schneller sein und eine verbesserte Datenbankintegration, Skalierbarkeit und Produktivität bei Datenanalytik anbieten.

Die neue Version 9.5 steigert die Leistung und Skalierbarkeit, verbessert die Produktivität bei der Datenanalytik sowie die Integration mit anderen Datenbanklösungen und schafft so die Grundlage zur Unterstützung horizontaler Skalierung über mehrere Server hinweg.

EDBs Benchmark-Tests zeigen, dass PostgreSQL 9.5 über beachtliche Leistungsverbesserungen für Arbeitslasten verfügt, die im hohen Maße gleichzeitig ausgeführt werden.

www.enterprisedb.com



Die neue Version 9.5 von PostgreSQL bietet mehr Leistung

Umfrage

Rückläufige Investitionsbereitschaft

Laut einer Umfrage von Accenture wollen die Verbraucher im kommenden Jahr weltweit weniger in neue Technik investieren als im Vorjahr.

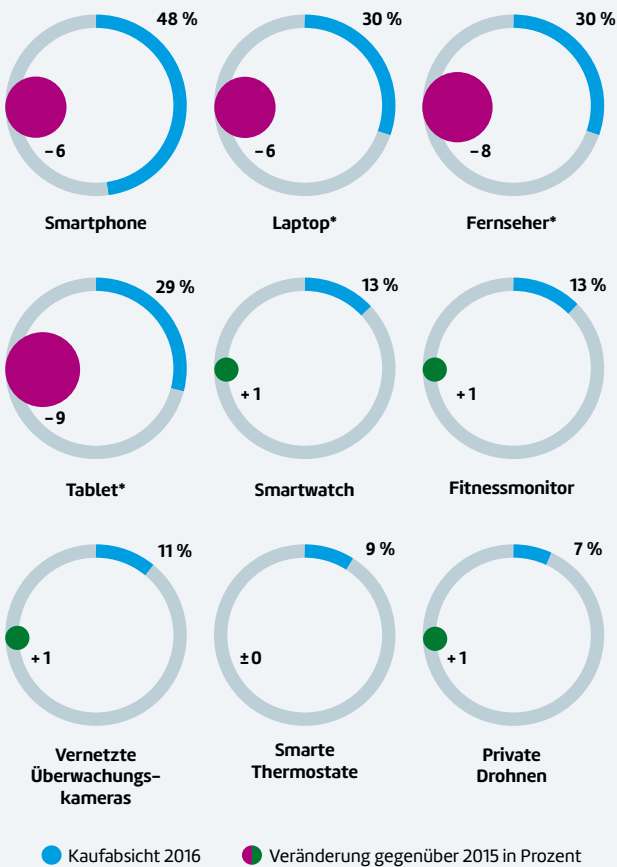
2016 könnte ein schwieriges Jahr für die Tech-Branche werden, zumindest wenn es nach einer aktuellen Umfrage der Consultants von Accenture geht. Demnach sind die Verbraucher weltweit deutlich weniger geneigt, in neue Technik zu investieren, als im Vorjahr.

Gerade bei den Dauerbrennern der vergangenen Jahre – zum Beispiel Smartphones oder Flachbildfernseher – sind die Kaufabsichten rückläufig. Zu den Ursachen zählen laut Ansicht der Analysten Marktsättigung und ein niedrigeres Innovations-tempo. Hinzu kommt, dass die sogenannten »the next big thing«-Technologien die Lücke noch nicht schließen können: Derzeit planen weniger als 15 Prozent der Befragten die Anschaffung von Geräten wie smarten Thermostaten oder Fitnessstrackern.

www.meltwater.com/de

Ein schwieriges Jahr für die Tech-Branche?

Anteil der Konsumenten, die 2016 den Kauf folgender Geräte planen.



Umfrage: Verbraucher wollen 2016 weniger investieren.

Basis: n = 28.000 Konsumenten aus 28 Ländern

web & mobile developer 3/2016

Quelle: Accenture

Top-Trainings für Entwickler

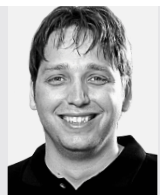
Cross-Plattform-Apps mit C# und Xamarin

Trainer: Sebastian Seidel
07.-09.03.2016, München



Webanwendungen mit HTML, CSS und JavaScript

Trainer: David Tielke
14.-15.03.2016, München



Webentwicklung mit ASP.NET, MVC & WebAPI

Trainer: David Tielke
16.-18.03.2016, München



Entwicklung von Universal Windows Apps

Trainer: Lars Heinrich
21.-22.03.2016, Köln



Angular 2 mit TypeScript

Trainer: Johannes Hoppe, Gregor Woiwode
20.-22.04.2016, München



Hybrid-Apps mit Ionic, Cordova und AngularJS

Trainer: Hendrik Lösch
25.-26.04.2016, Köln



JavaScript für C#-Entwickler

Trainer: Golo Roden
3 Tage, Termin und Ort n. V.



Weitere Informationen und Anmeldung unter
www.developer-media.de/trainings

State-of-the-Internet-Bericht

Europa ist bei IPv6 Spitze

Akamai hat seinen State-of-the-Internet-Bericht für das dritte Quartal 2015 veröffentlicht. Der aktuelle Report enthält unter anderem Daten zu Verbindungsgeschwindigkeiten, Breitbandverbindungen, schwerwiegenden Internetstörungen, zum Status von IPv4-Verbindungen und der Einführung von IPv6. Über die Akamai Intelligent Platform werden Daten gesammelt, die Einblick in relevante Statistiken zur weltweiten Internetnutzung geben. Daraus entsteht in jedem Quartal der State-of-the-Internet-Bericht.

»Während die Zahl der eindeutigen IPv4-Adressen, die sich mit der Akamai-Plattform verbinden, anstieg, erreichte der verfügbare IPv4-Adressraum des nordamerikanischen Regional Internet Registry (RIR) seine Kapazitätsgrenzen«, sagt David Belson, Autor des State of the Internet Report. »Der anhaltende Rückgang der Zahl verfügbarer IPv4-Adressen, sowohl in Nordamerika als auch im Rest der Welt, sollte für Unternehmen Anlass sein, die IPv6-Einführung zu beschleunigen, insbesondere, da die Kosten zur Erlangung von IPv4-Adressen bei zunehmender Verknappung steigen.«

Die weltweiten Top Ten bei Anfragen an die Akamai-Plattform über IPv6 wurden auch im dritten Quartal 2015 wieder von europäischen Ländern dominiert. Sie belegten mit der größten prozentualen Zunahme bei den Anfragen acht von zehn Plätzen in der Top-Ten-Liste.

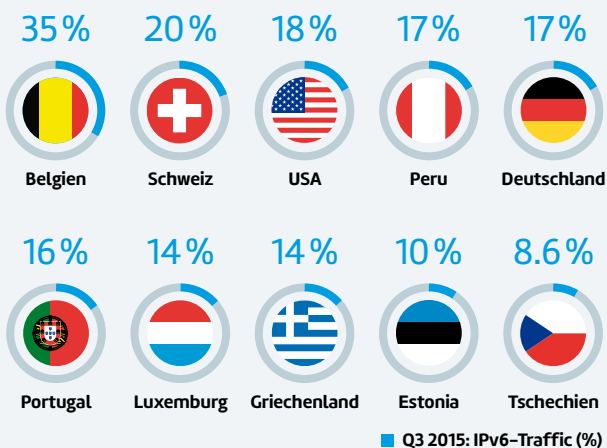
Trotz eines Rückgangs um 8,4 Prozent blieb Belgien Spitzenreiter; 35 Prozent der Anfragen erfolgten über IPv6.

Spitzenreiter unter den Netzbetreibern mit den höchsten Anteilen an Anfragen an die Akamai-Plattform über IPv6 waren Verizon Wireless (72 Prozent) und Telenet (53 Prozent) aus Belgien; jeweils mehr als die Hälfte der Anfragen an Akamai kamen über IPv6.

Bei neun der Top-20-Provider – im zweiten Quartal waren es elf – kam zumindest jede vierte Anfrage an Akamai via IPv6. Mindestens 10 Prozent der Anfragen aller 20 Provider kamen über IPv6; im zweiten Quartal 2015 war dies nur bei 17 Providern der Fall.

www.stateoftheinternet.com/soti-report

In Belgien erfolgen 35 Prozent der Anfragen über IPv6



Quelle: Akamai

All-IP-Leitfaden

Vorteile von All-IP für den Mittelstand

In dem jetzt zum Download zur Verfügung stehenden E-Book »Vorteile von All-IP für den Mittelstand« erläutert der Dortmunder Unified-Communications-Anbieter Swyx anhand anschaulicher Praxistipps,



Das E-Book beleuchtet Vorteile und Herausforderungen der anstehenden Abstellung von ISDN

welche Chancen der Übergang von ISDN zu All-IP für den deutschen Mittelstand bietet und wie zukunftssichere, IP-basierte Lösungen zu mehr Wachstum verhelfen.

Die Publikation geht detailliert auf praxisnahe Fragen zu All-IP ein und beleuchtet die Vorteile und Herausforderungen der anstehenden Abstellung von ISDN.

Sie zeigt zudem auf, wie IP-Kommunikation den deutschen Mittelstand unterstützt. Das gilt sowohl für das daraus resultierende Plus an Flexibilität wie auch für die Beschleunigung aller Prozesse und die Optimierung bestehender Kommunikationsinfrastrukturen.

Das kostenlose E-Book bietet natürlich auch übersichtliche Feature-Checks und detailliert beschriebene Einsatzszenarien.

www.swyx.de/all-ip

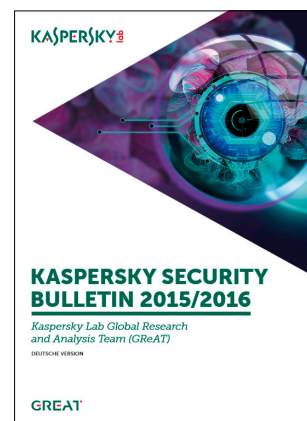
Kostenloses E-Book

Kaspersky Security Bulletin 2015/2016

Auch in diesem Jahr stellt Kaspersky Lab das Kaspersky Security Bulletin 2015/2016 als kostenloses E-Book zur Verfügung. Neben Jahresanalysen und der Entwicklung der Cybergefahren im Jahr 2015 gibt der IT-Sicherheitsexperte auch einen Ausblick auf zukünftige Internetgefahren und Angriffsszenarien.

Neben allen bisher veröffentlichten Jahresanalysen, umfasst die E-Book-Version des Kaspersky Security Bulletins 2015/2016 auch Bildmaterial und Videos. Das E-Book kann kostenfrei unter <https://kas.pr/SecBulletin15/16> heruntergeladen werden.

In puncto Malware zeichnete sich im Jahr 2015 insbesondere zwei Trends ab: die Ausbreitung von Banking-Trojanern auf neue Plattformen wie Android sowie ein deutlicher An-



Kaspersky Security Bulletin 2015/2016: Jahresanalysen und die Entwicklung der Cybergefahren im Jahr 2015

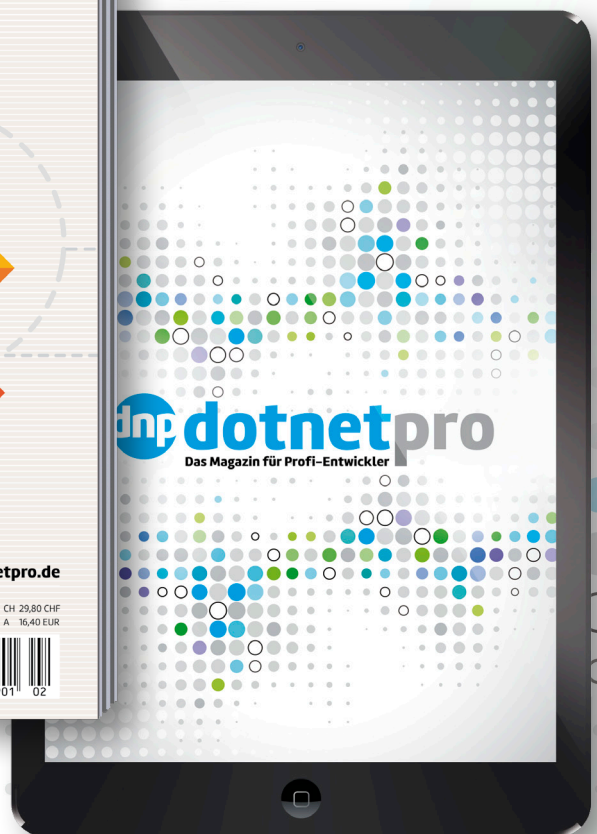
stieg von Erpresser-Schadsoftware, sogenannter Ransomware. Kaspersky Lab geht in seinen Analysen und Statistiken auch auf die geografischen Besonderheiten – unter anderem auch in Deutschland, Österreich und der Schweiz – ein.

www.kaspersky.eu/de

Jetzt kostenlos testen!



2x
gratis!



Das Fachmagazin für .NET-Entwickler

Testen Sie jetzt 2 kostenlose Ausgaben und erhalten Sie unseren exklusiven Newsletter gratis dazu.

www.dotnetpro.de/probeabo

News

Windows 10: Schon auf über 200 Millionen Geräten installiert

Jeden Monat kommen 40 Millionen neue Windows-10-Installationen hinzu, berichtet die Webseite Winbeta.org. Microsoft wollte die Zahlen noch nicht bestätigen.

In den vergangenen drei Monaten ist die Zahl der Geräte, die weltweit mit Windows 10 laufen, von 110 auf über 200 Millionen ge-



Windows 10 ist schon auf über 200 Millionen Geräten installiert

stiegen. Die Zahl wurde aber noch nicht von Microsoft bestätigt. Sie stammt von Zac Bowden, der für die unabhängige Windows-Webseite Winbeta.org schreibt. Bowden hat sie nach eigenen Angaben von nicht näher genannten Kontakten bekommen.

Wenn die Zahl stimmt, dann wurde Windows 10 jeden Monat im Schnitt auf 40 Millionen neuen Geräten installiert. Das Betriebssystem ist erst vor fünf Monaten erschienen. Bis zum von Microsoft selbst gesetzten Ziel von mehr als einer Milliarde Geräte, die mit Windows 10 bis zum Jahr 2018 laufen sollen, ist es aber trotzdem noch ein weiter Weg.

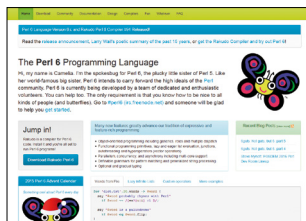
www.microsoft.com

Programmiersprachen**Perl 6 soll eigenständige Programmiersprache werden**

Nach 15 Jahren Entwicklungsarbeit hat der Erfinder der Sprache, Larry Wall, die Version 6 der Programmiersprache vorgestellt.

Einerseits sieht sich Perl 6 zwar in der Entwicklungsgeschichte von früheren Versionen der Programmiersprache, will jedoch kein reines Nachfolgeprodukt sein, sondern der erste Release einer völlig neuen Programmiersprache. Perl 5.0 erschien zum ersten Mal 1994 und liegt derzeit in der Version 5.22 vor.

Konsequente Objektorientierung ist das herausragende Merkmal von Perl 6. Daneben erlaubt die neue Sprache funktionale, prozedurale und parallele Programmierung. Seine



Perl 6 soll sich als eigenständige Programmiersprache etablieren

großen Stärken zeigte Perl bisher vor allem beim Parsen von Texten und bei der Verwendung von regulären Ausdrücken (Regex). Die Syntax dieser regulären Ausdrücke wurde in Perl 6 erweitert.

<https://perl6.org>

HPI Hasso-Plattner-Institut**Geraubte Identitätsdaten aufspüren**

Sicherheitsforscher des Hasso-Plattner-Instituts (HPI) haben im vergangenen Jahr in speziellen Internetforen fast 35 Millio-



Der Identity Leak Checker prüft, ob persönliche Identitätsdaten geraubt worden sind

nen geraubte Identitätsdaten aufgespürt. Diese kamen aus 15 Quellen. Cyberkriminelle hatten die von ihnen gestohlenen Identitäten veröffentlicht.

Wie HPI-Direktor Prof. Christoph Meinel mitteilte, kann jedermann mit dem Identity Leak Checker des Instituts überprüfen, ob seine persönlichen Identitätsdaten betroffen sind. Wer auf <https://sec.hpi.de/ilc> seine E-Mail-Adresse eingibt, erfährt nach einem Abgleich sofort, ob diese in Verbindung mit anderen persönlichen Daten offengelegt wurde und so missbraucht werden könnte.

Insgesamt verzeichnete der kostenlose Identity Leak Checker-Dienst des Hasso-Plattner-Instituts in den vergangenen zwölf Monaten fast 100.000 Besucher. In nahezu 13.000 Fällen mussten diese per E-Mail darüber informiert werden, dass Identitätsdaten von ihnen frei im Internet kursieren und welches Verhalten im speziellen Fall empfehlenswert ist. Seit dem Start des Dienstes im Mai 2014 nahmen bislang insgesamt rund 1,7 Millionen Besucher den HPI-Identity Leak Checker in Anspruch. 160.000 Warnmeldungen wegen unrechtmäßig veröffentlichter Identitätsdaten wurden bislang versandt.

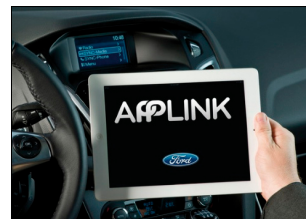
»Selbst wenn nichts gefunden wurde, teilen wir das den Anfragern mit. Das bietet aber keine Garantie, dass persönlichen Informationen nicht doch gestohlen wurden«, betonte Meinel. Denn nicht alle geraub-

ten Daten würden auch veröffentlicht. Die Daten selbst gibt das Institut aus Sicherheitsgründen nicht preis. Allerdings nennt es für jede betroffene Information die Kategorie und ein ungefähres Veröffentlichungsdatum.

<https://sec.hpi.de/leak-checker>

Car-Software**Open-Source-Software von Ford wird Branchenstandard**

Die Toyota Motor Corporation will künftig die von Ford entwickelte Open-Source-Lösung SmartDeviceLink zum Einbinden von Smartphones in die Fahrzeug-Kommunikation nutzen. Zudem evaluieren derzeit PSA Peugeot Citroën, Honda, Subaru und Mazda die Software. Mehrere namhafte Automobilhersteller und Zulieferer, allen voran die Toyota Motor



AppLink ist in mehr als fünf Millionen Fahrzeugen von Ford im Einsatz

Corporation als volumenstärkster Pkw-Produzent der Welt, übernehmen die von Ford entwickelte Softwarelösung.

Die Software eröffnet Autofahrern eine interessante Option, um ihre Smartphone-Apps vom Lenkrad aus zu nutzen. Bei SmartDeviceLink handelt es sich um eine Open-Source-Software, auf der auch die Plattform Ford SYNC AppLink basiert. Sie ermöglicht die Einbindung und Bedienung von Smartphone-Apps per Sprachsteuerung im Fahrzeug.

<http://projects.genivi.org>



Developer Week 2016

20.-23. Juni 2016,
Messe Nürnberg



Das Event 2016 für .NET-,
Web- & Mobile-Entwickler

Mit Code

DWX16wmd

€ 200,-
sparen!

Das AdBoard und Track Chairs der DWX 2016 (u. a.)



Gregor Biswanger



Jan Fellien



Dr. Ronald Hartwig



Johannes Hoppe



Patrik Lobacher



Hendrik Lösch



André Krämer



Björn Schotte



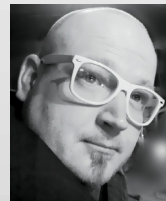
Ulrike Stirnweiß



David C. Thömmes



David Tielke



Holger Wendel

developer-week.de



DeveloperWeek

Aussteller &
Sponsoren:

adorsys



brainLight
LIFE IN BALANCE



Veranstalter:
developer
media

Präsentiert von:

dotnetpro



MATHEMA

SDX

SPARX
SYSTEMS

TEXTCONTROL

Neue
Mediengesellschaft
Ulm mbH

web & mobile
DEVELOPER

News

**Panda Security:
IT-Sicherheitsprognose**

Die IT-Experten von Panda Security haben die Entwicklungen der vergangenen zwölf Monate im Bereich der Cybersicherheit analysiert. Auf Basis dieser Daten haben sie eine Prognose darüber getroffen, mit welchen Risiken und Herausforderungen die IT-Welt in 2016 voraussichtlich konfrontiert wird.

Die Entwicklung und Verbreitung von neuen Malware-Exemplaren, die darauf abzielen, Nutzer aller Arten von digitalen Geräten zu infizieren, wird auch in 2016 weiterhin exponentiell zunehmen.

2016 gibt es voraussichtlich einen Anstieg von Infektionen via JavaScript sowie eine zunehmende Anzahl von Cyberkriminellen, die die Powershell nutzen.

Vorrangiges Ziel von Cyberkriminellen ist es, eine große Anzahl von Personen und Unternehmen anzugreifen. Aus diesem Grund werden sie Tools wie Exploit Kits verwenden.



Experten von Panda Security haben die Risiken und Herausforderungen in 2016 untersucht

Aus demselben Grund werden auch Mobilgeräte zunehmend mit Malware infiziert. Hier werden insbesondere Geräte mit Android betroffen sein.

www.pandasecurity.com

**animago Award & Conference
Umzug nach München**

Die Fachkonferenz für 3D Animation & Still, Visuelle Effekte und Visualisierung feiert im nächsten Jahr das 20-jährige Jubiläum – und zieht nach München um.



Die Fachkonferenz für 3D Animation & Still, Visuelle Effekte und Visualisierung findet 2016 in München statt

Nach sieben Ausgaben in Potsdam-Babelsberg verlegt animago Award & Conference den Veranstaltungsort nach München und findet am 27. und 28. Oktober im Kultur- und Veranstaltungszentrum Gasteig statt. Gleichzeitig wird am neuen Veranstaltungsort das 20-jährige Jubiläum der Auszeichnung animago Award gefeiert. Das Bayerische Staatsministerium für Wirtschaft und Medien, Energie und Technologie und das Kulturreferat der Landeshauptstadt München unterstützen die Veranstaltung mit insgesamt 200.000 Euro.

www.animago.com

Verband der Internetwirtschaft e.V.**Die Internetrends 2016**

Smart, Mobility, New Work und Entertainment – das sind für den Verband der Internetwirtschaft eco die Trendthemen des Jahres 2016.

»Die Internet-Industrie hat sich aus einer kleinen Nische heraus zum Taktgeber in Sachen Innovation und Wachstum

entwickelt. Es gibt schon heute kaum noch ein Unternehmen, egal aus welcher Branche, das ohne Internet denkbar wäre – jetzt befindet sich das Netz der Netze in einer Metamorphose zum Herr aller Dinge«, sagt eco-Geschäftsführer Harald A. Summa.

In der smarten Welt ist das Internet nicht mehr nur Teil unseres Lebens wie bisher, sondern es stellt die Grundlage für nahezu alle anderen Aktivitäten dar, denen wir privat und beruflich nachgehen, postuliert der eco-Chef. Die allumfassende Digitalisierung erschüttert nicht nur eine Branche nach der anderen, sondern dringt auch in alle privaten Lebensbereiche ein, das Smart Home steht beispielhaft für diese Entwicklung. »Wir werden ab 2016 sehen, dass die Insellösungen von der Lichtsteuerung per App bis zur Medienzentrale zu smarten Ökosystemen zusammenwachsen.«

Das Smart Car steht für die wachsende Bedeutung des Internets im Bereich Mobility. Bis 2030 sollen 60 Prozent der Weltbevölkerung in Ballungsräumen leben. Intermodale Mobilitätskonzepte sind laut eco unabdingbar, um den Verkehrskollaps zurückzudrängen. Car-Sharing, ÖPNV-Konzepte mit Call-a-Bike-Verlängerung bis zur Haustür und verkehrsmittelübergreifendes Payment stehen exemplarisch für neue Mobility-Ansätze. Als einen Game Changer stuft eco das selbststeuernde Automobil ein. »Wir stehen an der Schwelle zur massenhaften Verbreitung des ersten Automobils, das diesen Namen wirklich verdient, weil es sich ohne unmittelbares menschliches Zutun im Verkehr zurechtfindet«, verdeutlicht eco-Chef Harald A. Summa die Tragweite. Auch auf dem Logistiksektor stehen umfassende Innovationen wie Same-day-

Delivery vor der Tür. Der Einzug der Shared Economy in die Logistikbranche ist ebenfalls unübersehbar. Ganz selbstverständlich nutzen Geschäftsleute mittlerweile die Mobilitätsangebote der Car-Sharing-Anbieter und Transportdienstleister wie zum Beispiel Uber, um die Fixkosten der eigenen Fahrzeugflotte zu reduzieren. Als Pionierunternehmen der Shared Logistik ist die Frachten- und Laderaumbörse TimoCom bereits seit Jahren europaweit erfolgreich. In den USA macht in diesem Zusammenhang das kalifornische Start-up-Unternehmen Cargomatic von sich reden. Das Unternehmen vermittelt freien Laderaum



Harald A. Summa, Geschäftsführer beim Verband der Internetwirtschaft e.V. eco

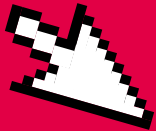
auf LKWs gegen Kostenbeteiligung an andere Unternehmen.

Das Internet wird in dieser neuen Ära der Smart Mobility eine Schlüsselrolle als Innovationsmotor spielen. »Die Internetwirtschaft ist wie kaum eine andere Branche darin geübt, innovative Geschäftsmodelle zu entwickeln und umzusetzen«, erklärt Summa.

Die Technologieveränderungen bringen zudem einen vielfältigen Einfluss auf die Arbeitswelt mit sich und werden zahlreiche Arbeitsplätze schaffen. Neue Arbeitskulturen haben das Potenzial, zu mehr wirtschaftlichem Erfolg, Lebensqualität und Chancengleichheit zu führen, heißt es bei eco.

www.eco.de

Developer Newsletter



Top-Informationen für Web- und Mobile-Entwickler.
Klicken. Lesen. Mitreden.

web & mobile
DEVELOPER

Newsletter

Probleme mit der Darstellung | Aktuelles Heft

// news



Stellenbörse für Open Source-Unternehmen

Der Open Source-Branche geht es gut, und mit dem Erfolg wächst der Bedarf an weiteren Mitarbeitern. Dem trägt die Open Source Business Alliance (OSB Alliance) nun Rechnung und startet auf ihrer Website eine Stellenbörse und veröffentlicht in einem ersten Schritt Stellenangebote ihrer Mitgliedsunternehmen.



Add-on-Marktplatz für Node.js-Entwickler

Die Progress-Tochtergesellschaft Modulus hat eine Reihe von Zusatzprodukten auf ihrem Add-on-Marktplatz veröffentlicht. Damit ist es für Node.js-Entwickler einfacher, neue Funktionalitäten schneller in ihre Applikationen einzubauen.



HPI will Benchmarks für Big-Data-Leistungsvergleiche erarbeiten

Das Hasso-Plattner-Institut (HPI) ist Gastgeber des fünften internationalen Workshops zu Leistungsvergleichen im Bereich Big Data, dem so genannten Big Data Benchmarking. Das Treffen, zu dem rund 80 Teilnehmer erwartet werden, findet am 5. und 6. August am HPI in Potsdam statt.

Jetzt kostenlos anmelden:



webundmobile.de



twitter.com/webundmobile



facebook.de/webundmobile



gplus.to/webundmobile



Foto: Shutterstock / Macrovector

Sicherheit 2016

Das Bekanntwerden unerwartet ernster Verwundbarkeiten aufgrund uralter Bugs in OpenSSL (Heartbleed), in der Bash-Shell (Shellshock), verwundbarer Hypervisors (Venom) und das mittlerweile leicht hackbare Duo aus HTTP/1.1 und SSL/TLS hat die IT-Gemeinde im Kern erschüttert. **web & mobile developer** nimmt den aktuellen Stand der Dinge unter die Lupe.

chen Wettbewerbsfähigkeit legen Unternehmen verstärkt Wert auf ein ganzheitliches Verständnis der Datensicherheit. Das größte Problem im Hinblick auf die Sicherheit von Webanwendungen und mobilen Apps besteht schlicht und einfach darin, dass das Bewusstsein für höhere Sicherheitsanforderungen zwar vorhanden ist, aber das Wissen um die technischen Möglichkeiten der Implementierung bei vielen Entscheidungsträgern nicht allzu stark ausgeprägt ist.

Da bei jedem Vorfall eine Vielzahl von Faktoren eine Rolle spielt – auf der einen Seite die Code-Qualität, auf der anderen Seite die Qualität der Systemkonfiguration wie auch die Zuteilung von finanziellen Mitteln zu den verschiedenen Bereichen der IT –, wäre eine verbesserte Kommunikation zwischen der Führungsetage, den Web- und App-Entwicklern und den Administratoren für das Gesamtergebnis von Vorteil.

Web- und Mobile-Entwickler können ihr umfassendes Wissen über die Sicherheitsanforderungen ihrer Webanwendun-

gen und webangebundener mobiler Apps zum Tragen kommen lassen, indem sie sich an dem Design der Backend-Anbindung aktiv beteiligen. Das Komitee-Syndrom, das auftritt, wenn sich niemand für das Gesamtergebnis hinreichend zuständig fühlt, um Fehler außerhalb des eigenen, mehr oder weniger eng abgesteckten Verantwortungsgebiets behoben zu bekommen, können nur diejenigen Entwickler kontern, die sich auch mit der Cybersecurity-Seite ihrer Anwendungen auskennen.

Entwickler von Web- und Mobile-Anwendungen benötigen demnach zumindest etwas Know-how rund um eine durchgängige, lückenlose Sicherheit ihrer Webanwendungen und mobilen Apps. Dieses Wissen umfasst unter anderem Kompetenzen rund um die fachgerechte Implementierung der Datenverschlüsselung sowohl bei der Übertragung als auch im Ruhezustand.

Rasanter Fortschritt und IoT

Im Bereich der Cybersecurity von webangebundener Software dürften drei massive Trends die nahe Zukunft prägen:

- das Internet der Dinge (IoT) als Risikofaktor für amokgelaufene Elektronik,
- mobile Apps mit Fähigkeiten zur Datenerfassung als Angriffsvektor mit steigender Bedeutung,
- Webshop-Raub und Erpressung als Instrumente organisierter Kriminalität, denen eine größtenteils hilflose Strafverfolgung gegenübersteht.

Das Internet der Dinge (IoT) verleiht dem Thema Softwaresicherheit eine völlig neue Ernsthaftigkeit. Der größte anzunehmende Unfall beim Einbruch in IT-Systeme bestand bisher im Daten- und Imageverlust. Im IoT-Zeitalter mit Zulieferungsdrohnen, selbstfahrenden Autos, funkenden Herzschrittmachern und Internet-angebundenen Smart-Home-Einrichtungen stehen vielmehr Menschenleben der Endbenutzer und Dritter auf dem Spiel.

Das Internet der Dinge liegt keinesfalls in der fernen Zukunft. Bereits heute sind weltweit allein in der privaten Wirtschaft schätzungsweise 25 Milliarden IoT-fähige Endgeräte aktiv im Umlauf. Auch sind bereits die ersten Verwundbarkeiten ans Tageslicht gekommen.

ConnectedDrive und PKW-Hijacking

Das Internet der Dinge hält zunehmend bei Fahrzeugen Einzug. Immer mehr Autos verlassen die Produktionshallen mit einem Modem zur drahtlosen Datenübertragung. Die Anwendungen sind sehr vielfältig: Die einen Autohersteller möchten den Insassen diverse Features über eine mobile App zur Verfügung stellen, die anderen verschaffen sich Möglichkeiten zur Ferndiagnose oder modellieren das Nutzungsverhalten anhand diverser Messwerte, die sie auch ohne die Zustimmung des Autobesitzers anzapfen können.

Autos mit einer drahtlosen Anbindung können beim Unfall die Rettungskräfte verständigen und im Nachhinein möglicherweise auch die Schuld und Unschuld eines Autofahrers unter Beweis stellen. Doch Konnektivität ist ein zweischneidiges Schwert: Bei einem Datenleak kann dieselbe Technologie äußerst unangenehme Folgen nach sich ziehen. Letzteres ist gar nicht so weit hergeholt, wie es manche Befürworter dieser Technologien gerne hätten, zumal das Betätigungsfeld Datenerfassung, -verwertung und -übertragung von den Regulierungsbehörden größtenteils vernachlässigt wird.

Der ADAC fordert lautstark, dass die Autofahrer die Datenhoheit über ihre Fahrzeuge haben sollten. Bisher ist dies immer noch nicht der Fall.

Nachdem sich Volkswagen, Audi und Porsche den Zorn internationaler Umweltbehörden im peinlichen Emissionsskandal durch serienmäßig manipulierte Abgasmesswerte zugezogen haben, konnten BMW, MINI und Rolls-Royce noch im Bereich der Sicherheit nachkarten. Eine Untersuchung des



Foto: Shutterstock / a-image

Das Internet der Dinge (IoT) erhöht die Relevanz von Sicherheit in webangebundenen Anwendungen (Bild 1)

deutschen Sicherheitsexperten Dieter Spaar im Auftrag des ADAC hat haarsträubende Sicherheitsmängel ans Tageslicht gebracht.

BMW's bedauernswerte Softwaresicherheit ist in der Autobranche beim Weiten kein Ausreißer. Im Sommer 2015 konnten Hacker über eine Funkverbindung die Kontrolle über einen Jeep Cherokee (Hersteller: Ford) an sich reißen. Bei einer Geschwindigkeit von über 110 Stundenkilometern im dichten Verkehr auf der Autobahn verstellten sie das Radio auf die maximale Lautstärke, drehten die Scheibenwischer voll auf, starteten die Klimatisierung und beamten ihre Gesichter auf das digitale Dashboard.

Dieser Vorfall war dem Fahrer vorab aber bewusst; bei den beiden Hackern handelte es sich um zwei Sicherheitsexperten, Charlie Miller und Chris Valasek. Zuvor hatten sie auch schon Erfolgserlebnisse beim Deaktivieren des Gaspedals und des Bremssystems in einem Ford Escape und einem Toyota Prius ebenfalls via Funk verzeichnen können. ►

Das Problem der IoT-Sicherheit ist leider weit verbreitet. Das völlige Fehlen von Sicherheitsbewusstsein bei zahlreichen Anbietern führt dazu, dass sie diese Problematik viel zu oft auf die leichte Schulter nehmen, bis sich ein Sicherheitsexperte des Themas annimmt (Bild 2).

BMW nutzte für alle Fahrzeuge im ADAC-Test identische symmetrische (anstatt fahrzeugspezifischer) Sicherheitsschlüssel. Zur Datenverschlüsselung der Notruf-SMS kamen der klar veraltete DES-Algorithmus (56-Bit-Schlüssel) und AES128 (128-Bit-Schlüssel) zum Einsatz. Bei einigen Diensten verzichtete BMW gänzlich auf die Transportverschlüsselung.

ADAC-Tests der ferngesteuerten Türentriegelung haben gleich mehrere recht gravierende Sicherheitslücken ans Tageslicht gefördert (Bild 3).

Nach dem Mitschneiden der Datenkommunikation zwischen dem Fahrzeug und den Backend-Servern konnte der Sicherheitsforscher des ADAC die Datenpakete problemlos entschlüsseln. Resultat: Die verwendeten Algorithmen seien zwar etwas einbruchssicherer (AES 128 anstelle von 56-Bit-DES und HMAC-SHA256 anstelle von DES CBC-MAC), dafür nutze BMW dieselbe Schlüsseltabelle wie bei der nahezu unverschlüsselten Notruf-SMS.

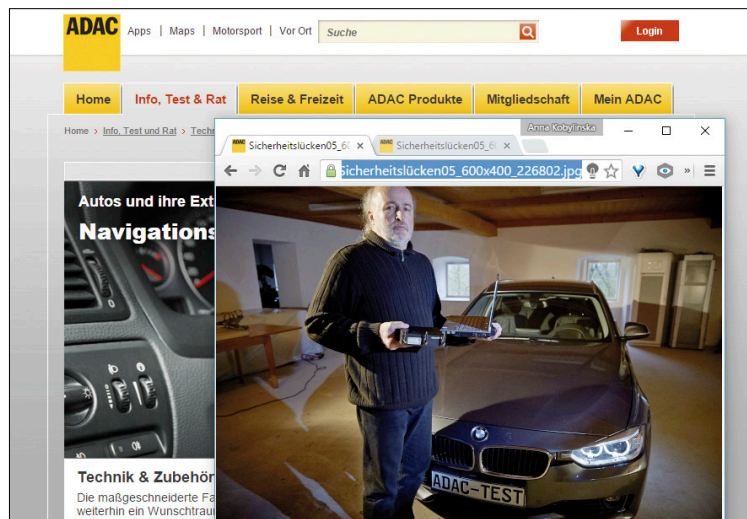


Foto: ADAC

Der Sicherheitsexperte Dieter Spaar hat gravierende Sicherheitslücken in der ConnectedDrive-Steuerung aufdecken können (Bild 2)

Mit einer gefälschten SMS ließe sich das Fahrzeug dazu verleiten, mit einem einfachen HTTP-GET-Request die Konfigurationsdaten des Fahrzeugs vom BMW-Backend als unverschlüsseltes XML einzuholen. Zwar bestünde für den Besitzer die Möglichkeit, Remote-Services abzuschalten, doch soll

Sicherheitslücken im Rückspiegel

Eine Reihe von kürzlich entdeckten Sicherheitslücken, die niemand in diesem Umfang erwartet hatte, hat das Vertrauen in die bisherigen Sicherheitskonzepte des Web erschüttert.

Heartbleed

Ein Bug in der TLS-Implementierung von OpenSSL, der es ermöglicht, Daten wie Passwörter oder private Schlüssel aus dem Arbeitsspeicher eines entfernten Servers zu extrahieren, warf Zweifel an der Sicherheit administrativer Remote-Verbindungen auf und lieferte eine gute Begründung für die Implementierung von Perfect Forward Secrecy.

Shellshock

Eine inzwischen behobene Verwundbarkeit in der Unix/Linux-Shell Bash rund um die Remote-Execution von Code via Bash rief die Zeiten von Netzwerk-Hacking in Erinnerung und wurde mit einem Update behoben.

XSS/CSRF

Das beharrliche Wiederauftreten von Cross-Site-Scripting- und Cross-Site-Request-Forgery-Angriffen machte Webentwicklern klar, dass der Kampf um die Sicherheit des E-Commerce immer wieder neu errungen werden muss.

Venom

Virtualized Environment Neglected Operations Manipulation, ein Bug in dem Emulator QEMU, auf dem die Hypervisors Xen, KVM,

und Oracles VirtualBox aufsetzen, hat die Sicherheit virtualisierter Umgebungen in Frage gestellt, bis es mit einem Update behoben werden konnte.

FREAK-Angriffe

Die FREAK-Angriffe auf den RSA-Schlüssel durch ein vom Browser ausgelöstes Downgrade der Verschlüsselungsebene machte die Neukonfiguration rund jedes dritten Webserver erforderlich (noch rund 11 Prozent aller Webserver sind weiterhin betroffen).

LOGJAM

Eine Downgrade-Angriffe auf TLS hat zahlreiche Websites zum Upgrade auf den 2048 Bit starken Diffie-Hellman-Algorithmus inspiriert (<http://weakdh.org>).

HTTP

Zahlreiche Verwundbarkeiten von HTTP 1.1 mit SSL/TLS, darunter POODLE, eine Man-in-the-middle-Angriffe in HTTP 1.1, haben der Umstellung auf HTTP/2 eine neue Dringlichkeit verliehen.

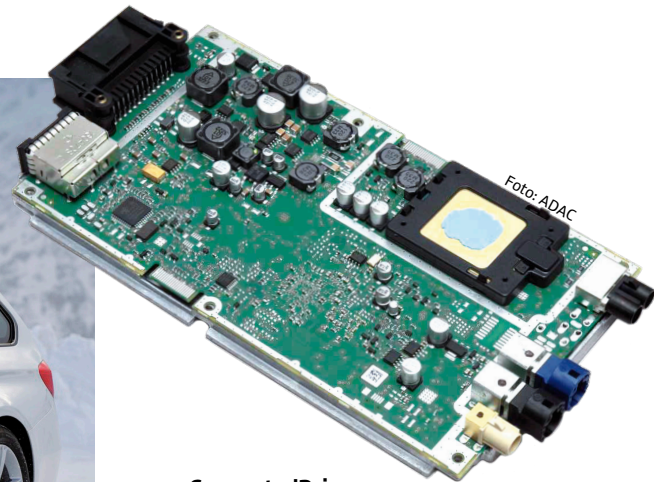
CRIME

Compression Ratio Info-leak Made Easy in HTTPS und SPDY hat bisherige Annahmen zur Sicherheit GZIP-komprimierter Datenverschlüsselung durch Session-Hijacking widerlegen können und lässt sich am wirksamsten durch den Umstieg auf HTTP/2 verhindern.



Foto: BMW

Bewundernswerte Fahrtechnik, doch an der Sicherheit der Remote-Steuerung haperte es, meint der ADAC nach einem Sicherheitsaudit von Dieter Spaar (Bild 3)



ConnectedDrive:

Die Combox liefert die sicherheitskritische Identifikationsnummer des Fahrgestells eines Fahrzeugs dem Angreifer in einer überaus hilfreichen Fehlermeldung (Bild 4)

sich diese Funktionalität problemlos mit einer ebenfalls gefälschten SMS-Nachricht via Mobilfunk freischalten lassen. Hierzu sei zwar die Fahrgestellnummer erforderlich (die sogenannte VIN), die das jeweilige Fahrzeug eindeutig identifiziere, doch diese ließe sich bei vielen der betroffenen Modelle aus Fehlermeldungen der Combobox ablesen (Bild 4).

Empfängt ein betroffenes Fahrzeug eine gültige Nachricht ohne die richtige VIN, meldet es sich mit der korrekten VIN im Absenderfeld zurück. Dieses Verhalten beschränkt sich auf Autos mit der sogenannten Combobox. Bei anderen ConnectedDrive-fähigen Fahrzeugen soll sich die benötigte Identifikationsnummer zwar nicht einer Fehlermeldung entnehmen, dafür aber mehr oder weniger bequem direkt am Fahrzeug abfotografieren lassen.

Wer also die Verschlüsselung für ein einziges ConnectedDrive-Fahrzeug geknackt haben sollte, könne nicht nur

die Konfiguration für das betroffene Auto auslesen, sondern gleich bei allen ConnectedDrive-Fahrzeugen mit einer gefälschten SMS aus der Ferne die Türen öffnen.

Als Antwort auf die Enthüllungen habe BMW die SSL-Verschlüsselung aktiviert; weitere Spezifikationen wurden jedoch nicht bekanntgegeben.

Warum auch diese Maßnahme wohl kaum ausreicht, um eine Man-in-the-middle-Attacke gänzlich zu verhindern, lässt sich anhand zahlreicher Sicherheitslücken und Ausrutscher in den gängigsten Webanwendungen illustrieren (Bild 5).

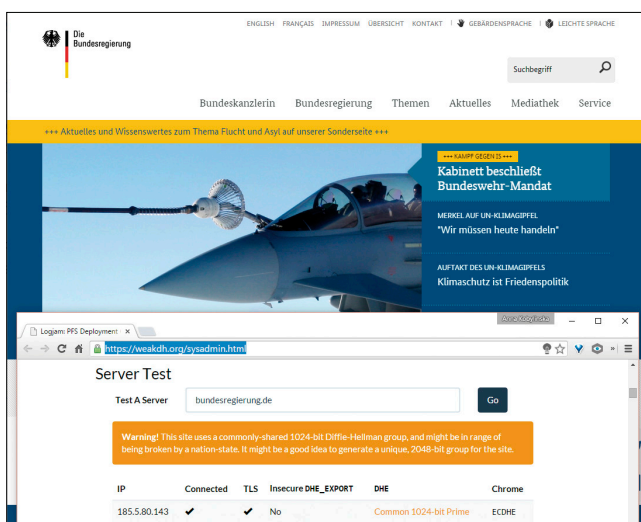
Ob der TÜV jetzt Software-Sicherheitsexperten beauftragen muss, um die Sicherheit von Fahrzeugen zu gewährleisten, sei einmal dahingestellt.

Auf der Achterbahn mit sensiblen Benutzerdaten

Zahllose Sicherheitslücken bei Anbietern wie Apple, Facebook, Google, Microsoft und anderen haben die inzwischen recht dramatische Lage rund um die Datensicherheit ins Rampenlicht geholt. Zum Teil handelte es sich um Verwundbarkeiten, die seit Jahren existierten und auch beim Bekanntwerden meist nur zögerlich behoben wurden. Die World Bank wurde sagenhafte 419-mal gehackt, bevor die Sicherheitslücke an der Quelle behoben werden konnte (Bild 6, Bild 7).

Das Dreiste an der Sache: Die Hacker konnten ein gültiges Extended-Validation-SSL-Zertifikat der prominenten Website missbrauchen, um eine PayPal-Phishing-Applikation zu hosten. Ein EV-Zertifikat stellt die höchste Qualitätsstufe eines SSL-Zertifikats dar; dieser Art von Zertifikat liegen umfassende Identitätsprüfungen zugrunde.

Bei einigen bekannten Anbietern von Verbraucherelektronik kamen gezielte Verletzungen der Privatsphäre der Endbenutzer durch eine Mischung aus gezieltem Unwissen (Dell mit dem selbstsignierten Zertifikat eDellRoot) beziehungsweise unlauteren Absichten (Sony BMG mit der XCP-Spyware) zu Stande. Andere bekannt gewordene Fehlgriffe ließen sich wiederum auf reines Versehen zurückführen. Das ►

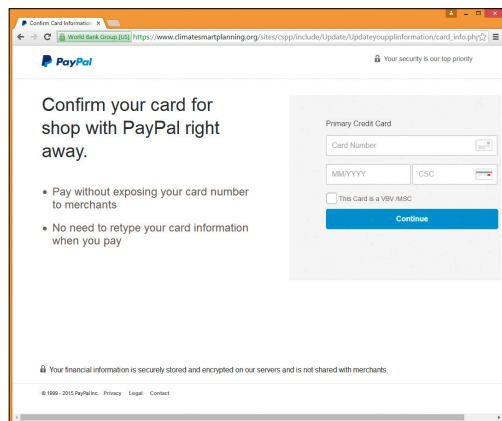


Vorsicht Stolperfalle: Die unter anderem von Bundesregierung.de eingesetzte Diffie-Hellman-Gruppe mit einer Schlüssellänge von 1024 Bit gilt als unsicher (Bild 5)

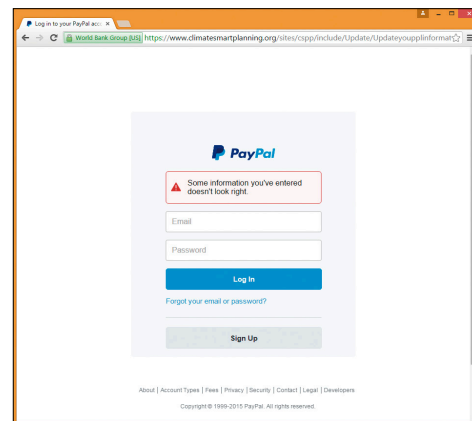
Backend-System des Werbenetzwerks Chitika (<http://chitika.com>) setzte eine Zeitlang benutzereigene Einstellungen für den strikten Schutz der Privatsphäre nach zehn Tagen wieder stillschweigend auf die Standardwerte zurück und zog sich dadurch den Unmut der Anwender (und einen Vergleich mit der US-Behörde FTC) zu.

Nicht eines der führenden Finanzinstitute Großbritanniens kann von sich behaupten, in den vergangenen zwei Jahren ohne gravierende Sicherheitsvorfälle, die die Datensicherheit ihrer Benutzer verletzten, davongekommen zu sein. Die Hacker machen nicht einmal vor der Prominenz der IT-Industrie halt.

Larry Ellison (Bild 8), der im Lauf seiner 37 Jahre langen Laufbahn als Chef von Oracle praktisch niemals Fehler oder



Sicherheitszertifikat: Eine PayPal-Phishing-Seite mit dem EV-Sicherheitszertifikat der Weltbank (Bild 6)



Die PayPal-Phishing-Seite mit dem EV-Sicherheitszertifikat der Weltbank lieferte sogar recht überzeugende Fehlermeldungen (Bild 7)



Oracles Larry Ellison bejammert neuerdings lautstark das traurige Ergebnis auf dem Cybersecurity-Schlachtfeld westlicher Nationen (Bild 8)

Verwundbarkeiten zugab, jammerte lautstark, dass die westliche Welt das Gros der Cybersicherheitsschlachten einfach sang- und klanglos verlieren würde. In seiner Keynote auf der Oracle OpenWorld-Konferenz in San Francisco am 27. Oktober 2015 meinte Ellison: »Wir brauchen eine deutlich bessere Sicherheit. Wir brauchen die nächste Generation von Sicherheitstechnologie, da wir gelinde gesagt nicht allzu viele Cybersicherheitsschlachten gewinnen (...)«.

Rund einen Monat später fiel das von Oracle eingesetzte System zur Vergabe von Zertifizierungen für IT-Kompetenzen, Pearson VUE (das im Übrigen auch IBM und Cisco benutzen), einer Hackerattacke zum Opfer.

Inzwischen haben alle Sicherheitslücken, die es in die Schlagzeilen schaffen, eines gemeinsam: Sie betreffen ge-

waltige Benutzerzahlen. Eine Woche vor der Vorstellung der neuesten Generation des iPhone 6 am 9. September 2014 musste Apple zähneknirschend gravierende Sicherheitslücken zugeben, die laut einem Bericht des Wall Street Journals die Accountdaten von über 320 Millionen iCloud-Anwendern in Mitleidenschaft gezogen haben sollen.

Wenn finanziell gut aufgestellte Großunternehmen bereits ihre liebe Mühe damit haben, Sicherheitslücken in ihren Anwendungen aufzuspüren (geschweige denn zu schließen), sind kleine und mittlere Softwareschmieden umso mehr von dem Problem betroffen. Viele Sicherheitslücken entstehen zum Teil dadurch, dass der IT- und der Entwicklungsabteilung oft ein viel zu knappes Budget zur Verfügung steht. So entstand PHP ursprünglich als eine kostengünstigere Alternative zu Java, deren Ansprüche im Hinblick auf die Rechenleistung wie auch auf die erforderliche Entwicklungszeit den verfügbaren Budgetrahmen vieler Anwender sprengten. Java hat sich vor allem bei multinationalen Großkonzernen durchgesetzt, die ihre hohen IT-Kosten nicht sonderlich rechtfertigen müssen (Banken, Versicherungen et cetera).

Die Schlussfolgerung, dass zu knappe IT-Budgets die Entstehung von Verwundbarkeiten fördern, ist sicherlich nachvollziehbar. Den betroffenen Kleinunternehmern und Mittelständlern bleibt rein nichts anderes übrig, als mit geballtem Fachwissen und quelloffener Software zu kontern und sich bloß nicht als ein leichtes Opfer zu erkennen zu geben. ■



Filipe Pereira Martins und Anna Kobylinska

sind international anerkannte IT-Berater mit dem Schwerpunkt auf Cloud-Lösungen. Sie stehen den Lesern der **web & mobile developer** gern per Twitter via **@D1gitalPro** und **@D1gitalInfo** zur Verfügung.



6 MONATE
0%
 €/Monat
 danach 9,99 €/Monat*

MANAGED WORDPRESS

100 % POWER

- **NEU!** Full SSD Webspace
- **NEU!** Full SSD Datenbanken
- **NEU!** PHP 7 mit OPcache
- **Unlimited** Traffic
- **Unlimited** E-Mail Accounts

100 % SICHERHEIT

- **NEU!** 1&1 DDoS Protection auf NGINX für noch mehr Performance, Zuverlässigkeit und höchste Sicherheit
- **Georedundanz:** Parallelbetrieb in räumlich getrennten Rechenzentren
- 1&1 CDN
- 1&1 SiteLock

100 % KOMFORT

- **NEU!** 1&1 WP Assistent – einzigartige Nutzerführung bei Installation und Design
- **Inklusive Ready-to-use Templates**
- **24/7 Hotline**
- 1&1 WP Experten-Support
- 1&1 Community



DE: 02602 / 96 91
 AT: 0800 / 100 668



1und1.info

* Preisvorteil bei 12 Monaten Mindestvertragslaufzeit: 1&1 Managed WP PLUS 6 Monate für 0,- €/Monat, danach 9,99 €/Monat. 14,90 € Einrichtungsgebühr. Preise inkl. MwSt. Rubik's Cube® used by permission of Rubik's Brand Ltd. 1&1 Internet SE, Elgendorfer Straße 57, 56410 Montabaur.

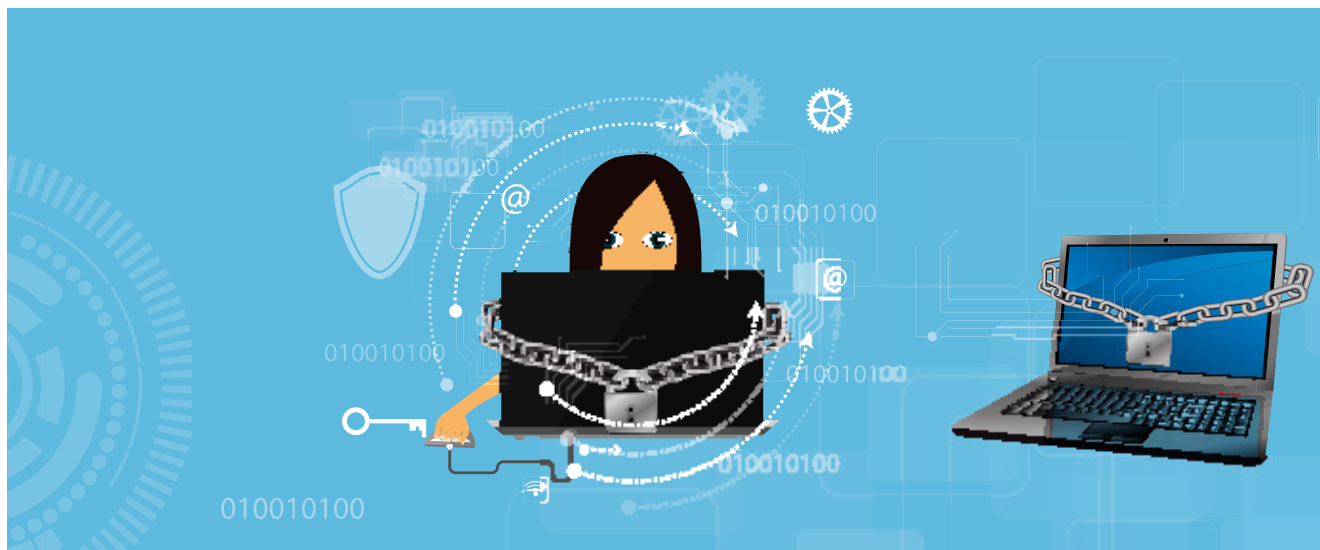


Foto: Shutterstock / julRose

BEDROHUNGSSZENARIOEN UND IHRE BEWÄLTIGUNG

Konkrete Bedrohung

Mangelndes Sicherheitsbewusstsein stellt eine Problemquelle dar.

Eine Kette ist bekanntlich nur so stark wie das schwächste ihrer Glieder. Konkret bedeutet dies, dass es kein Allheilmittel für die perfekte Sicherheit gibt, sondern leider nur eine lange Zu-erledigen-Liste möglicher Katastrophenfelder, die es sorgfältig abzuarbeiten gilt und die nicht zuletzt in regelmäßigen Abständen auch immer wieder einer gründlichen Aktualisierung bedarf.

Deserialisierung-Sicherheitslücken

Deserialisierung-Sicherheitslücken bilden eine eigene Klasse von Problemen.

Der Prozess der Konvertierung der Applikationsdaten in ein anderes – zum Beispiel binäres – Format wird allgemein als Serialisierung bezeichnet. Der umgekehrte Prozess, also das Einlesen der Daten, nennt sich entsprechend Deserialisierung (siehe Kasten »Serialisierung und Deserialisierung«).

Die Serialisierung und Deserialisierung sind unvermeidlich: Die meisten Programmiersprachen, darunter Java, PHP und C++, haben eine eingebaute Funktionalität, die es den Programmierern erlaubt, Daten zu schreiben oder diese Daten über das Netzwerk zu streamen. Problematisch wird es, sobald ein Programm serialisierte Dateneingaben vom Anwender akzeptiert und diese zur Nutzung im Programm deserialisiert. Je nach eingesetzter Sprache können sich die unterschiedlichsten, meist unerwünschten Nebeneffekte ergeben.

Im schlimmsten Fall kann die misslungen umgesetzte Deserialisierung (zum Beispiel durch Apaches Commons Collections-Bibliothek) eine Remote-Code-Ausführung zur Folge haben (Remote Code Execution).

Deserialisierung-Sicherheitslücken sind leider kein neues Problem; sie wurden auch bereits im Spring Framework, in Groovy und in den *commons fileupload*, wenn auch nur sehr zögerlich, behoben. Leider sind die Sicherheitslücken in Apaches Commons Collections-Bibliothek aber nach gut einem Jahr seit der erstmaligen Veröffentlichung eines Exploits durch Chris Frohoff (@frohoff) und Gabriel Lawrence (@gebl) immer noch vorhanden, obwohl sie einen potenziell enormen Schaden anrichten könnten.

Von der Java-Bibliothek Commons Collections der Apache Software Foundation sind leider unter anderem zahlreiche quelloffene wie auch proprietäre Java-Applikationen betroffen. Sowohl die Java-Superstars WebLogic, WebSphere, JBoss, Jenkins und OpenNMS als auch all die weniger prominenten Webapplikationen beziehungsweise Webdienste, die auf den verwundbaren Bibliotheken aufbauen, erben deren Sicherheitslücken.

Zu guter Letzt gibt es auch eine beachtliche Anzahl von quelloffenen Java-Apps kleinerer und mittlerer Relevanz, die eben ganz selbstverständlich die Commons Collections-Bibliothek verwenden. Diese Bibliothek führt bei der Deserialisierung von Datenpaketen nicht nur keine Bereinigung der Eingabe durch, sondern unterlässt auch noch jegliche Prüfung im Hinblick auf eine mögliche Man-in-the-middle-Attacke.

Das Problem mit der Commons Collections-Bibliothek tritt überall dort auf, wo quelloffene Java-Apps zum Einsatz kommen, also in der privaten Wirtschaft, in der Lehre und Forschung, in der Verwaltung und in militärischen Einrichtungen. Genaue Angaben gibt es leider nicht, weil sich die Be-

troffenen vorzugsweise in Schweigen darüber hüllen und lieber die enormen Sicherheitsrisiken auf die leichte Schulter nehmen, als das Problem an der Wurzel anzupacken. Dabei ist das Problem genauso gravierend wie durchaus zu beheben.

Obwohl sich viele Unternehmen die Cyber-Security auf die Fahnen geschrieben haben, merkt man bei gängigen WebSphere-Installationen kaum etwas davon.

Um mögliche Skeptiker hinsichtlich der Sicherheitslücken in der Commons-Collections-Bibliothek aufzuklären, hilft es, mit einem konkreten Exploit in die Materie einzusteigen. Der Einstiegspunkt ist ein TCP-Verwaltungsport an der Nummer 8880. Dieser Port ist in der Regel nicht über das Internet zugänglich, doch sollte es Ihnen doch einmal gelingen, mit einer WebSphere-Box dasselbe Subnetz zu teilen, hat für diese einzelne WebSphere-Installation das letzte Stündlein geschlagen. Oft kann dies der Fall sein, wenn WebSphere vielleicht im Rahmen einer Messe mal eben über Docker mit gestümperter Sicherheit gestartet wird oder wenn im firmenweiten Unternehmensnetzwerk ein einziges Subnetz den Regelfall darstellen sollte.

Der schwierigste Schritt für den Angreifer besteht darin, einen WebSphere-Prozess ans Laufen zu bekommen. Cybersecurity-Experten haben es aber meistens nicht allzu schwer, die WebSphere-Sicherheitslücke dem erstaunten Publikum zu präsentieren, denn nur zu oft laufen diese Systeme mit der Standardeinstellung vor sich hin. In diesem Fall ließe sich ein Prozess zum Beispiel wie folgt starten:

```
/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/bin# ./startServer.sh server1
```

Jetzt muss man lediglich nachprüfen, ob die Commons-Collections-Bibliothek auch zum Einsatz kommt, etwa so:

```
find . -iname "*commons*collection*"
```

Das Resultat könnte nun wie folgt aussehen:

```
./WebSphere/AppServer/optionalLibraries/Apache/Struts/1.1/commons-collections.jar
./WebSphere/AppServer/optionalLibraries/Apache/
```

Tipp: Penetrationstests, aber richtig

So umgehen Sie bekannte Stolperfallen beim Durchführen von Penetrationstests.

Vermeiden Sie Tests am Produktionsserver, damit Sie nicht versehentlich Sicherheitslücken eröffnen. Zur Durchführung von Penetrationstests an Systemen eines externen Auftraggebers lassen Sie sich die Aufgabe schriftlich bestätigen. In der Public Cloud (Amazon AWS, Google, Microsoft Azure et cetera) gelten besonders rigide Vorschriften: In der Regel müssen Sie sogar für Penetrationstests Ihrer eigenen Systeme vorab eine Genehmigung des Cloud-Anbieters unter Angabe konkreter IP-Adressen per Webformular einholen; im Fall der Zuwiderhandlung können nicht vorab genehmigte Penetrationstests sogar zur Schließung des betreffenden Cloud-Accounts führen. Stellen Sie sicher, dass Sie über lückenlose Logdateien Ihrer Systeme verfügen, damit Sie nicht nur Ihre Aktivitäten belegen können, sondern auch im Nachhinein nachweisen können, dass sich Ihr Unternehmen im Rahmen der verbindlichen Vertragsrichtlinien des Diensteanbieters (ToS/AGBs) korrekt verhalten hat.

```
Struts/1.2.4/commons-collections.jar
./WebSphere/AppServer/plugins/com.ibm.ws.prereq.commons-collections.jar
./WebSphere/AppServer/systemApps/LongRunningScheduler.ear/JobManagementWeb.war/WEB-INF/lib/commons-collections.jar
./WebSphere/AppServer/systemApps/isclite.ear/commons-collections.jar
./WebSphere/AppServer/deploytool/itp/plugins/com.ibm.websphere.v85_2.0.0.v20120621_2102/wasJars/com.ibm.ws.prereq.commons-collections.jar
```

Um jetzt konkreter zu Werke schreiten zu können, hilft der *ls-of*-Befehl (list open files) weiter:

```
/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/bin# ls-of -i -P | grep LISTEN |grep java
```

Serialisierung und Deserialisierung

Unter der Serialisierung versteht man in der Informatik die Abbildung von strukturierten Daten auf eine sequenzielle Darstellungsform.

Die Serialisierung ist primär für die Persistierung von Datenobjekten in Dateien und zudem auch für die Übertragung von Datenobjekten über das Netzwerk von verteilten Softwaresystemen vorgesehen, zum Beispiel die Umwandlung von Benutzereingaben in eine statische Binärform. Speichermedien (Festplatten, SSDs, Tapes et cetera) werden dann genutzt, um Datenströme zu speichern. Möchte man ein Datenobjekt dauerhaft haltbar

machen, so spricht man im IT-Jargon von Serialisierung. Nach der Serialisierung liegt ein Datenobjekt gewöhnlich mehrfach vor, zum Beispiel als Datei und auch aus Performancegründen im Arbeitsspeicher. Beachtenswert ist hierbei, dass sich Modifikationen an den Objekten im Arbeitsspeicher nicht ohne Weiteres auf das serialisierte Objekt in der externen Darstellung (zum Beispiel eine Datei) auswirken.

Auch die Umkehrung der Serialisierung, also die Deserialisierung, ist möglich und sinnvoll, falls sie korrekt umgesetzt wird. Bei der Deserialisierung handelt es sich um die Transformation eines Datenstroms in Objekte.

Anhand der mittels *Isop* aufgestellten Liste von geöffneten Ports kann der Angreifer informierter und damit gezielter vorgehen. Die resultierende Ausgabe könnte zum Beispiel wie folgt aussehen:

```
java 15269 root 121u IPv6 2352865 0t0 TCP *:8880
(LISTEN)
java 15269 root 125u IPv6 2352866 0t0 TCP localhost:9633
(LISTEN)
java 15269 root 320u IPv6 2353308 0t0 TCP *:9100
(LISTEN)
java 15269 root 322u IPv6 2352862 0t0 TCP *:9403
(LISTEN)
java 15269 root 323u IPv6 2352863 0t0 TCP *:9402
(LISTEN)
java 15269 root 325u IPv6 2352864 0t0 TCP *:2809
(LISTEN)
java 15269 root 585u IPv6 2350042 0t0 TCP *:9060
(LISTEN)
java 15269 root 586u IPv6 2350043 0t0 TCP *:9080
(LISTEN)
java 15269 root 587u IPv6 2350044 0t0 TCP *:9443
(LISTEN)
java 15269 root 588u IPv6 2352868 0t0 TCP *:9043
(LISTEN)
```

In anderen Worten: Standardmäßig lauscht WebSphere an allen Netzwerkschnittstellen auf jeweils 10 TCP-Ports.

Der Angreifer könnte jetzt anhand der Portnummern in der WebSphere-Dokumentation nachlesen, wozu diese Ports da sind, und sich so Schritt für Schritt einarbeiten. Leider ist so viel Mühe auch gar nicht nötig, denn IBM hat für den WebSphere Application Server ein Administrationsskript namens *wsadmin.sh* entwickelt. Das *wsadmin*-Tool versteht sich sowohl auf die Ausführung von Skripten, die in Jacl oder Jython verfasst wurden, als auch auf die interaktive Ausführung einzelner WebSphere-Befehle. Die Software ist zudem Open Source und lässt sich unter anderem via SourceForge herunterladen, und viele Administratoren tun dies auch und installieren sie (wie auch IBM selbst) gleich mit.

IBM dokumentiert den IBM WebSphere Application Server aktiv weiter und liefert dazu regelmäßig aktuelle Informationen. Obwohl der WebSphere-Exploit unter Open-Source-Experten viel Aufsehen erregt hat, scheint sich IBM nicht allzu sehr daran zu stören, wie nützlich das kleine Tool sein kann.

Um die Arbeitsweise des WebSphere-Exploits zu illustrieren, benötigen Sie als Nächstes die sogenannte Burp Suite. Bei Burp handelt es sich um ein Werkzeug, das es in zwei Fassungen gibt: Burp Suite Free Edition und Burp Suite Professional. Die Free Edition ist kostenlos unter der Adresse <https://portswigger.net/burp/download.html> erhältlich und reicht für das Arbeiten im manuellen Modus vollkommen aus.

Als Erstes bearbeiten Sie die Konfigurationsdatei des *wsadmin*-Tools */opt/IBM/WebSphere/AppServer/profiles/AppSrv01/properties/wsadmin.properties* in einem Texteditor Ihrer Wahl. Wie diese Datei im Ausgangszustand aussieht, können Sie dem [Listing 1](#) entnehmen. Wie Sie diese anpassen, um

WebSphere via *wsadmin* mittels Burp als Proxy anzusprechen, entnehmen Sie dem [Listing 2](#).

Da der WebSphere Application Server seine Daten am Verwaltungsport 8880 erwartet, muss in Burp eine Umlenkung vom Port *:1880 auf 127.0.0.1:8880 vollzogen werden. Mittels der Umlenkung von Port *:1880 auf 127.0.0.1:8880 gelingt es, den Datenverkehr mittels des *wsadmin*-Tools abzufangen, zu verändern und schließlich mit Burp wieder an die gewohnte Adresse zu leiten.

Bevor Burp zu Werke schreiten kann, meldet es sich mit einem Zertifikat zu Wort, welches Sie akzeptieren müssen. Sobald dies geschehen ist, wird die Kommunikation zwischen Burp und dem WebSphere Application Server SSL-verschlüsselt aufgenommen. Sie sollten nun in der Lage sein, die Übertragung roher, serialisierter Java-Objekte zu überwachen.

Im Hexcode-Viewer geben sich diese Objekte an ihrem Präfix zu erkennen; *r00* lässt beispielsweise auf ein base64-verschlüsseltes Objekt schließen. Der Java-Exploit ermöglicht es, die base64-codierten Daten durch beliebige base64-codierte Ersatzdaten auszuwechseln (*ysoserial* hilft dabei) und so etwa eine bösartige Payload abzufeuern.

Die Datenersetzung ließe sich zum Beispiel folgendermaßen umsetzen:

```
~/Desktop$ cat ersatzdaten.out | base64 -w 0 >
ersatzdaten.out.b64
```

Jetzt kommt Burp Repeater zu Ehren, denn dort, wo der base64-codierte Datenbereich ausgeschnitten wurde, lassen sich jetzt die nun ebenfalls base64-codierten Ersatzdaten einsetzen. Hierzu markieren Sie den gewünschten Ausschnitt

Listing 1: Standardport von wsadmin.sh

```
#-----
# The port property determines what port is used
when attempting
# a connection.
# The default SOAP port for a single-server
installation is 8880
#-----
com.ibm.ws.scripting.port=8880
```

Listing 2: Portanpassung in wsadmin.sh

```
#-----
# The port property determines what port is used
when attempting
# a connection.
# The default SOAP port for a single-server
installation is 8880
#-----
com.ibm.ws.scripting.port=1880
```

und fügen an seiner Stelle mittels *Paste from file* den Inhalt der Datei *ersatzdaten.out.b64* ein. Ein Klick auf *Go*, und schon können Sie den Vorgang in der Kommandoshell vor sich hinlaufen sehen.

Um die Risiken Ihrer WebSphere-Applikationen zu minimieren, seien Ihnen folgende Gegenmaßnahmen empfohlen:

- Richten Sie WebSphere in zwei separaten Subnetzen wie folgt ein: Trennen Sie den administrativen Zugang von den WebSphere-Diensten durch eine Aufteilung in zwei physikalisch getrennte Subnetze.
- Tragen Sie dafür Sorge, dass sich der Port der Applikation WebSphere von dem Verwaltungsport unterscheidet.
- Verzichten Sie auf den Einsatz von Standardportnummern.

Manch einer mag jetzt argumentieren, dass WebSphere lediglich eine rare Ausnahme darstelle. Doch leider ist dem bei Weitem nicht so, denn die Liste der möglichen Java-basierten Commons-Collection-Exploits ist so lang, dass man sich bei der Exploit-Besprechung auf ein paar prominente Java-Superstar-Applikationen beschränken muss, um nicht in einer Flut von Exploits förmlich zu ertrinken.

Zudem ist der Exploit der Commons-Collections-Bibliothek in Java ein so schwerwiegendes Problem grundlegender Natur, dass nur ein paar der wenigen Java-Schwergewichte wie der Java-Besitzer Oracle und vielleicht noch IBM hinreichend viel Druck auf die Apache Software Foundation ausüben könnten, um das Problem direkt an der Quelle zu lösen: in der Apache Commons Collection Library.

Allen anderen bleibt nur, sich mehr oder weniger mühsam um die Fehler der Commons-Collection-Bibliothek herumzuhangeln. Für den Jenkins-Integrationsserver hat der Anbieter ein Hotfix-Skript entwickelt, mit dem sich die Sicherheitslücke in dieser speziellen Bereitstellungslösung für Java-Software ausnahmsweise schließen lässt, und stellt es unter der Adresse <https://github.com/jenkinsci-cert/SECURITY-218> bereit. Damit der Workaround über einen Neustart von Jenkins hinaus persistent wirksam bleibt, müssen Sie das Skript in der Umgebungsvariablen `$JENKINS_HOME/init.groovy.d/cli-shutdown.groov` verzeichnen.

Exploit der Commons-Collections-Bibliothek von Java

Um die Verwundbarkeit der Commons-Collections-Bibliothek in Java am Beispiel von Red Hat JBoss zu demonstrieren, greifen wir auf das JMXInvokerServlet zurück, das in vielen JBoss-Installationen nicht abgesichert ist. Genau wie bei dem WebSphere Application Server lässt sich die Verwundbarkeit in JBoss mittels des Burp-Repeater nachweisen. Bei JMX handelt es sich um ein Java-Management-Protokoll, das es erlaubt, über den Verwaltungsport 8080 auf die laufende JBoss-Installation zuzugreifen:

```
http://localhost:8080/invoker/JMXInvokerServlet
```

Wer diesen URL in einem Webbrowser aufruft, bekommt ein rohes, serialisiertes Java-Objekt zugestellt. Dies ist gewöhnlich ein todsicheres Zeichen, dass es sich hier wieder um ei-

ne gravierende Verwundbarkeit handelt, insbesondere wenn der Standardport 8080 aus dem Internet zugänglich sein sollte. Die Verfügbarkeit der Commons-Collection-Bibliothek lässt sich in diesem Fall wie folgt in Erfahrung bringen:

```
/opt/jboss-6.1.0.Final# grep -R InvokerTransformer
```

Die Bibliothek müsste daraufhin in der Ausgabe des Befehls auftauchen, zum Beispiel:

```
Binary file ./server/standard/deployers/jsf.deployer/MyFaces-2.0/jsf-libs/commons-collections-3.2.jar matches
Binary file ./server/all/deployers/jsf.deployer/MyFaces-2.0/jsf-libs/commons-collections-3.2.jar matches
Binary file ./server/default/deployers/jsf.deployer/MyFaces-2.0/jsf-libs/commons-collections-3.2.jar matches
Binary file ./common/deploy/admin-console.war/WEB-INF/lib/commons-collections-3.2.jar matches
Binary file ./common/lib/commons-collections.jar matches
Binary file ./client/commons-collections.jar matches
```

Da das JMXInvokerServlet auf demselben Port wie die Applikation JBoss läuft, genießt es keinen Schutz durch die Firewall.

Jetzt geht's wieder ans Eingemachte. Da die betreffende Anfrage nur aus einem einzigen Objekt besteht, erstellen Sie den POST-Request mit einem gültigen HTTP-Header in Burp und benutzen nun *Paste from file*, um die mittels *ysoserial* erzeugten Binärdaten der Java-Payload einzubinden. So lässt sich mit einem einzigen böartigen Datenpaket Apaches Commons-Collections-Bibliothek kompromittieren.

Exploit der fehlerhaften Deserialisierung

Ein weiteres Beispiel für eine Art und Weise, wie die verwundbare Commons-Collection-Bibliothek ausgenutzt werden kann, bietet die E-Commerce-Transaktionsplattform WebLogic. Mit einem Webdienst wie zum Beispiel Builtwith.com finden Hacker einen WebLogic-basierten Webserver leider viel schneller, als es Ihnen lieb sein kann. Der Exploit von WebLogic macht sich die Verfügbarkeit der Commons-Collections-Bibliothek in Java zunutze, und darüber hinaus auch den Umstand, dass sich die WebLogic-Entwickler anscheinend redlich bemüht haben, den potenziellen Hackern so weit wie möglich den Weg zu ebnen.

Die Standardinstallation von WebLogic lauscht am Port 7001. Es ist der standardmäßige und auch der einzige Port, den WebLogic für die sämtliche Kommunikation mit der Außenwelt nutzt, was ja die Wirksamkeit einer externen Firewall zunichte macht. Sie könnten meinen, dass so viel geballte Inkompetenz gar nicht möglich sei, doch WebLogic belehrt Sie schnell eines Besseren, denn alle Anfragen, zum Beispiel HTTP, SNMP, T3 und einige mehr, gehen alle durch den einen einzigen Standardport, und praktisch immer handelt es sich dabei um den Standardport 7001. Intern werden die Daten dann erkannt und weitergeroutet.

Da die WebLogic-Applikation für alle unterstützten Protokolle nur einen einzigen Port vorgesehen hat, können Sie ►

den WebLogic-Server nicht ohne Weiteres durch eine Firewall vor Angriffen von außen schützen, denn sobald Sie dies versuchen sollten, kommt jegliche Kommunikation zum Erliegen.

Sie wollten nun sicherlich prüfen, ob WebLogic die Java-Bibliothek *commons-collections.jar* nutzt und dadurch verwundbar ist. Doch sollten Sie sich dabei auf eine Suche nach dem Namen der Bibliothek beschränken, würden Sie wohl kaum fündig werden. Offenbar hat es sich bei WebLogic schon herumgesprochen, dass jemand danach suchen könnte. Leider stellt bloßes Umbenennen einer Java-Bibliothek nur einen rein illusorischen Schutz gegen Hackerangriffe dar. Sie müssen lediglich Folgendes in Ihre Kommandozeile eingeben, um im Bilde zu sein:

```
/opt/OracleHome# grep -R InvokerTransformer
```

Als Resultat erhalten Sie die folgende Ausgabe:

```
Binary file ./oracle_common/modules/com.bea.core.apache.commons.collections.jar matches
```

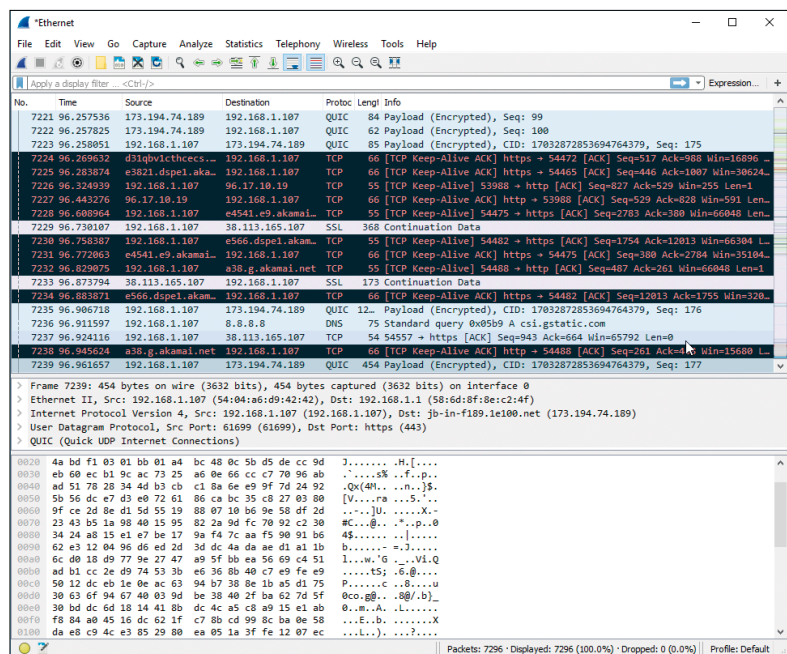
Konkret bedeutet es für Sie, dass Sie sich im Fall von WebLogic nicht darauf verlassen können, dass potenzielle Hacker unverrichteter Dinge wieder abdrehen, denn bei der Datei *com.bea.core.apache.commons.collections.jar* handelt es sich um eine ganz normale Commons-Collections-Java-Bibliothek, die lediglich verschleiert wurde. Der Abschreckungsfaktor für potenzielle Hacker ist damit nicht gegeben. Stattdessen stellt WebLogic einen besonders klassischen Fall dar, der Hacker geradezu auf den Plan ruft: Alle Anfragen werden an einem einzigen Port abgearbeitet. Mit dem Wechseln der Portnummer ist Ihnen auch nicht geholfen, denn mit *lsof* können sich potenzielle Hacker in Windeseile Gewissheit verschaffen, auf welchen Ports die betreffende Java-Applikation läuft:

```
lsof -i -P | grep LISTEN | grep java
```

Ein potenzieller Hacker wird dann versuchen, einen Angriffspunkt in WebLogic zu finden, um die frisch serialisierten Payload-Daten wie ein Kuckucksei unterzuschmuggeln. Möchten Sie in Echtzeit die Datenpakete beobachten können, die durch die verschiedenen Schnittstellen durchlaufen, sei Ihnen das kostenfreie Wireshark 2 von der Wireshark Foundation empfohlen (Bild 1). Mittels Wireshark können Sie den Netztraffic im Blick behalten, und dann geben Sie einen Befehl ein, um den WebLogic-Server gezielt zu stoppen:

```
/opt/OracleHome/user_projects/domains/base_domain/bin#  
./stopWebLogic.sh
```

Zum Stoppen des WebLogic Servers müssen Sie sich als Administrator authentifizieren. Wenn Sie sich jetzt bewusst einmal beim Passwort vertippen, dann wird die Authentifizierung natürlich fehlschlagen.



Wireshark kann beliebige Protokolle überwachen und anzeigen, sodass Sie alle Datenpakete auf Herz und Nieren prüfen können (Bild 1)

In Wireshark können Sie die fehlgeschlagene Authentifizierung an ein paar sogenannten magischen Bytes *ac ed 00 05* zu erkennen. Ähnlich wie zuvor werden mögliche Angreifer versuchen, die Payload so geschickt zu tarnen, dass es nicht weiter auffällt. Anders als beim JBoss- oder Jenkins-Exploit kommen Angreifer beim WebLogic-Exploit mit der Base64-Codierung nicht ans Ziel. Mit dem T3-Protokoll kommt man jedoch problemlos ans Ziel.

Sicherheit trifft Performance: HTTP/2

Das Transportprotokoll HTTP/1.1 ist das heutige Herzstück des Internets: Es definiert, wie Server und Clients miteinander kommunizieren. HTTP/1.1 datiert bis ins Jahr 1999 zurück und hat sich seitdem praktisch kaum geändert. Nahezu zwei Jahrzehnte lange herrschte Stagnation nicht nur im Hinblick auf die Performance, sondern auch im Hinblick auf die Sicherheit des Protokolls. Jetzt steht der Nachfolger von HTTP/1.1 vor der Tür: HTTP/2 mit konzeptionellen Anleihen bei Googles SPDY.

HTTP/2 soll das Web sowohl sicherer als auch schneller machen. Da HTTP/1.1 prinzipiell nicht über die Fähigkeit verfügt, die Datenübertragung effizient zu koordinieren, leidet es unter akuten Performance-Engpässen. Diese machen sich insbesondere bei SSL-Verbindungen schmerzhaft bemerkbar (denn der Overhead ist hier noch höher als sonst). Dies hat wiederum dazu geführt, dass HTTPS bisher sehr ungenutz und auch dann meist mit veralteten Verschlüsselungsmethoden genutzt wird.

Das Resultat dieser Kompromisse waren zahlreiche schlagzeilenreife Sicherheitspannen bei prominenten Anbietern von Webdiensten und Webshops. Wenn es nach den Browserherstellern geht, soll damit jetzt Schluss sein. Während bei HTTP/1.1 verschlüsselte Datenverbindungen optional sind,

gibt es bei HTTP/2 erstmals keine Wahl: In HTTP/2 ist HTTPS erstmals Pflicht. Als Trostpflaster bekommen Sie zu der verbesserten Sicherheit einen Turbobooster.

Die zusätzliche Belastung durch den Einsatz von SSL machte sich bei HTTP/1.1 aufgrund der zahlreichen Engpässe des Protokolls deutlich bemerkbar. In HTTP/2 konnte das Problem dann zumindest zum Teil behoben werden. Beim Umstieg von HTTP/1.1 in der Variante mit SSL auf HTTP/2 (mit SSL) können die meisten Webseiten einen merklichen Performanceschub fest verbuchen (**Bild 2**). Problematisch wird es bei der Umstellung einer Webseite von HTTP/1.1 ohne SSL auf HTTP/2 (mit SSL).

In vielen Fällen müssen Sie mit einem messbaren, wenn auch geringen Performanceverlust rechnen, da die Verschlüsselung etwas Rechenzeit kostet. Im besten Fall lässt sich der zusätzliche Rechenaufwand im Zusammenhang mit SSL durch die effizientere Ressourcennutzung von HTTP/2 kompensieren (statt einer Verbindung pro Webserver baut der Client jetzt je eine Verbindung pro Datei auf).

Als grobe Faustregel ist diese Feststellung bezüglich der Performance in den meisten Fällen zutreffend. Trotzdem empfiehlt es sich, die Kombo aus HTTP/2 und dem eigenen Webserver für die eigenen Bedürfnisse auf die Probe zu stellen. Sollte der Test unter einer vergleichbaren Belastung die gewünschten Resultate ergeben, dann kann man ruhigen Gewissens die betroffenen Webserver auch in der Produktionsumgebung umstellen (siehe Kasten »HTTP/2-Unterstützung in aktuellen Webservern«).

Multiplexing, Nebenläufigkeit, Sicherheit

Da das Problem mit den stetig wachsenden Latenzzeiten und langsameren Websites im Lauf der Zeit an einer gewissen Dringlichkeit gewann, hatte Google zwischendurch das eigene SPDY-Protokoll als technischen Nachfolger von HTTP/1.1 gefördert. Doch Stream-Kompression (zum Beispiel mit Gzip) beim SPDY-Protokoll zeigte sich als anfällig für die berüchtigte CRIME-Attacke (Compression Ratio Infoleak Made Easy). CRIME hat das Ende von SPDY besiegelt: Ein komplett neuer Ansatz war vonnöten. Google hat sein ganzes Gewicht hinter HTTP/2 geworfen und plant sogar, das hauseigene

HTTP/2-Unterstützung in aktuellen Webservern

HTTP/2-Unterstützung bieten bereits unter anderem die folgenden Webserver:

- **Akamai Edge Server**
- **Apache** (ab der Version 2.4.17)
- **CDN77**, ein Content Delivery Network (aber nur mit NGINX)
- **Citrix NetScaler**
- **Microsoft IIS** (ab Windows Server 2016 und Windows 10)
- **NGINX** (ab Version 1.9.5)

Vorteile von HTTP/2

Mit dem Einsatz von HTTP/2 ergeben sich für Website-Betreiber insbesondere die folgenden fünf Vorteile:

- eine einzige, persistente Datenverbindung pro Webseite und Benutzersitzung;
- Multiplexing mit Priorisierung von Anfragen und Antworten;
- verbesserte Sicherheit dank TLS (statt SSL v3);
- deutlich höhere TLS-Performance gegenüber HTTP/1.1;
- Wegfall verwirrender HTTP/1.1-Optimierungen bei Webseiten und HTML5-basierten Apps
- HPACK-Datenkompression von HTTP/2 verhindert CRIME-Angriffe auf Webanwendungen.

SPDY-Protokoll – in Google Chrome bereits ab Januar 2016 – durch den offiziellen Nachfolger HTTP/2 abzulösen.

HTTP/2 lässt mehrere Objekte im Multiplex-Verfahren in einem einzigen Datenstream ineinanderfließen, um sie über eine einzige Verbindung pro Webseite an den Client zu übertragen. HTTP/2 benötigt daher ebenfalls nur einen – anstatt bisher zahlloser – TLS-Handshakes (siehe Kasten »Vorteile von HTTP/2«). Auf diese Weise entfällt der bisher mit HTTP/1.1 unvermeidliche enorme Overhead durch den wiederholten Aufbau einzelner Verbindungen ohne eine eindeutige Zuordnung zur jeweiligen Anfrage (das Head-of-Line-Blocking-Problem). Dies fällt insbesondere beim Verbindungsaufbau von verschlüsselten SSL/TLS-Verbindungen ins Gewicht.

Dank des Multiplex-Verfahrens und einer intelligenten Kompression mit dem HPACK-Algorithmus (Header Compression für HTTP/2) kann das Protokoll die Übertragungsleistung einer einzigen Verbindung maximieren. Zudem kann der Webserver bestimmte Inhalte einer Webseite – also solche, auf die besonders oft zugegriffen wird – priorisieren, sodass die Webseite im Browser möglichst schnell gerendert und angezeigt werden kann.

Mit HTTP/2 entfallen endlich auch die zahllosen (und nun nutzlosen) HTTP/1.1-Optimierungen, die vor allem für Entwickler Web-Apps unnötig kom-



Testen möglicher Performance-Änderungen bei der Umstellung einer Webseite von HTTP/1.1 auf HTTP/2 (<https://loadimpact.com>) (**Bild 2**)

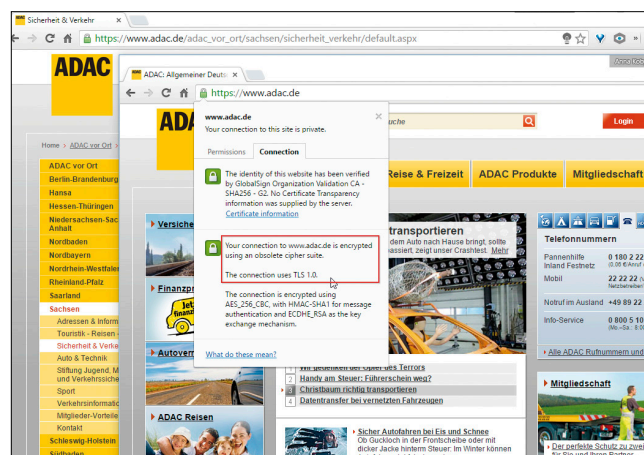
pliziert gemacht haben (siehe Kasten »HTTP/2 – störende Website-Optimierungen entschärfen«).

Perfect Forward Secrecy in HTTP/2 mit TLS statt SSL

HTTP/2 erfordert an sich keine Datenverschlüsselung, aber alle Implementierungen in den führenden Webbrowsern (Firefox, Chrome, Edge et cetera) sollen nur verschlüsselte HTTP/2-Verbindungen unterstützen. Ob sich das automatisch in einer ausreichenden Sicherheit niederschlägt, hängt von Ihrer konkreten Implementierung ab: der Verschlüsselungsebene, der Cipher-Suite und Ihrem Zertifikat. Die Krux besteht darin, dass jedes Detail der Implementierung zählt. Viele Entwickler wiegen sich bei SSL 3 in missverständlicher Sicherheit.

SSL 3 (ob mit HTTP/1.1 oder HTTP/2) kann die Datenübertragung einer Webanwendung oder einer mobilen App nicht zuverlässig absichern, weil diese Verschlüsselungsebene bereits mehrmals kompromittiert wurde. Auch TLS 1.0 gilt als nicht sicher genug. Die niedrigste gerade noch akzeptable Verschlüsselungsebene ist TLS 1.1 (Bild 3). Wer auf TLS 1.2 setzt, fährt am besten (Bild 4). Die scheinbar hohe Versionsnummer 3.0 bei SSL stiftet da nur unnötig Verwirrung.

Verbindet sich ein Client mit einem Webserver über eine verschlüsselte Verbindung, also HTTPS, so entscheidet sich in den ersten Millisekunden, wie sicher die verschlüsselte Verbindung wirklich sein wird. Wenn der Webserver HTTPS anbietet, aber der Webbrowser die verwendete Verschlüsselungsmethode nicht versteht, entscheidet sich also, ob dieser Client sozusagen mitspielt. Der Webbrowser liefert dem Webserver eine Liste der bevorzugten, also unterstützten Verschlüsselungsoptionen (der sogenannten Cipher-Suites). Daraufhin antwortet der Server mit einem Überprüfungszertifikat und wählt eine vom Browser unterstützte Cipher-Suite aus. Ist eine sichere Verbindung gewährleistet,



TLS v.1.0 ist Webbrowsern nicht sicher genug. TLS ist zurzeit nur in der Version 1.2 sicher (<https://www.adac.de>) (Bild 3)

spricht man von perfekt fortgesetzter Geheimhaltung (auch als Perfect Forward Secrecy bekannt). Leider ist dies nicht immer der Fall.

Veraltetes SSL 3.0

SSL 3.0 wurde mehrmals gehackt, gilt als veraltet und sollte niemals zum Einsatz kommen. Seit der POODLE-Angriffe sind sogar manche Behörden aufgewacht. Das US-amerikanische Wirtschaftsministerium (US Department of Commerce), das ja nicht gerade durch zeitnahe Handlungsbereitschaft auf sich aufmerksam macht, hat inzwischen auf Empfehlung des NIST (National Institute of Standards and Technology) hin den Einsatz der SSL-Verschlüsselung in der Version 3.0 als grobe IT-technische Fahrlässigkeit eingestuft und rät US-Unternehmen davon ausdrücklich ab.

POODLE (Padding Oracle On Downgraded Legacy Encryption) bezeichnet eine Man-in-the-middle-Angriffe. Es

HTTP/2 – störende Website-Optimierungen entschärfen

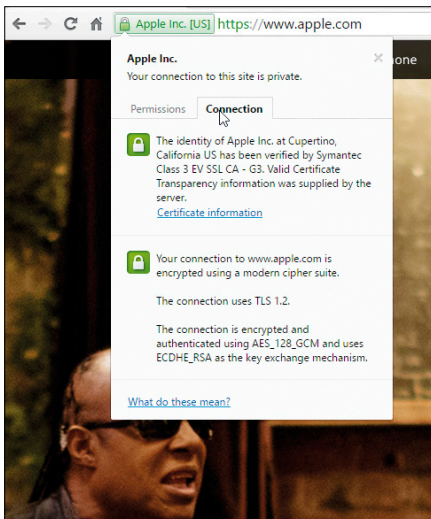
Mit dem Umstieg auf HTTP/2 werden die folgenden Optimierungen entweder ganz überflüssig oder müssen angepasst werden:

- **Domain Sharding**, also das Bereitstellen von Webinhalten über mehrere Subdomains, um die Anzahl der zulässigen HTTP-Verbindungen zu maximieren;
- **Sprites**, das Zusammenführen von kleinen Icons zu größeren Bildern zum Minimieren des Overheads;
- **Dateiverkettung**: Zusammenführen von CSS- und JavaScript-Dateien zur Minimierung der Anzahl von HTTP-Anfragen;
- **Inlining**, das Einbinden des Inhalts von CSS- und JavaScript-Dateien, manchmal sogar Bildern, in das HTML-Dokument zur Minimierung der Anzahl von HTTP-Verbindungen (eine Methode nicht ohne ihre eigenen Nachteile, denn sie konnte den Seitenaufbau verlangsamen);
- **Cookie-lose Domains**: Da statische Ressourcen wie Bilder, CSS-

und JavaScript-Dateien keine Cookies erfordern, gingen viele Entwickler dazu über, diese Objekte aus Cookie-losen Domains zu servieren, um die Bandbreite zu schonen; seit HTTP/2 werden alle Header, einschließlich Cookies, gemultiplext und effizient komprimiert, sodass die Anfragen im Vergleich zu HTTP/1.1 sehr klein ausfallen und sich somit auch diese Optimierung erübrigt.

Falls Sie für HTTP/1.1-Verbindungen Domain Sharding beibehalten möchten, ohne die HTTP/2-Performance zu beeinträchtigen, stellen Sie sicher, dass sich alle verwendeten Sharding-Domains in dieselbe öffentliche IP-Adresse auflösen und mit einem einzigen Wildcard-Zertifikat abgedeckt sind, damit HTTP/2 eine gemultiplexte Verbindung aufbauen kann. So nutzten Sie die volle Funktionsvielfalt von HTTP/2 und das effiziente Caching des Protokolls.

Anstelle der Dateiverkettung in HTTP/1.1 empfiehlt sich JavaScript-Packaging in HTTP/2.



Apple präsentiert sich von der besten Seite:

Apple nutzt ein EV-Zertifikat mit Zertifikatstransparenzinformatio-
nen, kommuniziert
via TLS 1.2 und
nutzt die ECDHE-
Cipher-Suite
(Bild 4)

handelt sich dabei um eine SSL 3.0-Verwundbarkeit gemäß CVE-2014-3566. Hierbei macht sich der Angreifer das viel zu kooperative Verhalten des Webserver beim Einsatz von TLS zunutze. Der Webserver versucht nämlich, einen fehlgeschlagenen TLS-Handshake dadurch auszugleichen, dass er dem Client SSL 3.0 anbietet und der POODLE-Sicherheitslücke Tür und Tor öffnet.

Dieser Angriff wird auch Version-Rollback-Attacke genannt. Die bisher einzige Lösung besteht darin, das Herunterschalten auf SSL abzublocken (siehe Kasten »SSL-v3 zugunsten von TLS-v1.x abschalten«).

Leider werden frühe Warnungen meist einfach in den Wind geschlagen statt als Wegweiser zum Absichern der Systeme genutzt zu werden.

Die unterschätzte Gefahr: SHA1-Kollisionsattacken

HTTP/2 mit TLS gehört zwar ab sofort auf die Zu-Erledigen-Liste einer jeden IT-Abteilung, aber damit erschöpft sich noch bei Weitem nicht die Liste erforderlicher Sicherheitsmaßnahmen. Sofortiger Handlungsbedarf besteht nicht zuletzt auch im Hinblick auf die Qualität der verwendeten Zertifikate.

In 2016 dürften unternehmungslustige Hacker vermehrt auf die Jagd gehen und versuchen, sich die ausgiebig dokumentierte Sicherheitslücke, welche der Einsatz von SHA1-Zertifikaten darstellt, zunutze zu machen. SHA1-verschlüsselte Kommunikation lässt sich mit Hilfe von Brute-Force-Attacken auf den SHA-Schlüssel aushebeln (Beispiel: die Flame-Malware und MD5).

Hierbei errechnet der Angreifer eine sogenannte Kollision, die ihm erlaubt, ein gefälschtes Zertifikat auszustellen, mit dem sich die Daten – ob Passwörter, Kreditkartendaten oder sonstige Informationen – decodieren lassen. Angreifer können sich unter Umständen sogar in das Backend solch unsicherer Webanwendungen hineinhacken.

Die führenden Browser-Hersteller haben auch bereits angekündigt, die inzwischen leicht hackbaren SHA1-Zertifikate ab 2016 als nicht vertrauenswürdig zu kennzeichnen. Viele Website-Betreiber konnten ihre SHA1-Zertifikate bereits

SSL-v3 zugunsten von TLS-v1.x abschalten

Um sicherzustellen, dass das höchst unsichere SSL-v3-Protokoll auf Ihrem Server nicht mehr zum Einsatz kommt, müssen Sie SSL-v3 als Verschlüsselungsoption im Server-Backend abschalten.

In den gängigsten Webservern Apache, Microsoft IIS und NGINX ist dies zum Glück ein Leichtes.

Bei Apache müssen Sie die folgende Zeile in die betreffende Konfigurationsdatei einfügen:

```
SSLProtocol All -SSLv2 -SSLv3
```

Alternativ müssen Sie jede Zeile, die in den zahlreichen Konfigurationsdateien mit `SSLProtocol` beginnt, entsprechend abändern. Danach speichern Sie die betroffenen Dateien ab und starten Sie Apache neu, zum Beispiel mittels:

```
sudo apache2ctl configtest && sudo service apache2 restart
```

Im Fall von IIS fügen Sie die folgenden Zeilen in die Datei `disable_ssl3.reg` ein:

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\
Control\SecurityProviders\SCHANNEL\Protocols\
SSL 3.0\Server]
"Enabled"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\
Control\SecurityProviders\SCHANNEL\Protocols\SSL
2.0\Server]
"Enabled"=dword:00000000
```

Bei NGINX öffnen Sie die betreffende Konfigurationsdatei in einem Texteditor Ihrer Wahl und ersetzen Sie alle Zeilen, die mit `ssl_protocols` anfangen, wie folgt:

```
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
```

Speichern Sie die neue Konfiguration ab und starten Sie mittels

```
sudo service nginx restart
```

NGINX neu, um die geänderte Konfiguration einzulesen.

durch zeitgemäße SHA2-Zertifikate austauschen. Wem diese Aufgabe noch bevorstehen sollte, der ist gut beraten, sich ihr umgehend und mit höchster Priorität anzunehmen.

Bei seriösen Zertifikatausstellern ist der Umtausch übrigens kostenlos, solange die Gültigkeit des ursprünglichen Zertifikats noch nicht aufgebraucht ist und Sie über den zugehörigen privaten Schlüssel verfügen. ►

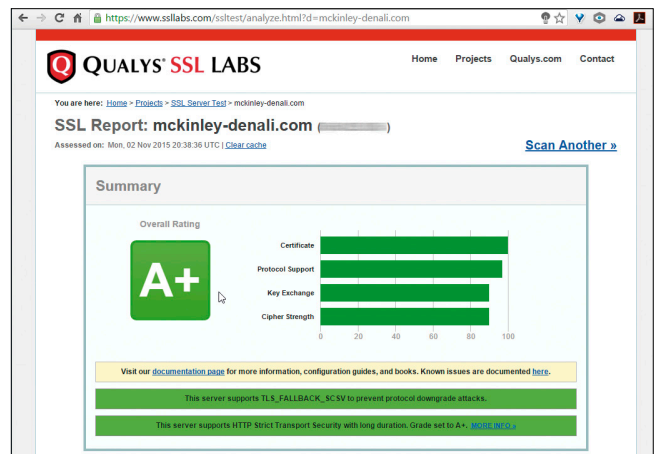
Ab dem Stichtag 1. Januar 2016 dürften auch die letzten maroden Zertifikatsstellen (CAs) endlich aufhören, unsichere SHA1-Zertifikate auszustellen, und ihren Kunden nur noch die deutlich sicheren SHA2-Zertifikate anbieten. Die Zertifikate auf dem Server aktualisieren sich jedoch nicht von selbst; ein Tausch durch den Administrator ist hierzu erforderlich.

In der Praxis hat sich das Thema unsicherer Zertifikate mit dem Austausch auf SHA-2 nicht erledigt. Ganz im Gegenteil: Das ganze Konzept der Ausstellung und Beglaubigung der Gültigkeit von SSL-Zertifikaten durch Zertifikatsbehörden (CAs) hat sich in der Praxis der vergangenen Jahre als äußerst fehlerträchtig erwiesen. Nichts deutet derzeit darauf hin, dass sich eine Besserung abzeichnen könnte.

CA-Zertifikatsbehörden

CA-Zertifikatsbehörden unterliegen selbst praktisch keinerlei Sicherheitsvorschriften. Unter den CA-Zertifikatsbehörden finden sich zum Teil tyrannische Regierungen, welche entweder eigene Root-Zertifikate ausstellen oder sich in den Zertifikatsketten regulärer CAs durch Intermediate-Zertifikate einnisten, um den Datenverkehr anderer Nationen entschlüsseln zu können. Beim Einsatz solcher Zertifikate gibt es zwar einen Anschein von Verschlüsselung, aber das war es dann auch schon.

Einige Sicherheitsexperten, darunter Matthew Green von der Johns Hopkins-Universität im US-Bundesstaat Maryland,



Qualys SSL Tester: Das Test-Tool im Einsatz (Bild 5)

vertreten sogar die Meinung, dass die vielen und recht gravierenden konzeptionellen Fehler der frühen Versionen des SSL-Protokolls beabsichtigt gewesen sein müssten. Da hilft es nur, auf die neueste TLS-basierte Verschlüsselung aufzurüsten.

Doch so einfach ist es nun auch nicht, denn die Webserver wurden so entworfen, dass sie sich von außen her auf eine niedrigere Sicherheitsstufe zwangsweise zurückschalten lassen, wie es unter anderem die (immer noch relevante) FREAK-Attacke unter Beweis stellte.

Listing 3: SSL-Konfiguration für NGINX 1.9.5

```
server {
listen 443 ssl;

# Konfiguration der Zertifikate
ssl_certificate /pfad/zum/
signierten_Zertifikat_mit_den_Intermediates;
ssl_certificate_key /pfad/zum/privaten_schluessel;
ssl_session_timeout 1d;
ssl_session_cache shared:SSL:50m;
ssl_session_tickets off;

# Diffie-Hellman-Parameter für die DHE-Ciphersuites,
# empfohlen wird mindestens 2048 Bits
ssl_dhparam /pfad/zum/dhparam-Zertifikat.pem;

# modernste unterstützte Protokolle; Konfiguration
# zum Anpassen an eigene Bedürfnisse
ssl_protocols TLSv1.1 TLSv1.2;
ssl_ciphers 'ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-
ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:
DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-
AES128-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-
AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-
SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-
```

```

SHA:ECDSA-AES256-SHA:DHE-RSA-AES128-SHA256:
DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-
AES256-SHA256:DHE-DSS-AES256-SHA:DHE-RSA-AES256-SHA:!a
NULL:!eNULL:!EXPORT:!DES:!RC4:!3DES:!MD5:!PSK';
ssl_prefer_server_ciphers on;

# HSTS zum Schutz vor einem Downgrade und
# Cookie-Hijacking
# (erfordert das Modul ngx_http_headers_module)
# (15768000 Sekunden = 6 Monate)
add_header Strict-Transport-Security max-age=15768000;

# OCSP-Stapling
ssl_stapling on;
ssl_stapling_verify on;

# Vertrauensketten der OCSP-Antworten mit
# Hilfe des Root-CA
# und der Intermediate-Zertifikate überprüfen
ssl_trusted_certificate /pfad/zum/
root_CA_Zertifikat_mit_Intermediates;
resolver <IP-Adressen_von_DNS-Resolvern>;

# die übrige Konfiguration
}

```


Tipp: Qualys SSL Labs bietet Ihnen unter der Adresse <https://www.ssllabs.com/ssltest> einen kostenfreien Webdienst, mit dem Sie Ihren Webserver auf Herz und Nieren prüfen und konkrete Hinweise bezüglich möglicher Verbesserungen erhalten können (Bild 5).

Microsoft Research, IMDEA (eine Forschungsgruppe aus der Europäischen Union) und INRIA (ein französisches IT-Unternehmen) haben herausgefunden, dass immer noch Millionen von Webbrowsers weltweit gezwungen werden können, auf eine schwache Verschlüsselung herunterzuschalten. Diese ließe sich innerhalb von wenigen Stunden mittels EC2-Instanzen auf AWS aushebeln.

Factoring attack on RSA-EXPORT Key

Diese Verwundbarkeit wurde auf den Namen FREAK getauft (Factoring attack on RSA-EXPORT Key) und bedrohte unter anderem Webseiten des Weißen Hauses, des Kreditkarteninstituts American Express, der NSA und des FBI. Das Problem, das FREAK zugrunde liegt, wurde inzwischen in OpenSSL 1.0.2 behoben. Dennoch ist nach wie vor eine von zehn verschlüsselten Webseiten betroffen, darunter so hoch frequentierte Portale wie ZDnet.com.

Um das Herunterschalten der Verschlüsselungsstufe (und somit die FREAK-Angriffe) zu verhindern muss man lediglich die Konfiguration des Webserver anpassen. Die erforderlichen Anpassungen für NGINX 1.9.5 mit OpenSSL 1.0.2d (für eine einzelne Webseite) illustriert Listing 3, und im Fall von Apache 2.4.17 mit OpenSSL 1.0.2d können Sie die Anpassungen dem Listing 4 entnehmen. In beiden Fällen funktioniert die betroffene Webanwendung dann in Firefox ab der Version 27, in Chrome ab 22, in IE ab 11 und Edge, in Opera ab

Version 14, Safari ab Version 7, Android ab der Version 4.4 und in Java 8 oder höher.

XSS und XSRF

Cross-Site Scripting (XSS oder auch CSS) ist eine stark verbreitete Angriffsmethode, bei der eine Webapplikation böartigen Code entgegennimmt, der einer anderen Domain entstammt. An einem XSS-Angriff ist meistens ein vollkommen ahnungsloser Webbesucher beteiligt, bei dem es sich gar nicht um einen Angreifer handelt. Der beteiligte Benutzer versteht die Auswirkungen seiner Aktivitäten auf Ihre Webapplikation nicht und hat auch nichts Böses im Sinne.

Der betroffene Webbesucher mag eine Verknüpfung in einer E-Mail-Nachricht, in sozialen Netzen oder in einem Diskussionsforum angeklickt haben, oder hat einfach nur eine verseuchte Webseite gelesen, die einen Aufruf der Zieladresse Ihres Webserver auslöst.

Die betroffene Webapplikation, die den böartigen Code empfangen hat, muss korrekt entscheiden, wie sie auf diesen Code reagiert. Wenn sie ihn verarbeitet, bekommt der betroffene Benutzer eine Webseite zu sehen, die den Anschein von Legitimität erweckt, als ob alles mit Ihrem Webserver vollkommen in Ordnung sei. Dabei führt der Webserver nun gerade böartigen Code aus, indem er zum Beispiel unberechtigt Cookies des betroffenen Besuchers einsammelt und an den Angreifer sendet.

Bösartiges Verhalten einer von XSS betroffenen Website kann unter anderem dazu führen, dass der Angreifer Zugriff auf das Benutzerkonto des betroffenen Besuchers erhält. Der Kreativität der Angreifer sind an dieser Stelle kaum Grenzen gesetzt. ►

Listing 4. SSL-Konfiguration für Apache 2.4.17 mit OpenSSL 1.0.2d

```
<VirtualHost *:443>
...
SSLEngine on
SSLCertificateFile /pfad/zum/signierten_Zertifikat_
mit_den_Intermediates
SSLCertificateKeyFile /Pfad/zum/privaten/Schluesseel
SSLCACertificateFile /Pfad/zu/all_ca_certs

# modernste unterstützte Protokolle; Konfiguration
# zum Anpassen an eigene Bedürfnisse
SSLProtocol all -SSLv3 -TLSv1

SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-
SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-
GCM-SHA256:DHE-DSS-AES128-GCM-
SHA256:KEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-
AES128-SHA256:ECDHE-RSA-AES128-SHA:ECDSA-
AES128-SHA:ECDSA-AES256-SHA384:ECDSA-AES256-
SHA384:ECDSA-AES256-SHA:ECDSA-AES256-
SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-
AES128-SHA256:DHE-RSA-AES256-SHA256:DSS-AES256-
SHA:DHE-RSA-AES256-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4
:!3DES:!MD5:!PSK
SSLHonorCipherOrder on
SSLSessionTickets off
SSLCompression off

# OCSP-Stapling, erst ab httpd 2.3.3 und neuer
SSLUseStapling on
SSLStaplingResponderTimeout 5
SSLStaplingReturnResponderErrors off
SSLStaplingCache shmcb:/var/run/ocsp(128000)

# HSTS (erfordert das Modul mod_headers)
# (15768000 Sekunden = 6 Monate)
Header always set Strict-Transport-Security
"max-age=15768000"
...
</VirtualHost>
SSLStaplingCache shmcb:/var/run/ocsp(128000)
```

Die Wahrscheinlichkeit, dass Ihre Webanwendung betroffen ist, könnte höher sein, als Sie denken. Viele Blogs, Wikis, CMS-Systeme und Plug-ins schleppen mit sich seit Jahren Code herum, der in der Vergangenheit XSS-Angriffen ermöglichte. Zu den am stärksten verwundbaren CMS-Systemen in dieser Kategorie zählte historisch PHPNuke.

Cross-Site Request Forgery

Wenn sich ein böswilliger Angreifer als einer der legitimen (bereits authentifizierten) Benutzer Ihrer Webseite ausgibt, haben Sie es mit einer Cross-Site Request Forgery zu tun (kurz: XSRF oder CSRF). Das Schlimme daran: Der Angreifer kann in dem System eine Menge Unheil anrichten, doch gehen diese Tätigkeiten allesamt zu Lasten eines (völlig ahnungslosen) legitimen Benutzers.

Der Websitebetreiber geht dann möglicherweise hin und sperrt dem vermeintlichen Täter sein Benutzerkonto, da dieser seine Unschuld nicht beweisen kann, doch das Problem hört nicht auf. Der Angreifer kam ja ohne Schaden davon, und so stehen Ihnen weitere Vorfälle noch bevor, bis Sie den Fehler in der Webanwendung finden und beheben...

Im Rahmen einer CSRF-Angriffe kann ein Angreifer beispielsweise die Kontaktdaten des Benutzers auslesen (Gmail), bösartige Malware in Internetforen posten, bei Online-Auktionen mitbieten, Bestellungen an die falsche Adresse schi-

cken (Netflix), Geldüberweisungen auslösen (ING Direkt), oder Spam versenden, alles im Namen des Betroffenen und mit gültiger Authentifizierung durch Ihre Webapplikation.

Im Zentrum einer CSRF(XSRF)-Angriffe steht üblicherweise ein HTML-``-Tag oder ein JavaScript-`Image()`-Objekt. Der bösartige Code taucht in Spam-Nachrichten und im Quelltext von Webseiten auf. Sobald ein Benutzer eine solche Spam-Nachricht in seinem E-Mail-Client liest oder sich auf eine verseuchte Webseite verirrt, wird dieser Code ausgeführt und ruft eine bestimmte URL-Adresse einer Webapplikation auf, um eine bestimmte Aktion auszulösen. Wenn sich in dem Webbrowser des Benutzers ein Authentifizierungs-cookie einer offenen Session einfindet, wird die Webseite der gefälschten Anfrage vertrauen und die betreffende Aktion im Auftrag des authentifizierten Benutzers ausführen.

Inwieweit eine bestimmte Webseite betroffen ist, können Sie schnell in Erfahrung bringen. Wenn Ihre Webseite Benutzer anmeldet und dabei bestimmte Funktionen über den Aufruf eines statischen URL oder einer sich nicht ändernden POST-Anfrage oder gar einer GET-Anfrage durchführen lässt, ist sie verwundbar. Eine Webapplikation müsste jedes Mal prüfen, ob eine bestimmte Anfrage tatsächlich durch den authentifizierten Benutzer ausgelöst wurde, dem die Applikation vertraut.

Tipp: Wenn Sie nicht sicher sind, ob Ihre Webapplikationen von dem XSS- beziehungsweise XSRF-Problem oder anderen Verwundbarkeiten wie SQL-Injektionen betroffen sind, können Sie mit einer Applikation wie Paros (<http://parosproxy.org>), einem HTTP/HTTPS-Proxy, einen Test durchführen. Dieser Test ist eigentlich ganz einfach.

Im ersten Schritt navigieren Sie zu Ihrer Webseite und führen Sie einige typische Aktivitäten aus, während Paros alle Anfragen und Zugriffe zwischen Ihrem Client und dem Server mitschneidet. Lassen Sie dann etwas Zeit vergehen und wiederholen Sie den Test, indem Sie genau dieselben Schritte befolgen und wieder die gesamte Client-Server-Kommunikation durch Paros aufzeichnen lassen. Wenn Sie dieselbe Aufgabe durchführen und dabei genau dieselben GET- oder POST-Anfragen erfolgen, müssen Sie damit rechnen, dass Ihre Webseite gegen CSRF nicht immun ist.

Sie können auch Webseiten anderer Anbieter, auf denen Dienste Sie vertrauen, auf ihre Verwundbarkeit durch XSRF hin überprüfen.

Jede Anfrage mit einem Token versehen

Als Erstes sollten Sie sogenannte Tokens implementieren. Jede Anfrage sollte mit einem Token versehen werden, der an die Sitzung gekoppelt ist und eine beschränkte Gültigkeit besitzt.

Ein einfacher, statischer Token erfüllt den Zweck nicht, denn sollte es einem Angreifer gelingen, den Token abzufangen, würde der ganze Sinn und Zweck zunichte gemacht.

Auch korrekt implementierte Tokens können – zum Beispiel mittels XSS oder durch eine Browser-Verwundbarkeit – ausgehebelt werden. Das ist kein Grund zum Verzweifeln. Korrekt implementierte Tokens bedeuten eine erhebliche zusätzliche Hürde für den Angreifer. Wenn Sie den Einsatz von

Links zum Thema

- Durchführen einer Freestart-Kollision zum Entschlüsseln eines kompletten SHA-1-Keys
<https://eprint.iacr.org/2015/967.pdf>
- Mozilla SSL-Generator
<https://mozilla.github.io/server-side-tls/ssl-config-generator>
- SSL 3.0-Verwundbarkeit gemäß CVE-2014-3566
<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566>
- Hilfestellung zum Entfernen des eDellRoot-Zertifikats
<https://dellupdater.dell.com/Downloads/APP009/eDellRootCertificateRemovalInstructions.pdf>
- Download von Dells Software zum Entfernen des eDellRoot-Zertifikats:
<https://dellupdater.dell.com/Downloads/APP009/eDellRootCertFix.exe>
- HPACK: Header Compression für HTTP/2
<https://http2.github.io/http2-spec/compression.html>
- Test der Unterstützung von Diffie-Hellman für TLS:
<https://weakdh.org/sysadmin.html>
- Wireshark Download für Windows und OS X in 32- und 64-Bit (Dritthersteller-Packages gibt es für Linux, BSD, Solaris, HP-UX und andere Unix-Varianten):
<https://www.wireshark.org/download.html>

Listing 5: SSLSocket

```
URL url = new URL("https://webundmobiledeveloper.de");
URLConnection urlConnection =
    (URLConnection) url.openConnection();
try {
    InputStream in = new BufferedInputStream(
        urlConnection.getInputStream());
    readStream(in);
    finally {
        urlConnection.disconnect();
    }
}
```

Tokens mit TLS1.1/1.2-verschlüsseltem HTTP/2 verbinden können, machen Sie sich zusätzlich zur Verschlüsselung die POODLE-sichere Kompression und das Multiplexing zunutze (das funktioniert aber nur beim Einsatz von Tokens; das Verschlüsseln der Datenübertragung alleine nützt an dieser Stelle nichts.)

Fernbedienung per SQL

Wenn Ihre Webseite Daten in einer Datenbank speichert oder aus einer Datenbank bezieht, müssen Sie zusätzlich auch noch mit der Gefahr einer SQL Injection rechnen. Datenbanken sind ein sehr beliebtes Ziel für Hacker-Attacken. Bei einer SQL Injection nutzt der Angreifer eine Sicherheitslücke in Ihren Skripts oder auch in externen Bibliotheken aus, die es ihm erlaubt, seine eigenen SQL-Befehle an Ihre Datenbank zu senden.

In der Syntax einer SQL-Abfrage haben bestimmte Metazeichen, darunter Anführungszeichen, der Apostroph, der umgekehrte Schrägstrich und das Semikolon, eine eigene Bedeutung. Werden (Benutzer-)Eingaben – zum Beispiel aus Formularfeldern – ungeprüft in unveränderter Form an den SQL-Interpreter weitergegeben, so werden sie von ihm ausgeführt, wenn sie der Syntax entsprechen. Dadurch kann ein Angreifer andere Sicherheitsvorkehrungen Ihrer Webapplikation wie etwa Authentifizierungsmechanismen zur Anmeldung von Benutzern zwar nicht aushebeln, dafür aber umgehen. Auch beim Abrufen von Daten aus einer Datenbank besteht ein Risiko: Werden bei einer Datenbankabfrage die angeforderten Einträge unmaskiert ausgegeben, können die betreffenden Inhalte mit der Syntax der Datenbankabfrage in Kollision treten und Datenbankfehler verursachen.

SQL Injections kommen meistens über ein Webformular zustande, aus dem Daten ungeprüft an die SQL-Datenbank weitergeleitet werden, sodass diese sie als SQL-Befehle erkennt und pflichtbewusst ausführt.

App-Sicherheit: HTTPS in Android

In mobilen Apps ist HTTPS noch nicht die Regel. Mobile Apps kommunizieren aber immer öfter mit Backend-Diensten in der Cloud. Vielleicht führen auch Ihre Apps eine Remote-Au-

thentifizierung durch, synchronisieren Einstellungen, übertragen Benutzerdaten wie Bilder zur Aufbewahrung in der Cloud, sammeln Messwerte oder berichten über relevante UI-Ereignisse.

Wenn Ihr Backend-Server über ein Sicherheitszertifikat einer bekannten CA (Zertifizierungsstelle) verfügt, können Sie eine Verbindung Ihrer App damit wie folgt erstellen:

```
URL url = new URL("https://webundmobiledeveloper.de/");
URLConnection urlConnection = url.openConnection();
InputStream in = urlConnection.getInputStream();
copyInputStreamToOutputStream(in, System.out);
```

Der Aufruf von `URL.openConnection()` unter Angabe des HTTPS-Protokolls initiiert einen neuen `HttpsURLConnection`-Verbindungsaufbau.

Dadurch lassen sich die standardmäßigen `HostnameVerifier`- und `SSLConnectionFactory`-Klassen überschreiben. Eine durch die Anwendung bereitgestellte `SSLConnectionFactory`-Klasse zur `SSLSocket`-Erstellung, die vom `SSLContext`-API abgeleitet wurde, kann eine maßgeschneiderte Instanz der `X509TrustManager`-Klasse – zwecks Überprüfung von Zertifikatsschlüsseln – wie auch der `X509KeyManager`-Klasse – zur Bereitstellung von Client-Zertifikaten – ins Leben rufen.

Falls Sie Ihre HTTP-Anfragen maßgeschneidert aufbauen möchten, können Sie die `URLConnection`-Klasse zum Einsatz bringen (Listing 5). Dadurch verschaffen Sie sich die Möglichkeit, Anfrage- und Antwort-Header auszuwerten, Inhalte abzuschicken oder zu cachen, Cookies zu verwalten, Proxy-Server zu nutzen und dergleichen mehr.

Von dieser Klasse erbt die `HttpsURLConnection`-Klasse, die Ihnen einen Zugriff auf die zu verwendende Cipher-Suite, die Zertifikatskette und – falls verfügbar – das Client-Zertifikat ermöglicht.

Das Verifizieren von Zertifikaten und Hostnamen übernimmt für Sie das Android Framework unter Verwendung der bereitgestellten APIs – einfach gesagt. Die Realität ist leider etwas komplizierter. Die aktuelle Implementierung von Sicherheitszertifikaten ist weit davon entfernt, problemlos zu funktionieren.

Fazit

Ab dem Jahr 2016 dürfte weltweit mindestens eine Viertelmilliarde Anwender (vielleicht auch mehr) über Warnungen und Browserfehler auf verschlüsselten Websites ins Staunen geraten. Die einen oder anderen werden sicherlich daran abprallen. Überraschungen sind auch bei mobilen Apps nicht auszuschließen. ■

Filipe Pereira Martins und Anna Kobylinska

sind international anerkannte IT-Berater mit dem Schwerpunkt auf Cloud-Lösungen. Sie stehen den Lesern der **web & mobile developer** gern per Twitter via **@D1gitalPro** und **@D1gitalInfo** zur Verfügung.

FORMULARGENERIERUNG MIT JAVASCRIPT

Dynamische Generierung

Das jQuery-Plug-in AlpacaJS erlaubt die dynamische Generierung von Webformularen.

Das manuelle Erstellen von Webformularen kann mitunter recht langweilig sein. Oftmals sind Formulare ähnlich aufgebaut (beispielsweise Bestellformulare, Login-Formulare oder Formulare zur Registrierung), sodass man über kurz oder lang wahrscheinlich einen Online-Formulargenerator zu Hilfe nimmt oder zumindest aus einem Fundus von eigenen Formularvorlagen wählt und diese den jeweiligen Bedürfnissen anpasst.

Bei AlpacaJS (<http://alpaca.js.org>) handelt es sich um ein Plug-in für jQuery, mit Hilfe dessen sich automatisch Formulare per JavaScript generieren lassen. Die Konfiguration erfolgt über JavaScript-Objekte oder über JSON-Dateien.

Um AlpacaJS zu verwenden, müssen wie in Listing 1 gezeigt verschiedene Bibliotheken eingebunden werden: jQuery als Basis, Bootstrap für das Layout der Formularelemente, Handlebars, die standardmäßig als Templating-Engine zum Einsatz kommen, und schließlich AlpacaJS selbst.

Da man die Konfiguration von Formularen komplett über JavaScript durchführt, benötigt man im HTML lediglich einen Container (etwa ein `<div>`-Element), in den das entsprechende Formular hineingeneriert werden soll.

Über die Plug-in-Methode `alpaca()` lässt sich dann die Generierung eines Formulars vornehmen (Listing 2). Die Konfiguration erfolgt dabei im Wesentlichen über das der Methode übergebene Konfigurationsobjekt. Über dessen Eigen-

schaft *schema* lässt sich etwa der generelle Aufbau des zu generierenden Formulars (das Schema) definieren. Hier lassen sich unter anderem der Titel des Formulars (über die Eigenschaft *title*) und die einzelnen enthaltenen Formularelemente (über die Eigenschaft *properties*) bestimmen.

Über den folgenden Code beispielsweise veranlasst man die Generierung eines Texteingabefelds, das außerdem als Pflichtfeld gekennzeichnet ist:

```
schema: {
  ...
  properties: {
    ...
    'firstName': {
      'type': 'string',
      'title': 'Vorname',
      'required': true
    }
  }
}
```

Auf die gleiche Weise lassen sich auch Auswahllisten definieren (abhängig von der Anzahl an Auswahlmöglichkeiten generiert AlpacaJS entweder Radiobuttons oder Dropdown-Auswahllisten). Darüber hinaus lassen sich alle Standardfor-

Listing 1: Basis-HTML

```
<!DOCTYPE html>
<html>
  <head>
    <!-- jquery -->
    <script type="text/javascript" src="//code.jquery.
com/jquery-2.1.4.min.js"></script>

    <!-- bootstrap -->
    <link type="text/css" rel="stylesheet" href=
"//maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/
bootstrap.min.css" />
    <script type="text/javascript" src="//maxcdn.
bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.
js"></script>

    <!-- handlebars -->
    <script type="text/javascript" src="//cdnjs.
cloudflare.com/ajax/libs/handlebars.js/3.0.3/
```

```
handlebars.js"></script>

    <!-- alpaca -->
    <link type="text/css" href="//code.cloudcms.com/
alpaca/1.5.14/bootstrap/alpaca.min.css"
rel="stylesheet" />
    <script type="text/javascript" src="//code.
cloudcms.com/alpaca/1.5.14/bootstrap/
alpaca.min.js">
    </script>
  </head>
  <body>
    <div id="form"></div>
    <script type="text/javascript" src="scripts/
main.js"></script>
  </body>
</html>
```


Nutzer anlegen

★**Vorname**

ⓘ Bitte geben Sie Ihren Vornamen ein.

ⓘ This field is not optional.

★**Nachname**

ⓘ Bitte geben Sie Ihren Nachnamen ein.

★**Geschlecht**

☒ Männlich

☐ Weiblich

ⓘ Bitte wählen Sie Ihr Geschlecht aus.

Beispiel für ein generiertes Formular (Bild 1)

Fragebogen

★**Wie sind Sie auf XYZ aufmerksam geworden?**

☐ Durch Freunde oder Bekannte.

☐ Durch den Newsletter.

☐ Durch die Webseite von XYZ.

★**Wie häufig bestellen Sie bei XYZ?**

★**Welche Produktart bestellen Sie bei XYZ?**

★**Würden Sie XYZ weiterempfehlen?**

☐ Ja

☐ Nein

Ein Anwendungsfall von AlpacaJS: die Generierung von Fragebögen (Bild 2)

mularelemente wie Checkboxes, Radiobuttons, Dateiauswahlen, Datums- und Zeitauswahlen, Textbereiche und Texteingabefelder sowie auch komplexere beziehungsweise speziellere Formularelemente wie Adresseingaben, Farbauswahlen und Länder- oder Währungsangaben.

Der Eigenschaft *options* des Konfigurationsobjekts kann man ein weiteres Konfigurationsobjekt hinterlegen, über das

sich etwa die Breite von Textfeldern, Platzhalter oder Hilfetexte für jedes zuvor im Schema definierte Formularfeld definieren lassen (Bild 1).

Der wesentliche Vorteil von AlpacaJS gegenüber statischen HTML-Formularen wird erst dann richtig deutlich, wenn es an die dynamische Generierung von Formularen geht, etwa um die Generierung eines Fragebogens, ab- ►

Listing 2: Einfaches Formular

```
'use strict';

$(document).ready(() => {
  $('#form').alpaca({
    'schema': {
      'title': 'Nutzer anlegen',
      'type': 'object',
      'properties': {
        'firstName': {
          'type': 'string',
          'title': 'Vorname',
          'required': true
        },
        'lastName': {
          'type': 'string',
          'title': 'Nachname',
          'required': true
        },
        'gender': {
          'type': 'string',
          'title': 'Geschlecht',
          'enum': ['Weiblich', 'Männlich'],
          'required': true
        }
      }
    },
    'options': {
      'fields': {
        'firstName': {
          'size': 20,
          'helper': 'Bitte geben Sie Ihren Vornamen ein.',
          'placeholder': 'Vorname'
        },
        'lastName': {
          'size': 40,
          'helper': 'Bitte geben Sie Ihren Nachnamen ein.',
          'placeholder': 'Nachname'
        },
        'gender': {
          'helper': 'Bitte wählen Sie Ihr Geschlecht aus.'
        }
      }
    }
  });
});
```

hängig von einem Fragenkatalog. Listing 3 zeigt ein solches Beispiel. Die einzelnen Fragen liegen hier direkt im Code vor, könnten aber auch ohne größeren Aufwand aus einer lokalen Datei oder von einem Server zur Laufzeit geladen werden. Mit Hilfe von ein wenig funktionaler Programmierung (*map()* und *reduce()*) wird das Array in ein Objekt umgewandelt, das direkt als Eingabe für die Eigenschaft *properties* des *schema*-Objekts verwendet werden kann (Bild 2).

Generierung von Wizards

Ein besonders nettes und nützliches Feature ist die Erstellung von Wizards, sprich mehrschrittigen Formulareingaben, die aus mehreren einzelnen Formularmasken bestehen.

Was – wenn man es selbst in HTML umsetzt – durchaus aufwendig sein kann, ist mit Hilfe von AlpacaJS schnell umgesetzt, wie in Listing 4 zu sehen ist.

Die Formularkomponenten an sich werden auch hier über das *schema*-Objekt definiert. Neu hinzu kommt lediglich die Eigenschaft *view*: Darüber konfiguriert man die Titel und Be-

schreibungen der einzelnen Formularmasken (Eigenschaft *steps*) und die Zuordnung von Formularkomponenten zu diesen Masken (Eigenschaft *bindings*). So werden die Formularkomponenten *firstName*, *lastName* und *age* der ersten Formularmaske zugewiesen, die Formularkomponenten *street*, *number*, *zipcode*, *city* und *state* der zweiten Formularmaske und *phone* und *email* der dritten Formularmaske. Alles Weitere übernimmt das jQuery-Plug-in AlpacaJS im Hintergrund über Templates, die sich bei Bedarf auch anpassen lassen.

Eigenschaften von Wizards

Wizards (und Formulare im Allgemeinen) können mit AlpacaJS relativ schnell über viele der zur Verfügung stehenden Eigenschaften konfiguriert werden.

Beispielsweise lässt sich über die Eigenschaft *showProgressBar* festlegen, ob eine Fortschrittsanzeige im Wizard angezeigt werden soll. Eine ausführliche Dokumentation aller weiteren Eigenschaften findet man auf der Homepage der Bibliothek.

Listing 3: Generierung eines Fragebogens

```
'use strict';
$(document).ready(() => {
  let questions = [
    {
      question: 'Wie sind Sie auf XYZ aufmerksam geworden?',
      answers: [
        'Durch die Webseite von XYZ.',
        'Durch den Newsletter.',
        'Durch Freunde oder Bekannte.'
      ]
    },
    {
      question: 'Wie häufig bestellen Sie bei XYZ?',
      answers: [
        '1x pro Jahr',
        '1x pro Monat',
        '1x pro Woche',
        '1x pro Tag'
      ]
    },
    {
      question: 'Welche Produktart bestellen Sie bei XYZ?',
      answers: [
        'Produktart A',
        'Produktart B',
        'Produktart C',
        'Produktart D',
        'Produktart E'
      ]
    }
  ],
  [
    {
      question: 'Würden Sie XYZ weiterempfehlen?',
      answers: [
        'Ja',
        'Nein'
      ]
    }
  ];

  let properties = questions
    .map((question, index) => {
      return {
        name: 'question' + index,
        content: {
          'type': 'string',
          'title': question.question,
          'enum': question.answers,
          'required': true
        }
      };
    })
    .reduce((object, property) => {
      object[property.name] = property.content;
      return object;
    }, {});

  $('#form').alpaca({
    'schema': {
      'title': 'Fragebogen',
      'type': 'object',
      'properties': properties
    }
  });
});
```

Listing 4: Erstellen eines Wizards

```

'use strict';
$(document).ready(() => {
  $('#form').alpaca({
    'schema': {
      'title': 'Bestellformular',
      'description':
        'Bestellformular.',
      'type': 'object',
      'properties': {
        'firstName': {
          'type': 'string',
          'title': 'Vorname',
          'required': true
        },
        'lastName': {
          'type': 'string',
          'title': 'Nachname',
          'required': true
        },
        'gender': {
          'type': 'string',
          'title': 'Geschlecht',
          'enum': ['Weiblich',
            'Männlich'],
          'required': true
        },
        'street': {
          'type': 'string',
          'title': 'Straße',
          'required': true
        },
        'number': {
          'type': 'number',
          'title': 'Nr.',
          'required': true
        },
        'zipcode': {
          'title': 'PLZ',
          'type': 'zipcode',
          'required': true
        },
        'city': {
          'type': 'string',
          'title': 'Ort',
          'required': true
        },
        'state': {
          'type': 'string',
          'title': 'Bundesland',
          'required': true
        },
        'phone': {
          'title': 'Telefon',
          'type': 'string',
          'format': 'phone'
        },
        'email': {
          'title': 'Email-Adresse',
          'type': 'string',
          'format': 'email'
        }
      },
      'view': {
        'parent': 'bootstrap-edit-
          horizontal',
        'wizard': {
          'title': 'Bestellung starten',
          'bindings': {
            'firstName': 1,
            'lastName': 1,
            'age': 1,
            'gender': 1,
            'street': 2,
            'number': 2,
            'zipcode': 2,
            'city': 2,
            'state': 2,
            'phone': 3,
            'email': 3
          },
          'steps': [{
            'title': 'Nutzerdaten',
            'description': 'Eingabe der
              persönlichen Daten'
          }, {
            'title': 'Adresse',
            'description': 'Eingabe der
              Adressinformationen'
          }, {
            'title': 'Kontakt',
            'description': 'Eingabe der
              Kontaktdaten'
          }],
          'showSteps': true,
          'showProgressBar': false,
          'validation': true,
          'buttons': {
            'first': {
              'title': 'Zu Schritt 1',
              'align': 'left',
              'click': function(e) {
                this.trigger('moveToStep', {
                  'index': 0,
                  'skipValidation': true
                });
              }
            },
            'previous': {
              'validate':
                function(callback) {
                  callback(true);
                }
            },
            'next': {
              'validate':
                function(callback) {
                  callback(true);
                }
            },
            'submit': {
              'title': 'Absenden',
              'validate':
                function(callback) {
                  callback(true);
                },
              'click': function(e) {
                console.log(JSON.stringify(
                  this.getValue(), null,
                  '  '));
              },
              'id': 'submit'
            }
          }
        }
      }
    }
  });
});

```

Fazit

AlpacaJS ermöglicht die dynamische und automatisierte Generierung von HTML-Formularen. Es unterstützt die Generierung von Wizards, Mehrsprachigkeit, Verwenden eigener Templates, Abhängigkeiten zwischen einzelnen Formular-komponenten, Validierung von Formulareingaben und vieles mehr. Wenn man also als Webentwickler wieder einmal mit Formularen zu tun hat und diese nicht manuell erzeugen möchte, ist AlpacaJS auf jeden Fall einen Blick wert. ■



Philip Ackermann

arbeitet beim Fraunhofer-Institut für Angewandte Informationstechnologie FIT an Tools zum teilautomatisierten Testen von Web Compliance und ist Autor zweier Fachbücher über Java und JavaScript.

<http://philipackermann.de>

EXT JS 6 – FRAMEWORK FÜR ALLE PLATTFORMEN

Eines für alle

Sencha verfügt mit dem neuen Ext JS 6 über ein durchgängiges JavaScript-Framework für Desktop, Tablets und mobile Endgeräte.

Bis zu den Vorgängerversionen existierten bei Sencha noch zwei getrennte Welten: das Ext-JS-Framework ausschließlich für Desktop und Tablets und das speziell auf mobile Endgeräte ausgerichtete Sencha-Touch-Framework. Für die neue Version 6 von Ext JS entschloss sich der Hersteller, die beiden getrennten Frameworks zu einem gemeinsamen Framework für die Entwicklung verschiedener Anwendungstypen zu vereinen.

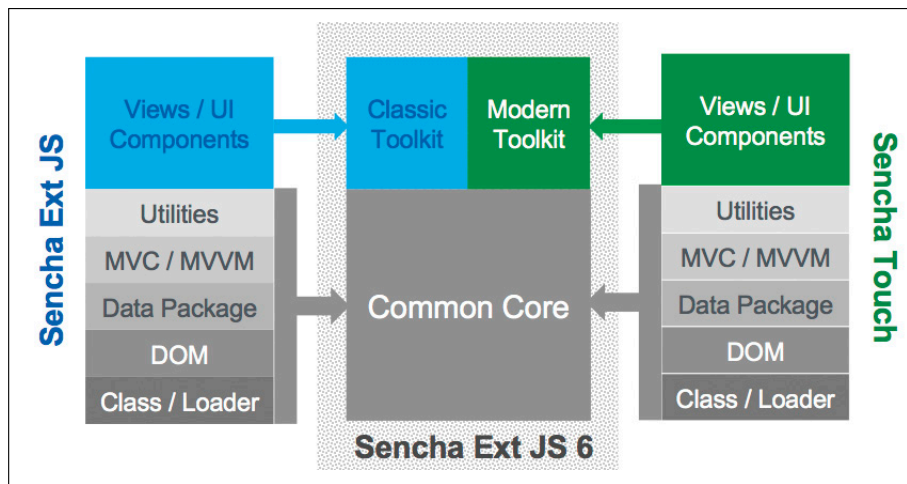
Sencha unterstützt mit Ext JS 6 die ganze Palette heute möglicher Hardware und Software, beginnend mit den älteren Webbrowsern für Desktops bis zu den verschiedenen modernen Bedienoberflächen für Smartphones. Der Hersteller verfolgt dabei das Ziel, Legacy/Alt-Systeme für ältere Browserversionen und moderne Webanwendungen gleichberechtigt einzubeziehen. Für diesen Zweck bietet Sencha verschiedene Migrationspfade zur Umstellung bestehender Legacy/Alt-Anwendungen von Ext JS und Sencha Touch auf die neue Version 6 von Ext JS an.

Ergänzend umfasst das vereinheitlichte Framework auch neue Features wie einen Theme Compiler (Fashion), Promises für asynchrones JavaScript, App Templates (Anwendungsvorlagen), ein verbessertes Pivot Grid, IDE-Plug-ins (für JetBrains, Eclipse und Visual Studio) sowie erstmals einen eigenen Debugger: Sencha Inspector.

Vereintes Framework bringt viele Vorteile für die Zukunft

Bei der Konzeption von Ext JS Version 6 waren zwei Aspekte zentral: eine einzige Code-Basis und ein neuer sogenannter Universal-App-Typ auf dieser vereinheitlichten Code-Basis. Damit sollten Entwickler Anwendungen für alle Gerätetypen vom Desktop bis Smartphones programmieren können, die dem Endbenutzer eine optimierte Oberfläche für das jeweilige Endgerät mit möglichst wenig Quellcode implementieren.

Zusätzlich gestaltete Sencha ein neues Theme, Triton genannt, um auch für Font-Icons einen umfassenden Geräte- und Browser-Support bieten zu können. Die bisher notwendige Überarbeitung der Größe oder Farbe von Bildern und Icons mittels einer Bildbearbeitung wird damit hinfällig.



Version 6 von Ext JS besteht aus einer einheitlichen Code-Basis, die aus den Grundfunktionen von Ext JS Version 4/5 und Sencha Touch Version 2 aufgebaut wurde (Bild 1)

Die ersten Schritte für diese einheitliche Framework-Welt unternahm Sencha bereits mit der Weiterentwicklung von Ext JS Version 5. Dabei führte man die Basis der beiden Frameworks Ext JS und Sencha Touch zu einem sogenannten Core-Package zusammen (Bild 1). Mit Ext JS 5 vereinte man Klassen mit gleichen Aufgaben wie Class/Loader, DOM, Data Package, MVC/MVVM und weitere Hilfsklassen/Utilities in einem gemeinsamen Package. Für Version 6 von Ext JS nutzte man diese Ausgangsbasis von Ext JS 5 und verein-

Workflow-Automatisierung für HTML/JavaScript

Sencha nutzt Packages, um den Build- und Deploy-Prozess möglichst flexibel und effizient durchzuführen.

Seit Version 4.2 von Ext JS unterstützt Sencha mit seinen SDK-Tools (heute Sencha Cmd genannt) den Build- und Management-Prozess von der Generierung bis zum Packaging. Die neue Version 6 von Sencha Cmd verwaltet automatisch sämtliche Abhängigkeiten und bindet Arbeitsergebnissen anderer Entwickler über Packages ein – auch für Testzwecke.

Damit automatisiert Sencha Cmd den kompletten Workflow der Software-Entwicklung, sodass sich Continuous Integration auf einer gemeinsamen Code-Basis inklusive Continuous Delivery realisieren lässt.

heitlichte unter anderem die grundlegenden APIs wie Ext.data und Ext.app. Auf Basis dieses neuen, vereinten Common-Core-Packages können Anwendungen erstmals Quellcode gemeinsam für das Daten-Handling, für ViewModels oder die Control-Logik (für die Kommunikation mit dem Backend) sowohl auf dem Desktop als auch dem Smartphone nutzen.

Leider verdeutlichten die Vorarbeiten für Version 6 sehr bald, dass sich die Oberfläche einer Desktop-Anwendung nicht optimal für eine Smartphone-App eignet. Umgekehrt gestaltete sich die Oberfläche einer Smartphone-App nicht optimal für eine Desktop-App; während die optimale Oberfläche einer Tablet-App sich irgendwo zwischen der eines Smartphones oder Desktops befindet.

Dieser Sachverhalt rückte das Ziel einer vollkommen einheitlichen Code-Basis für Desktop und mobile Welt in die weitere Ferne. Aufgrund dieser Erkenntnisse schuf Sencha für die vorliegenden Unterschiede bei den visuellen Aspekten des letztendlich angestrebten komplett vereinten Frameworks eigenständige Packages. Um diese Oberflächen-Aspekte der Endgeräte optimal nutzen und sie von den anderen Packages unterscheiden zu können, führte man für diese die Bezeichnung Toolkit ein.

View Components

Während für die nicht auf der Oberfläche sichtbaren Komponenten des Frameworks schon jetzt eine einheitliche Welt entstand, besteht auf der Ebene der Bedienoberfläche (der View Components) nach wie vor eine Zweiteilung in Classic Toolkit für Ext-JS- und in Modern Toolkit für Sencha-Touch-Oberflächen. Nichtsdestotrotz können Entwickler über das Cmd-Werkzeug von Sencha eine sogenannte Universal Application anlegen, die die beiden Welten Desktop und Mobile wieder zusammenführt.

Eine Universal Application greift auf beide Toolkits Classic und Modern zurück, sodass man mit einer einzigen Code-Ba-

Empfehlungen zur Installation von Ext JS 6

Sencha stellt Ext JS wie bisher als ZIP-Archiv in Form eines SDK-Packages zum Download bereit.

Ext JS Version 6 installiert man wie die Vorgängerversionen durch ein einfaches Unzip in einen beliebigen Ablageordner. Leider muss man auch für Version 6 immer noch eine individuelle Umgebungsvariable (zum Beispiel `ext_home`) manuell einrichten, die auf das Installationsverzeichnis von Ext JS weist.

Diese neu eingerichtete Umgebungsvariable erlaubt es, die gewünschte Ext-JS-Version mittels der `sdk`-Option von Sencha Cmd direkt (ohne ausführliche nochmalige Referenz des Installationsordners) anzusprechen.

sis sowohl mobile Endgeräte wie Smartphones und Tablets als auch gängige Hardware wie Desktop/Laptops unterstützen kann. Sencha Cmd greift beim Anlegen einer Universal Application für die Mobile-Welt auf das Modern Toolkit und für den Desktop/Laptop auf das Classic Toolkit von Ext JS Version 6 zurück.

Sencha Cmd 6 rückt verstärkt ins Zentrum

Mit Hilfe von Sencha Cmd verfolgt der Hersteller seit Längerem das Ziel, die Produktivität der Entwickler und die Performance der Anwendungen durch maximale Automatisierung zu verbessern. Das Tool besitzt eine Kommandozeilen-Schnittstelle, um alle zu erledigenden Aufgaben auch im Batch/Stapel-Betrieb durchführen zu können.

Die Kommandozeile bietet dem Entwickler eine große Auswahl an Optionen, beginnend mit der Generierung von Quellcode, Test/Debugging bis zum abschließenden Na- ►

Installationshinweise für Sencha Cmd 6

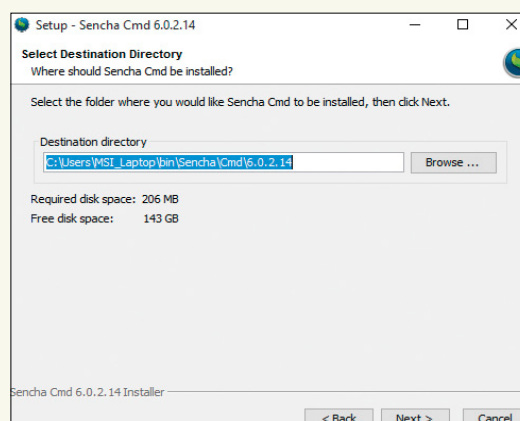
Sencha Cmd 6 basiert auf der Java Runtime Environment (JRE)

1.7. Bei bereits installierter JRE oder JDK wählt man für den Download eine Cmd-Version ohne integrierte JRE aus.

Für Sencha Cmd gibt es analog der JRE 32- und 64-Bit-Versionen. Die Installation von Sencha Cmd erfolgt mittels eines Setup-Wizards des Multi-Plattform-Java-Installers `install4j`. Soll das Styling einer Anwendung die Vorgängerversionen (Ext JS 4.x, Ext JS 5.x und Sencha Touch 2.x) unterstützen, muss man bei der Installation zusätzlich die Komponente *Compass extension* über das Setup auswählen.

Eine korrekte Installation von Sencha Cmd lässt sich mittels des Befehls `sencha which` feststellen. Nach dessen Ausführung gibt Sencha Cmd auf der Ebene des Betriebssystems die aktuell installierte Versionsnummer aus. Zusammen mit der Versionsnummer erscheint das komplet-

te Installationsverzeichnis in der Ausgabezeile. Der Setup-Wizard erzeugt automatisch den notwendigen Eintrag für die Umgebungsvariable `Path`.



Mittels des Setup-Wizards legt man den Ablageordner von Sencha Cmd fest und wählt gegebenenfalls das Package *Compass extension* für die Installation aus.

tive-Packaging. Als Kommandozeilen-Tool lässt sich Sencha Cmd in jede Umgebung innerhalb einer gewachsenen Infrastruktur einer Entwicklungsorganisation integrieren. Hierfür stellt es einen großen Umfang an Befehlen und auch eine Einbindung des Ant-Build-Systems zur Verfügung.

In Version 6 von Ext JS enthält das Classic Toolkit die speziellen visuellen Elemente von Ext JS, während die besonderen GUI-Elemente von Sencha Touch im Modern Toolkit enthalten sind. Die gemeinsame Basis dieser GUI-Toolkits bildet der Common Core.

Eine Entwicklung mit Ext JS Version 6 macht damit Sencha-Touch-Anwendungen das bereits mit Version 5 von Ext JS eingeführte Multi-Device-Event-System zugänglich. Damit werden mit der Maus ausgelöste Events in entsprechende Event Listener überführt, ohne vorher die Fähigkeiten des jeweiligen Endgeräts erkunden zu müssen. Zusammen mit der neuen Version 6 von Sencha Cmd kann man für Ext JS 6 zusätzlich sogenannte Universal Applications erzeugen, die sowohl das Classic als auch das Modern Toolkit benutzen können.

Sencha Cmd 6 generiert mittels des nachfolgenden Kommandos über die `-sdk`-Option der bei der Ext-JS-Installation manuell definierten `ext_home`-Umgebungsvariablen, den Befehlskategorien `generate` und `app`, standardmäßig eine Universal Application (manchmal auch All-in-one App genannt) mit dem Name `meineAnwd` und legt dabei alle zugehörigen Dateien in den Ordner `pfadAnwd` (Application Root genannt) ab:

```
sencha -sdk %ext_home% generate app meineAnwd ./pfadAnwd
```

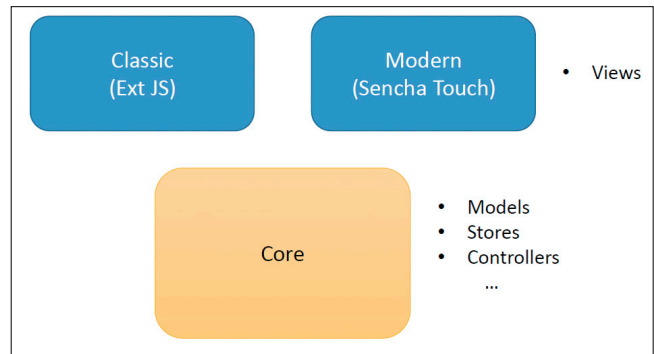
Die Universal Application besteht im wesentlichen aus den drei folgenden Bereichen (Bild 2):

- **Shared App Area:** Umfasst Stores, Models und Controller entsprechend der Konventionen des Common Core. Die zugehörigen Dateien befinden sich im `app`-Ordner unterhalb des Application-Root-Verzeichnisses. Ebenfalls zum Shared Content rechnet man alle anderen Dateien in den Unterverzeichnissen des Anwendungspaths wie Ressourcen- und Sass-Dateien.

Sencha Cmd – das Fundament der Entwicklung

Als Werkzeug für das Lifecycle-Management stellte Sencha Cmd schon länger die Basis der Entwicklung mit Sencha Ext JS und Sencha Touch dar.

Sencha Cmd erhöht die Produktivität und optimiert die Performance bereits während des Application Builds. Das Tool übernimmt Aufgaben wie Code Generation (inklusive Image Spriting), Scaffolding, Package Management (auch nativ mittels einer Schnittstelle zu PhoneGap/Cordova), Test inklusive Fehlersuche und Application Tuning. Zudem besitzt Sencha Cmd einen JavaScript-Compiler, der die Besonderheiten und auch die Semantik der Frameworks kennt und direkt unterstützt.

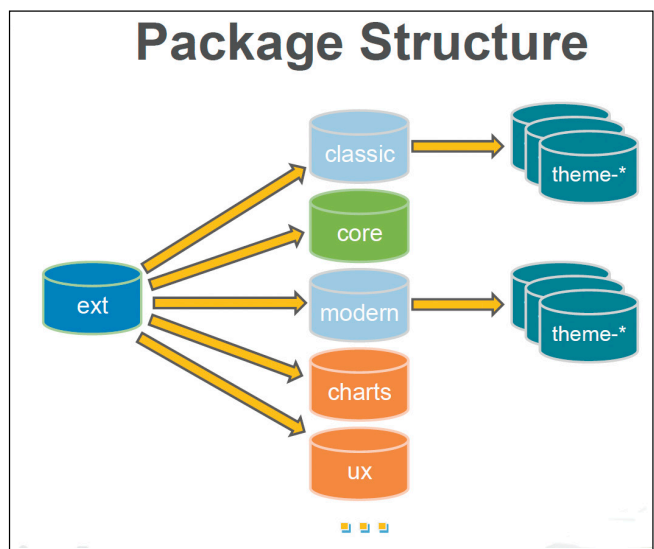


Die Anwendungsarchitektur von Ext JS Version 6 setzt sich aus drei Bereichen zusammen: Classic (ehemals Ext JS bis Version 5.x), Modern (ehemals Sencha Touch bis Version 2.x) und Core (gemeinsam genutzter Quellcode) (Bild 2)

- **Classic App Area:** Hierzu gehören alle Views und die Logik zu den visuellen Aspekten der Desktop/Laptop-Anwendung. Zu diesem Bereich zählen alle Dateien im Ordner `classic/src`, den man unterhalb des Application Root findet.
- **Modern App Area:** Sie beinhaltet alle Views mit Logik zu den visuellen Aspekten der Smartphone-Anwendung. Der Bereich mit zugehörigen Dateien entspricht dem `modern/src`-Verzeichnis des Application Root.

Ausrichtung auf alle Zielplattformen mit maximaler Automatisierung

Während die Classic und Modern App Area lediglich den toolkitspezifischen Content enthält, findet man den mehrfach verwendbaren Quellcode einer Universal Application im Shared-Content-Bereich. Zu den visuellen toolkitspezifischen Inhalten rechnet man auch alle Themes, die sowohl seitens des Classic- als auch des Modern Toolkits unterschiedlich behandelt werden (Bild 3).



Die Weiterentwicklung von Ext JS führte zu einer neuen Ablagestruktur für eine Anwendung, die Sencha Cmd umfassend behandeln kann (Bild 3)

Ergänzend können zu diesen drei Bereichen noch weitere hinzukommen, zum Beispiel *charts* für Diagramme oder *UX* für Ext JS User Extensions. Die neue Version 6 von Sencha Cmd kennt alle Neuerungen in der Package-Struktur und kann deren neuen Aufbau/Zusammensetzung sowie auch die spezifischen Inhalte verarbeiten.

Sencha vereinfachte die Entwicklung für die unterschiedlichen Hardwareplattformen von Ext JS 6 sowohl für die Programmierung als auch für Test/Debugging und Deployment. Neben dem Anwendungstyp der Universal Application, der quasi alle Welten vereint, unterstützt die Version 6 von Sencha Cmd gleichberechtigt sowohl das Classic als auch das Modern Toolkit.

Es spielt keine Rolle, ob eine Entwicklung mit Test/Debugging oder Deployment auf einem mobilen Endgerät oder Desktop/Laptop gewünscht wird. Viele Befehlskategorien von Sencha Cmd, wie *generate* für den Quellcode-Generator, *inspect* für Test/Debugging oder *app* für das Application-Management, besitzen jetzt bei ihren Kommandos Schalter, um eine Automatisierung vollständig plattformgerecht vornehmen zu können.

Somit führt Sencha Cmd alle erforderlichen Arbeiten gänzlich automatisch gezielt auf die jeweilige Plattform abgestimmt und damit optimal für das Classic als auch das Modern Toolkit aus. Hat man beispielsweise eine Anwendung mittels des *app*-Befehls der *generate*-Kategorie angelegt, kann man im Pfad der Anwendung mittels der Befehlsfolge:

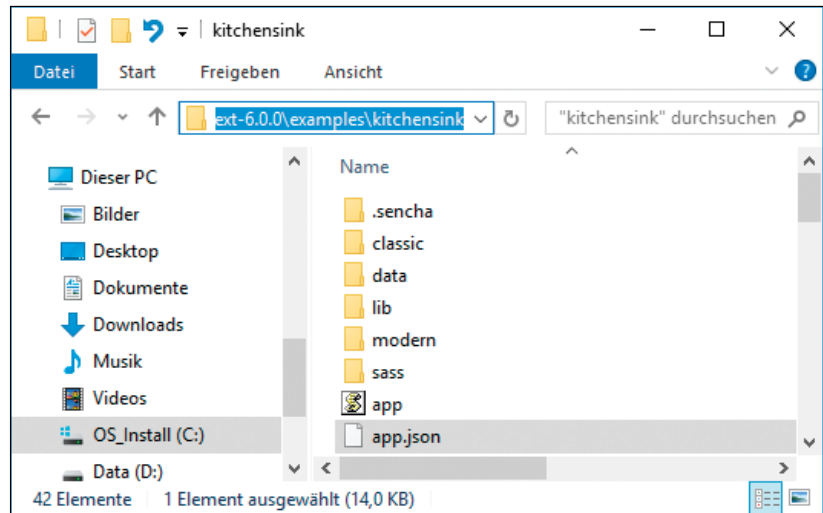
```
cd pfadAnwd
sencha app watch classic
```

ein Packaging und ein Deployment der Classic-Anwendung auf den über Sencha Cmd verfügbaren Webserver durchführen. Seit Version 3.1.x hat Sencha Cmd den Jetty-Webserver integriert; diesen erreicht man über die IP-Adresse `http://localhost:1841`. Der in Sencha Cmd integrierte Webserver startet mittels des Befehls: *sencha web start* auch direkt und hält mittels *sencha web stop* wieder an.

Das Hilfesystem von Sencha Cmd 6 nutzen

Aufgrund der Mächtigkeit von Sencha Cmd gibt das allgemeine Hilfe-Kommando *sencha help* eine Übersicht zu den vielen Optionen, Kategorien und Befehlen aus.

Nähere Information zu einer konkreten Option, Kategorie oder Befehl erhält man durch eine spezielle *help*-Abfrage. Zum Beispiel liefert *sencha help generate* nähere Hilfestellungen zur Kategorie *generate*. Benötigt man nähere Angaben zum *app*-Befehl der *generate*-Kategorie, so zeigt man sich diese über *sencha help generate app* an.



Beispiel: Die neue Version des Frameworks Ext JS 6 enthält im Unterverzeichnis *examples* ein fertig lauffähiges Beispiel für eine Universal App (Bild 4)

Ohne einen der Zusätze *classic* oder *modern* bei der *app*-Befehlskategorie wird ein Compile/Build-Prozess für eine Universal App durchgeführt. Diese startet man im Webbrowser direkt durch Eingabe des URL `http://localhost:1841`. Der Sencha Cmd-Befehl *app watch* zeigt mittels der Informationsmeldung *Waiting for changes ...* an, dass Veränderungen am Quellcode automatisch aktualisiert und danach beim Deployment berücksichtigt werden. Die Universal App führt diese Änderungen allerdings nur aus, wenn man im Webbrowser die Seite über den URL neu lädt.

Entscheidende Änderungen und Restrukturierungen im Framework

Intern achtete Sencha beim Framework darauf, dass die Vereinheitlichung nur minimale Änderungen in Ext JS mit sich brachte. In Anlehnung an eine zu entwickelnde App be- ▶

Neuerungen von Sencha Cmd 6 im Überblick

Version 6 von Sencha Cmd kennt alle mit Ext JS 6 eingeführten neuen Anwendungstypen und unterstützt jetzt auch Test/Debugging sowie Theming.

Um die Migration für Ext-JS-Anwender zu erleichtern, kann Sencha Cmd Version 6 auch zusammen mit den Framework-Versionen Ext JS 4.1.1a (und höher) sowie Sencha Touch 2.1 (und höher) eingesetzt werden.

Der *generate app*-Befehl kennt zwei neue, mit Version 6 eingeführte Optionen: *-classic* und *-modern*, um eine Classic- (ursprünglich: Ext-JS-Anwendung) oder eine Modern- (ursprünglich: Sencha-Touch-) App zu erzeugen.

Der *app watch*-Befehl besitzt zwei neue Optionen *inspect* (für Instrumentierung von Tests mittels Sencha Inspector) und *fashion*, um den Saas-Compiler (für das Theming) einzuschalten.

steht die ZIP-Datei des einheitlichen Frameworks aus drei Bereichen: *classic*, *modern* und *common*. Auch die beiden Unterverzeichnisse *build* (für das Package Management) und *examples* (mit den Beispielanwendungen) einer Ext-JS-Installation gliedern sich gemäß dieser Dreiteilung. Beispielsweise findet man für eine Universal Application im *examples*-Unterverzeichnis den Ordner *kitchensink* mit zugehörigem Quellcode (Bild 4); wobei die Datei *app.json* für beide Anwendungstypen Classic und Modern verschiedene Build-Profile enthält.

Wie bei den Vorgängerversionen von Ext JS umfasst der *build*-Ordner die *ext-all*.js*-Dateien; sie besitzen nach wie vor auch in der neuen Version 6 dieselbe Funktionalität beziehungsweise implementieren wie früher die damit verbundenen Aufgaben. Im Unterschied hierzu ersetzte man die *sencha-touch*.js*- durch die *ext-modern*.js*-Dateien; diese bauen nun direkt auf dem neuen Common-Core-Package und nicht mehr auf Sencha Touch auf.

Einheitliche Namensgebung bei den Package-Namen

Alle früher für die Unterscheidung zwischen Ext JS und Sencha Touch bei den Package-Namen vorhandenen *ext*- und *sencha*-Präfixe entfernte man. So heißen im einheitlichen Framework die Packages *sencha-charts* nun einfach nur *charts* oder *ext-theme-neptune* nur noch *theme neptune*.

Wegen dieser Namensänderungen muss man ein Refactoring für alle Anwendungen durchführen. Auch in der *app.json*-Datei müssen die neuen Bezeichnungen berücksichtigt werden. Zum Beispiel darf anstatt *ext-charts* im *requires*-Block dort jetzt nur *charts* stehen.

Seitens des API schuf man für den *Ext.ux.**-Quellcode ein eigenständiges *ux*-Package; dies machte die bisher notwendige Code-Generierung seitens Sencha Cmd für das *Ext.ux*-API hinfällig. Das bisherige *ext-aria*-Package für die erleichterte Bedienung der Oberflächen-Elemente und die WAI-ARIA-Unterstützung (Web Accessibility Initiative – Accessible Rich Internet Applications gemäß W3C) integrierte man vollständig in das Core-Framework. Somit kann man jetzt al-

Refactoring oder Restrukturierung des Quellcodes

Eine manuelle oder automatisierte Umstrukturierung von Quellcode bei gleichbleibendem Programmverhalten nennt man Refactoring/Restrukturierung.

Als Entwickler verfolgt man damit das Ziel, die Erweiterbarkeit, Lesbarkeit, Verständlichkeit oder Wartbarkeit des eigenen Quellcodes zu verbessern. Auch die Testbarkeit soll erleichtert und das Auftreten eventuell vorhandene Redundanzen im Quellcode beseitigt werden.

In der Regel besitzen integrierte Entwicklungsumgebungen (IDEs) spezielle Werkzeuge, um das Refactoring zu unterstützen, oder führen diese anhand individueller Vorgaben durch den Programmierer sogar automatisch durch.

Flexibles, wiederverwend- und erweiterbares Styling

Sass (Syntactically Awesome Stylesheets) beseitigt als Präprozessor die Nachteile von CSS.

Im Unterschied zu CSS ermöglicht Sass Variablen, Funktionen, Nested Rules und Mixins, um Code wiederverwenden zu können.

Sencha Cmd führt den Webbrowser PhantomJS aus, der im Hintergrund ohne GUI-Oberfläche läuft. Fashion überführt das Theme mittels PhantomJS vollständig in eine große JavaScript-Funktion.

Die Funktionalität oder das Verhalten von Fashion lässt sich durch JavaScript-Module erweitern oder ändern.

le notwendigen Verweise auf *ext-aria* im *requires*-Block der *app.json*-Datei entfernen.

Vorhandene Anwendungen nach Ext JS Version 6 migrieren

Kennt man die zuvor beschriebenen Restrukturierungen und weiß man, dass Ext JS 6 sowohl aus Sicht von Ext JS 5.1.x als auch seitens Sencha Touch 2.4.x bereits alle notwendigen Bestandteile enthält, so lässt sich abhängig vom Typ einer Anwendung in relativ kurzer Zeit die Migration des Quellcodes auf die neue Version 6 von Ext JS durchführen. Im ersten Schritt muss man den *.sencha*-Ordner (enthält Konfigurationen und Build-Dateien für Sencha Cmd) und andere wichtige Bestandteile der Anwendung auf die neue Version 6 umstellen. Dazu gibt es zwei Alternativen:

- **Automatisches Upgrade:** Ausgehend vom Anwendungsverzeichnis der zu migrierenden App führt man den Befehl *sencha app upgrade ../ext6-dateipfad* aus.
- **Manuelle Umstellung:** Neuanlage einer Classic-App (im Fall einer Ext-JS-5.1-App) oder einer Modern-App (im Fall einer Sencha-Touch-2.4-App) mit der in Version 6 eingeführten *classic*- beziehungsweise *modern*-Option des *generate app*-Befehls; anschließend ersetzt man den *app*-Ordner (und falls notwendig auch den *sass*-Ordner) durch die neu generierten Anwendungsteile.

Zusätzlich passt man noch die Vorgaben der Datei *app.json* entsprechend der mit Ext JS 6 durchgeführten Namensänderungen der Packages, Themes und den damit verbundenen Umstrukturierungen an. Im Fall einer Sencha-Touch-Anwendung muss man noch das Model der App anpassen. Version 6 macht die bisher in Sencha-Touch-Apps erforderliche Schachtelung für *proxy* und *fields* im *config*-Block eines Models hinfällig. Daher kann man diese Schachtelung entfernen oder als Kommentarzeilen (mittels *//* im JavaScript-Code) definieren. Zum Schluss sollte man noch den Befehl *sencha app refresh* durchführen; dieser integriert eventuell noch benötigte Packages in die zu migrierende App.

Soll eine Ext-JS-5.1.x- oder eine Sencha-Touch-2.4.x-App auf eine Universal Application umgestellt werden, so ver-

wendet man anstatt der *classic/modern*-Option beim *generate app*-Befehl die neue Option *-starter=false*. Durch diese neue Option erzeugt Sencha Cmd die erforderliche Infrastruktur für die App, legt aber keinen *app*-Ordner an.

Eine Universal Application besitzt als wichtigste Ablageordner: *app*, *classic/src* und *modern/src*. Im *app*-Ordner befindet sich der mehrfach verwendete Code für Models und ViewController. Die *classic*- und *modern*-Verzeichnisse sind entsprechend bisherigen Ext-JS-5.1.x- und Sencha Touch-2.4.x-Apps aufgebaut. Diese Unterverzeichnisse enthalten den *model*-, *store*- und *view*-Ordner sowie oberhalb des *src*-Unterverzeichnis den zur App gehörigen *sass*-Ordner.

In diese *src*-Unterverzeichnisse übernimmt man die zu migrierenden *app*- und *sass*-Ordner der zugehörigen Ext-JS-beziehungsweise Sencha-Touch-App. Danach kopiert man den *resources*-Ordner sowie die Datei *Application.js* der Ext-JS-5.1.x-App in das Hauptverzeichnis der neuen Universal Application.

Sollen View-Models und View-Controller zwischen der Classic- und der Modern-App wiederverwendet werden, so genügt es, diese einfach in das Wurzelverzeichnis der neuen Universal Application zu verschieben.

Als Letztes erzeugt man mittels *sencha generate view-modern main.Main* im Hauptverzeichnis eine Haupt-View, einen ViewController und ein ViewModel innerhalb des *modern*-Teils der Universal Application und übersetzt die komplette Anwendung (Bild 5).

Performantes Theming on-the-fly mittels Fashion

Ext JS verwendet Sass (Syntactically Awesome Stylesheets), um CSS flexibel und äußerst dynamisch zu erstellen sowie die Wiederverwendung von Stylesheets zu ermöglichen. Für Version 6 des einheitlichen Ext-JS-Frameworks entwickelte

```
[INF] Application available at http://localhost:1841
[INF] Appending content to C:\Users\MSI_Laptop\bin\Sencha\MeineAnwd\bootstrap.js
[INF] Writing content to C:\Users\MSI_Laptop\bin\Sencha\MeineAnwd\modern.json
[INF] Copying page resources to C:\Users\MSI_Laptop\bin\Sencha\MeineAnwd\build\pro
[INF] Applying version stamps to json resource entries
[INF] Generating file deltas to C:\Users\MSI_Laptop\bin\Sencha\MeineAnwd\build\pro
[INF] Writing content to C:\Users\MSI_Laptop\bin\Sencha\MeineAnwd\build\production
[INF] Writing content to C:\Users\MSI_Laptop\bin\Sencha\MeineAnwd\build\production
[INF] Appending content to C:\Users\MSI_Laptop\bin\Sencha\MeineAnwd\build\producti
[INF] Building output markup to C:\Users\MSI_Laptop\bin\Sencha\MeineAnwd\build\pro
[INF] Writing content to C:\Users\MSI_Laptop\bin\Sencha\MeineAnwd\build\production
[INF] generating cache manifest to C:\Users\MSI_Laptop\bin\Sencha\MeineAnwd\build\
```

Testen: Mittels *sencha app build modern* stellt Sencha Cmd eine Anwendung zum Testen bereit (Bild 5)

Sencha einen neuen Sass-Compiler direkt in JavaScript, Fashion genannt. Diese Konzeption von Fashion erleichtert die Migration bestehender Themes für vorhandene Anwendungen, da diese nach wie vor auf Saas-Vorgaben basieren. Themes kann man deshalb aus vorhandenen Anwendungen in Ext JS 6 direkt und unverändert übernehmen.

Fashion wurde als Bestandteil von Sencha Cmd 6 realisiert, sodass man im Unterschied zu Ext JS Version 4 und 5 keine eigenständige Compass-Installation mit einem zusätzlichen Ruby-System mehr benötigt. Damit entfällt das bisher in der Regel recht zeitaufwendige Zusammenspiel von Ruby und Compass beim Übersetzen eines Themes. Daher lässt sich ein Compile-Vorgang mit Fashion wesentlich schneller durchführen. Zudem macht Fashion die bisherige Übersetzung zum Build-Zeitpunkt hinfällig, da der neue Saas-Compiler CSS sofort on-the-fly im Webbrowser übersetzt.

Lediglich die erste Ausführung des Compile-Vorgangs im Webbrowser dauert noch etwas länger; alle nachfolgenden Übersetzungsvorgänge gestalten sich jedoch wesentlich schneller. Damit kann auch das Anstoßen eines Refreshs im Webbrowser von Hand entfallen.

Um einen Compile-Vorgang mit Fashion zu aktivieren, erweitert man den URL der Anwendung (<http://localhost:1841>), um die Vorgabe eines Parameters *?platformTags=fashion:true*. Alternativ lässt sich Fashion auch über die Datei *app.json* beim *development*-Objekt als *tags*-Eintrag *fashion* einschalten. Setzt man zusätzlich den Befehl *sencha app watch* mit der Option *--fashion* ab, so befindet sich die Entwicklungsumgebung im Live-Upates-Mode, sodass Änderungen im Theme beinahe in Echtzeit in der App angezeigt werden.

Über einen weiteren Toolkit-Parameter im URL kann man direkt ein spezielles Theme anziehen. Beispielsweise führt der URL *host:1841/?toolkit=modern&platformTags=fashion:true* direkt das Modern Toolkit aus. Vorausgesetzt, der *sass*-Ordner des Modern Toolkits gibt in der *app.json*-Datei der *builds*-Konfiguration von Modern durch den Array-Eintrag *theme* das anwendungsseitig gewünschte Theme passend vor. Steht dort *theme-neptune*, so zieht Fashion das Default-Theme Neptune von Sencha Touch an.

Migration bereits vorhandener Themes

Um eine Anwendung auf Ext JS 6 umzustellen, genügt es nicht, nur den Quellcode zu berücksichtigen. Vielmehr muss man auch eventuell vorhandene Themes auf die neue Ver- ►

Workspaces als Basis der Wiederverwendung

Sencha Cmd nutzt Workspaces als Ausgangsbasis für die Mehrfachverwendung von Quellcode.

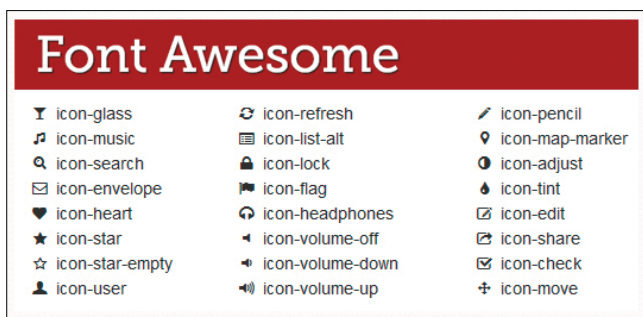
Seit Version 3.1 kennt Sencha Cmd Workspaces und Packages. Workspaces entsprechen einem Ordner, der mehrfach verwendbare Bestandteile einer Anwendung enthält. Packages wurden zwar ursprünglich speziell als Einheiten für Themes eingeführt, dienten aber bald als generelle Einheiten für Build und Deployment.

Nach dem Einrichten eines Projektordners erzeugt *sencha generate workspace* in diesem ein Workspace-Verzeichnis. Innerhalb eines Workspace kann Sencha Cmd mehrere Anwendungen verwalten, um Komponenten zwischen ihnen gemeinsam nutzen zu können.

sion 6 von Ext JS migrieren. Zunächst generiert man über Sencha Cmd mittels *generate theme* das Theme und kopiert anschließend den *sass/src*- und *sass/var*-Ordner inklusive der zugehörigen Resources in die passenden Zielverzeichnisse. Falls man für mehrere Anwendungen über Workspaces wiederverwendbare Teile im Theme einsetzt, überarbeitet man die Datei *package.json* und passt das *requires*-Array in *app.json* entsprechend an.

Über die *app.json*-Datei legt man ebenfalls über das *builds*-Array fest, welches Toolkit und welches Theme benutzt wird. Damit muss man nicht selbst auf die korrekte Verwendung von Theme und Toolkit bei Sencha-Cmd-Befehlen achten:

```
"toolkit": "classic", // oder "modern"
"builds": {
  "classic": {
    "toolkit": "classic",
```



Font Awesome stellt skalierbare und über CSS anpassbare Vector-Icons zur Verfügung (Bild 6)

```
"theme": "theme-name"
},
"modern": {
  "toolkit": "modern",
  "theme": "theme-name"
}
}
```

Ext JS 6 macht die bisher notwendige Ausführung des *slice*-Befehls (ein Bestandteil von Sencha Cmd bei der *fs*-Kategorie, um Images für bestimmte Bestandteile eines Custom-Themes zu generieren, damit ältere Webbrowser diese anzeigen können) hinfällig. Für diese Zwecke kommt jetzt Font Awesome mit entsprechenden Glyphen (grafische Darstellung des Schriftzeichens als Icon) im Theme anstatt Images zum Einsatz (Bild 6). Das macht nicht nur das Theme beim Herunterladen in der Anwendung kleiner und schneller, sondern, da die Glyphen vektor- und nicht pixelbasiert sind, auch perfekt skalierbar.

Neues Default-Theme Triton für beide Welten

Mit Version 6 ersetzte Sencha das bisherige Default-Theme durch ein neues, Triton genannt. Das neue Default-Theme Triton unterstützt die beiden Toolkits Classic und Modern

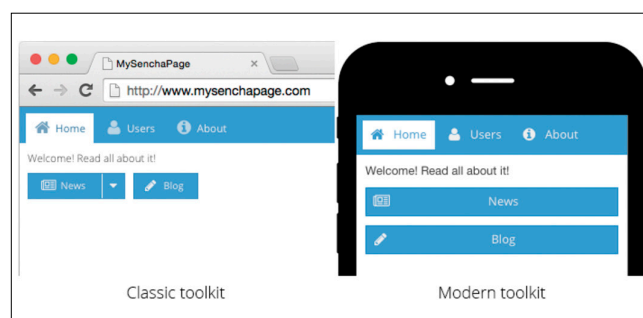
(Bild 7), sodass mit Ext JS 6 entwickelte Anwendungen über alle Plattformen nicht nur einheitlich arbeiten, sondern auch das gleiche Erscheinungsbild nach außen anbieten. Triton ist wie Neptune und Crisp (seine beiden Vorgänger) flach und minimalistisch ausgerichtet und benötigt wegen des Einsatzes von Font Awesome keine Images mehr für ältere Webbrowser.

Als Default-Font setzt Triton Open Sans ein, da diese Schrift freundlicher und optisch runder wirkt als die zuvor verwendeten anderen Schriften. Mit Triton verfolgt Sencha zudem das Ziel, Größendarstellungen auch für Desktop und Tablet zu optimieren. Dies gestaltete sich früher beim Desktop beziehungsweise Tablet schwieriger als bei den mobilen Endgeräten, da man über Sencha Touch mittels CSS-Layouts die Größen schon immer flexibler einstellen konnte. Das Triton-Theme basiert direkt auf Font Awesome und den Sencha Font Packages, die Ersteres nicht umfasst.

Sencha Font Packages bestehen konzeptionell aus dem ExtJS Font Package und dem Pictos Font Package; Letzteres war ursprünglich bereits im Sencha-Touch-Framework enthalten. Im ExtJS Font Package fasste Sencha einige Icons, Codes und Namen zu einer Einheit zusammen. Während das ExtJS Font Package mit dem Framework ausgeliefert wird, stellt das Pictos Font Package ein separates Package dar. Einer Anwendung macht man Pictos über die *package.json*-Datei durch den *requires*-Array-Eintrag *font-pictos* zugänglich. Den CSS-Style-Class-Namen für Pictos und Font Awesome vergibt man mittels der *iconCls*-Eigenschaft.

Vorgefertigte Schablonen für User-Interface-Designer

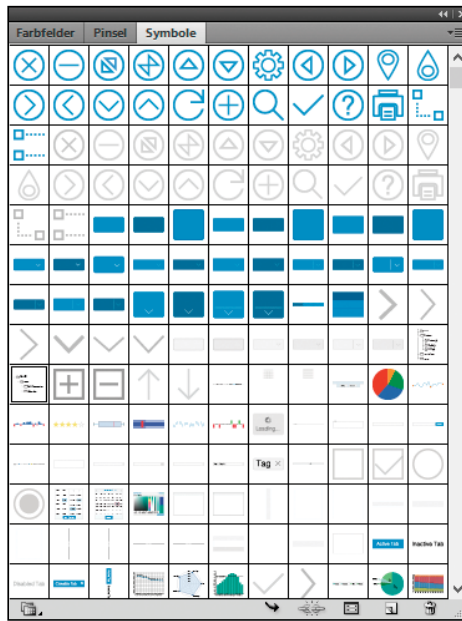
Zusammen mit Ext JS 6 entwickelte Sencha als neues Produkt einen Vorrat an sogenannten Schablonen, die Ext JS Stencils genannt werden. Mit ihnen können User-Interface-Designer Komponenten und andere Bestandteile der in einem Theme verwendeten Symbole und Zeichenstile an die eigenen Wünsche anpassen oder gänzlich neue definieren. Dabei lassen sich auch komplett neue Komponenten, die aus GUI-Elementen (zum Beispiel Schaltflächen) bestehen, und nicht nur Symbole anlegen. Derzeit unterstützt Ext JS Stencils ausschließlich das Classic Toolkit und die drei Themes Triton, Crisp und Neptune.



Triton bietet als standardmäßig voreingestelltes Theme auf dem Desktop/Laptop (Classic) und auf mobilen Endgeräten/Smartphones (Modern) ein einheitliches Erscheinungsbild (Bild 7)

Design-Komponenten eines jeden Themes befinden sich in einer separaten Datei. Für die Arbeit mit Ext JS Stencils benötigt man eines der Grafik-/Illustrationsprogramme Adobe Illustrator, Omnigraffle oder Sketch. Zudem muss der OpenType-Font FontAwesome als Schrifttyp installiert sein. Über die Website von Sencha zur Anforderung eines Download-Links der Ext-JS-Stencils-Schablonen muss man in der Dropdown-Liste *Select Version ...* die installierte Version des installierten Grafik-/Illustrationsprogramms auswählen. Um Fragen oder Probleme zu Ext JS Stencils diskutieren zu können, hat Sencha für User-Interface-Designer ein eigenes Forum eingerichtet.

Sencha legte beim Aufbau der Ext JS Stencils in Adobe Illustrator die Symbole als Vektoren an. Anschließend wandelte man diese Vektoren in Bildobjekte um, die auf einer 9-Slice-Skalierung basieren. Damit minimiert man die Dateigröße, spart Zeit beim Arbeiten mit den Symbolen und gewährleistet aufgrund der 9-Slice-Skalierung ihre korrekte Skalierung als Komponente einer Bedienoberfläche. Ein eigenes Symbol-Fenster in Illustrator fasst alle Bildobjekte übersichtlich zu einer Bildbibliothek (Bild 8) zusammen. Ändert man direkt ein Bildobjekt, so werden alle anderen bereits vorhandenen Instanzen innerhalb der Illustrator-Datei automatisch angepasst.



Adobe Illustrator zeigt alle Bildobjekte einer Symbol-Bibliothek im Symbol-Panel (Fenster > Symbole) an (Bild 8)

Asynchrone Programmierung jetzt nativ im Framework integriert

Für die Asynchrone Programmierung greift Sencha auf das Feature Promises and Deferreds der seit länger existierenden Erweiterungen von Deft JS zurück und integriert diese direkt in Ext JS 6. Die JavaScript-Bibliothek der Deft-JS-Erweiterungen basiert auf der CommonJS Promises/A-Spezifikation (dem De-facto-Standard), den auch die Ecma International für ihre Sprachspezifikation nutzt. Damit bleibt seitens des einheitlichen Frameworks die Konformität zum ECMAScript-6/2015-Standard erhalten, während Sencha einem Programmierer gleichzeitig die über Deferreds verbundenen weiteren Vorteile zugänglich macht.

Damit erübrigt sich die bisherige oft unübersichtliche Callback-Alternative für eine asynchrone Programmierung in JavaScript. Mit Ext JS 6 kann man erstmals Promises direkt in JavaScript als Objekte nutzen, die asynchrone Operationen kapseln und native implementieren. Somit verbessert sich in zukünftigen ECMAScript-6-kompatiblen Webbrowsern wesentlich die Performance, da diese quasi direkt Promises verstehen und man keine weitere Bibliothek für die Seite laden muss. In Ext JS 6 kann ein Promise derzeit über zwei verschiedene Vorgehensweisen erzeugt werden:

- **Ext.Promise():** Instanziert ein Promise-Objekt gemäß dem ECMA-Script-6-Standard; diese Vorgehensweise entspricht einem Aufruf mit dem *new*-Konstruktor der *Promise*-Klasse (*new Ext.Promise(function (resolve, reject))*).
- **Ext.Deferred():** Die zweite Vorgehensweise macht mittels *new Ext.Deferred()* einem Entwickler weitere nützliche Verbesserungen für die asynchrone Programmierung zugänglich.

Im Framework erfolgt die Realisierung beider Vorgehensweisen über eine Erweiterung des *Ext.data.request.Ajax()*-APIs von Ext JS. Dazu deklarierte Sencha in der Ext-JS-Klassenhierarchie über *Ext.Base* zwei neue Klassen *Ext.Promise* und *Ext.Deferred*. Während Erstere über die an *Ext.Promise()* übergebene Funktion sofort eine asynchrone Aktion startet,

erzeugt *Ext.Deferred()* zuerst ein *Deferred*-Objekt. Über dieses zusätzliche Objekt greift man mittels *deferred.promise* auf ein natives *Promise*-Objekt zu, womit indirekt neben dem *Deferred*- auch ein *Promise*-Objekt verfügbar ist.

Das *Ext.Deferred*-Objekt macht zusätzlich Management-Funktionen verfügbar, um den Zustand eines asynchronen Prozesses verwalten, abfragen und beeinflussen zu können. Im Unterschied hierzu kennt das *Ext.Promise*-Objekt nur ausgewählte Zustände quasi als Callback-Hook wie *success*, *failure* et cetera, die der Entwickler nutzen kann. Damit fasst das *Deferred*-Objekt alle asynchronen Aktivitäten unter eine gemeinsame Hülle zusammen; es stellt softwaretechnisch ►

Responsive Design

Ein Paradigma zur Erstellung eines User-Interfaces, das entsprechend der Bedienung auf aktuelle Gegebenheiten des jeweiligen Geräts geeignet reagiert.

Anfangs nur für die Gestaltung von Websites gefordert, erwarten inzwischen viele Endbenutzer die Erfüllung der mit Responsive Design verbundenen Anforderung auch für Mobile-Apps.

Ext JS unterstützt seit Version 5 mittels eines speziellen Features *responsiveConfig* die Anpassung des User-Interfaces auf dynamische Änderungen (Auflösung/Orientierung des Bildschirms).

Responsive Design soll das einheitliche Anzeigen von Inhalten im User-Interface gewährleisten. Mit den neuen Webstandards HTML5, CSS und JavaScript steht die technische Basis dafür bereit.

gesehen einen Wrapper dar. Der Wrapper verbirgt dank seines Interfaces zwar die Komplexität, trotzdem sollte man das *Deferred*-Objekt als nichtöffentliche/private Schnittstelle des *Promise*-Objekts betrachten. Das *Promise*-Objekt stellt die öffentliche Sicht auf das *Deferred*-Objekt dar und sollte daher vorrangig zum Einsatz kommen.

Application Templates als fertige Vorlagen für Quellcode

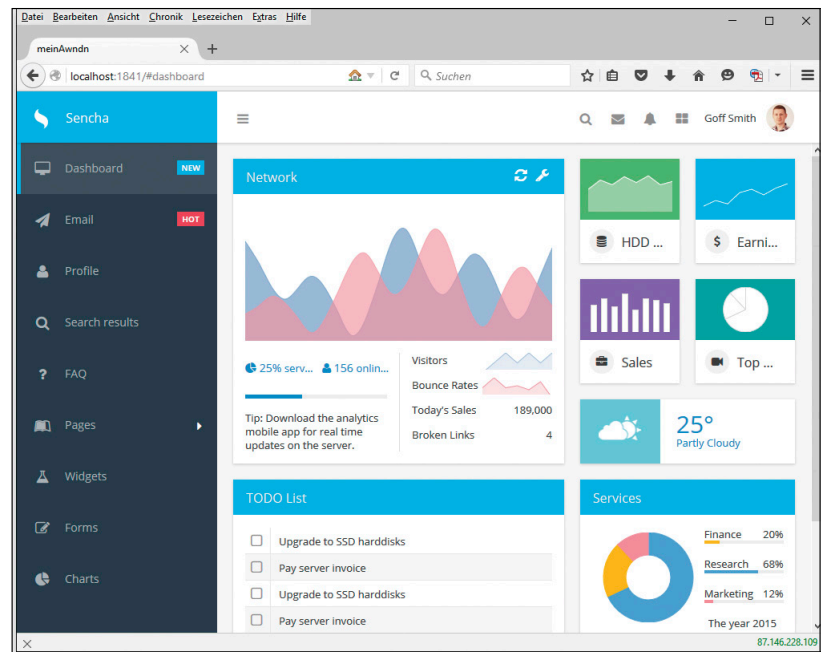
Den Begriff Templates findet man in der Sencha-Welt an unterschiedlichen Stellen für verschiedene Zwecke. So gibt es in beiden Frameworks (Ext JS und Sencha Touch) *Ext.XTemplates*, die ein pre-compiltes HTML-Fragment zur Verbesserung der Performance darstellen.

Ferner unterstützt der GUI-Builder Sencha Architect sogenannte App Templates als Basis der Wiederverwendung von GUI-Bestandteilen. Neu in Version 6, dem vereinheitlichten Framework, verwendet Sencha einen ähnlichen Begriff, Application Templates (kurz: App Templates oder Starter Apps genannt), für vorgefertigte Apps, die zusammen mit Ext JS ausgeliefert werden.

Diese neue, weitere Art von App Templates zielt auf eine schnelle Demonstration der Leistungsfähigkeit des Frameworks im Sinne von Prototyping ab.

Ein App Template besteht aus dem vollständigen Quellcode einer eigenständig lauffähigen App, den man (vorausgesetzt, man besitzt Grundkenntnisse über Ext JS) an die eigenen Vorstellungen anpassen kann. Im Idealfall muss man lediglich die Oberfläche und die Datenschnittstelle der App verändern. Sicherlich beabsichtigt Sencha, die mit Ext JS ausgelieferten Apps in Zukunft auch dem eigenen GUI-Tool Sencha Architect verfügbar zu machen. Derzeit allerdings unterstützt Sencha Architect leider noch nicht das vereinheitlichte Framework Ext JS 6.

In der aktuellen Ext-JS-6.x-Auslieferung befinden sich zurzeit nur zwei App Templates im *templates*-Ordner des Installationsverzeichnisses: das Admin Dashboard und das Executive Dashboard. Während Admin Dashboard über eine Web-App vielfältige Administrationsaufgaben bereitstellt; enthält



Webbrowser: Um ein App Template wie das Admin Dashboard im Webbrowser auszuführen, muss der Webserver gestartet sein (Bild 9)

das Executive Dashboard eine Tablet-App als Beispiel für die Umsetzung von Responsive Design. Über die Homepage der Ext-JS-Beispielsammlung kann man sich direkt Funktionalität und Arbeitsweise eines Application Templates ansehen und diese erkunden.

Sencha plant, diese beiden zukünftig um weitere fertig programmierte App Templates zu erweitern. Somit will man einen Fundus in Anlehnung an die über die Beispiel-Homepage zugänglichen Combination Examples für App Templates aufbauen.

Die Vorgehensweise der Wiederverwendung eines App Templates gestaltet sich immer gleich:

- **App-Generierung:** Zuerst generiert man sich über Sencha Cmd eine App in einen passenden Ordner mit dem gleichen Namespace wie das App Template (*sencha -sdk %ext_home% app Namespace Ordner*). Die Bezeichnung des

Pivot-Tabellen für die Analyse von Massendaten

Eine Pivot-Tabelle reduziert große Datenmengen auf eine überschaubare Größe und wertet diese im Sinne einer Big-Data-Analyse aus.

Eine Auswertung dieser Massendaten verändert keine Ausgangsdaten der Pivot-Tabelle. Somit dienen Pivot-Tabellen lediglich zur Abfrage und nicht zur Erzeugung oder Änderung von Daten.

Version 6 verfügbar in drei neuen Produkt-Editionen

Seit der Markteinführung von Version 6 gibt es neben der kostenlosen Open-Source-Version auch drei kommerzielle Ext-JS-Editionen: Standard, Pro und Premium.

Das Open-Source-Paket enthält das Basis HTML5/JavaScript-Framework inklusive der App Templates. Zusätzlich umfasst die Standard-Edition Sencha Cmd, Ext JS Stencils (Design Kit) und eine auf ein Jahr begrenzte Wartungs- und Support-Zeit über ein geschlossenes Sencha-Forum. Des Weiteren kommen zur Pro-Edition die IDE-Plug-ins (für JetBrains, Eclipse und Visual Studio) sowie für die Premium-Edition zusätzlich noch das Pivot Grid und der Sencha Inspector hinzu.

Namespace findet man in der JavaScript-Datei *Base* beim *schema*-Eintrag *namespace*.

- **Kopieren des Quellcodes:** Danach wählt man über den *template*-Ordner der Ext-JS-Installation das gewünschte App Template aus, kopiert dessen gesamten Inhalt in den Ordner der zuvor generierten App und überschreibt alle bereits vorhandenen Dateien.
- **Refresh der Metadaten:** Im nächsten Schritt aktualisiert man alle Metadaten der neuen App über *sencha app refresh*.
- **Build der App:** Im letzten Schritt macht man durch *sencha app build* das App Template dem eigenen Webserver zugänglich (Bild 9).

Neuerungen beim Tabellen-Handling und Pivot Grid

Die Zusammenfassung der beiden Frameworks zu Ext JS 6 brachte erstmals die Unterscheidung zwischen Classic Grid und Modern Grid beim Tabellen-Handling mit sich. So wurde die Klassenbibliothek des Modern Toolkits umstrukturiert, um als Basis das Common Core nutzen zu können. Als wesentliche Neuerung erhielt eine Zelle im Classic Grid das Attribut *extensible*; ist es auf *true* gesetzt, kann die Zelle horizontal oder vertikal analog einer Zelle in einer Tabelle erweitert werden.

Die Funktionalität beim Modern Grid befindet sich nun in der Unterklasse *Ext.dataview.List*. Das Grid selbst stellt eine Komponentenliste dar, bei der jeder Eintrag auf *Ext.grid.Row* zurückgreift. Zellen einer Row-Komponente entsprechen Instanzen von *Ext.grid.cell*, einer Zelle kann man Widgets über die *widgetcell*-Methode zuordnen.

Im Frühjahr 2015 erwarb Sencha die Pivot-Technologie *mzPivotGrid* von *mz<Solutions>*, einem Softwarehaus aus Rumänien. Dieses bereits für Ext JS 4.x und 5.x verfügbare Produkt bietet einem Entwickler die Funktionalität der aus MS Excel bekannten Pivot-Tabellen. Ausgehend von den bekannten Ext JS Grids lassen sich mit Pivot Grids Business-Intelligence-Features für Ext-JS-Anwendungen realisieren, die Ext-JS-Charts zur Visualisierung der Daten nutzen.

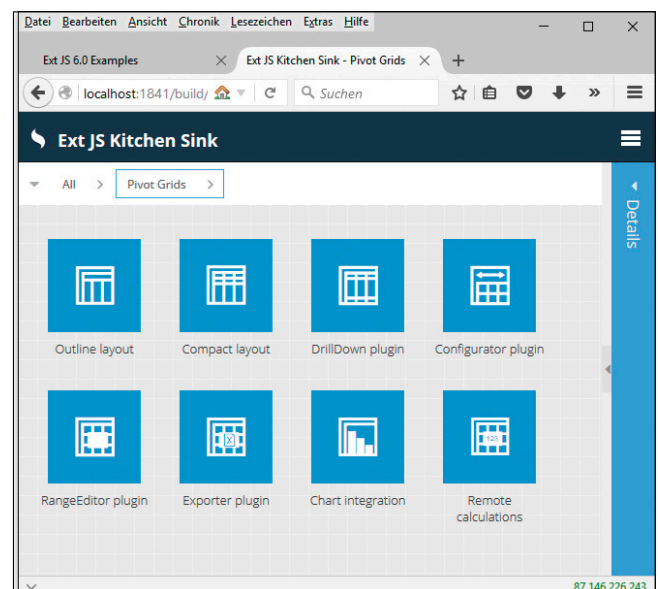
Das neue Pivot Grid steht als eigenständiges Add-on für Ext JS 5 zur Verfügung und ist ausschließlich über die Premium-Edition von Ext JS 6 für das Classic Toolkit erhältlich. Die Integration von *mzPivotGrid* erfolgte mittels eines neuen *pivot-package* über den Namespace *Ext.pivot*. Gleichzeitig wurde das bisher für den Excel-Export verfügbare Plug-in durch ein neues *pivotexporter*-Plug-in ersetzt.

Grundsätzlich muss man für Testzwecke von Ext-JS-Produkten eine Download-Anfrage über die Trial-Website (Try Sencha Ext JS for 30 Days) absetzen. Als Antwort erhält man eine E-Mail von Sencha mit verschiedenen Download-Links. Über diese E-Mail lädt man für Testzwecke das neue Pivot Grid entweder direkt als Evaluierungsversion von Ext JS Premium oder als eigenständiges Add-on für Ext JS 6 herunter. Nach dem Entpacken der ZIP-Datei und Anlegen einer neuen App mittels *Sencha Cmd* kopiert man alle Ordner und Dateien aus *packages* in den gleichnamigen Ordner (*packages*) der neu generierten Ext-JS-App. Danach steht der volle Funktionsumfang des Pivot Grid für die Anwendungsent-

wicklung in Ext JS auf Basis des Frameworks zur Verfügung. Zum Testen der Beispiele des Pivot Grid von Kitchen Sink (Bild 10) genügt ein Refresh und der anschließende Build der App mittels *Sencha Cmd*.

Direkte IDE-Anbindung mittels dreier neuer Plug-ins

Sencha richtet sich mit seinen Produkten bewusst auf die Entwicklung von Enterprise-Anwendungen aus. In diesem Umfeld setzen Entwickler für eine produktivere und effiziente Arbeitsweise immer eine integrierte Entwicklungsumgebung (Integrated Development Environment, IDE) ein. Zusammen mit IDE und Plug-in stehen einem Programmierer sämtliche Features der jeweiligen IDE und direkt auch alle Ext-JS-/Sencha-Touch-Besonderheiten zur Verfügung. Sencha bietet seit mehreren Jahren sowohl für Sencha Touch als auch für



Die Anwendungssammlung Kitchen Sink der Ext-JS-Installation enthält mehrere Fallbeispiele zum Pivot Grid (Bild 10)

Ext JS ein Eclipse-Plug-in an. Mit der Vereinheitlichung beider Frameworks baut das Softwarehaus die IDE-Unterstützung mit zwei neuen Plug-ins weiter aus, sodass mit Version 6 von Ext JS insgesamt drei IDE-Plug-ins verfügbar sind:

- **JetBrains-Plug-in:** Sencha macht damit Ext JS 6 für die gängigsten IDE-Produkte von JetBrains verfügbar. Jetzt kann man Ext JS zusammen mit Java (IntelliJ), HTML/JavaScript (WebStorm), PHP (PhpStorm), Ruby (RubyMine) oder Python (PyCharm) nutzen.
- **Visual-Studio-Plug-in:** Sencha trat dem Visual Studio Programm von Microsoft als Premier Level Partner bei. In diesem Rahmen erschließt Sencha das Ext-JS-Framework Webentwicklern für die .NET-Umgebung. Derzeit stellt Sencha ein IDE-Plug-in für die beiden Editions Enterprise und Community von Visual Studio bereit.
- **Eclipse-Plug-in:** Sencha programmierte die bisherige Version des Plug-ins neu und eliminierte alle Abhängigkei- ►

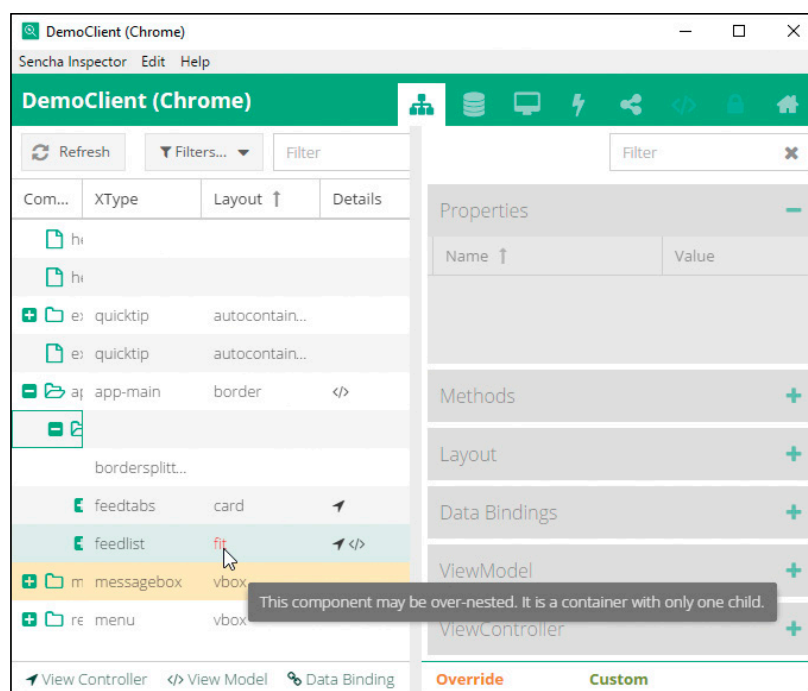
ten von anderen Software-Bibliotheken. Somit lässt sich das Eclipse-Plug-in auch in allen gängigen, auf Eclipse-basierenden IDEs wie JBoss Developer Studio, MyEclipse oder Spring Tool Suite nutzen. Speziell unterstützt dieses Plug-in das ältere Lune- sowie das neue Mars-Release und die in Entwicklung befindliche Neon-Version von Eclipse.

Sencha vertreibt zusammen mit der neuen Ext-JS-6-Version die drei IDE-Plug-ins (als sogenannte Productivity Tools) in der Pro- und Premium-Edition von Ext JS 6. Leider kann man für die frei zugängliche GNU-Version von Ext JS keines der drei IDE-Plug-ins separat erwerben. Dafür lassen sich die IDE-Plug-ins zu Testzwecken über eine Download-Anforderung – »Try Sencha Ext JS for 30 Days« – für dreißig Tage nutzen. Sencha gewährleistet für alle IDE-Plug-ins eine Rückwärtskompatibilität von Ext JS bis zur Version 4.2.0 und von Sencha Touch bis Version 2.1.0. Zudem macht das Plug-in für die Entwicklung in Teams die Versionskontrolle sowohl für das Ext-JS- als auch das Sencha-Touch-Framework (über die IDE) zugänglich.

Optional kann man jedes der drei IDE-Plug-ins zusammen mit Sencha Cmd ab Version 4 einsetzen. Einige IDE-Features wie Ext JS Workspaces, Ext JS Packages, Generierung eines neuen Projekts oder einer neuen Ext-JS-App setzen zwingend die Installation einer Sencha-Cmd-Version voraus. Leider gehört dazu auch ein wichtiges IDE-Feature: das schnelle Testen von Änderungen im Quellcode während der Entwicklung, das dem Sencha Cmd *app watch* entspricht. Daher empfiehlt es sich immer, um den vollen Funktionsumfang der jeweiligen IDE nutzen zu können, ein Plug-in zusammen mit Sencha Cmd zu installieren.

Intelligentes Ext-JS-basiertes Debugging und Theming

Seit beinahe zwei Jahren entwickelt Sencha einen Debugger speziell für Framework-Anwendungen, anfangs App Inspector genannt, mit Ext JS 6 unter neuer Bezeichnung als Sen-



Mittels Filter kann man im Debugging-Fenster gezielt nach bestimmten Werten/Namen oder Komponenten in der App oder im Framework suchen (Bild 11)

cha Inspector freigegeben. Ursprünglich war der Debugger als reine Erweiterung der Google Chrome Dev Tools gedacht; inzwischen ist er aber als eigenständige Desktop-Anwendung für Mac OS X und Windows (32 und 64 Bit) verfügbar; eine Linux-Version befindet sich noch in Entwicklung. Browserseitig unterstützt Sencha Inspector nicht nur Chrome, sondern auch Firefox, Internet Explorer (ab Version 10) und Safari (ab Version 6). Dabei lässt sich Sencha Inspector für Anwendungen einsetzen, die mit Ext JS ab Version 4.2.0 und Sencha Touch ab Version 2.1.0 entwickelt wurden.

Im Unterschied zu den für HTML/JavaScript verfügbaren Debuggern bietet Sencha Inspector ein intelligentes Debugging und Theming. Das Tool kennt sämtliche Bestandteile der Anwendungsarchitektur einer App wie Klassen, Views, Stores, Models oder Proxies; zudem auch sämtliche Internas wie Komponenten, XType oder Layout des Ext-JS/Sencha-Touch-Frameworks.

Sass-Compiler Fashion

Ab Ext JS 6 ermöglicht Sencha Inspector einem Entwickler, geänderte Bestandteile des aktuellen Themes einer App direkt anzuzeigen. Dieses Theming realisiert Sencha Inspector über den mit Version 6 von Ext JS bereitgestellten neuen Sass-Compiler Fashion und dessen Live-Update-Feature.

App und Sencha Inspector laufen auf verschiedenen Webservern, die über eine WebSocket-Verbindung kommunizieren. Sencha unterstützt automatisches Logging über die JavaScript-Konsole sowie mehrere Vorgehensweisen für den Verbindungsaufbau zwischen einer App und Sencha Inspector.

Die gewählte Vorgehensweise hängt von verschiedenen Faktoren ab: beispielsweise, ob die App bereits auf den Produktionsserver verteilt wurde, oder ob sie sich noch in Ent-

Verfügbarkeit und Installation von Sencha Inspector

Sencha Inspector wird als Bestandteil nur mit Ext JS Premium ausgeliefert und lässt sich zu Testzwecken 30 Tage per Download für Ext JS 6 nutzen.

Die Installation setzt eine Java Runtime Environment (JRE) ab Version 1.7 voraus; zusätzlich muss ein Zugriff durch die Firewall des Betriebssystems sowie ein Sencha-Account eingerichtet werden. Einen Sencha-Account erhält man, wenn man sich als Benutzer des Sencha-Forums registriert.

wicklung befindet, oder ob die Mobile App in einem Container wie Sencha Web Application Manager oder einem mobilen Webbrowser läuft. Klickt man nach Start des Sencha Inspector über die Toolbar ganz rechts auf das Icon *Connect your app* ..., so baut der Debugger eine Verbindung über den Port 1839 zur über die Installation bereitgestellten Demo-App auf.

Danach erscheint im Sencha Inspector unterhalb der Toolbar eine neue Zeile, die den verwendeten Browser als Icon, den Namen der zu untersuchenden Anwendung, darunter den URL und rechts das bei der Entwicklung verwendete Sencha Framework inklusive Versionsnummer anzeigt. Mittels des URL lässt sich die App auch in jedem anderen Browser starten. Ein Klick auf diesen Zeileneintrag im Hauptfenster von Sencha Inspector öffnet ein Debugging-Fenster mit näheren Informationen über die Anwendung (Bild 11).

Mit Hilfe des Debugging-Fensters erkundet man auch detailliert die aktuell vorhandenen Einstellungen der App. Sencha Inspector erkennt häufige Problemfälle und hebt diese durch eine rote Kennzeichnung hervor. Ihre Anzeige in einer automatisch erstellten Liste unterstützt Sencha Inspector derzeit noch nicht; allerdings erscheinen Ratschläge zur Fehlerursache.

Links zum Thema

- Homepage von Sencha Ext JS
<https://www.sencha.com/products/extjs>
- Homepage von Sencha Touch
<https://www.sencha.com/products/touch>
- Homepage von Sencha Cmd
<https://www.sencha.com/products/sencha-cmd>
- Download-Site für GNU General Public License von Ext JS
<https://www.sencha.com/legal/gpl>
- Download-Site für Sencha Cmd
<https://www.sencha.com/products/extjs/cmd-download>
- Installationshandbuch für Ext JS 6
<http://se.sencha.com/setup-guide>
- Homepage von Font Awesome
<http://fontawesome.io>
- Download-Anforderung – Ext JS Stencils
<http://pages.sencha.com/Stencils.html>
- Homepage der Ext-JS-/Sencha-Touch-Erweiterungen von DeftJS
<http://deftjs.org>
- Homepage der Beispielanwendungen von Ext JS 6
<http://examples.sencha.com/extjs/6.0.0/examples>
- Download-Anforderung – Try Sencha Ext JS for 30 Days
<https://www.sencha.com/products/extjs/evaluate>
- Einrichtung eines Sencha-Accounts durch Forums-Registrierung
<https://www.sencha.com/forum/register.php>

Mit der neuen Ext JS Version 6 bietet Sencha einem Entwicklerteam erstmals eine Auswahlmöglichkeit zwischen verschiedenen Strategien beziehungsweise Vorgehensweisen in der Anwendungsentwicklung. Mit einem Toolkit-Ansatz verfolgt Sencha das Ziel, den Grad an Wiederverwendbarkeit des Quellcodes zu maximieren. Daraus resultieren automatisch weitere Vorteile für die Wartung, Weiterentwicklung und Wirtschaftlichkeit. Vor allem sinkt innerhalb der verschiedenen Projekte auch der Bedarf, unterschiedliches Programmier-Know-how im Entwicklungsteam vorzuhalten. Grundsätzlich unterstützt Ext JS 6 die beiden wichtigsten Entwicklungsstrategien:

- **Mobile First:** Statt Websites vorrangig für Desktop-Systeme zu entwickeln und anschließend auf kleinere Displays zu adaptieren, arbeitet der Mobile First-Ansatz andersherum: Zuerst entwickelt man für Mobilgeräte und danach für Desktops. Bei dieser Strategie sollte der Fokus auf den Anforderungen Inhalte, Navigation und Performance liegen. Nachgelagert portiert man die Mobile App auf größere Displays wie Tablet und Desktop (Progressive Enhancement).
- **Graceful Degradation:** Ausgehend von bereits länger im Unternehmen eingesetzten Websites bestimmt man diejenigen Funktionen/Anforderungen, die Endbenutzer auf einem Mobilgerät benötigen. Anschließend priorisiert man diese Kernfunktionen und portiert sie vom Desktop auf Tablets und verschiedene Mobilgeräte.

Bei der Auswahl der geeigneten Entwicklungsstrategie sollte sich das Projekt an den tatsächlichen Gegebenheiten wie vorhandene Anwendungen, Know-how des Entwicklungsteams, Infrastruktur oder Budget/Kosten orientieren. Ext JS 6 drängt den Entwicklern keine fixe Vorgehensweise auf, sondern erlaubt es, eine individuell passende Strategie eigenständig zu wählen. Ext JS 6 unterstützt die beiden wichtigsten Vorgehensweisen (Mobile First/Graceful Degradation) gleichwertig und flexibel.

So kann die Anwendung beim Start bereits Device Profile für ausgewählte Geräte anziehen, zur Laufzeit über Platform Configs spezielle Anpassungen aufgrund der aktuellen Geräteklassifikation vornehmen, ein Responsive Design zur Laufzeit der App mittels Responsive Config realisieren oder über Device Themes dem Endbenutzer das entsprechend dem jeweiligen Betriebssystem gewohnte Look and Feel anbieten. Aufgrund dieser flexiblen Einstellungsvarianten eignet sich das Ext-JS-Framework von Sencha besonders für die professionelle Entwicklung von Enterprise-Apps. ■



Frank Simon

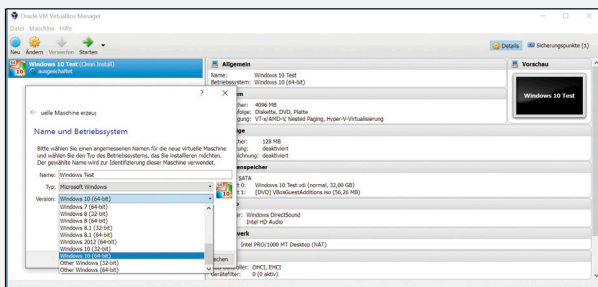
Frank Simon arbeitet in der Software-Entwicklung mit den Arbeitsgebieten Entwicklung, Programmierung, Test und Debugging von Cloud-, Rich-Internet-, Mobile- und Webanwendungen inklusive deren Systemmanagement.
web_mobile_developers@gmx.eu

ÜBERBLICK

CD-Highlights

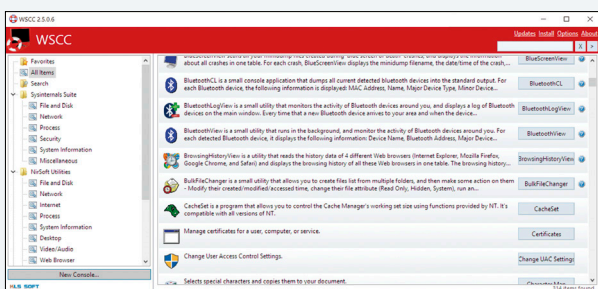
Auf der Heft-CD finden Sie diesmal Tools für mehr Sicherheit und für Netzwerk-Profis.

Oracle VirtualBox 5.0.12



Die kostenlose Software stellt Ihnen virtuelle PCs (VPCs) zur Verfügung, auf denen Sie beliebige Windows- und Linux-Betriebssysteme und Anwendungen installieren und gefahrlos testen können. Mit Hilfe eines solchen VPC probieren Sie dann etwa Linux-Distributionen oder Live-CDs/-DVDs direkt unter Windows aus, ohne den echten PC neu zu starten oder vorhandene Windows-Installationen zu überschreiben. Auch neue oder unbekannte Software testen Sie in der isolierten Umgebung. Schädlinge bleiben im virtuellen PC und können keinen Schaden anrichten. Auf Mausklick setzen Sie einen virtuellen PC zurück und laden einen zuvor gesicherten Systemzustand.

Windows System Control Center 2.5.0.6



Das Windows System Control Center – kurz WSCC – ist eine Tool-Sammlung, die vor allem für versierte Anwender und Netzwerkprofis interessant ist. Mit dem WSCC hat man Zugriff auf über 250 System-Tools. Ein Update-Manager hält die Hilfsprogramme aktuell. Das Hauptmenü von WSCC ist übersichtlich. In der linken Spalte sind die Tools nach Herstellern in die Rubriken *Sysinternals Suite*, *NirSoft Utilities* und *Windows* sowie in zahlreiche Unterrubriken eingeteilt.



Weitere Highlights:

AkelPad 4.9.7

Der Gratis-Texteditor bietet volle Unicode-Unterstützung und kann neben dem klassischen Einfenster-Modus auch im Mehrfenster-Modus geöffnet werden. Gute Extras sind das Vergrößern und Verkleinern der Ansichtsgröße mit dem Mause, das Konvertieren von Schreibweisen sowie die Suchen- und Ersetzen-Funktion.

Double Commander 0.6.6

Der kostenlose Dateimanager bringt viele praktische und hilfreiche Funktionen sowie Shortcuts mit. Double Commander bietet zwei nebeneinanderliegende Fenster sowie die Möglichkeit, zusätzliche Reiter zu öffnen, um bequem zwischen mehreren Ordnern zu wechseln. ZIP- und RAR-Archive entpackt das Tool automatisch.

Free File Sync 7.8

Das Gratis-Tool synchronisiert Ordner auf Festplatten, USB-Sticks, NAS-Servern und Online-Speichern. Dank eingebauter Versionskonfliktkontrolle werden keine Dateien versehentlich überschrieben. Das Tool erleichtert es, Daten an unterschiedlichen Speicherorten auf dem gleichen Stand zu halten.

Network Scanner 6.1

Das Programm zeigt unter anderem offene TCP-Ports sowie freigegebene Dateien und Geräte im Netzwerk an. Außerdem prüft das Tool vom Benutzer vorgegebene Ports und ermittelt die aktuelle IP-Adresse des Rechners. Auch ein Ping auf andere Rechner ist möglich, um deren Erreichbarkeit zu prüfen.

Synwrite 6.19.2170

Ein Quelltext-Editor für über 80 Programmiersprachen von ADA bis XML. Ein Highlight ist die automatische Syntaxhervorhebung während der Eingabe. Synwrite richtet sich vor allem an Software-Entwickler und Webdesigner. In großen Dateien helfen Lesezeichen sowie Such- und Textmarken. Außerdem gibt es einen FTP-Client.

Jetzt kostenlos testen!



**2x
gratis!**



Praxiswissen für Entwickler!

Testen Sie jetzt 2 kostenlose Ausgaben und erhalten Sie exklusiven Zugang zu unserem Archiv.

webundmobile.de/probelesen

FOUNDATION 6

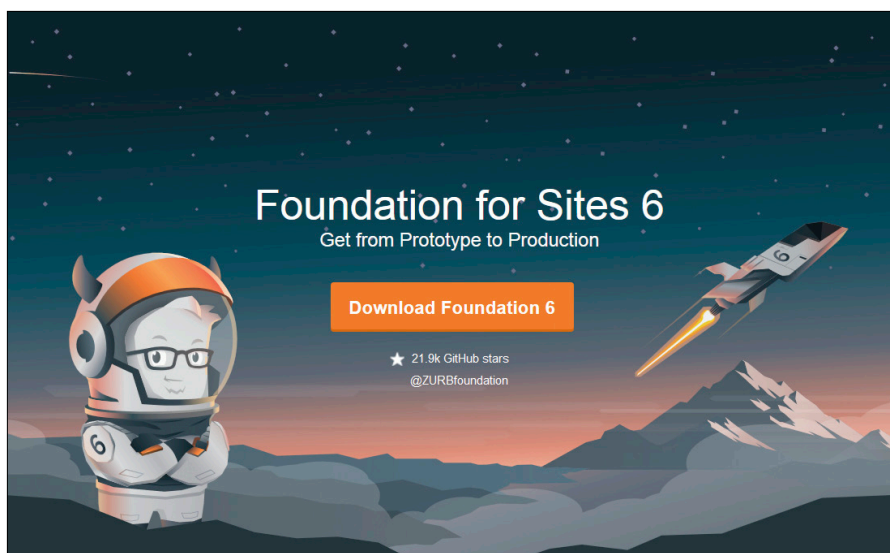
Die neue Basis

Das Framework Foundation 6 hat Starter-Templates mit nützlichen Tools an Bord.

Foundation – von seinen Entwicklern als fortschrittlichstes Frontend-Framework bezeichnet – ist in Version 6 (**Bild 1**) grundlegend überarbeitet worden und bringt vielversprechende Neuerungen mit:

- **Schlanker:** Die Dateigröße wurde auf die Hälfte reduziert, insgesamt sind es jetzt 60 KByte CSS und 84 KByte JavaScript. Sie können die Dateigröße weiter verringern, indem Sie nicht benötigte Komponenten entfernen.
- **Schnellstartoptionen:** Zum Starten und zur komfortablen Erstellung von Projektvorlagen bringt Foundation 6 die Foundation CLI und Yeti Launch mit – mit beiden beschäftigen wir uns noch genauer.
- **Bewegungsbibliothek:** Speziell für Animationen und Transitions gibt es Motion UI, eine Sass-basierte Animationsbibliothek.
- **Alles für die Zugänglichkeit:** Foundation integriert konsequent die für die Accessibility so wichtigen ARIA-Attribute.
- **Mehr Rasteroptionen:** Das bewährte responsive Raster ist auch weiterhin Teil von Foundation 6. Alternativ dazu lässt sich aber das Raster über Flexbox umsetzen, wodurch Sie mehr Möglichkeiten für die Anordnung und Ausrichtung von Elementen haben.

Sehen wir uns die einzelnen Neuerungen im Folgenden genauer an.



Die Projektseite von Foundation 6 (**Bild 1**)

Es gibt mehrere Möglichkeiten, wie Sie an den Code von Foundation kommen. Prinzipiell können Sie wählen, ob Sie mit CSS pur arbeiten wollen oder lieber auf Sass setzen; darüber hinaus gibt es Starterprojekte mit weiteren Komponenten und Tools.

CSS-basierte Varianten

Beginnen wir mit den CSS-basierten Varianten: Die Variante Complete beinhaltet den vollständigen CSS- und JavaScript-Code. Essential ist die kleinere Variante, bei der nur die wichtigsten Komponenten mit dabei sind. Bei Custom können Sie selbst auswählen, welche Bestandteile integriert sein sollen

(**Bild 2**).

Foundation selbst ist mit Sass erstellt. Den Sass-Code können Sie wahlweise über NPM, Bower, Meteor oder Composer herunterladen. Vollständige Starter-Projekte erstellen Sie am besten über Foundation CLI oder Yeti Launch (MacOS), darin integriert ist dann auch schon ein Sass-Compiler.

Sehen wir uns die Installation und die Optionen, die Ihnen bei Foundation CLI (**Bild 3**) zur Verfügung stehen, einmal genauer an.

Foundation CLI setzt Node.js und einen Git-Client voraus, die Sie gegebenenfalls zuerst auf Ihrer Arbeitsumgebung instal-

Pick and choose the features you want or have the whole enchilada.

Choose Your Components:

☒ All Foundation Components

Grid:

☒ Grid

☐ Flex Grid

☐ No Grid

General:

☒ Typography

☒ Forms

☒ Controls

☒ Buttons

☒ Close Button

☒ Button Group

☒ Float Classes

☒ Visibility Classes

☒ Slider

☒ Switch

Set Your Defaults:

The Grid

of Columns:

Gutter:

Max-Width:

Colors

Primary Color:

Alert Color:

Body Font Color:

Secondary Color:

Success Color:

Header Font Color:

Angepasste Foundation-Version erstellen (**Bild 2**)

```

C:\Users\Florence\entwicklung\foundation_1>Foundation new
? What are you building today? A website (Foundation for Sites)
? What's the project called? meinefoundation
? Which template would you like to use? Basic Template: includes a Sass compiler

  /  \  /  \  /  \  /  \  /  \
 /    \/    \/    \/    \/    \
[  --  ]  [  --  ]  [  --  ]
 \    /\    /\    /\    /\    /
  \  /  \  /  \  /  \  /  \

Thanks for using ZURB Foundation for Sites!
-----
Let's set up a new project.
It shouldn't take more than a minute.

Downloading the project template...
Done downloading!
Installing dependencies...

```

Foundation CLI setzt Node.js und einen Git-Client voraus (Bild 3)

lieren müssen. Dann können Sie Foundation CLI über die Eingabeaufforderung laden. Geben Sie dazu das folgende Kommando ein:

```
npm install -g foundation-cli
```

Nun lässt sich über *foundation new* ein neues Projekt anlegen. Dabei können Sie die Art des Projekts wählen – neben Foundation für Webseiten, welches das Thema dieses Artikels ist, gibt es auch noch Foundation für E-Mails und Foundation für Apps. Außerdem müssen Sie einen Namen für Ihr Projekt vergeben.

Schließlich können Sie sich zwischen zwei Projektvorlagen entscheiden: dem Basic-Template oder dem Foundation-Template. Beide Template-Typen setzen auf Sass, beim Foundation-Template sind allerdings weitere Tools wie Panini oder Browsersync integriert – dazu gleich mehr.

Wenn Sie das Basic-Template gewählt haben, finden Sie dann Ihre Projektvorlage inklusive *index.html*-Beispieldatei. Allerdings müssen Sie das Stylesheet erzeugen, auf das verwiesen wird. Dafür wechseln Sie in das Verzeichnis Ihres Projekts und schreiben *npm start*.

Yeti Launch

Noch komfortabler als Foundation CLI ist Yeti Launch (Bild 4), das eine grafische Oberfläche für die Arbeit mit Projekten bereitstellt. Wenn Sie Yeti Launch installiert haben und starten, müssen Sie den Namen Ihres Projektordners angeben.

Außerdem werden Sie aufgefordert, einen Notable-Account anzulegen. Notable ist ein kostenpflichtiger Dienst der Firma Zurb, über den Sie Prototypen für Diskussionen veröffentlichen können. Diesen Schritt können Sie aber auch auf später verschieben.

Nun können Sie Ihr Projekt benennen, die Art des Projekts wählen und das Template bestimmen. Damit kann Yeti Launch Ihr Projekt anlegen. Über die Oberfläche von Yeti Launch

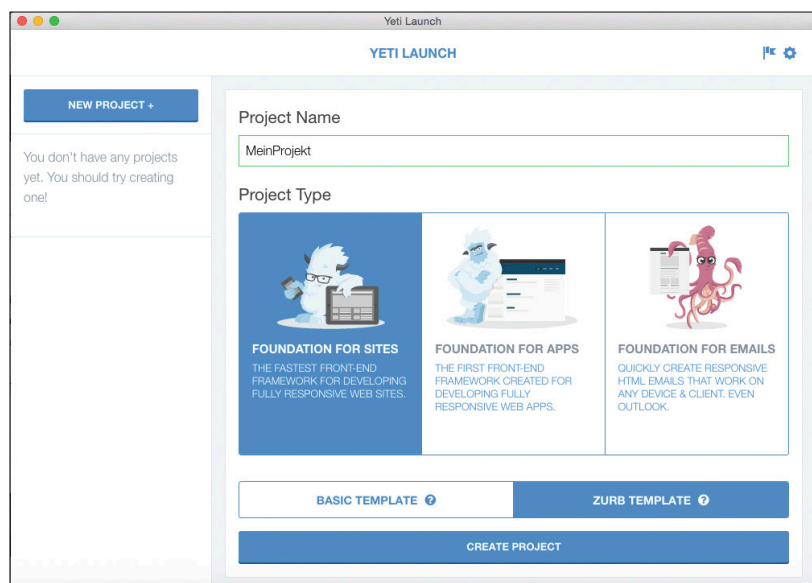
können Sie dann direkt einen Editor, ein Terminal, den Finder oder einen Browser öffnen. Außerdem werden Ihnen diverse Links zur Dokumentation und zu den Foren angeboten.

Zurb-Template

Neben dem Basic-Template steht Ihnen sowohl bei Foundation CLI als auch bei Yeti Launch das Zurb-Template zur Verfügung.

Das Zurb-Template ist ein praktisches Basisprojekt, das viele Komponenten, die man bei einem typischen Workflow braucht, bereits enthält – es ist auch das Template, das die Firma hinter Foundation, Zurb, selbst einsetzt. Es beinhaltet ein Gulpfile, in dem die notwendigen Module angegeben sind und der Build-Prozess definiert ist:

- Die HTML-Seiten können Sie über Templates erstellen lassen – dabei kommt die Panini-Bibliothek von Foundation zum Einsatz.
- Mit Libsass werden Sass-Dateien in CSS umgewandelt.
- JavaScript-Dateien werden kompiliert, eine Source-Map wird kreiert und die Dateien werden mit UglifyJS minimiert.
- Assets werden aus dem Ordner *src/assets* in den Ordner *assets* kopiert.
- Bilder werden automatisch mit Hilfe von *gulp-imagemin* komprimiert. Berücksichtigt werden dabei die Formate JPEG, PNG, SVG und GIF-Dateien.
- Außerdem ist Browsersync mit an Bord und direkt ohne weitere Konfigurationen einsatzbereit. Dank Browsersync werden Änderungen am HTML- oder CSS-Code automatisch an alle Browserfenster weitergereicht, in denen diese Datei geöffnet ist – es ist nicht notwendig, die Aktualisierung händisch anzustoßen. Zusätzlich finden weitere Synchronisierungen statt: Wenn Sie etwa in einem Browser scrollen, wird automatisch auch in weiteren geöffneten Browserfenstern zur entsprechenden Stelle gescrollt. ►



Yeti Launch: Die Starter-App für Foundation (Bild 4)

- Ebenfalls nützlich ist das integrierte UnCSS, das nicht benötigten CSS-Code entfernt.

Sehen wir uns die Panini-Komponente zur Erzeugung von HTML-Dateien einmal genauer an.

Panini – schnelle Prototypen

Bei Projekten – und auch bei Prototypen – haben Sie üblicherweise mehrere Seiten, die in Teilen übereinstimmen, da sie zum Beispiel dasselbe Grundgerüst und dieselben Inhalte im Kopf- und Fußbereich haben. Dank Panini können Sie diese Inhalte zentral anlegen und verwalten. Panini setzt dabei auf die Template-Sprache von Handlebars.

Nun zur Funktionsweise von Panini: Beim Zurb-Template befinden sich die zentralen Dateien innerhalb des Unterordners *src*. Innerhalb des Ordners *layouts* befindet sich das *default.html*-Beispiel-Template, das die Basis aller Seiten bildet. Den Inhalt von *src/layouts/default.html* zeigt **Listing 1**.

Das Dokument besteht aus normalem HTML-Code und Handlebar-Code – das sind die Angaben in geschweiften Klammern. Diese Inhalte werden später ersetzt.

Sie können das Dokument natürlich an Ihre Anforderungen anpassen. Beispielsweise wäre es sinnvoll, bei einer deutschen Seite die Angabe beim *lang*-Attribut im HTML-Start-Tag in *de* abzuändern.

Jetzt lassen sich einzelne Seiten erstellen, die auf diesem Template basieren. Eine Beispielseite ist schon definiert; Sie finden sie unter *src/pages/index.html*. Entscheidend ist, dass diese Seite kein HTML-Grundgerüst enthält, da dieses durch das Default-Template eingefügt wird. In *src/pages/index.html* steht nur der Inhalt, der bei `{{> body}}` im Default-Template ergänzt wird.

Um das Grundprinzip zu demonstrieren, wollen wir eine weitere Seite im Ordner *src/pages* mit dem Namen *beispiel.html* anlegen:

```
<h1>Titel</h1>
<p>Lorem ... </p>
```

Danach können Sie die Generierung der Seiten anstoßen, indem Sie im Projektverzeichnis *npm start* eingeben. Es dauert einen kleinen Moment, dann finden Sie die neuen Seiten im Ordner *dist*. **Listing 2** zeigt den Inhalt von *beispiel.html*.

Die Template-Angaben sind ersetzt – Sie sehen, dass der Seiteninhalt bei `{{> body}}` eingefügt ist und die Pfadangaben zum Stylesheet und zur JavaScript-Datei erzeugt wurden – vorher stand hier nur der Platzhalter `{{root}}`.

Sie erreichen die Seiten des Projekts im Browser über *http://localhost:8000/beispiel.html*.

Mehr Optionen bei Panini

Es gibt eine Reihe von weiteren Steuerungsmöglichkeiten bei der Arbeit mit Panini, wie etwa die *Partials*. *Partials* sind Code Teile, die an den angegebenen Stellen eingefügt werden. Wenn Sie in Ihrer Datei `{{> header}}` schreiben, wird an dieser Stelle der Code aus *header.html* eingefügt, sofern sich dieses im Ordner *src/partials* befindet.

Listing 1: Panini-Beispiel-Template

```
<!doctype html>
<html class="no-js" lang="en">
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible"
    content="ie=edge">
    <meta name="viewport" content=
    "width=device-width, initial-scale=1.0">
    <title>Foundation for Sites</title>
    <link rel="stylesheet"
    href="{{root}}assets/css/app.css">
  </head>

  <body>
    {{> body}}
    <script src="{{root}}assets/js/app.js"></script>
  </body>
</html>
```

Außerdem gibt es vordefinierte Variablen, die Sie nutzen können, wie `{{page}}` für den Namen der aktuellen Seite (= Dateiname ohne Endung *.html*). Eine weitere nützliche Variable ist `{{root}}`. Sie kam bereits im Beispiel vor und sorgt dafür, dass Pfadangaben funktionieren, unabhängig davon, wo sich die Datei befindet. Zusätzlich können Sie über *ifpage* Bedingungen formulieren oder *repeat* für Schleifen nutzen.

Foundations Raster

Foundation 6 bietet das gewohnte responsive Raster, bei dem sich unterschiedliche Aufteilungen je nach Viewport-Breite definieren lassen. Im folgenden Beispiel beansprucht das erste Element bei kleinen Screens zwei Spalten, das zweite vier Spalten und das dritte sechs Spalten – bei viel verfügbarem Platz sind hingegen alle Elemente gleich breit:

```
<div class="row">
  <div class="small-2 large-4 columns"></div>
  <div class="small-4 large-4 columns"></div>
  <div class="small-6 large-4 columns"></div>
</div>
```

Außerdem können Sie Offsets mit *large-offset-X*, *medium-offset-X* et cetera definieren oder Spalten über die Klassen *large-centered*, *medium-centered* et cetera zentrieren.

Bis zu einem gewissen Grad besteht darüber hinaus die Option, die Anzeigereihenfolge durch *push*- und *pull*-Klassen zu verändern. Im Beispiel wird die zweite Spalte vor der ersten angezeigt:

```
<div class="row">
  <div class="small-10 small-push-2 columns">10
</div>
  <div class="small-2 small-pull-10 columns">2, last
```



```
</div>
</div>
```

Bis auf kleine Details wie die Behandlung der Block-Grids – diese sind jetzt Teil des normalen Rasters und über Klassen wie **-up-X* aktivierbar – funktioniert das responsive Raster in Foundation wie bisher. Die entscheidende Neuerung ist, dass Sie statt des klassischen Rasters auch ein mit Hilfe der CSS3-Layouttechnik Flexbox definiertes Raster einsetzen können.

Rastern mit Flexbox

Das neue Flexbox-Raster setzt auf *display: flex* und weitere Eigenschaften, die anstelle von *float* und *position* beim klassischen Raster zum Einsatz kommen. Bevor Sie das Flexbox-Raster benutzen, sollten Sie überprüfen, ob die Browserunterstützung für Sie akzeptabel ist – das Flexbox-Raster funktioniert beispielsweise erst ab Internet Explorer 10 und ab Android 4+, in Android 2 ist es noch zu unzuverlässig.

Das Flexbox-Raster ist nicht standardmäßig aktiviert. Zur seiner Aktivierung müssen Sie in Ihrer Sass-Datei das Standard-Raster auskommentieren und das Flexbox-Raster einbinden:

```
// @include foundation-grid;
@include foundation-flex-grid;
```

Eine Kombination von Standard- und Flexbox-Raster ist eigentlich nicht vorgesehen, da beide Raster dieselben Klassennamen benutzen. Falls Sie eine Kombination beider Layouttechniken brauchen, müssen Sie sich eigene Sass-Mixins erstellen.

Auch wenn Sie die reine CSS-Variante von Foundation nutzen, können Sie das Flexbox-Raster einsetzen; eine entsprechende Option gibt es beim Custom-Download.

Listing 2: Die erzeugte Datei beispiel.html

```
<!doctype html>
<html class="no-js" lang="de">
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible"
    content="ie=edge">
    <meta name="viewport" content=
    "width=device-width, initial-scale=1.0">
    <title>Foundation for Sites</title>
    <link rel="stylesheet"
    href="assets/css/app.css">
  </head>
  <body>
    <h1>Titel</h1>
    <p>Lorem ...</p>
    <script src="assets/js/app.js"></script>
  </body>
</html>
```

Die klassischen Beispiele des normalen Grids funktionieren auch bei der Flexbox-Variante. Sie bestimmen die Breite von Elementen und können unterschiedliche Varianten für die verschiedenen Viewports angeben, indem Sie auf Klassen wie *small-6* oder *large-2* zurückgreifen.

Das Flexbox-Raster bietet jedoch noch weitergehende Möglichkeiten und zudem eine größere Flexibilität. So müssen Sie beispielsweise nicht bei allen Elementen angeben, über wie viele Spalten sie sich erstrecken sollen. Wenn Sie keine Spaltenbreite definieren, nimmt das Element einfach den übrig bleibenden Platz ein. Im folgenden Beispiel erstreckt sich das erste Element über vier Spalten, das zweite erstreckt sich über die bei einem zwölfspaltigen Raster verbleibenden acht Spalten (12 minus 4):

```
<div class="row">
  <div class="small-4 columns">4 Spalten</div>
  <div class="columns">der Rest</div>
</div>
```

Die Aufteilung funktioniert aber genauso, wenn sich das erste Element nur über zwei Spalten erstreckt – am Code des zweiten Elements gibt es keine Änderung:

```
<div class="row">
  <div class="small-2 columns">2 Spalten</div>
  <div class="columns">der Rest</div>
</div>
```

Im Beispiel erstreckt sich das Element ohne Angabe der Spaltenanzahl über allen verfügbaren Platz. Wenn Sie wollen, dass es nur den minimalen Platz einnimmt, können Sie dafür die Klasse *shrink* ergänzen. Möglich wird dieses Verhalten durch die *flex*-Eigenschaft des Flexbox-Layoutmoduls.

Dank dieser *flex*-Eigenschaft werden allerdings Elemente nicht automatisch bei wenig verfügbarem Platz untereinander angeordnet. Sollen auf kleinen Screens die Elemente untereinander stehen, dann müssen Sie die Klasse *small-12* ergänzen. Da Foundation dem Mobile-First-Ansatz folgt, gelten Angaben mit *small* jedoch automatisch auch bei größeren Viewports. Sollen jetzt die Elemente bei größeren Viewports nebeneinander dargestellt werden, und zwar ohne vordefinierte Breiten, dann müssen Sie die Klassen *medium-expand* oder *large-expand* ergänzen. Durch folgenden Code haben wir bei kleinen Viewports eine Darstellung untereinander, bei größeren werden die drei Spalten nebeneinander platziert – ohne dass die Breite der Spalten spezifiziert ist:

```
<div class="row">
  <div class="small-12 large-expand columns">Eins</div>
  <div class="small-12 large-expand columns">Zwei</div>
  <div class="small-12 large-expand columns">Drei</div>
</div>
```

Wenn Sie hingegen wissen, über wie viele Spalten sich ein Element auf großen Screens erstrecken soll, können Sie wie beim klassischen Raster *large-X* angeben. ►

Ein weiteres Plus neben der flexibleren Breiten sind die komfortablen Möglichkeiten, Elemente auszurichten. Zur Auswahl stehen für die horizontale Anordnung die Klassen *align-right*, *align-center*, *align-justify* und *align-spaces*. Die Zentrierung in der Vertikalen funktioniert mit Hilfe der Klassen *align-middle*, *align-stretch*, *align-bottom* und *align-top*.

Ein schönes Feature des Flexbox-Rasters ist die einfache und intuitive Art, wie sich die Anzeigenreihenfolge ändern lässt. Beim klassischen Grid müssten Sie mit *push*- und *pull*-* arbeiten. Beim Flexbox-Raster schreiben Sie stattdessen die *order-X-Klassen*, die es auch noch als *medium-order-X* und *large-order-X* gibt. Im folgenden Beispiel befindet sich das Element auf kleinen Screens an erster Stelle, bei größeren an zweiter.

```
<div class="column order-1 medium-order-2"></div>
```

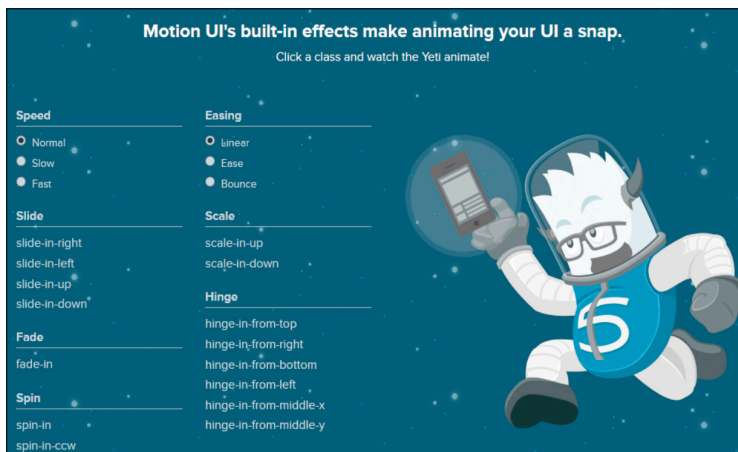
Neu in Foundation 6 ist Motion UI, eine eigenständige Bibliothek, die zum Beispiel auch für die Effekte zuständig ist, die

nen und Verschwinden ist die Angabe *data-animate="fade-in fade-out"* verantwortlich:

```
<div data-toggler="n6zm8v-toggler"
data-animate="fade-in fade-out" id="motion-example-1">
  <p>ein-/ausgeblendet</p>
</div>
```

- Vordefiniert sind fünf grundlegende Effekttypen:
- **Slide** – Einschieben
- **Fade** – Einblenden
- **Hinge** – Scharniereffekte: Diese wirken so, als wäre das Element an einer Seite festgemacht und würde von dort hineinschwingen.
- **Scale** – Größenveränderungen
- **Spin** – Drehungen

Zu jedem der grundlegenden Effekttypen gibt es Varianten; so kann etwa der Slide-Vorgang von unterschiedlichen Seiten erfolgen. Diese Effekte können Sie direkt auf der Webseite des Projekts ausprobieren (Bild 5) und Sie können auch eigene Effekte definieren. Übrigens lassen sich diese Effekte auch bei CSS3-Animationen nutzen; auch die Kombination mehrerer Animationen zu eigenen Animationsreihen ist möglich.



Motion UI für Transitions/Animations (Bild 5)

bei den Foundation-Komponenten Reveal (modale Fenster) und Orbit (Karussell) benutzt werden.

Damit Sie die Transition-Klassen nutzen können, brauchen Sie in Ihrer Sass-Datei die folgende Angabe:

```
@include motion-ui-transitions;
```

Die gewünschten Effekte schreiben Sie bei *data-animate*. Ein Beispiel: Wir erstellen zuerst einen Button, der das Einblenden beziehungsweise Ausblenden des Elements bewirkt:

```
<button
aria-controls="motion-example-1"
type="button" class="button"
data-toggle="motion-example-1">
  Fade</button>
```

Außerdem haben wir noch das Element, das eingeblendet beziehungsweise ausgeblendet wird. Für das animierte Erschei-

Accessibility

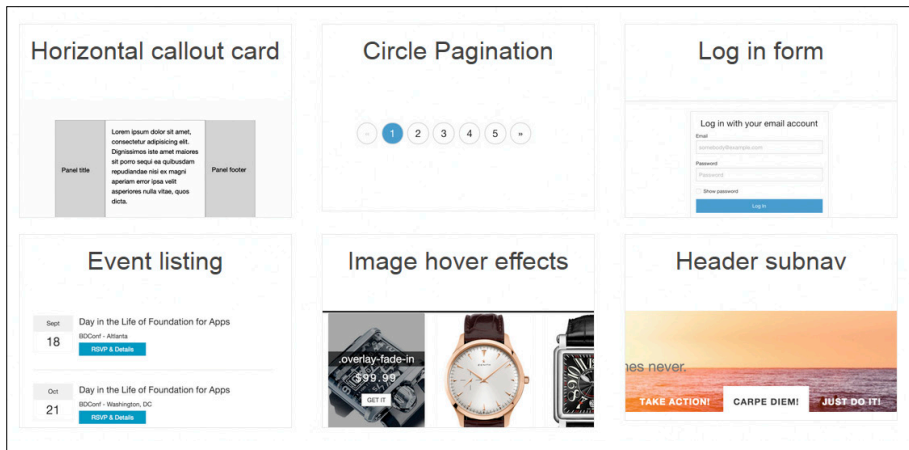
Die Zugänglichkeit von Webseiten (Accessibility) ist und bleibt ein wichtiges Thema, das jedoch gerade von Frameworks oft sträflich vernachlässigt wird. Foundation 6 macht es besser.

Screenreader können gängige HTML-Elemente wie Formularelemente „verstehen“. Schwierig wird es aber bei Komponenten, die aus mehreren HTML-Elementen bestehen, um eine besondere Funktionalität bereitzustellen. Normalsichtige erkennen die Funktionalität an der besonderen Optik, aber für Blinde ist das natürlich nicht möglich. Abhilfe bieten

die in der WAI-ARIA-Recommendation vom W3C definierten Attribute, die die Funktionen einzelner Elemente und ihre Bezüge untereinander spezifizieren. Diese werden nun konsequent bei Foundation eingesetzt.

Links zum Thema

- Foundation for Sites
<http://foundation.zurb.com/sites/docs>
- Motion UI
<http://foundation.zurb.com/sites/docs/motion-ui.html>
- Panini-Komponenten
<http://handlebarsjs.com>
<https://www.browsersync.io>
- Building Blocks
<http://zurb.com/building-blocks>



Building Blocks: Bereit zum Einsatz per Copy and Paste (Bild 6)



Weniger Formatierung: Vergleich zwischen Komponenten von Foundation 5 (links) und Foundation 6 (rechts) (Bild 7)

Sehen wir uns als Beispiel den HTML-Code für ein Accordion in Foundation 6 an. Im Beispiel wird ein Accordion-Element gezeigt:

```
<ul class="accordion" data-accordion role="tablist">
  <li class="accordion-item is-active">
    <a href="#panel1d" role="tab"
      class="accordion-title" id="panel1d-heading"
      aria-controls="panel1d">Accordion 1</a>
    <div id="panel1d" class="accordion-content"
      role="tabpanel" data-tab-content
      aria-labelledby="panel1d-heading">Panel 1.</div>
  </li>
</ul>
```

Zum Einsatz kommen eine Reihe von ARIA-Attributen – beim umfassenden Container, beim einzelnen Accordion-Link und beim Accordion-Inhalt:

- Neu ist `role="tablist"` beim Container-Element; dadurch können Screenreader beispielsweise die Anzahl der enthaltenen Elemente angeben.
- Die einzelnen Links, das sind die anklickbaren Accordion-Titel, benötigen `role="tab"` sowie eine eindeutige ID und die Angabe `aria-controls="panel1d"`. Über `aria-controls` verweisen Sie auf die ID eines weiteren Elements, das über dieses kontrolliert wird.
- Das Element mit dem Inhalt erhält `role="tabpanel"`. Die Zugehörigkeit zum Link wird zusätzlich durch `aria-labelledby="panel1d-heading"` hergestellt.

Als sehr nützliche Helfer können sich die Building Blocks (Bild 6) erweisen. Dabei handelt es sich um fertige Komponenten, die man direkt einsetzen kann. Zu den über 100 Building Blocks zählen Hover-Effekte für Bilder, Overlay-Menüs, Log-

in-Formulare, aber auch so kleine Details wie runde Buttons für die Seitennummerierung et cetera. Bei jeder Komponente finden Sie den HTML- sowie den CSS- oder SCSS-Code zur direkten Verwendung in einem Foundation-Projekt.

Das allgemeine Problem bei Frameworks ist, dass sie üblicherweise viele CSS-Angaben mitbringen, die für eine hübsche Default-Optik sorgen. Wenn Sie aber individuelle Formatierungen festlegen möchten, kommen Ihnen die Default-Angaben in die Quere. Dadurch kann es passieren, dass Sie bei der individuellen Gestaltung einer Komponente mehr Code schreiben müssen als ohne Framework. Um diesen Missstand zu beheben, bringt Foundation von Haus aus weniger Formatierungen mit, was an vielen Komponenten wie bei-

spielsweise den Breadcrumbs zu erkennen ist (Bild 7).

Eine weitere Änderung betrifft die Menü-Komponente, die anders organisiert ist. Standardmenüs, die durch CSS gebildet werden, können durch drei verschiedene Menü-Plug-ins verbessert werden: Dropdown, Drilldown und Accordion. Das responsive Menü-Plug-in ermöglicht es dann, verschiedene Navigationsmuster für unterschiedliche Viewport-Größen zu definieren. So könnten Sie etwa bei viel verfügbarem Platz ein Dropdown-Menü anzeigen lassen und bei weniger Platz ein Drilldown-Menü:

```
<ul class="vertical menu"
  data-responsive-menu="drilldown medium-dropdown" >
```

Ebenfalls praktisch sind die typografischen Helferklassen wie zum Rechts- oder Links-Floaten, für die Textausrichtung oder für Unterüberschriften. Neu ist außerdem die Standardkomponenten Badge. Und schließlich gibt es das Sticky-Plug-in, das dafür sorgt, dass Elemente stehen bleiben, wenn sie sonst aus dem Browserfenster verschwinden würden.

Neben den hier behandelten Neuerungen beinhaltet Foundation 6 die vielen nützlichen, aus vorherigen Versionen bekannten Komponenten. ■



Dr. Florence Maurice

ist Autorin, Trainerin und Programmiererin in München. Ihr aktuelles Buch ist dem Thema »CSS3« gewidmet und bei dpunkt erschienen (ISBN 978-3-86490-118-8). Außerdem bloggt sie regelmäßig zu CSS und verwandten Themen unter: <http://maurice-web.de/blog>

DAS NAVIGATION TIMING API

Zeitinformationen

Mit Hilfe des Navigation Timing API kann man die Performance einer Website messen.

Das Navigation Timing API (www.w3.org/TR/navigation-timing) bietet Webentwicklern die Möglichkeit, per JavaScript verschiedene Aspekte bezüglich der Performance einer Webseite zu messen. Die Spezifikation ist schon relativ lange eine Empfehlung des W3C und wird daher auch vom Großteil der verschiedenen Browsern unterstützt (Bild 1). Dennoch ist der Bekanntheitsgrad des API unter Webentwicklern eher gering. Eine gute Gelegenheit also, einmal einen etwas genaueren Blick darauf zu werfen.

Den Einstieg in das Navigation Timing API bildet die Eigenschaft *performance* des *window*-Objekts (Bild 2). Das dieser Eigenschaft hinterlegte Objekt (vom Typ *performance*) wiederum enthält seinerseits zwei Eigenschaften, und zwar die beiden titelgebenden Eigenschaften *timing* und *navigation*.

Erstere enthält zahlreiche Zeitinformationen zu bestimmten Ereignissen, die während des Ladens einer Webseite (oder allgemeiner: eines Webdokuments) auftreten (daher der Name *timing*). Letzteres enthält Informationen darüber, wie ein Nutzer auf die entsprechende Webseite gelangt ist, sprich auf welche Art und Weise der Nutzer zu der Webseite navigiert hat (daher der Name *navigation*).

Navigation Timing API - REC

API for accessing timing information related to navigation and elements.

Global: 87.99%
unprefixed: 87.87%

Current aligned Usage relative Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8			43					4.1	
9		40	44					4.3	
10	12	41	45	8	32	8.4		4.4	
11	13	42	46	9	33	9.1	8	4.4.4	
	14	43	47		34			46	46
		44	48		35				
		45	49						

Quelle: <http://caniuse.com>

Das Navigation Timing API wird von den meisten Browsern unterstützt (Bild 1)

Zeitinformationen abfragen

Insgesamt stellt das *timing*-Objekt 21 Eigenschaften bereit, die allesamt verschiedene Zeitpunkte markieren, an denen bestimmte Ereignisse ausgelöst wurden, beispielsweise Start und Ende von Redirects, Start und Ende des DNS-Lookups, Start und Ende des HTTP-Requests, Start und Ende der Verarbeitung des DOM und einiges mehr. Bild 3 und Tabelle 1 geben hierzu eine detaillierte Übersicht.

Anhand dieser verschiedenen Zeitinformationen, die jeweils durch Zeitstempel repräsentiert werden, können verschiedene Aspekte berechnet werden, beispielsweise die Dauer des DNS-Lookups oder einer HTTP-Anfrage oder auch die Verweildauer des Nutzers auf der entsprechenden Webseite (Listing 1).

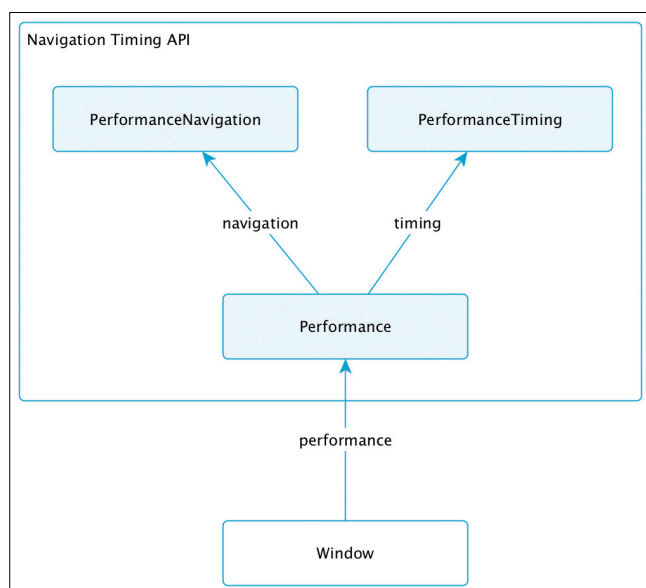
Navigationsinformationen abfragen

Die zweite Eigenschaft des *Performance*-Objekts beziehungsweise das dort hinterlegte Objekt vom Typ *PerformanceNavigation* ermöglicht es, an Informationen bezüglich der Navigation des Nutzers zu gelangen.

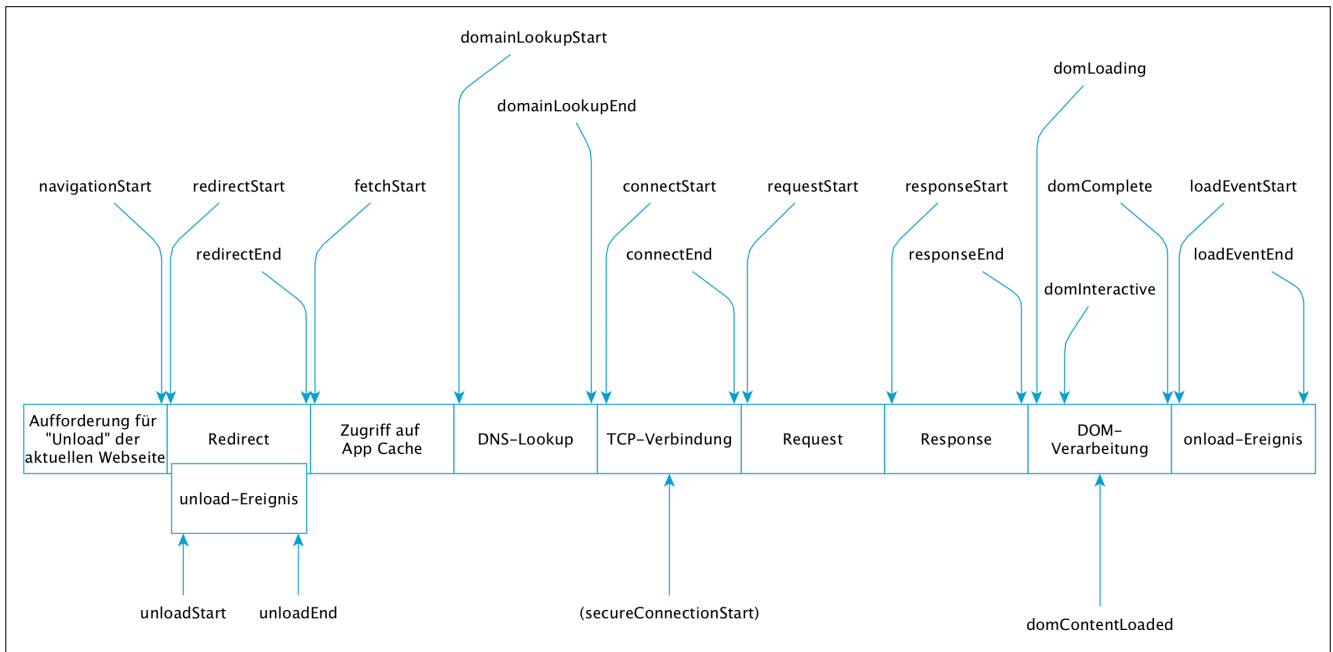
Das Objekt hat zwei Eigenschaften: *type* und *redirectCount*. Über Erstere lässt sich ermitteln, wie ein Nutzer auf die aktuelle Webseite gekommen ist.

Die Eigenschaft kann dabei einen von vier möglichen Werten annehmen: Hat der Nutzer aktiv den URL der Webseite eingegeben, so besitzt die Eigenschaft den Wert 0. Wurde die Webseite lediglich neu geladen (also beispielsweise durch Drücken des Refresh-Buttons), so hat die Eigenschaft den Wert 1. Wurden dagegen die Vor- beziehungsweise Zurück-Buttons des Browsers verwendet, so besitzt die Eigenschaft den Wert 2.

Alle anderen Arten der Navigation schließlich werden durch den Wert 255 repräsentiert.



Zusammenhang der verschiedenen Interfaces, die durch das Navigation Timing API definiert und erweitert werden (Bild 2)



Die einzelnen Zeitpunkte, zu denen das API Informationen bereitstellt (Bild 3)

Über die Eigenschaft `redirectCount` kann zudem ermittelt werden, wie häufig der Nutzer durch Redirects weitergeleitet wurde, bevor er auf die jeweilige Webseite kam. Listing 2 zeigt ein Beispielprogramm, das alle relevanten Eigenschaf-

ten der beiden Objekte `PerformanceTiming` und `PerformanceNavigation` als Tabelle ausgibt. Das Ergebnis zeigt der Screenshot in Bild 4.

Während die Information in dieser Form zugegebenermaßen etwas schwierig zu interpretieren ist, geht es auch deutlich nützlicher, wie die Webseite <https://github.com/kaaes/timing> zeigt. Das dort hinterlegte Skript visualisiert die In- ►

Listing 1: Verwenden des Navigation Timing API

```
'use strict';
if ('performance' in window) {
  window.addEventListener('load', () => {
    setTimeout(() => {
      let timing = window.performance.timing;
      let userTime = timing.loadEventEnd -
        timing.navigationStart;
      let dnsTime = timing.domainLookupEnd -
        timing.domainLookupStart;
      let connectionTime = timing.connectEnd -
        timing.connectStart;
      let requestTime = timing.responseEnd -
        timing.requestStart;
      let fetchTime = timing.responseEnd -
        timing.fetchStart;
      console.log('Dauer Nutzer auf Webseite: ' +
        userTime);
      console.log('Dauer DNS-Lookup: ' + dnsTime);
      console.log('Dauer Verbindungsaufbau: ' +
        connectionTime);
      console.log('Dauer Request: ' + requestTime);
      console.log('Dauer Fetch: ' + fetchTime);
    }, 0);
  });
}
```

Eigenschaft	Wert
navigationStart	1448814279271
unloadEventStart	1448814279277
unloadEventEnd	1448814279277
redirectStart	0
redirectEnd	0
fetchStart	1448814279271
domainLookupStart	1448814279271
domainLookupEnd	1448814279271
connectStart	1448814279271
connectEnd	1448814279271
secureConnectionStart	0
requestStart	1448814279273
responseEnd	1448814279276
domLoading	1448814279297
domInteractive	1448814279633
domContentLoadedEventStart	1448814279633
domContentLoadedEventEnd	1448814279694
domComplete	1448814279771
loadEventStart	1448814279771
loadEventEnd	0
type	1
redirectCount	0

Ausgabe aller relevanten Eigenschaften des Navigation Timing API (Bild 4)

Tabelle 1: Eigenschaften des PerformanceTiming-Interfaces

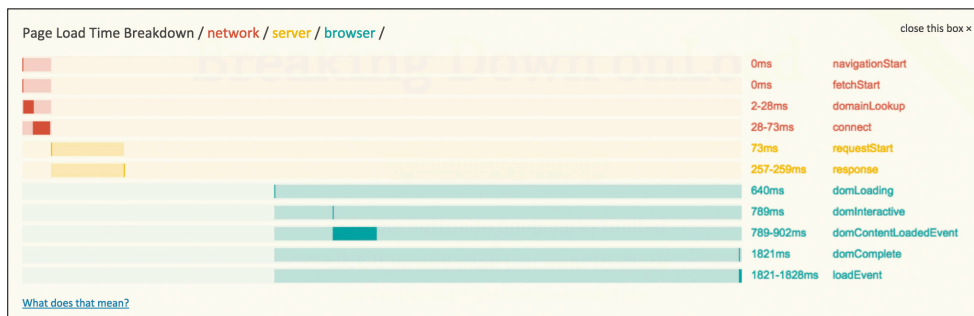
Eigenschaft	Beschreibung
<i>navigationStart</i>	Repräsentiert den Zeitpunkt, zu dem das vorher im Browser geladene Webdokument entladen wurde. Für den Fall, dass es kein vorheriges Webdokument gibt, entspricht der Wert dem Wert <i>fetchStart</i> .
<i>unloadEventStart</i>	Repräsentiert den Zeitpunkt, unmittelbar bevor der Browser das <i>unload</i> -Event auslöst.
<i>unloadEventEnd</i>	Repräsentiert den Zeitpunkt, unmittelbar nachdem das <i>unload</i> -Event beendet wurde.
<i>redirectStart</i>	Repräsentiert im Fall eines Redirects den Zeitpunkt, an dem dieser startet.
<i>redirectEnd</i>	Repräsentiert im Fall eines Redirects den Zeitpunkt, an dem dieser endet.
<i>fetchStart</i>	Repräsentiert den Zeitpunkt, an dem eine <i>HTTP-GET</i> -Anfrage startet.
<i>domainLookupStart</i>	Repräsentiert den Zeitpunkt, an dem der Browser eine DNS-Abfrage für das Webdokument startet.
<i>domainLookupEnd</i>	Repräsentiert den Zeitpunkt, an dem die DNS-Abfrage für das Webdokument endet.
<i>connectStart</i>	Repräsentiert den Zeitpunkt, an dem der Browser eine Verbindung zu dem entsprechenden Server herstellt, um das Webdokument zu laden.
<i>connectEnd</i>	Repräsentiert den Zeitpunkt, an dem die Verbindung zu dem entsprechenden Server hergestellt wurde, um das Webdokument zu laden.
<i>secureConnectionStart</i>	Repräsentiert im Fall von HTTPS den Zeitpunkt, an dem der Browser den Handshake-Prozess startet.
<i>requestStart</i>	Repräsentiert den Zeitpunkt, an dem der Browser mit dem Laden des entsprechenden Webdokuments beginnt.
<i>responseStart</i>	Repräsentiert den Zeitpunkt, an dem der Browser das erste Byte der Antwort vom Server erhält.
<i>responseEnd</i>	Repräsentiert den Zeitpunkt, an dem der Browser das letzte Byte der Antwort vom Server erhält.
<i>domLoading</i>	Repräsentiert den Zeitpunkt, an dem der Browser die Eigenschaft <i>readyState</i> auf <i>loading</i> setzt.
<i>domInteractive</i>	Repräsentiert den Zeitpunkt, an dem der Browser die Eigenschaft <i>readyState</i> auf <i>interactive</i> setzt.
<i>domContentLoadedEventStart</i>	Repräsentiert den Zeitpunkt, unmittelbar bevor der Browser das <i>DOMContentLoaded</i> -Event auslöst.
<i>domContentLoadedEventEnd</i>	Repräsentiert den Zeitpunkt, unmittelbar nachdem das <i>DOMContentLoaded</i> -Event beendet wurde.
<i>domComplete</i>	Repräsentiert den Zeitpunkt, an dem der Browser die Eigenschaft <i>readyState</i> auf <i>complete</i> setzt.
<i>loadEventStart</i>	Repräsentiert den Zeitpunkt, unmittelbar bevor der Browser das <i>load</i> -Event auslöst.
<i>loadEventEnd</i>	Repräsentiert den Zeitpunkt, unmittelbar nachdem das <i>load</i> -Event beendet wurde.

formationen auf einer Zeitachse (Bild 5) und kennzeichnet farblich, welche Aktivitäten die Netzwerkkommunikation betreffen (zum Beispiel *domainLookup* oder *connect*), welche den Server (zum Beispiel *requestStart* oder *responseStart*) und welche den Browser (zum Beispiel *domLoading* oder *domComplete*). Auf diese Weise kann man bei Performance-

problemen schnell herausfinden, wo die Ursache liegt. Nichts, was man nicht auch mit den Chrome Developer Tools ermitteln kann, aber trotzdem gut zu wissen.

Der Hauptanwendungszweck für die Verwendung des Navigation Timing API dürfte jedoch die Erfassung von Nutzerdaten sowie die Performance-Messung in der produktiven Version einer Website sein. Die an dieser Stelle gesammelten Informationen könnten anschließend im Hintergrund per Ajax zur Auswertung an einen Server gesendet werden.

Das Navigation Timing API bildet die Grundlage für eine Reihe weiterer Performance-relevanter APIs, die jeweils das Performance-Interface um zusätzliche Eigenschaften erweitern:



Das Script unter <https://github.com/kaaes/timing> visualisiert die Informationen aus dem Navigation Timing API (Bild 5)

Listing 2: Ausgabe aller relevanten Informationen des Navigation Timing API

```

'use strict';
if ('performance' in window) {
  window.addEventListener('load', () => {

    let table = initTable();
    let tbody = document.createElement('tbody');
    table.appendChild(tbody);

    let propertiesTiming = [
      'navigationStart',
      'unloadEventStart',
      'unloadEventEnd',
      'redirectStart',
      'redirectEnd',
      'fetchStart',
      'domainLookupStart',
      'domainLookupEnd',
      'connectStart',
      'connectEnd',
      'secureConnectionStart',
      'requestStart',
      'responseEnd',
      'domLoading',
      'domInteractive',
      'domContentLoadedEventStart',
      'domContentLoadedEventEnd',
      'domComplete',
      'loadEventStart',
      'loadEventEnd'
    ];
    propertiesTiming.forEach(
      (
        (object) => {
          return (property) => {
            return createTableRow(property, object, table);
          };
        }
      )(window.performance.timing)
    );

    let propertiesNavigation = [
      'type',
      'redirectCount'
    ];
    propertiesNavigation.forEach(
      (
        (object) => {
          return (property) => {
            return createTableRow(property, object, table);
          };
        }
      )(window.performance.navigation)
    );

    document.getElementById('timing').
    appendChild(table);

    function initTable() {
      let table = document.createElement('table');
      table.className = 'table table-striped
      table-bordered';
      let thead = document.createElement('thead');
      let tr = document.createElement('tr');
      let th1 = document.createElement('th');
      th1.appendChild
      (document.createTextNode('Eigenschaft'));
      let th2 = document.createElement('th');
      th2.appendChild(document.createTextNode('Wert'));
      tr.appendChild(th1);
      tr.appendChild(th2);
      thead.appendChild(tr);
      table.appendChild(thead);
      return table;
    }

    function createTableRow(property, object, table) {
      let tableRow = document.createElement('tr');
      let td1 = document.createElement('td');
      td1.appendChild(document.createTextNode(property));
      let td2 = document.createElement('td');
      td2.appendChild(document.createTextNode
      (object[property]));
      tableRow.appendChild(td1);
      tableRow.appendChild(td2);
      tbody.appendChild(tableRow);
    }
  });
}

```

- das Performance Timeline API (www.w3.org/TR/performance-timeline),
- das User Timing API (www.w3.org/TR/user-timing),
- das Resource Timing API (www.w3.org/TR/resource-timing),
- das Frame Timing API (www.w3.org/TR/frame-timing),
- das Server Timing API (www.w3.org/TR/server-timing).

Darüber hinaus arbeitet man beim W3C derzeit an einer zweiten Version (www.w3.org/TR/navigation-timing-2).

Das Navigation Timing API bietet Zugriff auf verschiedene Performance-relevante Informationen, die das Laden einer Webseite betreffen. Es wird von allen gängigen Browsern un-

terstützt und kann dazu genutzt werden, um in Produkktivsystemen Performance-Statistiken zu generieren. ■



Philip Ackermann

arbeitet beim Fraunhofer-Institut für Angewandte Informationstechnologie FIT an Tools zum teilautomatisierten Testen von Web Compliance und ist Autor zweier Fachbücher über Java und JavaScript.

<http://philipackermann.de>

FORMULARVALIDIERUNG MIT ANGULAR 2

Komfortables Handling

Formulare sind eine der wichtigsten Aufgaben moderner Webframeworks.

Dies ist der vierte Artikel in unserer Reihe zum Framework Angular 2. In den vorangegangenen Artikeln haben wir bereits SystemJS, Templates, Dependency Injection, Unit-Testing und asynchrone HTTP-Kommunikation kennengelernt. Mit dabei ist stets das Car Dashboard, das kontinuierlich neue Funktionen erhält. In diesem Artikel soll nun die Verarbeitung von Formulardaten im Mittelpunkt der Betrachtung stehen (Bild 1).

Als Webentwickler haben wir regelmäßig mit HTML-Formularen zu tun. Sie dienen der Gewinnung relevanter Informationen durch Benutzereingaben, welche anschließend weiterverarbeitet werden sollen.

Neben der bloßen Eingabe der Daten spielt es eine wichtige Rolle, ein adäquates visuelles Feedback an den Benutzer weiterzugeben. Fehleingaben sollen frühzeitig signalisiert werden. Auf Änderung der Eingabedaten muss ein Formular schnell reagieren können. Angular 2 hilft dabei, diese Anforderungen umzusetzen.

Wie gewohnt steht ein komplettes, lauffähiges Beispiel auf GitHub unter <https://github.com/Angular2Buch/angular2-forms> zur Verfügung, das die hier besprochenen Inhalte demonstriert.

Controls und ControlGroups als Basis der Formularverarbeitung

Die Basis der Formularverarbeitung sind die Klassen *Control* und *ControlGroup*.

Um sie in einer Komponente zu verwenden, müssen sie aus dem Namensraum *angular2/common* importiert werden:

```
import {Control, ControlGroup} from 'angular2/common';
```

Ein Control hat einen Wert und einen bestimmten Validierungsstatus, der durch eine zusätzliche Funktion bestimmt werden kann. Daraus ergeben sich für das Control drei optio-

Tabelle 1: Zustandsinformationen eines Controls

Status	Beschreibung
<i>dirty</i>	Der Wert wurde bearbeitet
<i>pristine</i>	Der Wert ist unberührt
<i>valid</i>	Der Wert ist gültig
<i>touched</i>	Das Control wurde verwendet/bedient
<i>untouched</i>	Das Control wurde noch nicht verwendet
<i>pending</i>	Eine asynchrone Eingabevalidierung läuft

Angular 2: Endlich im Beta-Status (Bild 1)

nale Parameter, ein Wert, sowie ein Validator und ein asynchroner Validator:

```
this.nameControl = new Control('Initialer Wert',
/* synchroner Validator -> */ Validators.required,
/* asynchroner Validator -> */ IdValidator.uniqueId);
nameControl.value; // -> 'Initialer Wert'
nameControl.errors; // -> undefined
```

Das Control kann im HTML-Template der Komponente über die Direktive *ngControl* gebunden werden:

```
<input type="text" ngControl="nameControl" required>
```

Je nach Zustand des Controls fügt *ngControl* automatisch CSS-Klassen an das jeweilige Eingabeelement hinzu:

```
<input type="text"
class="ng-pristine ng-valid ng-touched"
ngcontrol="nameControl"
required>
```

Da der Status des Controls (Tabelle 1) durch CSS-Klassen repräsentiert wird, kann das Element entsprechend gestaltet werden:

```
.ng-valid {
border-left: 5px solid #42A948;
}
```



```
.ng-invalid {
  border-left: 5px solid #a94442;
}
```

So erhält der Benutzer zu seiner getätigten Eingabe ein direktes visuelles Feedback.

Die Voraussetzung für das Funktionieren eines Controls ist, dass es innerhalb eines Formulars verwendet wird. Grund hierfür ist, dass Angular an jedes `<form>`-Tag eine Instanz der Klasse `NgForm` anfügt, um Daten im Formular automatisch aktualisieren zu können. Das kann zu Beginn verwirren, weil dieser Vorgang hinter den Kulissen geschieht.

In der Regel bestehen Formulare aus einer Vielzahl von Eingabefeldern. `ControlGroup` erlaubt es, beliebig viele Controls als Schlüssel-Wert-Paare zusammenzufassen:

```
class RegisterUserComponent {
  userForm: ControlGroup;
  nameControl: Control;
  emailControl: Control;
  constructor() {
    this.nameControl = new Control('John',
      Validators.required);
    this.emailControl = new Control('john@doe.de',
      Validators.required);
    this.userForm = new ControlGroup({
      name : this.nameControl,
      email : this.emailControl
    });
  }
}
```

Auch die `ControlGroup` verfügt über Eigenschaften, die bereits aus der Klasse `Control` bekannt sind:

```
this.userForm.value; // -> {"name":"John",
"email":"john@doe.de"}
this.userForm.valid; // -> true
```

In diesem Beispiel ist zu sehen, wie Validatoren direkt in einem Control verwendet werden (`Validators.required`). Die `ControlGroup` kann mit der Direktive `ngFormModel` an das Template gebunden werden:

```
<form [ngFormModel]="userForm">
  <input type="text"
    ngControl="name">
  <input type="text"
    ngControl="email">
  <button type="submit">
```

Name
<input type="text" value="J"/>
Der Name muss mindestens aus zwei Buchstaben bestehen.

Validierung: Eingabefeld mit definierter minimaler Länge (Bild 3)

Neuen Benutzer registrieren

Name

Email

Eingabeformular: Beispiel für eine Registrierung (Bild 2)

```
[disabled]="!userForm.valid">
  Registrieren
</button>
</form>
```

Da aus `userForm` die Gültigkeit der Eingaben bekannt ist, kann mit Hilfe der Eigenschaft `valid` gesteuert werden, ob der Benutzer das Formular absenden kann (Bild 2).

Vorhandene Validatoren

Derzeit verfügt das Angular-Framework über drei Validatoren, die sofort eingesetzt werden können. Diese können ebenfalls aus dem Modul `angular2/common` abgerufen werden:

```
import {Validators} from 'angular2/common';
```

So kann für ein beliebiges Control beispielsweise eine Mindestlänge verlangt werden:

```
this.nameControl = new Control('',
  Validators.minLength(2))
```

Im Formular kann ein Verstoß gegen diese Regel explizit ausgegeben werden (Bild 3):

```
<form [ngFormModel]="userForm">
  <input #name="ngForm"
    ngControl="name"
    type="text">

  <div *ngIf="!name.valid && name.errors.minLength">
    Der Name muss mindestens aus zwei Buchstaben bestehen.
  </div>

  <!-- ... -->
</form>
```

In diesem Beispiel wird eine lokale Referenz auf das Control angelegt. Sie ist als `#name` bezeichnet. Dadurch wird es möglich, den aktuellen Zustand des Controls in anderen Elementen zu verwenden. Mit `*ngIf` wird geprüft, ob ein Fehler für die Validierungsregel `minlength` vorliegt. Trifft das zu, wird die Fehlermeldung unter dem Eingabefeld eingeblendet. ►

Es ist erforderlich, dass bei der Prüfung auf das Fehlerobjekt *errors* vorher sichergestellt wird, dass das betreffende Control invalid ist. Ist das Control im validen Zustand, so ist das Fehlerobjekt *undefined*. Ein Zugriff auf einen der Schlüssel verursacht dann einen Fehler.

Zugegeben, nur drei vorhandene Validatoren sind wirklich nicht viel. Weitere eingebaute Überprüfungen werden folgen, mehrere Issues sind hierzu auf GitHub offen (zum Beispiel #2961, #2962 und #5411). Wahrscheinlich können wir mit *email*, *url*, *number*, *date*, *time*, *week*, *month*, *min*, *max* und *pattern* (Regex) rechnen.

Eigene Validatoren schreiben

Eigene Regeln und Eingabeformate gibt es reichlich, beispielsweise KFZ-Kennzeichen, ISBN, Postleitzahl et cetera. Für spezielle Anwendungsfälle können eigene Validatoren implementiert werden:

```
this.nameControl = new Control('', Validators.compose([Validators.required, Validators.minLength(2)]
```

Dabei handelt es sich um Klassen, die statische Methoden für die Validierung bereitstellen. Damit das Ergebnis der Überprüfung im Template ausgewertet werden kann, ist es erforderlich, dass ein Schlüssel-Wert-Paar vom Typ *<string, boolean>* zurückgegeben wird. Um die Typsicherheit zu gewährleisten, kann dafür ein Interface definiert werden:

```
interface ValidationResult{
  [key:string]:boolean; }
```

Als Beispiel wird ein Validator implementiert, der prüft, ob ein deutsches Datumsformat vorliegt. Der Methode *german-*

Date wird ein Control übergeben, dessen Wert mit einem regulären Ausdruck überprüft wird. Liegt kein Datum im korrekten Format vor, wird für das Fehlerobjekt des Controls der Eintrag *'germanDate'*: *true* erzeugt. Wird kein Fehler gefunden, wird *null* zurückgegeben:

```
import {Control} from 'angular2/common';
export class DateValidator {
  static germanDate(control: Control): ValidationResult {
    {
      let germanDatePattern = /^d{1,2}\.d{1,2}\.d{4}$/;
      if (!control.value.match(germanDatePattern))
        return { "germanDate": true };
      return null;
    }
  }
}
```

Der eigene Validator kann nun in der Komponente verwendet werden:

```
/* ... */
import {DateValidator} from 'date.validator.ts';
class RegisterUserComponent {
  /* ... */
  birthdateControl: Control;
  constructor() {
    /* ... */
    this.birthdateControl: new Control('',
      DateValidator.germanDate);
    this.userForm = new FormGroup({
      /* ... */
      birthDate: this.birthdateControl
    });
  }
}
```

Im Template der Komponente kann ebenfalls eine Fehlermeldung ausgegeben werden. Hierbei ist es wichtig, dass der exakte Schlüsselname des Validators verwendet wird:

```
<div *ngIf="!birthdate.valid &&
birthdate.errors.germanDate">
  Geben Sie ein Datum ein.
  Beispiel: 01.01.1970
</div>
```

Um die Validität von Eingaben zu prüfen, ist es manchmal erforderlich, einen externen Dienst anzufragen.

Asynchrone Validatoren

Bei der Vergabe von neuen E-Mail-Adressen oder eindeutigen Benutzernamen ist dieser Vorgang unabdingbar. Für solche Szenarien eignen sich asynchrone Validatoren. Sie zeichnen sich dadurch aus, dass sie eine Promise zurückgeben, in dem das Ergebnis der Überprüfung enthalten ist.

Das folgende Beispiel skizziert den Zeitversatz durch die asynchrone Prüfung anhand der Methode *setTimeout*. Es

Angular 2 im Beta-Status

Ende Dezember 2015 hat Angular 2 den Beta-Status erreicht. Die hier gezeigten Code-Beispiele sollten in Zukunft kaum noch tiefgreifende Änderungen aufweisen.

Die wesentlichen Änderungen, die auffallen werden:

Der Namensraum *angular2/angular2* existiert nicht mehr. Beispielsweise befinden sich die Dekoratoren für eine Komponente nun in *angular2/core*. Konstanten wie *CORE_DIRECTIVES* oder *FORM_DIRECTIVES* brauchen nicht mehr explizit referenziert werden. Die Direktiven **ngIf*, *ngModel* et cetera stehen nun sofort zur Verfügung. Direktiven werden nun in *lowerCamelCase* und nicht mehr in *kebab-case* geschrieben (*ng-model* wird zu *ngModel*).

Die Beispiele aus den vorherigen Artikeln haben wir für Sie auf den neuesten Stand gebracht:

<https://github.com/Angular2Buch/angular2-module>

<https://github.com/Angular2Buch/angular2-template-syntax>

<https://github.com/Angular2Buch/angular2-testing>

Links zum Thema

- Codebeispiel zum Artikel
<https://github.com/Angular2Buch/angular2-forms>
- Angular 2 CLI zur Erstellung der Projektstruktur
<https://github.com/angular/angular-cli>
- Blogartikel Form-Validation mit Angular 2
<https://medium.com/@daviddentoom/angular-2-form-validation-9b26f73fcb81>
- Angular-2-Dokumentation zur Formularverarbeitung
<https://angular.io/docs/ts/latest/guide/forms.html>
- Angular-2-Projekt auf GitHub
<https://github.com/angular/angular>
- Angular 2 Data Flow – Jeff Cross, Rob Wormald and Alex Rickabaugh
https://youtu.be/bVI5gGTEQ_U
- W3C – HTML: The Markup Language (an HTML language reference)
www.w3.org/TR/html-markup/syntax.html

wird geprüft, ob eine ID eindeutig ist oder nicht. Ist dies nicht der Fall, so wird erneut ein Eintrag für das Fehlerobjekt des Controls erstellt (`'uniqueId': true`):

```
import {Control} from 'angular2/common';
export class IdValidator
{
  static uniqueId(control: Control):
  Promise<ValidationResult> {
    return new Promise((resolve, reject) =>
    {
      setTimeout(() => {
        if (control.value === '42')
          resolve({'uniqueId': true});
        else
          resolve(null);
      }, 1000)
    });
  }
}
```

Bei der Verwendung asynchroner Validatoren ist lediglich zu beachten, dass sie als dritter Parameter im Konstruktor des Ziel-Controls referenziert werden:

```
/* ... */
import {IdValidator} from 'id.validator.ts';
class RegisterUserComponent {
  /* ... */
  id: Control;
  constructor() {
    /* ... */
    this.id = new Control('', Validators.required,
```

```
IdValidator.uniqueId));
this.userForm = new ControlGroup({
  /* ... */
  id: this.id
});
}
```

Für die Bindung im Template leisten asynchrone Validatoren eine komfortable Hilfestellung. Solange die Validierung läuft, wird im Control die Eigenschaft *pending* auf *true* gesetzt. Das ermöglicht zum einen visuelles Feedback für den Benutzer des Formulars und signalisiert andererseits, dass die Verarbeitung im Gang ist.

Beispielsweise ist ein rotierender Indikator so mittels CSS3 schnell implementiert. Das ungekürzte Beispiel mit der kompletten Verarbeitungslogik finden Sie auf GitHub:

```
<div *ngIf="id.pending">
  Überprüfe Verfügbarkeit...
</div>

<div *ngIf="!id.valid && !id.pending &&
id.errors.uniqueId">
  Diese Id existiert bereits.
</div>
```

Angular 2 ermöglicht es, einen Großteil des Verhaltens eines Formulars in der Komponente zu definieren. Templates enthalten somit weniger Validierungslogik als noch in Angular 1. Das bereinigt das Template und ermöglicht es, die Validierung stärker in Richtung der Domänenlogik anzusiedeln. Dies ist eine gute Entscheidung.

Eigene Validierungslogik kann schnell selbst entwickelt und in die Anwendung integriert werden. Sobald das Konzept der Controls, ControlGroups und Validatoren verstanden ist, ergibt sich ein geradliniger Entwicklungsprozess zum Erstellen und Erweitern von Webformularen. Wir wünschen viel Spaß beim Ausprobieren. ■

**Johannes Hoppe**

ist selbstständiger IT-Berater, Software-Entwickler und Trainer. Er arbeitet derzeit als Architekt für ein Portal auf Basis von .NET und AngularJS.

<http://blog.johanneshoppe.de>

**Gregor Woiwode**

ist als Software-Entwickler im Bereich des Competitive Intelligence beziehungsweise Enterprise Knowledge Managements für ein Software-Unternehmen in Leipzig tätig. Er veranstaltet Trainings zu AngularJS.

www.woiwode.info/blog

APPS FÜR TVOS

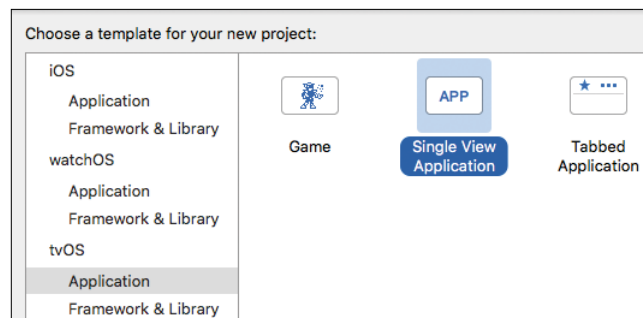
Apps im Fernseher

Apple TV hat ein neues Betriebssystem und einen eigenen App Store bekommen.

Eine neue Generation von Apple TV, die vierte, ist im Oktober 2015 in die Geschäfte gekommen. Davor war Apple TV für App-Entwickler eher uninteressant. Nur wenigen Entwicklern war es bisher erlaubt, Apps für Apple TV zu entwickeln und auf dem Gerät zu publizieren.

Mit der vierten Generation wird zum ersten Mal ein App Store auf dieser Plattform zur Verfügung gestellt, in dem (analog zu iPad, iPhone, iPod und Apple Watch) Apps von jedermann zum Download angeboten werden können. Auf dem Gerät kommt tvOS zum Einsatz. Es handelt sich hierbei um eine speziell angepasste Version von iOS 9. Durch diese Verwandtschaft ist es für erfahrene iOS-Entwickler kein Problem, Apps für Apple TV zu schreiben, und das natürlich mit Swift. Es wird lediglich Xcode ab Version 7.1 benötigt.

Aber es ist nicht nur möglich, Apps auf die herkömmliche Art und Weise zu entwickeln. Apple hat einen weiteren Weg erdacht, um Apps für Apple TV zu entwickeln. Das hat auch einen besonderen Grund. Eine App darf auf dem Flash-Speicher des Geräts maximal 200 MByte belegen. Das ist beispielsweise bei grafisch aufwendigen Spielen nicht gerade viel Platz. Diese Beschränkung liegt daran, dass das Gerät maximal 64 GByte als Flashspeicher zur Verfügung stellt. In der Basisversion sind es sogar nur 32 GByte. Eine Erweiterung

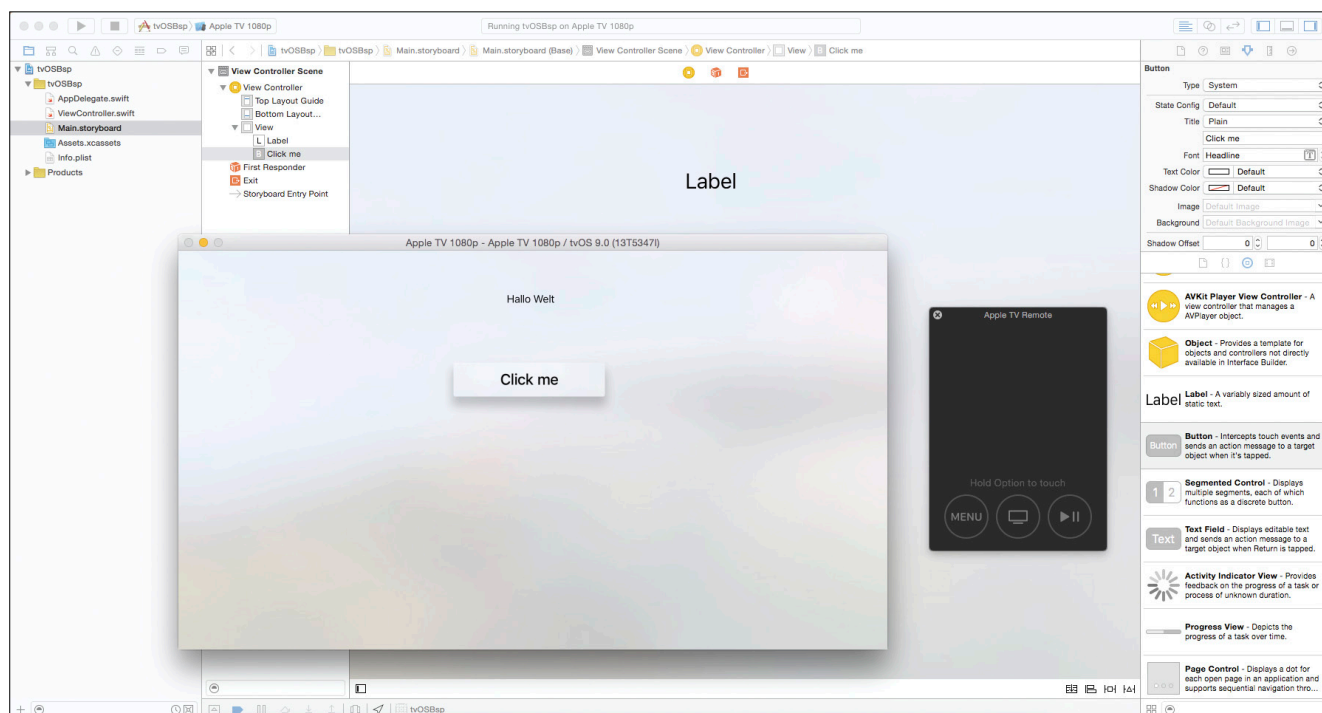


Die Projektvorlagen für eine tvOS-App in Xcode (Bild 1)

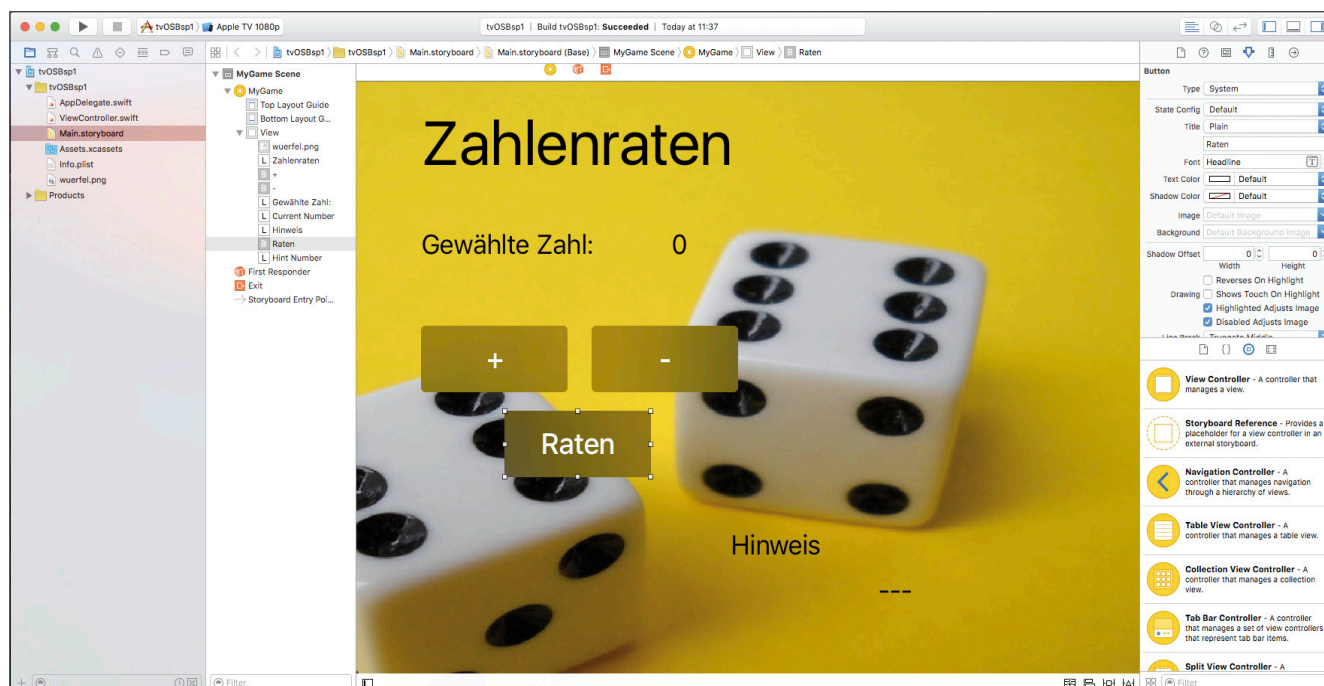
des Speichers ist – Apple-typisch – nicht möglich. App-Entwickler sollen ihre Apps so konzipieren, dass diese bei Bedarf zusätzliche Daten, zum Beispiel Grafiken, Sound et cetera, aus dem Netz nachladen.

Wege zur App (Xcode-Projektvorlagen)

Neben den typischen Projektvorlagen in Xcode gibt es noch eine weitere Möglichkeit, Apps für Apple TV zu entwickeln. Man spricht hier von sogenannten Client-Server-Apps. Dieser App-Typ (TVML – das steht für Apple TV Markup Lan-



Die Test-App im Apple-TV-Simulator (Bild 2)



Die Oberfläche der tvOS-App gestalten (Bild 4)

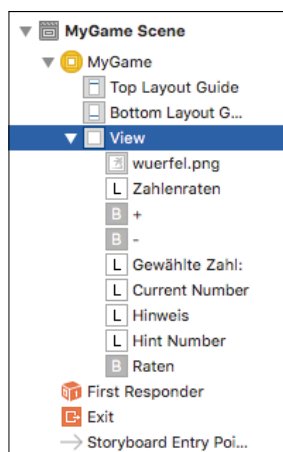
guage) ist vor allem dann interessant, wenn die App mit einer Datenbankverbindung arbeitet oder oft Daten von einem Server nachlädt.

Das ist zwar eine nette Neuerung, aber diese hat auch einen Haken. Dieser Typ von App lässt sich nicht so einfach mit den bekannten Mitteln (Swift, Xcode et cetera) beziehungsweise nur teilweise damit erstellen. Hauptsächlich basieren diese Apps auf einem HTML-Frontend, das zusätzlich JavaScript verwendet. Im Backend kann zum Beispiel Node.js verwendet werden.

Dieser Anwendungstyp wird, wie bereits erwähnt, noch nicht von Xcode unterstützt und wird daher auch nicht weiter betrachtet. Mehr Informationen zu diesem Thema finden Sie im Kasten »Links zum Thema«.

Wollen Sie eine App für tvOS auf dem klassischen Weg (mit Xcode und Swift) entwickeln, dann kann eine der neuen Projektvorlagen in Xcode 7.1 verwendet werden (Bild 1).

Drei Vorlagen findet man im entsprechenden Bereich: Game, Single View Application und Tabbed Application. Die Projektvorlage für ein Spiel unterscheidet sich nicht von der Variante für Standard-iOS. Benötigt die App nur eine View, so wird die Vorlage *Single View Application* verwendet. Soll eine App entwickelt werden, die mehrere Views verwendet, so wird am besten gleich die Projektvorlage *Tabbed Appli-*



Die Controls des Beispielprojekts (Bild 3)

cation verwendet. Diese Vorlage enthält bereits eine TabBar sowie zwei damit verknüpfte ViewController.

Auflösungsfragen

Das Beispiel verwendet die Vorlage *Single View Application*. Bevor man allerdings loslegen kann, sollte man sich ein wenig mit der neuen Umgebung beschäftigen, da diese sich doch an der ein oder anderen Stelle etwas anders verhält als gewohnt.

Die Unterschiede beginnen bereits bei der Auflösung der App. Während man bisher gewohnt ist, unterschiedliche Auflösungen bei der Entwicklung zu berücksichtigen und hierbei auch das Größenverhältnis (1x, 2x, 3x et cetera) beachten muss, so ist dies bei einer tvOS-App zurzeit noch etwas einfacher. Apple TV beziehungsweise tvOS verwendet eine Auflösung von 1920 x 1080 Punkten (1x). Die Apps sollten diese Auflösung natürlich berücksichtigen.

Diese Auflösung wird (im Querformat) derzeit auch vom iPhone 6s Plus und auch von den meisten HD-tauglichen Fernsehern verwendet. Außer der Auflösung gibt es beim Layout keine besonderen Punkte zur berücksichtigen. Die Controls werden aus der Object Library via Drag and Drop in die View befördert und können innerhalb der View mittels Constraints verankert werden. Eine Besonderheit gibt es hingegen bei der Implementierung der Steuerung der App zu beachten.

Fokus und Steuerung

Apple TV beziehungsweise die Apps auf dem Gerät werden natürlich nicht wie bei iOS üblich über Gesten gesteuert, die direkt auf dem Screen des jeweiligen Geräts ausgeführt werden. Abgesehen von der fehlenden Touchfähigkeit der wohl meisten TV-Geräte wäre das auch sehr unbequem. ►

Stattdessen liegt jedem Apple TV 4 eine neuartige Fernbedienung bei, welche neben der Sprachsteuerung mittels Siri auch die Steuerung über Gesten mittels einer kleinen Touch-Oberfläche auf der Fernbedienung ermöglicht. An diese Art der Steuerung muss man sich erst einmal gewöhnen. Am besten legt man hierzu auf der View erst einmal einen Button- sowie ein Label-Control ab, um mit der Steuerung etwas zu experimentieren.

Wie unter iOS, so wird auch unter tvOS eine Action für ein Control mittels [Ctrl] und Drag generiert. Generieren Sie also eine Action für den Button. Sobald dieser betätigt wurde, soll ein Text in das Label-Control geschrieben werden:

```
@IBAction func btnPressed(sender: AnyObject) {
    self.label.text = "Hallo Welt"
}
```

Wenn die Funktion umgesetzt worden ist, kann die App auf den Apple-TV-Simulator übertragen und durch Betätigung der Schaltfläche *Build and Run* gestartet werden.

Gleich nach dem Start wird die App im Apple-TV-Simulator geöffnet (Bild 2). Bewegt man intuitiv den Maus-Cursor über die Oberfläche der App, so passiert nichts. Die Interak-

tion mit Apple TV erfolgt ja auch über Apple Remote. Der Simulator stellt zur Simulation der Fernbedienung ebenfalls ein Stück Software bereit. Dieses muss aber zunächst im Simulator aktiviert werden.

Der Simulator und Apple Remote

Im Menü *Hardware* des Simulators gibt es den Menüpunkt *Show Apple TV Remote* (Shortcut: [Befehl Umschalt R]). Nun wird die Fernbedienung eingeblendet. Die Bedienung ist gewöhnungsbedürftig. Ein Mausklick aktiviert das zurzeit selektierte Control. Innerhalb des Simulators kann man diese Interaktion auch verfolgen. Wie aber kann man weitere Controls, die in der View sind, auswählen? Nur durch Bewegung der Maus ist das (im Vergleich zu den iOS-Simulatoren) nicht möglich.

Um von einem Element des GUI der App zu einem anderen zu springen, muss die [Alt]-Taste gedrückt gehalten werden. Erst dann werden Mausbewegungen in der Apple-Remote-Touchoberfläche an die App weitergegeben. Auch diese Form der Steuerung ist etwas gewöhnungsbedürftig. Am besten hält man während der Interaktion mit der Apple TV Remote im Simulator immer die Alt-Taste gedrückt, dann funktioniert es eigentlich recht einfach.

Listing 1: Quellcode des Spiels (Teil 1)

```
class ViewController: UIViewController {

    var numberToGuess = 0
    var inputNumber = 0
    var tryCounter = 0

    @IBOutlet var currentNumber: UILabel!
    @IBOutlet var hintNumber: UILabel!

    override func viewDidLoad() {
        super.viewDidLoad()
        initNumberToGuess()
    }

    override func
        didReceiveMemoryWarning() {
            super.didReceiveMemoryWarning()
        }

    @IBAction func
        increaseNumber(sender: AnyObject)
    {
        inputNumber += 1
        currentNumber.text =
            String(inputNumber)
    }

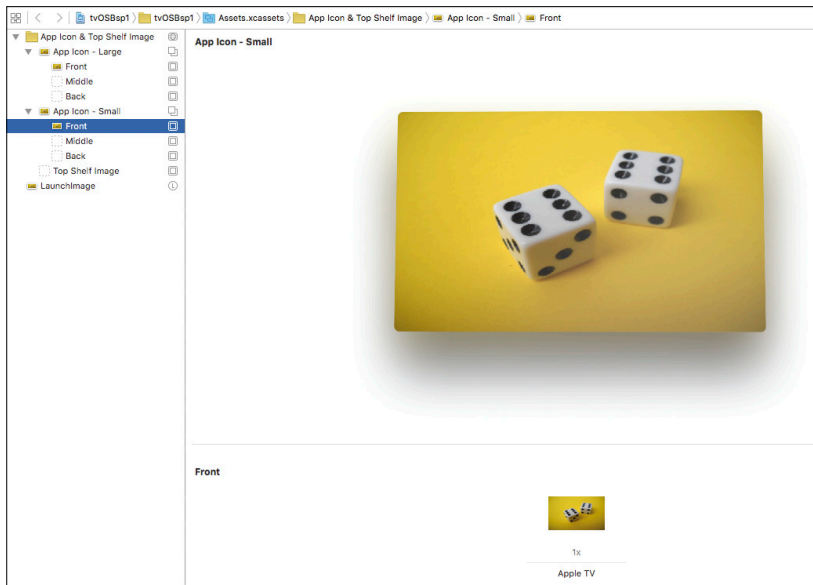
    @IBAction func
        decreaseNumber(sender: AnyObject)
    {
        inputNumber -= 1
        currentNumber.text =
            String(inputNumber)
    }

    @IBAction func guessNumber(sender:
        AnyObject) {
        startGuessing()
    }

    func initNumberToGuess() {
        numberToGuess = getRandomNumber(0,
            max: 60)
    }

    func getRandomNumber(min: Int,
        max: Int) -> Int {
        return min +
            Int(arc4random_uniform(
                UInt32(max - min + 1)))
    }

    func startGuessing() {
        if (inputNumber > numberToGuess) {
            tryCounter++
            hintNumber.text =
                "Zahl zu groß!"
        }
    }
}
```



Hinzufügen eines Icons zur App (Bild 5)

Wie kommt es nun zu dieser Form der Steuerung? Da es keine Touchsteuerung gibt, sondern die komplette Bedienung über die Fernbedienung erfolgt, hat sich Apple ein neues System zur Erkennung der Eingabe überlegt, das den Namen Focus Engine trägt. Das Konzept dahinter legt fest, dass immer nur ein Element der Oberfläche den Eingabefokus haben kann. Benutzer ändern den Fokus durch Verwendung von Swipe-Gesten (Swipe Left, Swipe Right) auf der Touchoberfläche der Fernbedienung. Die Focus Engine entscheidet

dabei, welches Control als Nächstes den Fokus erhält, nachdem der Benutzer eine entsprechende Geste auf der Fernbedienung gemacht hat. Somit muss man sich als Entwickler darüber keine weiteren Gedanken machen.

Würfeln auf dem TV

Eine kleine Beispiel-App, ein Spiel zum Erraten von Zahlen, soll einmal demonstrieren, wie eine App für tvOS entwickelt wird. Sie werden dabei feststellen, dass sich der Entwicklungsprozess eigentlich nicht wesentlich von dem einer normalen iOS-App unterscheidet.

Zu Beginn wird eine Projektvorlage vom Typ Single View Application als Basis verwendet. Gleich nach der Generierung des Projekts werden Sie je nach zur Verfügung stehendem Monitor feststellen, dass gegebenenfalls viel gescrollt werden muss.

Denken Sie daran, dass Sie eine verkleinerte Übersicht im Storyboard erhalten, sobald in einem leeren Bereich ein Doppelklick ausgeführt wird. Für das Beispiel werden einige Controls benötigt. Eine Auflistung können Sie Bild 3 entnehmen.

Damit die App auch ein wenig nach Spiel aussieht, wird in der View zuerst ein ImageControl eingefügt, das sich komplett über die View ausdehnt. In der View wird das Bild *wuerfel.png*, das zuvor dem Projekt hinzugefügt wurde, zur ►

Listing 1: Quellcode des Spiels (Teil 2)

```

if (inputNumber < numberToGuess)
{
    tryCounter++
    hintNumber.text =
        "Zahl zu klein!"
}
if (inputNumber == numberToGuess)
{
    tryCounter++;
    hintNumber.text =
        "Richtig geraten!";
    showAlertViewWithTitle(
        "Zahl gefunden!!",
        message:"Sie haben \(tryCounter) Versuche
        benötigt. Neue Zahl wird generiert.")
    tryCounter = 0;
    initNumberToGuess()
}
}

func
    showAlertViewWithTitle(
        title:String, message:String) {
    let alertController =
        UIAlertController(title:
            title, message: message,
            preferredStyle: .Alert)

    let OKAction = UIAlertAction(
        title: "OK", style: .Default)
        { (action) in
            //
        }

    alertController.addAction(
        OKAction)
    self.presentViewController(
        alertController, animated:
            true) {
    }
}

```



Die installierte App im tvOS-Simulator (Bild 6)

Anzeige ausgewählt. Das Bild hat eine Auflösung von 1920 x 1080 Punkten. Anschließend werden in die View fünf Label- und drei Button-Controls eingefügt. Diese sollten wie in **Bild 4** angeordnet werden. Wurde das GUI fertiggestellt, kann es an die Programmierung der Logik gehen.

Logik zum Würfeln

Für die drei Buttons werden ebenso viele Actions benötigt. Es müssen die Actions *increaseNumber* sowie *decreaseNumber* angelegt werden. Wie es die Bezeichnungen bereits vermuten lassen, werden diese beziehungsweise der darin enthaltene Code verwendet, um den Tipp des Spieles zu erhöhen oder zu verringern. Die letzte Action *guessNumber* wird mit dem Button *Raten* verknüpft.

Neben den Actions werden auch noch zwei Outlets für die Labels *currentNumber* und *hintNumber* benötigt. Das erste Label zeigt die Zahl, die den aktuellen Tipp des Spielers enthält. Das zweite Label wird benutzt, um dem Spieler einen Hinweis zu geben, nachdem dieser den Button *Raten* betätigt hat. Anschließend wird dem Spieler mitgeteilt, ob sein Tipp richtig gewesen ist oder aber die Zahl von ihm zu niedrig oder zu hoch angegeben wurde.

Die Klasse *ViewController* (**Listing 1**) wird auch in einer tvOS-App von der Basisklasse *UIViewController* abgeleitet. Generell fällt auf, dass es zwischen der Entwicklung einer tvOS- und einer reinen iOS-App nur wenig Unterschiede gibt. Diese sind eigentlich hauptsächlich im Storyboard und einigen anderen Bereichen des Projekts zu finden.

Zu Beginn der Klasse werden drei Variablen *numberToGuess*, *inputNumber* und *tryCounter* angelegt und initialisiert. Die Namen der Variablen sind selbsterklärend. Im Ereignis *ViewDidLoad* wird die Methode *initNumberToGuess* aufgerufen. Darin wird die zuvor generierte Zufallszahl der Variablen

numberToGuess zugewiesen. Es folgen drei Actions, die den Code enthalten, der nach Betätigung eines Buttons ausgeführt wird.

Interessant ist noch die Methode *startGuessing*. Diese enthält im Prinzip die komplette Logik des Spiels. Hier wird ermittelt, ob der in *inputNumber* (enthält den Tipp des Spielers) enthaltene Wert kleiner, gleich oder größer dem Wert in *numberToGuess* ist. Stimmen die Werte überein, wird eine entsprechende Meldung ausgegeben. Zuletzt gibt es noch die Methode *showAlertViewWithTitle*, die einen Dialog mit Text anzeigt. In diesem Fall wird dieser Dialog verwendet, um dem Spieler mitzuteilen, dass er die gesuchte Zahl gefunden hat und wie viele Versuche er dafür benötigt hat.

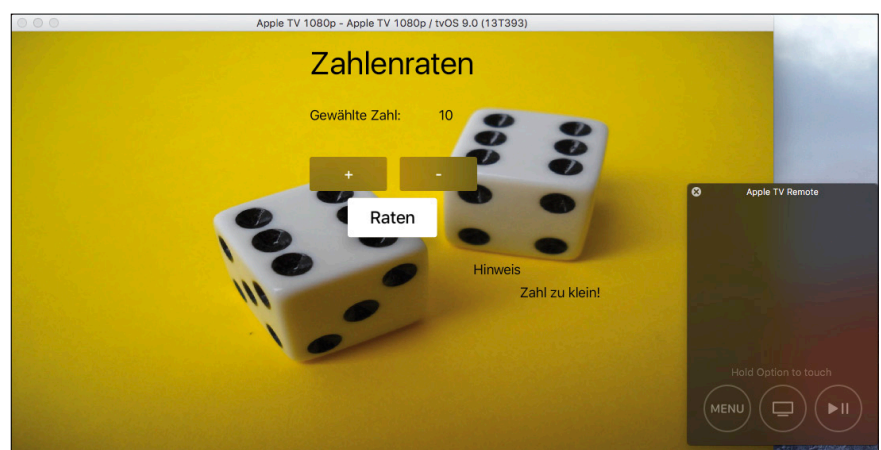
Icons

Nachdem das Programm fertiggestellt wurde, geht es an das Design der Icons der App. Dieser Punkt unterscheidet sich wieder etwas von der Umsetzung für Standard-iOS. An das App-Icon werden unter tvOS etwas andere Anforderungen gestellt, als dies unter iOS der Fall ist.

Dies liegt daran, dass ein Fernseher sich sicherlich nicht so dicht vor dem Gesichtsfeld des Anwenders befindet, wie das bei einem iOS-Gerät (zum Beispiel iPhone, iPad) der Fall ist. Das Icon muss also auch aus einiger Entfernung noch gut erkennbar sein.

Um die Aufmerksamkeit auf ein selektiertes Programm zu fokussieren, tanzt das Icon nach der App-Auswahl unter tvOS. Sie kennen diesen Effekt vom iPhone/iPad her. Dort wird er verwendet, um zu signalisieren, dass die Anwendung bearbeitet (zum Beispiel gelöscht oder verschoben) werden kann. Konfiguriert wird das App-Icon im Bereich *Assets.xcassets* des Projekts (**Bild 5**).

Man benötigt zur Konfiguration das Icon in unterschiedlichen Größen: Large (1280 x 768) und Small (400 x 240). Hierbei muss unbedingt auf die Qualität des Bildes geachtet werden. Denn das kleinere wird für die Selektion in der tvOS-



Die fertige App zur Laufzeit im tvOS-Simulator (Bild 7)

Links zum Thema

- Creating a Client-Server App
https://developer.apple.com/library/tvos/documentation/General/Conceptual/AppleTV_PG/YourFirstAppleTVApp.html
- Human Interface Guidelines Icon & Images
<https://developer.apple.com/tvos/human-interface-guidelines/icons-and-images>
- Testflight
<https://developer.apple.com/testflight>

Oberfläche in unterschiedlichen Größen angezeigt: 370 x 222 (Status: fokussiert) sowie 300 x 180 (nicht fokussiert). Das Icon lässt sich aber noch weiter bearbeiten. So kann es auch aus mehreren Ebenen bestehen.

Dies verstärkt den Eindruck der Selektion unter tvOS. Wenn mehrere Ebenen (drei) verwendet werden, so kann man festlegen, in welcher Ebene (Front, Middle, Back) ein bestimmtes Bild angezeigt werden soll. Testen Sie den Effekt, um ein befriedigendes Ergebnis zu erhalten (Bild 6).

Installation der App auf einem Gerät

Wie eine entwickelte iOS-App zum Test auf ein iOS-Gerät beispielsweise ein iPhone oder iPad, übertragen wird, ist klar. Das Gerät wird einfach an den Mac angeschlossen und fertig. Wie ist das aber mit dem Apple TV?

Es gibt zur Übertragung einer App zwei Möglichkeiten. Die erste ist, dass das Apple TV 4 mittels des beiliegenden USB-Kabels an den Mac angeschlossen wird. Xcode erkennt daraufhin, dass ein Apple TV 4 angeschlossen wurde, und erlaubt das automatische Kopieren und im Anschluss daran die Ausführung der App direkt auf dem Gerät. Eine weitere Möglichkeit besteht darin, das Apple-TV-4-Gerät, auf dem die App getestet werden soll, zu registrieren und die App mittels Testflight zu übertragen (Bild 7).

Fazit

Mit Apple TV 4 ist es allen iOS-Entwicklern erstmals möglich, Apps auch für die Set-Top-Box von Apple bereitzustellen. Die Entwicklung dieser Apps für Apple TV unterscheidet sich dabei nur in wenigen Aspekten von der Entwicklung für ein herkömmliches iOS-Gerät. ■



Christian Bleske

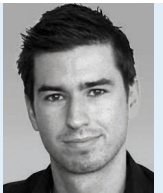
ist Autor, Trainer und Entwickler mit dem Schwerpunkt Client/Server und mobile Technologien. Sein Arbeitsschwerpunkt liegt auf Microsoft-Technologien.

cb.2000@hotmail.de

Komprimiertes Know-how für Entwickler

Entity Framework und C#

Referent: Christian Giesswein
18.02.2016, 60 min.



CQRS und Multi-Model-DB

Referent: Jan Fellien
23.02.2016, 60 min.



Smart Development mit Enapso

Referent: Alexander Schulze
25.02.2015, 60 min.



Smart Data mit Microsoft Azure

Referent: Sascha Dittmann
01.03.2016, 60 min.



MS SQL Server für Entwickler

Referent: Thorsten Kansy
10.03.2016, 120 min.



Cross-Plattform-Entwicklung mit Visual Studio 2015

Referent: André Krämer
On-demand, 120 min.



SOLID Prinzipien

Referent: David Tielke
On-demand, 60 min.



MUST-HAVE LIBRARIES FÜR ANDROID-ENTWICKLER

Zur freien Verfügung

Mit ein paar Open-Source-Libraries lassen sich schnell Android-Apps mit einem guten Nutzererlebnis entwickeln.

Anhand von konkreten Beispielen zeigt der Artikel die Verwendung der Bibliotheken auf. Die Beispielanwendung ist eine klassische Client-Anwendung, welche Daten von einem REST-Service lädt und diese dem Benutzer auch bei schlechter beziehungsweise keiner Internetverbindung ohne Verzögerung anzeigt (sofern die Daten bereits abgerufen wurden).

Einfache Lösungen

Ein guter Entwickler weiß, dass er das Rad nicht bei jedem Projekt neu erfinden muss, es sei denn, der Entwickler möchte sich näher mit der Erfindung des Rades beschäftigen. In diesem Fall steht dann aber nicht die Entwicklung der App im Vordergrund, sondern die Entwicklung einer eigenen Bibliothek.

Neben den gängigen Software-Design-Patterns, die für gängige Probleme einfache Lösungen bereitstellen, existieren eine ganze Reihe von Libraries, welche die Community frei zur Verfügung stellt und die Entwickler frei verwenden können.

Einige dieser Libraries lösen sehr spezifische Probleme, andere hingegen lassen sich für unterschiedlichste Projekte verwenden. Der Artikel möchte einige dieser Libraries vorstellen, mit denen man beispielsweise eine Android-App pro-



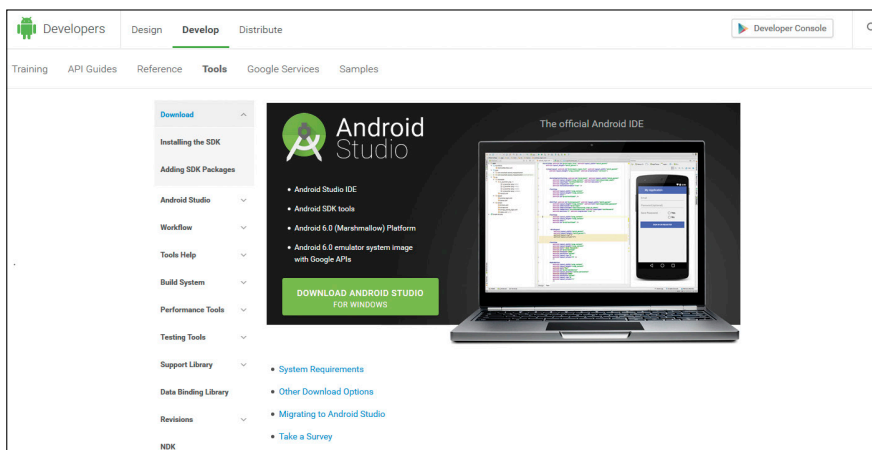
grammieren kann, die auf Daten eines Service im Internet zugreift (also potenziell mit schlechter Netzwerkverbindung zu kämpfen hat). Die App soll dabei jederzeit eine flüssige Bedienung ermöglichen und mit ansprechenden Effekten versehen sein.

Mit Android Studio (<https://developer.android.com/sdk/index.html>) steht Entwicklern eine mächtige und sehr gute Entwicklungsumgebung zur Verfügung (Bild 1). Mit einfachen Wizards lassen sich neue Projekte für unterschiedliche Plattformen (Smartphone und Tablet, Wear, TV, Android Auto und Glass) und unterschiedliche API-Levels erstellen (<http://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>). Neue Projekte enthalten aber immer nur die Standardfunktionen, die von Android selbst bereitgestellt werden. Der Entwickler muss weiterführende Funktionen – in Form von Libraries – hinzufügen (Bild 2).

Android Studio verwendet als Build-System Gradle (<http://gradle.org>). Neue Libraries werden dort als Dependency im Buildskript hinzugefügt, von der IDE automatisch heruntergeladen und dem Projekt hinzugefügt (Bild 3).

Das reicht zwar für den Einstieg in die Programmierung für das Android-Ökosystem und für kleine Hello-World-Projekte. Bei ambitionierten Projekten braucht man aber häufig weitere Funktionen, die man dank diverser Open-Source-Projekte nicht selbst zu programmieren braucht.

Neben den Libraries der Community gibt es auch bereits Erweiterungen von Google, die Funktionen von neueren API-Versionen in ältere portieren. Die Support Libraries (<http://developer.android.com/tools/support-library/index.html>) lassen sich wie alle anderen Libraries einfach als Abhängigkeit im Gradle Skript hinzufügen.



Android Studio – das Standardwerkzeug zum Entwickeln von Android-Apps (Bild 1)



In meinem GitHub-Repository, das Sie unter <https://github.com/tobiasamon/androidlibrariesexamples.git> erreichen, ist zu jeder Library ein Beispielprojekt abgelegt, das die Verwendung der jeweiligen Library in einer Minimalversion aufzeigt (Bild 4).

v7 Support Library

Es gibt aktuell zwölf verschiedene Support-Libraries, die diverse Funktionen nachliefern. Ich möchte hier als Beispiel aus der v7 Support Library beziehungsweise im Speziellen aus der v7 appcompat library die ActionBar (<http://developer.android.com/reference/android/support/v7/app/ActionBar.html>) zeigen, die in API-Level 21 grundlegend überarbeitet wurde und erheblich besser anpassbar ist.

Möchte man nun auch in niedrigeren API-Levels von den neuen Funktionen profitieren, so muss man im Gradle Buildskript des app-Moduls die Abhängigkeit `com.android.support:appcompat-v7` in der entsprechenden Version hinzufügen.

Developers

Develop

>

API Guides

>

<uses-sdk>

Introduction

App Components

App Resources

App Manifest

actions

activities

activities-alias

applications

categories

compatible-screens

dates

grant-uri-permission

instrumentation

intent-filters

manifests

meta-data

The table below specifies the API Level supported by each version of the Android platform. For information about the relative numbers of devices that are running each version, see the [Platform Versions dashboards page](#).

Platform Version	API Level	VERSION_CODE	Notes
Android 6.0	23	R	API Changes
Android 5.1	22	LOLLIPOP_P1	Platform Highlights
Android 5.0	21	LOLLIPOP	
Android 4.4W	20	KITKAT_WATCH	KitKat for Wearables Only
Android 4.4	19	KITKAT	Platform Highlights
Android 4.3	18	JELLY_BEAN_MR2	Platform Highlights
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1	Platform Highlights
Android 4.1, 4.1.1	16	JELLY_BEAN	Platform Highlights
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1	Platform Highlights
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH	
Android 3.2	13	HONEYCOMB_MR2	
Android 3.1.x	12	HONEYCOMB_MR1	Platform Highlights
Android 3.0.x	11	HONEYCOMB	Platform Highlights
Android 2.3.4	10	GINGERBREAD_MR1	Platform Highlights
Android 2.3.3			
Android 2.3.2	9	GINGERBREAD	
Android 2.3.1			
Android 2.3			
Android 2.2.x	8	FROYO	Platform Highlights
Android 2.1.x	7	ECLAIR_MR1	Platform Highlights

Mit einfachen Wizards lassen sich neue Projekte für unterschiedliche Plattformen und API-Level erstellen (Bild 2)

Die Library selbst lässt sich mit Hilfe des SDK-Managers auf den PC herunterladen. Das Buildskript sieht dann wie folgt aus:

```
}
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:22.2.1'
}
```

Mit dieser Erweiterung kann man nun als Basisklasse für die entsprechende Activity die Klasse `AppCompatActivity` verwenden. Das Menü, das man in der ActionBar anzeigen möchte, wird nun analog zu einem gewöhnlichen Options-Menu in einer XML-Datei im Ordner `res/menu` erstellt (Listing 1) und in der Activity als `OptionsMenu` hinzugefügt:

```
@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
```

In einer Standard-Anwendung für das API-Level 15 würden die Menüpunkte einfach am unteren Rand der Anwendung als Schaltflächen angezeigt. Das sieht aber nicht mehr zeitgemäß aus und muss auch nicht sein.

Fügt man die Komponenten zusammen wie oben beschrieben, erhält man auf Geräten mit API-Level 15 und auf Geräten mit API-Level 21 eine identische Anwendung. **Bild 5** zeigt, wie das Menü mit Hilfe der Support-Library in der Actionbar aussieht.

Platzierung von Elementen

Je nach Anwendung kommt es vor, dass man in seinem Layout sehr viele Elemente platziert, mit denen aus dem Code heraus interagiert werden muss. Der Standardweg ist normalerweise, dass man über die aktuelle Activity jedes Element per ID findet und dann noch auf die richtige Klasse casten muss.

Das bedeutet, dass man irgendwo im Code (zum Beispiel in der Methode `onCreate()`) für jedes Element eine Zeile unnötigen Code hinzufügen muss:

```
TextView myTextView = (TextView) findViewById(
    R.id.textview);
```

Wer mit vielen Elementen arbeitet, benötigt Butter Knife (<http://jakewharton.github.io/butterknife>). Mit dieser Library lassen sich die Views einfach in der aktuellen Activity binden und verwenden, ohne auf die richtige Klasse casten zu müssen (Bild 6).

Um Butter Knife verwenden zu können, muss man zunächst die Dependency `com.jakewharton:butterknife:7.0.1` im Buildskript hinzufügen. In der Activity wird anschließend jedes ►

Element, das man verwenden möchte, als Member-Variable mit der richtigen Klasse deklariert.

Das Binden des eigentlichen Objekts erfolgt dann einfach durch Hinzufügen einer Annotation, in der die Ressource-ID des GUI-Elements als Parameter übergeben wird:

```
@Bind(R.id.tv1) protected TextView textView1;
```

Im Layout lassen sich die Elemente wie gehabt definieren. Hier bedarf es keinerlei Anpassungen, was auch das nachträgliche Hinzufügen von Butter Knife zu einem bestehenden Projekt problemlos möglich macht.

Natürlich gibt es auch Alternativen zu Butter Knife, wie beispielsweise RoboGuice (<https://github.com/roboguice/roboguice>), die auch noch weitergehende Funktionen bieten. Für das einfache Injecten von Views aus dem Layout erscheint Butter Knife aber als beste Alternative, da man hier keine andere Activity-Basisklasse verwenden muss.

Glide

Häufig zeigen Android-Applikationen Bilder aus dem Internet an. Die Bilder direkt von einer Webseite einzubinden kann einige Probleme bereiten, beispielsweise dann, wenn die Netzwerkverbindung schlecht ist oder der Server aus irgendeinem Grund nicht verfügbar ist.

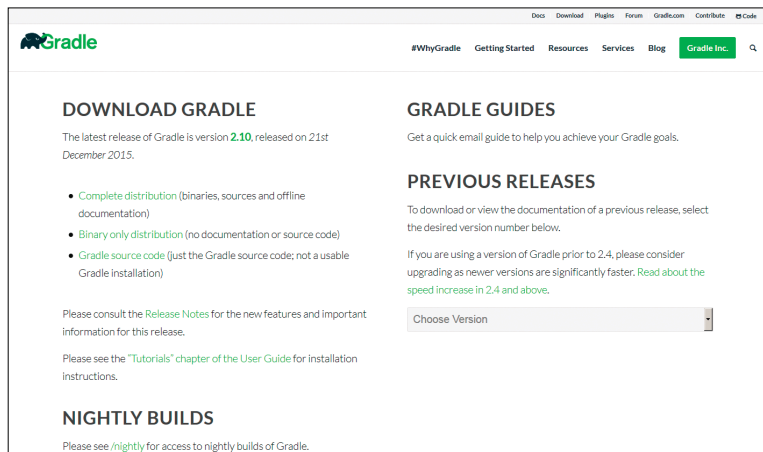
Mit Glide (<https://github.com/bumptech/glide>) braucht man sich über das Handling bezüglich des Ladens der Bilder, der Anzeige eines Ladeindikators und Verwaltung eines Disk Caches keine Sorgen mehr zu machen.

All dies deckt die Library ab. Glide lässt sich sowohl für einfache *ImageViews*, bei denen nur ein Bild von einem Webserver angezeigt wird, als auch für eine Liste von Bildern, die in einer *ListView* angezeigt werden, verwenden.

Alle Funktionen lassen sich durch ein einfaches API verwenden. Ein erster Schritt ist wie immer das Hinzufügen der

Listing 1: Definition der Menu-Items

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android=
"http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/
    apk/res-auto">
    ...
    <item
        android:id="@+id/action_backup"
        android:icon="@mipmap/ic_backup_black_24dp"
        android:orderInCategory="100"
        app:showAsAction="always"
        android:title="Backup" />
    ...
</menu>
```



Build-System: Android Studio verwendet hier Gradle (Bild 3)

richtigen Abhängigkeit zum Buildskript:

```
com.github.bumptech.glide:glide:3.6.1
```

Im Beispiel sind einige Bilder von einer Webseite in einer Liste angezeigt. Die Bilder werden direkt aus dem Internet in die View geladen. Alles, was man braucht, ist eine *ListView* und ein entsprechender *Array-Adapter*. Dem *ImageArrayAdapter* im Beispiel übergibt man im Konstruktor eine Liste von URLs als Strings.

Diese werden dann an der entsprechenden Position an Glide übergeben, damit diese geladen und angezeigt werden

Listing 2: ImageArrayAdapter für Glide

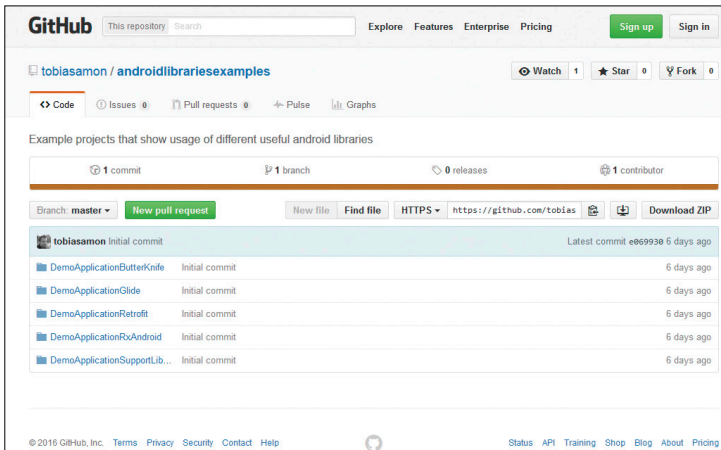
```
Class ImageArrayAdapter extends ArrayAdapter<String>
{
    private List<String> urls;

    ...

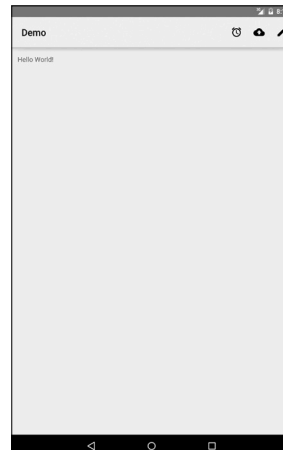
    @Override
    public View getView(int position,
        View convertView, ViewGroup parent) {
        ...

        Glide.with(getContext())
            .load(urls)
            .centerCrop()
            .placeholder(
                (R.drawable.ic_sync_black_24dp)
            ).crossFade()
            .into(imageView);

        return convertView;
    }
}
```

Ein Beispielprojekt zu jeder vorgestellten Library gibt es im GitHub-Repository des Autors (Bild 4)



App mit der Support-Library Butter Knife (Bild 5)

ment die Animation stattfinden soll:

```
YoYo.with(Techniques.
Tada)
    .duration(700)
    .playOn(findViewById
(R.id.item_to_
animate));
```

Anschließend ist noch zu entscheiden, wann die Animation ausgeführt werden soll. Dies kann zum Beispiel direkt beim Klick auf eine Schaltfläche geschehen oder nach dem Prüfen eines Eingabewerts. Dadurch kann man einfach ein positives oder negatives Feedback an den Benutzer auf seine Aktion geben.

können. In Listing 2 ist der relevante Teil des Array-Adapters enthalten, der die Integration von Glide zeigt.

Die ganze Logik für die Anzeige der Bilder, das Caching auf dem Gerät und das Tauschen des Ladeindikators mit dem richtigen Bild ist in einer einzigen Zeile enthalten. Würde man dies selbst implementieren, würde man sehr viel Zeit damit verschwenden, Code zu schreiben, den andere Entwickler bereits entwickelt und getestet haben.

Auch hier gibt es eine Alternative, nämlich Picasso (<http://square.github.io/picasso>). Verwendung und Syntax sind nahezu gleich. Welche Library man am Ende verwendet, ist Geschmackssache, beide Libraries sind jedoch nützlich (Bild 7).

AndroidViewAnimations

Google hat mit der Material Design Specification (<http://www.google.com/design/spec>) einen umfassenden Katalog an Vorgaben für das Design von Bedienoberflächen erstellt. Die Hauptelemente sind:

- ein neues Theme,
- neue Widgets für komplexe Views,
- neue APIs für selbsterstellte Schatten und Animationen.

Die Library AndroidViewAnimations (<https://github.com/daimajia/AndroidViewAnimations>) bietet für Standard-Elemente eine breite Palette an Animationen, die sich mit Hilfe eines einfachen API verwenden lassen. Eine Animation kann dem App-Benutzer ganz einfach ein optisches Feedback über seine Aktionen geben.

Als Abhängigkeit zusätzlich zur eigentlichen Library werden weitere Abhängigkeiten eingebunden, die als Grundlage für die Animationen dienen. Es sind also insgesamt drei Libraries, die hier verwendet werden:

- com.nineoldandroids.library:2.4.0,
- com.daimajia.easing.library:1.0.1@aar,
- com.daimajia.androidanimations.library:1.1.3@aar.

Die Verwendung ist im Grunde selbsterklärend. Wie die nachfolgenden Codezeilen zeigen, wählt man eine Animation sowie eine Dauer und gibt dann an, auf welchem Ele-

Retrofit

Bei Retrofit (<http://square.github.io/retrofit>) handelt es sich um eine Library, mit der man REST-Services ganz einfach als Java-Interfaces deklarieren kann (Bild 8). Ein REST-Service wird damit erstellt, indem die Methoden des Interfaces jeweils mit einer Annotation versehen werden. In Retrofit wird seit Version 2 der alternative HTTP-Client OkHttp (<http://square.github.io/okhttp>) als Standard-HTTP-Client verwendet. Damit hat der Entwickler unter anderem die Möglichkeit, einen Response-Cache zu konfigurieren, der die Daten speichert und zurückgibt, wenn beispielsweise die Netzwerkverbindung gerade nicht verfügbar ist.

Das Beispiel verwendet einen öffentlichen REST-Service (<http://jsonplaceholder.typicode.com>), der Beispieldaten zur Verfügung stellt und alle HTTP-Methoden unterstützt. Das Beispiel greift auf den Service zu, der Kommentare zurückliefert (<http://jsonplaceholder.typicode.com/comments>). ▶



Mit Butter Knife lassen sich die Views einfach in der aktuellen Activity binden und verwenden (Bild 6)

Der REST-Service gibt die Kommentare als JSON-Array zurück. Damit der Server die Antwort auch verarbeiten kann, ist es notwendig, noch einen Standard-Converter für die Verarbeitung von JSON-Objekten zu definieren. Hier gibt es für Retrofit bereits vorgefertigte Converter, die sich einfach einbinden lassen. Das Beispiel verwendet den GSON-Converter. Die Funktionsweise wird an späterer Stelle noch kurz beschrieben.

Für die Verwendung von Retrofit und des REST-Service, der JSON-Objekte zurückliefert, ist es notwendig, unserem Projekt zwei Abhängigkeiten hinzuzufügen:

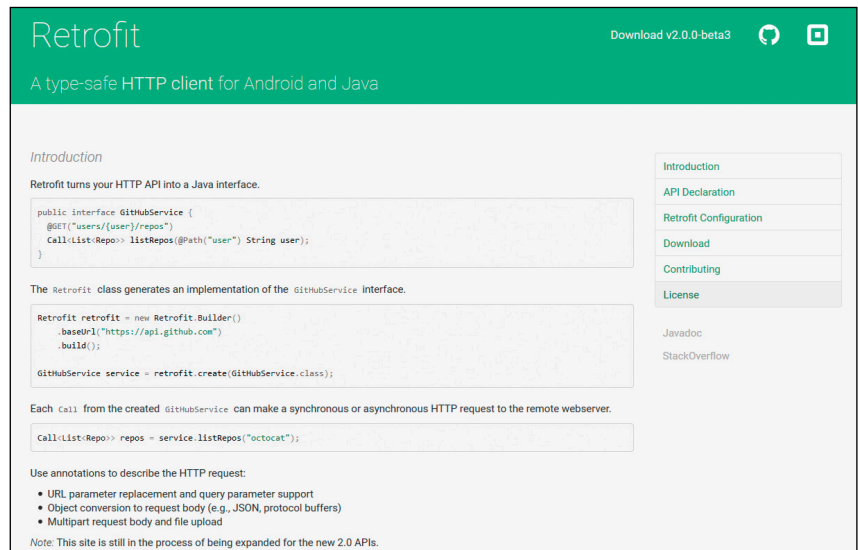
- `com.squareup.retrofit:retrofit:2.0.0-beta2`,
- `com.squareup.retrofit:converter-gson:2.0.0-beta2`.

Anschließend kann man ein Java-Interface definieren, das eine Methode enthält, die auf die Kommentarliste zugreift. Im Service-Interface wird nur der Pfad zum REST-Service selbst angegeben:

```
public interface CommentService {
    @GET("/comments")
    Call<List<Comment>> listComments();
}
```

Der Rückgabewert ist vom Typ *Call*. Dieses Interface erlaubt es, den eigentlichen Aufruf synchron oder auch asynchron durchzuführen. Als Typ-Parameter wird der eigentliche Ergebnistyp unseres Aufrufs angegeben. Zu erwarten ist hier eine Liste von Comment-Objekten. Das Comment-Objekt wird analog der Struktur im JSON-Objekt erstellt. Dies ermöglicht es, dem GSON-Converter, die Antwort vom Server automatisch in Java-Objekte zu konvertieren.

Nachdem der Service, inklusive Ergebnistyp, definiert wurde, lässt sich mit Hilfe des Retrofit-Builders die Instanz ei-

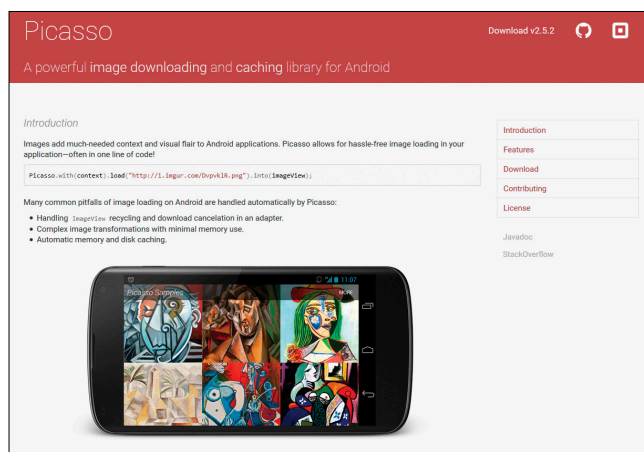


Mit Retrofit lassen sich REST-Services ganz einfach als Java-Interfaces deklarieren (Bild 8)

nes Retrofit-Objekts erstellen, mit dem sich wiederum eine Instanz eines Service erstellen lässt. Im Builder konfiguriert man zunächst den URL, unter welchem der Service erreichbar ist. Anschließend muss man noch bekannt geben, dass man den GSON-Converter verwenden möchte. Zuletzt erstellt man eine Retrofit-Instanz, um eine Service-Instanz zu erstellen (Listing 3).

Hat man nun seine Service-Instanz, kann man mit dem Methodenaufruf eine Instanz des Call-Objekts erstellen. Dabei wurde der Service selbst noch nicht aufgerufen. Dies geschieht nun, indem man den Call ausführt. Der Aufruf geschieht am besten asynchron. Befindet man sich momentan auf dem Main Thread (UI Thread), erreicht man dies durch den Aufruf der Methode *enqueue*. Befindet man sich bereits auf einem anderen Thread, kann man den Service auch synchron mit dem Aufruf von *execute* durchführen. Unser Beispiel verwendet die asynchrone Methode, bei der man ein Callback-Objekt angeben muss, dem das Ergebnis aus dem Aufruf übergeben wird. In diesem Fall implementiert die *MainActivity* das Callback-Interface.

Die Kommentare sollen in einer Liste auf der GUI angezeigt werden. Dazu wurde ein einfacher Array-Adapter implementiert, der diese Aufgabe übernimmt. Nachdem der Server geantwortet hat, wird in der *onResponse*-Methode zunächst der



Picasso stellt eine Alternative zu Glide dar (Bild 7)

Listing 3: Serviceklasse mit Retrofit konfiguriert

```
Retrofit retrofit = new Retrofit.Builder()
    .baseUrl("http://jsonplaceholder.typicode.com")
    .addConverterFactory
    (GsonConverterFactory.create())
    .build();
CommentService service = retrofit.
    create(CommentService.class);
```

Adapter geleert und anschließend mit den empfangenen Daten wieder befüllt.

Der Aufruf des REST-Service funktioniert an dieser Stelle bereits problemlos. Wenn man allerdings für ein mobiles Endgerät programmiert, ist immer zu bedenken, dass man die App auch einmal benutzen wird, wenn man keine Netzwerkverbindung hat. Hier kommt das Caching ins Spiel, das heißt, das Zwischenspeichern der Antworten auf dem Gerät beziehungsweise in der App. Der Benutzer soll idealerweise davon nichts mitbekommen.

Zum Glück lässt sich der Cache in Retrofit mit Hilfe von OkHttp sehr leicht einbauen beziehungsweise aktivieren. Die nötigen Funktionen werden direkt mitgeliefert und müssen nur konfiguriert werden. Zuerst muss auf dem Dateisystem ein Cache eingerichtet werden, der dann dem HTTP-Client bekannt gegeben wird. Anschließend ist dem REST-Client noch der angepasste HTTP-Client hinzuzufügen. Die Anpassungen sind im Listing 4 enthalten.

Das Beispiel enthält das minimale Setup für einen REST-Service-Client mit Cache. Was hier nicht abgedeckt ist, ist die Steuerung des Caches bezüglich der Aktualisierung der Daten et cetera. Dies bezieht jedoch ebenso die Anpassung des Servers mit ein, da dieser in den Netzwerk-Responses die Header entsprechend setzen muss.

RxAndroid

Zugriffe auf das Netzwerk sind aber nicht die einzigen Aufrufe, die potenziell länger dauern können. In Android lassen sich langlaufende Aufgaben meist mit einem AsyncTask lösen, der die Aufgabe auf einem anderen Thread im Hintergrund ausführt.

Die Fehlerbehandlung ist das erste Problem, mit dem AsyncTasks den Entwickler konfrontiert. Standardmäßig gibt es keine Lösung dafür, wie mit Fehlern umzugehen ist, die in der Methode *doInBackground* auftreten. Das Problem lässt sich beheben, indem man als Rückgabewert ein Paar aus dem eigentlichen Ergebnis und einem eventuellen Fehler definiert: *<ExpectedResult, Exception>*.

Das zweite Problem ist, dass ein AsyncTask nicht über den Lifecycle einer Activity informiert wird. Soll aufgrund des Ergebnisses aus dem AsyncTask das UI aktualisiert werden, muss der Entwickler auf den Status der Activity achten, damit er keinen unbehandelten Fehler bekommt. Alternativ kann die Activity eine Referenz auf den AsyncTask halten und diesen abbrechen, sofern die Activity pausiert oder beendet wird.

Zum Glück gibt es aber mit RxAndroid (<https://github.com/ReactiveX/RxAndroid>) eine Library, die diese und noch weitere Probleme auf elegante Weise löst.

Mit Retrofit erhält man zwar direkt die Möglichkeit, eine Netzwerkabfrage im Hintergrund auszuführen, die Verwendung von RxAndroid bietet darüber hinaus aber noch weitere Möglichkeiten.

Zunächst ist die folgende Abhängigkeit zum Gradle Buildskript hinzuzufügen: *io.reactivex:rxandroid:1.0.1*.

Die Verwendung von RxAndroid ist nun denkbar einfach. Es ist lediglich ein Observable-Objekt zu erzeugen, auf dem

man einen Observer registriert. Außerdem lässt sich der Thread Pool konfigurieren, auf dem die Berechnung ausgeführt wird. Zudem kann man definieren, auf welchem Thread das Ergebnis ausgeführt wird:

```
...
getObservable()
    .subscribeOn(Schedulers.newThread())
    // Do computation on a new thread
    .observeOn(AndroidSchedulers.mainThread())
    // Perform updates on main thread
    .subscribe(this);
// Set the current class as observer
...
```

Das erste Problem, mit dem ein Entwickler zu kämpfen hat, ist die Behandlung von Fehlern, die bei der Berechnung ►

Listing 4: Retrofit mit Caching

```
...
Retrofit retrofit = new Retrofit.Builder()
    .client(createClient())
    .baseUrl("http://jsonplaceholder.typicode.com")
    .addConverterFactory
        (GsonConverterFactory.create())
    .build();
CommentService service =
    retrofit.create(CommentService.class);
Call<List<Comment>> comments =
    service.listComments();
comments.enqueue(this);
}

private OkHttpClient createClient() {
    OkHttpClient client = new OkHttpClient();
    Cache cache = createCache();
    if (cache != null) {
        client.setCache(cache);
    }
    return client;
}

private Cache createCache() {
    File httpCacheDirectory = new File(getCacheDir(),
        "responses");
    Cache cache = null;
    if (!httpCacheDirectory.exists()) {
        httpCacheDirectory.mkdir();
    }
    if (httpCacheDirectory != null) {
        cache = new Cache(httpCacheDirectory,
            10 * 1024 * 1024);
    }
    return cache;
}
```

innerhalb eines anderen Threads auftreten können. Wie bereits beschrieben, ist dies bei AsyncTasks nur über Umwege möglich. RxAndroid hilft hier dem Entwickler, da bei einer Exception, die innerhalb des Observables auftritt, die `onError`-Methode des registrierten Observers aufgerufen wird. Der Fehler lässt sich an dieser Stelle ganz einfach behandeln.

Auch das zweite Problem der AsyncTasks, die nicht über den Lifecycle einer Activity Bescheid wissen, lässt sich mit RxAndroid lösen. Zwar kann auch RxAndroid nicht automatisch mit dem Pausieren und der Wiederaufnahme einer Activity umgehen. Es bietet aber eine einfache Möglichkeit, dies zu erreichen. RxAndroid beziehungsweise RxJava bietet einen einfachen Caching-Mechanismus, mit dem man sich bei einem bestehenden Observable subscriben und wieder unsubscribe kann. Damit hat man die Möglichkeit, bei der Wiederaufnahme einer Activity die Arbeit eines Observables ebenfalls wieder aufzunehmen. Das Beispiel verwendet wieder den Retrofit-REST-Service, der aber nun bei der Abfrage des Servers ein Observable zurückliefert:

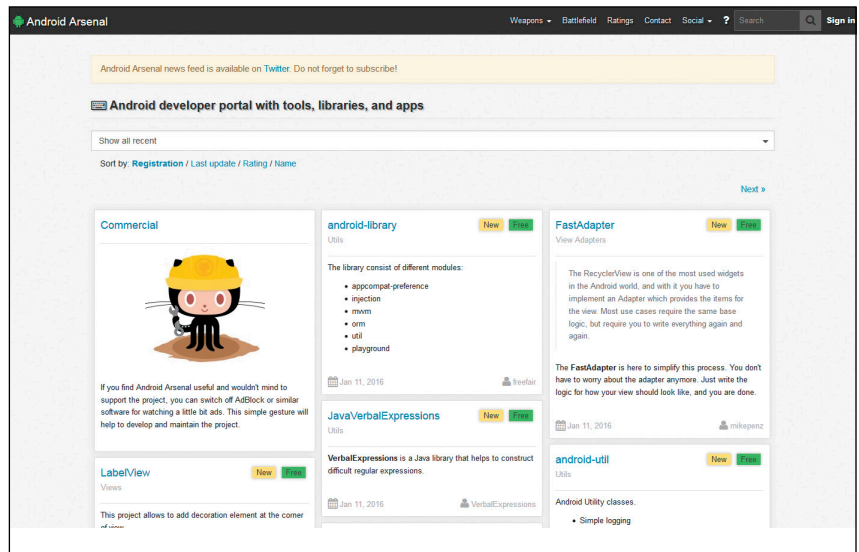
```
...
CommentService service = retrofit.create
(CommentService.class);
observer = new CommentsObserver();
commentObs = service.listComments()
    .cache()
    .subscribeOn(Schedulers.io())
    .observeOn(AndroidSchedulers.mainThread());
sub = commentObs.subscribe(observer);
...
```

Damit hat man nun die Vorbereitungen für die Behandlung des Activity-Lifecycles getroffen. Beim Pausieren und erneuten Starten der Activity kann man sich nun vom Observable unsubscribe beziehungsweise wieder subscriben:

```
@Override
protected void onPause() {
    sub.unsubscribe();
    super.onPause();
}
@Override
```

Links zum Thema

- GitHub-Repository mit Beispielprojekten
<https://github.com/tobiasamon/androidlibrariesexamples.git>



Android Arsenal bietet einen umfangreichen Fundus an Tools und Apps (Bild 9)

```
protected void onResume() {
    super.onResume();
    sub = commentObs.subscribe(observer);
}
```

Mit den beschriebenen Open-Source-Libraries ist eine Sammlung von Hilfsmitteln verfügbar, mit denen sich bei der Entwicklung von Android-Apps einige Probleme auf elegante Art und Weise lösen lassen. Wiederkehrende Aufgaben lassen sich einfach durch das Einbinden des richtigen Frameworks abbilden und müssen nur noch konfiguriert werden.

Fazit

Der Artikel hat gezeigt, wie sich einerseits mit den Android-ViewAnimations oder der Support-Library die Bedienoberfläche (auch für ältere Systeme) ansprechend gestalten lässt und andererseits mit Hilfe von Retrofit und RxAndroid eine flüssig zu bedienende Bedienoberfläche entsteht, die auch dann noch benutzbar ist, wenn keine Internetverbindung zur Verfügung steht.

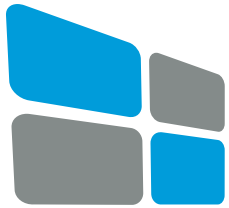
Neben den gezeigten Libraries gibt es noch eine ganze Reihe weiterer nützlicher Helfer, die mehr oder weniger spezielle Funktionen abbilden. Eine gute Quelle, um weitere Libraries zu finden, ist Android Arsenal (<https://android-arsenal.com>) (Bild 9). Dort sind sowohl freie als auch kommerzielle Bibliotheken gelistet und kategorisiert. ■



Tobias Amon

ist Senior Software Engineer bei der Adesso AG. Er verfügt über langjährige Erfahrung als Entwickler von mobilen Anwendungen für die Plattformen Android und iOS.

www.adesso.de



Internet World

Die E-Commerce Messe

01.-02. März 2016, München



Das Event des E-Commerce 2016

- 360 Aussteller
- 150 Vorträge
- 16.000 Besucher

internetworld-messe.de

**Jetzt
registrieren!**

**Besuch der
Messe
ist kostenlos.**

Wir sind
umgezogen:
**Halle
B5 + B6**
Eingang Ost

Partner Event
CMCX – Content-Marketing Conference & Exposition



ANDROID SUPPORT LIBRARY FÜR ANDROID 4.X

Material Design mit Ice Cream Sandwich

Die Android Design Support Library ermöglicht es, das Material Design auch auf Android 4.x lauffähig zu machen.

Mit der Veröffentlichung von Lollipop, Android 5, hat Google auch eine neue Designsprache eingeführt.

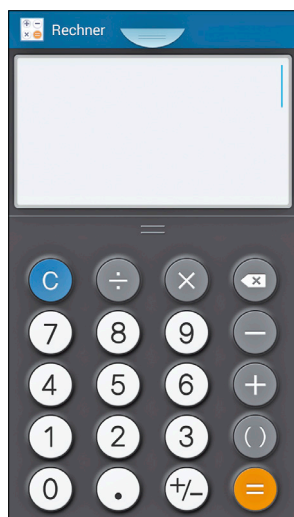
Material Design nennt sich dies, man erkennt es am runden schwebenden Call-to-Action-Button unten rechts, dem Burger-Menü oben links beziehungsweise dem Kebab Menü rechts, sowie an starken, kontrastreichen Farben und einer schlichten, klaren Schrift.

Die neuen Apps in Material Design sollen dem Nutzer mehr Feedback über seine Interaktion geben. Wird ein Element berührt, gibt es einen bestimmten Effekt dafür: Wird zum Beispiel ein Button angeklickt, so wechselt er die Farbe. Menüs werden jetzt optisch ausgefahren und vieles mehr.

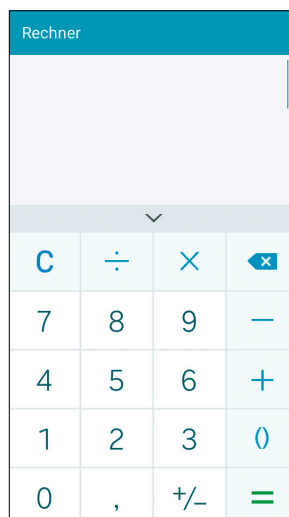
Material Design erreicht diese Optik, indem es Ebenen einführt. Buttons liegen eine Ebene höher als Inhalte, Menüs wiederum eine Ebene über den Buttons. Des Weiteren werden die Ebenen jeweils mit einem Schatten hervorgehoben, sodass der Anwender jederzeit erkennen kann, welche Elemente nun spezifisch zum gerade geöffneten Inhalt gehören.

Am besten lässt man das Design selbst sprechen. **Bild 1** zeigt den Google-Taschenrechner mit den typischen Design-Merkmalen von Android 4.0. Den Taschenrechner unter Material Design zeigt **Bild 2**. Hier fällt die neue Designsprache sofort ins Auge. Die Interaktion wird hier auf sehr dezente Weise umgesetzt.

Wenn Sie eine neue App mit der von Google favorisierten Entwicklungsumgebung Android Studio, nun aktuell in der Version 1.5, entwickeln, so erhalten Sie auch unter dem SDK Android 4.0 schon die entsprechende Unterstützung durch Einbindung der Support Library für Material Design (**Bild 3**). Als Entwickler stellt sich die Frage, ob es sinnvoll und vom Aufwand vertretbar ist, die Designphilosophie auch bei schon erstellten älteren Apps unter Android 4.x einzusetzen und diese anzupassen.



Android im typischen 4.0-Design (**Bild 1**)



Google Material Design im Einsatz (**Bild 2**)

Die Antwort lautet eindeutig ja. Die Vorzüge von Material Design liegen in klaren Formen und Interfaces, die dem Anwender helfen, sich zurechtzufinden. Auch der Wiedererkennungswert der typischen Designelemente ist hier hilfreich.

Um den Entwickler den Umstieg auf aktuelle Betriebssystemversionen zu erleichtern, pflegt Google seit Langem mehrere Support-Bibliotheken. Hierdurch sind Teile der neu hinzugekommenen Funktionen durch Backports auf früheren Plattformen verfügbar.

Support Library

Die Android Support Library Packages ermöglichen die Verwendung von später eingeführten Features, wie das Material Design oder auch Fragments und DrawerLayout, auf früheren Versionen des Betriebssystems.

Somit lassen sich diese Neuheiten dann auch auf älteren Android-Geräten nutzen.

Das heißt, die Support Library stellt eine Reihe von Codebibliotheken bereit, die abwärtskompatibel zu den Android Framework APIs sind. Beachten Sie hierbei, dass jede Support-Bibliothek nur bis zu einer bestimmten Android-API-Ebene abwärtskompatibel ist. Dies bedeutet, dass die zurzeit aktuelle Support Library V.23 nur noch mit Android-1.6-Geräten, das entspricht 4.x und höher, unterstützt wird.

Mehr über die Support Library APIs finden Sie unter *android.support-Packages* in der API-Referenz von Android.

Download

Die benötigte beziehungsweise aktuelle Support-Bibliothek lässt sich mit dem Android SDK Manager jederzeit nachinstallieren.

Starten Sie hierfür den Android SDK Manager (**Bild 4**), entweder über das Programmverzeichnis im Ordner *Android SDK Tools* oder über das

Menü *Tools, Android, SDK Manager* in Android Studio.

Scrollen Sie im SDK Manager bis ganz nach unten zum Ordner *Extras* und wählen Sie hier die *Android Support Library* aus. Wenn Sie lieber mit einem Maven Repository unter Android Studio arbeiten, installieren Sie stattdessen das *Android Support Repository*.

Der anschließende Download über *Install packages...* installiert die benötigte Library in Ihr Android-SDK-Verzeichnis.

Material Design

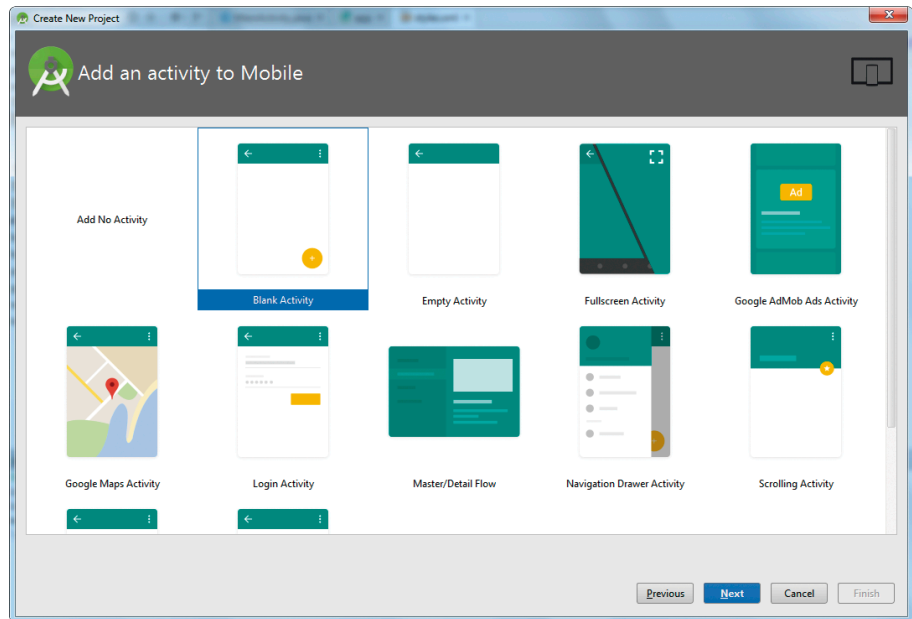
Mit der aktuell zur Verfügung stehenden Support Library können Sie somit auch das Material Design für ältere Android-Versionen verwenden. Das Material Design wird ab Support Library Version 21 unterstützt.

Die hierbei wichtigste Anpassung ist die Framework-Implementierung der bereitgestellten *AppCompat* (Basisklasse *ActionBarCompat*). Diese bietet das API und das Feature-Set von Android 5 für die entsprechende Rückportierung auf Android-4.x-Basis und löst somit die alte *ActionBar* ab.

Die Support Library führt somit ein neues Toolbar-Widget ein, das den allgemeinen Muster der Aktionsliste im Material Design entspricht. Es bietet den User somit mehr Kontrolle und Flexibilität in der Anwendung.

Bild 5 zeigt das aktuelle Aussehen der Toolbar im Material-Design-Stil mit dem entsprechenden Farbdesign und den benötigten Attributen für das Theming.

Des Weiteren wird durch die Einbindung des *AppCompat* API auch folgende Teilmenge der UI-Widgets automatisch auf das Material Design angepasst: *EditText*, *Spinner*, *CheckBox*, *RadioButton*, *Switch* (über *android.support.v7.widget.SwitchCompat*) und *CheckedTextView*.



Neues SDK jetzt mit Material-Design-Unterstützung (Bild 3)

Die Verwendung dieser Steuerelemente kann somit ohne Änderung und wie gewohnt im Layout eingesetzt werden. Die Anpassung für das richtige Verhalten unter der neuen Designsprache funktioniert ganz selbstverständlich.

Weiterhin finden Sie in der Support-Bibliothek weitere wichtige Material-Design-Komponenten für die bessere Interaktion des Anwenders mit Ihrer App. Hier zu zählen *NavigationViews*, *FloatingLabels*, *Snackbar*, *Tabs*, *CoordinatorLayout* und *CollapsingToolbars*.

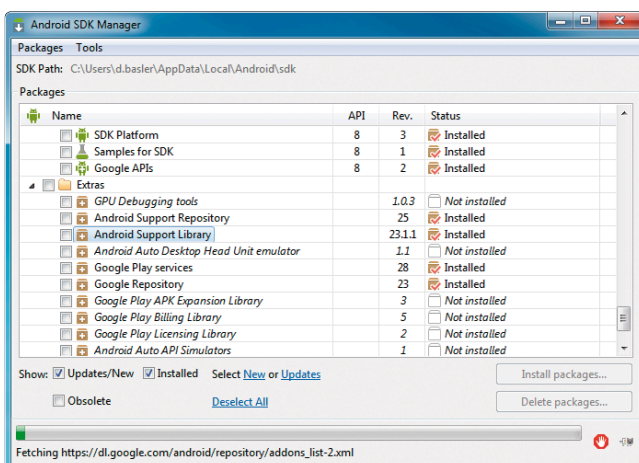
Die *NavigationView* kann als *DrawerLayout* verwendet werden. Sie bietet eine Sidebar als Navigationsmöglichkeit, durch die eine schnelle und effektive Navigation innerhalb der App möglich ist. Die Navigation ist bereits durch einen Wischer von der linken zur rechten Seite oder durch einen Druck auf den Toolbar-Titel geöffnet und ist somit von überall in der App erreichbar.

Die *EditText*-Komponente ermöglicht jetzt durch das *TextInputLayout* eine schwimmende Beschriftung (*Floating Label*) oberhalb des *EditText*-Elements. Somit verliert der Benutzer nie den Kontext für die Text-Eingabe.

Der *FloatingActionButton* stellt die neue runde Aktions-schaltfläche in der App dar. Dieser kann individuell dargestellt werden: durch Farbe, Mini-Größe und Images. Der Button ermöglicht eine einheitliche Bedienung der App.

Die *Snackbar* stellt unter Material Design das neue kleine Infofähnchen bereit und löst somit das *Toast*-Element ab. Weiterhin ermöglicht die *Snackbar* eine Interaktion durch Wischen. Auch die *Tabs* unterliegen im Material Design jetzt einer neuen Konzeption. Über das *TabLayout* implementieren Sie feste Tabs, bei denen die Ansicht gleichmäßig zwischen den Registerkarten aufgeteilt ist.

Das *CoordinatorLayout* stellt neue Möglichkeiten für Touch-Ereignisse in untergeordneten Ansichten zur Verfügung. Somit können *ViewGroups* innerhalb einer Ansicht angepasst und über Scroll-Techniken bedient werden. ►



Android SDK Manager: Zwei Varianten für den Start (Bild 4)

Über `CollapsingToolbar` können unter Material Design jetzt auch verschiedene Elemente einfach über Scrollen reduziert werden. Durch die neuen Widgets unter Material Design lassen sich Apps jetzt vom Anwender noch intuitiver bedienen. Material Design stellt durch die neue Designsprache eine viel bessere Unterstützung für den User zur Verfügung.

AppCompat

Um Ihre ältere App jetzt für das Material Design vorzubereiten, hilft Ihnen insbesondere das AppCompat Framework bei der Umsetzung.

Die Bibliothek stellt die benötigten Bedienelemente, hier besonders die neue Toolbar, für das Material Design zur Verfügung. Allerdings müssen Sie, um AppCompat zu nutzen, diese in jedes Projekt integrieren. Beachten Sie hierbei, dass

die neue Toolbar die aus älteren Android-Versionen bekannte `ActionBar` ersetzt.

Die `ActionBar` wurde standardmäßig in allen Apps verwendet, deren Manifestdatei für mindestens eines der beiden Attribute `android:minSdkVersion` und `android:targetSdkVersion` den Wert 11 oder höher enthält. Somit müssen die Activity-Klassen Ihrer App von der neuen Basisklasse `AppCompatActivity` abgeleitet werden.

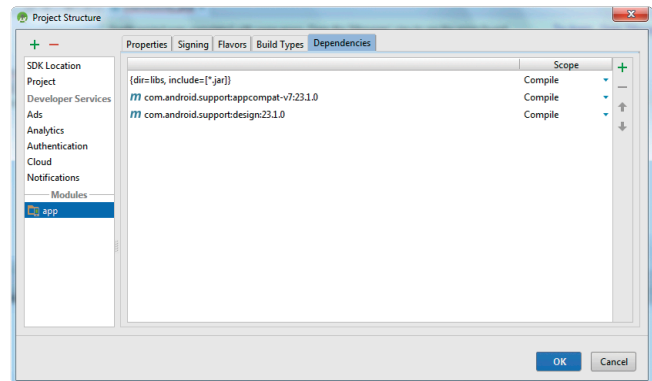
Um also die benötigte Support-Library in Ihr Projekt zu implementieren, gehen Sie am besten wie folgt vor: Starten Sie Android Studio und laden Sie das gewünschte Projekt Ihrer App, welches für das neue

Design angepasst werden soll.

Jetzt müssen Sie Ihrem Projekt die Abhängigkeit der Support-Library hinzufügen. Öffnen Sie hierfür die Projektstruktur Ihrer App und wählen Sie wie im **Bild 6** aufgezzeigt das Register *Dependencies*. Fügen Sie jetzt über *Add(+)* und *Library dependency* die benötigten Bibliotheken `appcompat-V7` und `design` hinzu. Schließen Sie das Dialogfenster *Project Structure* mit *Ok*. Android Studio fügt die Abhängigkeit jetzt automatisch zum Projekt hinzu und erweitert das *Build-Gradle*-File um die entsprechende Erweiterung:

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:23.1.0'
    compile 'com.android.support:design:23.1.0'
}
```

Alternativ können Sie diesen Eintrag auch direkt im *Build-File* vornehmen. Des Weiteren werden die Einträge von `ActionBar` durch `AppBar` ersetzt. Um die `ActionBar` durch die



Support Library: Abhängigkeiten hinzufügen (Bild 6)

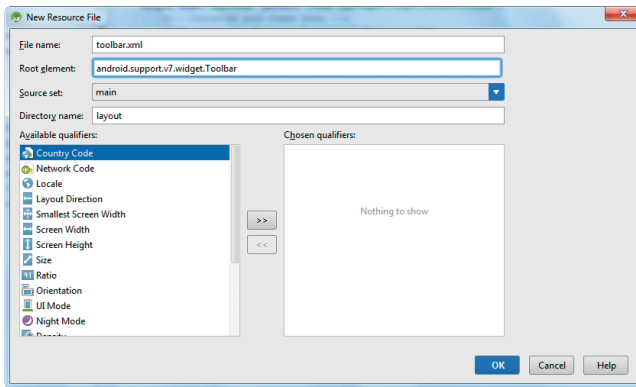
neue Toolbar zu ersetzen, müssen Sie in der *styles.xml*-Datei der Version 21 folgenden Eintrag vornehmen:

```
<resources>
    <style name="AppTheme.NoActionBar">
        <item name="windowActionBar">false</item>
        <item name="windowNoTitle">true</item>
        <item name="android:windowDrawsSystemBarBackgrounds">
            true</item>
        <item name="android:statusBarColor">@android:color/
            transparent</item>
    </style>
</resources>
```

Somit wird die `ActionBar` in Ihrem Projekt komplett deaktiviert. Sie finden die Datei in der Solution unter *res, values* im Ordner *styles.xml*. Nachdem diese Vorbereitungen getroffen

Listing 1: Anpassung des Farbschemas

```
<resources>
    <style name="AppTheme" parent=
        "Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary
        </item>
        <item name="colorPrimaryDark">@color/
            colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent
        </item>
    </style>
    <style name="AppTheme.NoActionBar">
        <item name="windowActionBar">false</item>
        <item name="windowNoTitle">true</item>
    </style>
    <style name="AppTheme.AppBarOverlay" parent=
        "ThemeOverlay.AppCompat.Dark.ActionBar" />
    <style name="AppTheme.PopupOverlay" parent=
        "ThemeOverlay.AppCompat.Light" />
</resources>
```

Eigenes Projekt: Neue Layout-Datei erstellen (Bild 7)

worden sind, können Sie das Farbschema für Ihr Projekt festlegen beziehungsweise anpassen. *ColorPrimaryDark* bestimmt die Grundfarbe für die Statusleiste. *ColorPrimary* stellt die Farbe der App dar und *TextColorPrimary* die Farbe des Textes. Dies gilt für den Titel in der Symbolleiste der App.

Dafür passen Sie die Datei *color.xml* unter *res, values* nach Ihren Wünschen an. Listing 1 zeigt das von Google vorgegebene Farbschema. Nach dem Festlegen der gewünschten Stilrichtung (Farbschema/Thema) können Sie dieses über die *AndroidManifest.xml* Datei Ihrer App zuweisen. Ändern Sie dazu einfach das *android:theme*-Attribut in der XML-Datei:

```
android:theme="@style/IhrMaterialTheme"
```

Wenn Sie Ihre App jetzt ausführen, wird das entsprechende neue Farbschema zur Laufzeit benutzt.

Toolbar implementieren

Nun kann die neue Toolbar in die App implementiert werden. Hierbei stellt die Toolbar ein Layout dar, das in der App-Bedienoberfläche platziert werden kann. Benutzen Sie nur eine Activity beziehungsweise eine Layout-Datei, so kann die Toolbar direkt implementiert werden. Andernfalls sollten Sie die Toolbar in einer eigenen Layout-Datei für Ihr Projekt erstellen.

Öffnen Sie hierfür unter *res, layout* das Dialogfenster *New Resource File* (Bild 7). Vergeben Sie einen Dateinamen und wählen Sie als *Root element* den Eintrag *android.support.v7.widget.Toolbar*.

Legen Sie jetzt die Hintergrundfarbe für Ihr Toolbar-Layout als Grundfarbe der App fest. Hier das Beispiel:

```
version="1.0" = encoding="utf-8"?>
<android.support.v7.widget.Toolbar xmlns:android=
"http://schemas.android.com/apk/res/android"
    android:layout_width = "match_parent"
    android:layout_height = "wrap_content"
    android:background = "@color/ColorPrimary"
    android:elevation = "4dp">
</android.support.v7.widget.Toolbar>
```

Das Attribut *android:elevation="4dp"* sorgt in diesem Fall für ein Schatten-Effekt der Toolbar.

Jetzt implementieren Sie das erstellte Toolbar-Layout über das *include*-Tag in Ihre gewünschte Layout-Datei der benötigten Activity. So zum Beispiel in die Layout-Datei der *main_activity.xml* Ihrer App:

```
<include
    android:id = "@+id/toolbar"
    layout = "@layout/toolbar">
</include>
```

Jetzt deklarieren Sie die Toolbar über

```
private Toolbar toolbar;
```

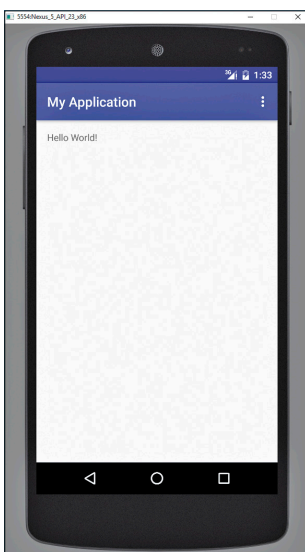
in der gewünschten Activity. Leiten Sie die Basisklasse der Activity auch von *AppCompatActivity* ab. In der *OnCreate*-Methode der Activity fügen Sie die Toolbar hinzu und rufen diese über *setSupportActionBar* auf:

```
toolbar = (Toolbar)findViewById(R.id.toolbar);
setSupportActionBar(toolbar);
```

Starten Sie jetzt Ihre App, so erscheint diese im neuen Material Design mit dem entsprechenden Farbschema. Bild 8 zeigt das Aussehen der neuen Toolbar in einer Beispiel-App.

Fazit

Wie dieser Workshop zeigt, ist es nicht so schwierig, die neue Designsprache von Google für ältere Apps umzusetzen. Es sollte daher möglich sein, die neuen Gestaltungsmöglichkeiten mit dem Material Design umzusetzen. Prüfen Sie daher auch, wie sinnvoll es sein kann, die neuen Widgets wie *NavigationView* oder auch den *FloatingActionButton* in Ihrer App zu verwenden. Die Anwender werden es Ihnen danken und Ihre App erfährt eine höhere Akzeptanz. ■



Material Design in der eigenen App (Bild 8)

```
<span style = "font-size:
x-small;" >? xml
```



Daniel Basler

ist Senior Consultant für Microsoft-Technologien und Autor zahlreicher Fachartikel. Darüber hinaus beschäftigt er sich mit Datenbanken und Compiler-Bau.

CHART-LIBRARIES FÜR ANDROID

Androides Diagramm

Wer Informationen in ein Diagramm packt, erleichtert den Nutzern das Leben.

Diensterfahrene Entwickler betrachten fertige Chart-Libraries als dekadentes Artefakt für mathematisch unbegabte Programmierer. Es mag durchaus stimmen, dass die hinter Diagrammen stehende Planargeometrie trivial ist. In Zeiten superschneller Produktzyklen ist es trotz allem sinnvoll, das Erzeugen von nichtspezifischen Diagrammen an eine Bibliothek auszulagern.

Alles Open Source

Philipp Jahoda pflegt mit MPAndroidChart ein Standardprodukt im Bereich der Diagrammbibliotheken. Das ab Android 2.2 lauffähige Produkt steht unter der liberalen Apache-2.0-Lizenz und kann somit auch in kommerziellen Programmen ohne Einschränkungen verwendet werden.

Der Entwickler bittet in der offiziellen Dokumentation lediglich um einen Link auf die im Play Store veröffentlichte Applikation. Sie sollten diesem Wunsch aus höchst eigennützigen Gründen nachkommen. Als Dank winkt ein Link auf der offiziellen Produktwebseite <https://github.com/PhilJay/MPAndroidChart/wiki/References>.

Da Google das Eclipse-ADK über kurz oder lang einstellt, basieren die in diesem Artikel erstellten Beispielprogramme auf einer unter Ubuntu gehosteten Version von Android Studio 1.3.2. Wer MPAndroidChart unter Eclipse verwenden möchte, kann ein fertiges JDK-File unter <https://github.com/PhilJay/MPAndroidChart/releases> herunterladen.

Erstellen Sie im nächsten Schritt ein normales Android-Applikationsprojekt namens *NMGACHart1*. MPAndroidChart ist bei der Auswahl der Zielpattform flexibel.

Dank des Gradle-Buildsystems lässt sich die Bibliothek mit minimalem Aufwand einbinden. Öffnen Sie Datei *build.gradle* des Gesamtprojekts und ergänzen Sie sie um die Deklaration des Maven-Repositorys:

```
allprojects {
    repositories {
        jcenter()
        maven { url "https://jitpack.io" }
    }
}
```

Im nächsten Schritt muss das *jar*-File in den Kompilationsvorgang eingebunden werden. Dies lässt sich in der *build.gradle*



Die Default-Parameter
neuer Steuerelemente behindern die automatische Ausbreitung des Widgets (Bild 1)

le-Datei des App-Moduls bewerkstelligen, in dem folgende Änderung erforderlich ist:

```
dependencies {
    compile fileTree(dir: 'libs',
        include: ['*.jar'])
    compile
        'com.android.support:appcompat-v7:23.0.1'
    compile
        'com.github.PhilJay:MPAndroidChart:v2.1.6'
}
```

Klicken Sie danach auf den Button *Sync Now*, um die IDE zur Regenerierung der Builddateien zu animieren. Bei bestehender Internetverbindung lädt Gradle die benötigten Informationen automatisch herunter; der Buildvorgang wird problemlos durchlaufen.

MPAndroidChart-Diagramme verhalten sich aus Sicht des Betriebssystems wie normale Widgets. Als Beispiel dafür dient folgendes Snippet,

das ein Liniendiagramm erstellt:

```
<com.github.mikephil.charting.charts.LineChart
    android:id="@+id/chart"
```

Listing 1: RelativeLayout

```
<RelativeLayout xmlns:android=
"http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    ... tools:context=".MainActivity">
    <TextView android:text="Hier folgt ein Diagramm"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textView" />
    <RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_below="@+id/textView"
        android:layout_alignParentStart="true"
        android:layout_marginTop="40dp"
        android:id="@+id/chartContainer" >
    </RelativeLayout>
</RelativeLayout>
```

```
android:layout_width="match_parent"
android:layout_height="match_parent" />
```

Diese Vorgehensweise erweist sich in der Praxis insofern als undankbar, als MPAndroidChart keine generische Zeichenfläche verwendet. Das von unserem Snippet erstellte Liniendiagramm bleibt also auf ewig ein Liniendiagramm: Der Diagrammtyp lässt sich beispielsweise nicht in Torten- oder Balkendiagramme ändern.

Zur Umgehung dieses Problems empfehlen die Entwickler die Verwendung eines Layout-Containers, der vom Hostprogramm während der Programm-Ausführung mit einem dynamisch erstellten Widget bevölkert wird. Öffnen Sie dazu die Datei *main_activity.xml* und schreiben Sie das im Snippet gezeigte *RelativeLayout* hinein (Listing 1).

ChartContainer agiert hier – nomen est omen – als Stellvertreter-Steuerelement, das den hineingeschriebenen Diagrammen den Weg in die Layouthierarchie ebnet.

Da die XML-Datei aus Gründen der Flexibilität keine Chart-Instanz generiert, erfolgt die eigentliche Erstellung der Chartview von Hand. Dazu ist in *onCreate()* die Änderung in Listing 2 notwendig.

Mit Androids Layoutsystem unerfahrene Entwickler stolpern an dieser Stelle mitunter über eine kleine Falle. Ein neu erstelltes Steuerelement hat von Haus aus eine minimale Layoutpriorität und wächst somit nicht. Der Aufruf von *setLayoutParams* sorgt dafür, dass sich das Diagramm maximal ausbreitet. Wer das Programm in der vorliegenden Form ausführt, wird mit dem in Bild 1 gezeigten Widget belohnt. Der gelbe Text ist eine Fehlermeldung, die auf das Nicht-Vorhandensein von Diagrammdaten hinweist (Bild 2).

Keine Dummy-Daten

MPAndroidChart versteht – im Unterschied zu anderen Bibliotheken – leere Diagramme nicht von selbst mit Dummy-Daten. Wer also unserem Beispielprogramm nützliche Informationen entlocken möchte, muss die Chartklasse mit einem Datenspeicher verbinden.

Dabei kommt ein dreistufiges Datenmodell zum Einsatz. Auf der oberen Ebene findet sich eine von *ChartData* abgeleitete Klasse. Eine Reflexion der *ChartData*-Basisklasse liefert folgende Struktur:

```
public abstract class
ChartData<T extends DataSet<?
extends Entry>> {
    protected float mYMax = 0.0f;
    protected List<String> mXVals;
    protected List<T> mDataSets;
    public void addDataSet(T d) {
        if (d == null)
            return;
    }
}
```

Fehlermeldung: Hier fehlen Chartinformationen (Bild 2)

ChartData-Instanzen fungieren als Speicherklassen, die die für die

Listing 2: onCreate()

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    LineChart chart = new
    LineChart(getApplicationContext());
    RelativeLayout rl = (RelativeLayout)
    findViewById(R.id.chartContainer);
    ViewGroup.LayoutParams myParams=new
    ViewGroup.LayoutParams(ViewGroup.LayoutParams.
    MATCH_PARENT,
    ViewGroup.LayoutParams.MATCH_PARENT);
    chart.setLayoutParams(myParams);
    rl.addView(chart);
}
```

Erzeugung eines Diagramms notwendigen Rohinformationen vorhalten. Zudem enthält sie einige Methoden, die das Einfügen und Entfernen von Informationen abbilden.

Die einzelnen Zeilen beziehungsweise Datengruppen basieren auf von *DataSet* abgeleiteten Klassen. Die Reflexion der Klasse erlaubt weitere Einblicke in die Struktur von MPAndroidChart:

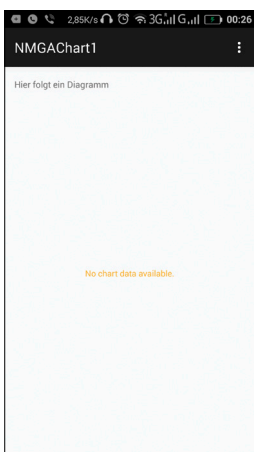
```
public abstract class DataSet<T extends Entry> {
    protected List<Integer> mColors = null;
    protected List<T> mYVals = null;
    protected float mYMax = 0.0f;
    ...
    public void setVisible(boolean visible) {
        mVisible = visible;
    }
}
```

DataSet exponiert einige Methoden, die das Beeinflussen der Darstellung der Datenreihen erlauben. Wir werden sie in den folgenden Abschnitten näher betrachten. Fürs Erste reicht es aus, wenn Sie sich die Verortung der Modifikationsfunktionen in den beiden Klassen veranschaulichen.

Entries stehen auf der untersten Reihe der Klassenhierarchie. Sie repräsentieren die einzelnen Punkte des Diagramms; nach Reflexion entsteht folgende Struktur:

```
public class Entry implements Parcelable {
    private float mVal = 0f;
    private int mXIndex = 0;
    private Object mData = null;
}
```

Neben dem eigentlich anzuzeigenden Wert – er liegt in *mVal* – nimmt *Entry* auch einen Index entgegen, der das Datenfeld auf der x-Achse verortet. Das von Haus aus genullte Feld *mData* kann Informationen über die zum *Entry* gehörende Engine-Klasse entgegennehmen, um Korrelation und Aktualisierung zu erleichtern. ►



Nach diesen einführenden Überlegungen wollen wir das bisher Gelernte zur Belegung des Liniendiagramms nutzen. Dazu sind im Konstruktor einige Anpassungen notwendig, die folgendermaßen aussehen:

```
@Override
protected void onCreate(Bundle
savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    ArrayList<Entry> line1 = new
    ArrayList<Entry>();
    line1.add(new Entry(20f, 0));
    line1.add(new Entry(50f, 1));
    line1.add(new Entry(10f, 2));
    line1.add(new Entry(80f, 3));
    line1.add(new Entry(20f, 4));
    line1.add(new Entry(3f, 5));
```

Als ersten Akt erstellen wir eine *ArrayList*, die die einzelnen Datenpunkte der Liste aufnimmt. Der Konstruktor von *Entry* nimmt den eigentlichen Zahlenwert und einen Arrayindex entgegen.

Dieser ist insofern von Bedeutung, als er zur Laufzeit mit dem x-Wert-Array korreliert wird. Es sollte unbedingt vor der Fertigstellung der *ChartData*-Klasse fertig sein. Nach der Erzeugung der *LineData*-Klasse lassen sich weitere Einträge nur sehr umständlich einpflegen (Listing 3).

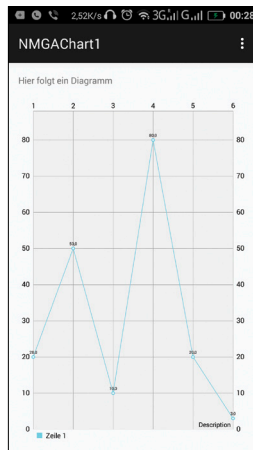
Als letzter Akt folgt die Initialisierung des Diagramms, die nun nach folgendem Schema abläuft:

```
LineData myData=new LineData(xVals,dataSets);
LineChart chart = new LineChart
(getApplicationContext());
chart.setData(myData);
RelativeLayout rl = (RelativeLayout)
findViewById(R.id.chartContainer);
ViewGroup.LayoutParams myParams=new
```

Listing 3: ArrayList

```
LineDataSet set1 = new LineDataSet(line1,
"Zeile 1");
ArrayList<String> xVals = new ArrayList<String>();
xVals.add("1");
xVals.add("2");
xVals.add("3");
xVals.add("4");
xVals.add("5");
xVals.add("6");

ArrayList<LineDataSet> dataSets =
new ArrayList<LineDataSet>();
dataSets.add(set1);
```



Die Linie erscheint im Diagramm (Bild 3)

```
ViewGroup.LayoutParams
(ViewGroup.LayoutParams.MATCH_PARENT,
ViewGroup.LayoutParams.MATCH_PARENT);
xChart.setLayoutParams(myParams);
rl.addView(chart);
}
```

Führen Sie das Programm danach in einer AVD oder auf einem echten Smartphone aus, um sich am gezeigten Ergebnis zu erfreuen (Bild 3).

Mehr Diagramme

Unser Beispiel bindet die Diagrammklassen dynamisch ein, um den flexiblen Wechsel zwischen verschiedenen Darstellungsformen anbieten zu können. Im Moment liegt diese Fähigkeit brach – Zeit, dies zu ändern. Zur Aktivierung der verschiedenen Diagrammtypen ist eine

Knopfreihe notwendig, die im Layout von *MainActivity* über ein *LinearLayout* entsteht:

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Line"
        android:id="@+id/CmdLine" />
    <Button ...
        android:text="Pie"
        android:id="@+id/CmdPie" />
</LinearLayout>
```

Die einzelnen Event Handler aktivieren Methoden, die den Inhalt des für die Chartdarstellung zuständigen *RelativeLayout*s mit dem gerade gewünschten Diagramm bevölkern. Aus Platzgründen drucken wir hier nur den Handler für das Liniendiagramm ab – die restlichen Buttons verhalten sich analog (Listing 4).

Der erste Schritt zur Multichart-Applikation besteht darin, die Generierung der Charts in Methoden auszulagern. Aus Gründen der rechentechnischen Effizienz speichern wir die Widgets als Membervariablen der Activity. Durch diese Referenz gehen Sie beim Entfernen aus dem *RelativeLayout* nicht verloren und können ohne aufwendige Neuinstanzierung abermals eingeblendet werden:

```
public class MainActivity extends AppCompatActivity
implements View.OnClickListener {
    Button CmdLine, CmdPie;
    LineChart lineChart;
    PieChart pieChart;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```



```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
lineChart=generateLineChart()
```

Als nächste Aufgabe wenden wir uns der Implementierung eines Tortendiagramms zu. Diese unter Visualisierungsexperten nicht sonderlich beliebte Darstellungsform setzt – natürlich – eine eigene Klassengruppe voraus, deren Aufbau folgendermaßen beginnt:

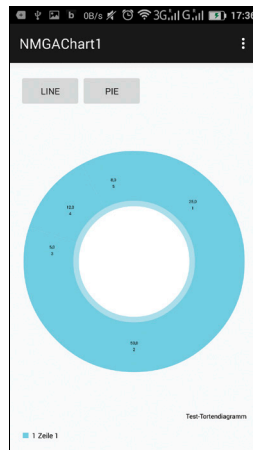
```
@SuppressWarnings("deprecation")
PieChart generatePieChart() {
    ArrayList<Entry> pie1 = new
    ArrayList<Entry>();
    pie1.add(new Entry(25f, 0));
    pie1.add(new Entry(50f, 1));
    pie1.add(new Entry(5f, 2));
    pie1.add(new Entry(12f, 3));
    pie1.add(new Entry(8f, 4));
```

Die Erstellung des Entry-Arrays erfolgt im Großen und Ganzen nach den vom Liniendiagramm bekannten Kriterien. Im Idealfall werden die angelieferten Daten direkt als Anteile von 100 dargestellt, um der Engine die Berechnung einer Summe zu ersparen:

```
PieDataSet set1 = new PieDataSet(pie1, "Zeile 1");
ArrayList<String> xVals = new ArrayList<String>();
xVals.add("1");
xVals.add("2");
xVals.add("3");
xVals.add("4");
xVals.add("5");
xVals.add("6");
PieData d1=new PieData(xVals, set1);
```

Auch bei Tortendiagrammen ist ein *xValues*-Array zwingend erforderlich: Fehlt es, so stürzt die Bibliothek im Rahmen der Initialisierung mit einem für Android Studio nicht erfassbaren Fehler ab.

Zu guter Letzt wird eine Instanz der *PieChart*-Klasse erstellt. Das Aufrufen von *setUsePercentValues* teilt *MPAndroid*



Das Tortendiagramm ist einsatzbereit **Bild 4**

idChart mit, dass die angelieferten Werte schon normalisiert sind. Mittels *setDescription* wird eine kleine Titelzeile eingeblendet:

```
PieChart xChart=new
PieChart(getApplicationContext());
ViewGroup.LayoutParams myParams=new
ViewGroup.LayoutParams
(ViewGroup.LayoutParams.MATCH_PARENT,
ViewGroup.LayoutParams.MATCH_PARENT);
xChart.setLayoutParams(myParams);
xChart.setData(d1);
xChart.setUsePercentValues(true);
xChart.setDescription
("Test-Tortendiagramm");
return xChart;
}
```

Damit sind wir an dieser Stelle fertig. Schicken Sie das Programm auf ein Smartphone Ihrer Wahl und erfreuen Sie sich am in **Bild 4** gezeigten Tortendiagramm. Das Loch in der Mitte lässt sich durch eine Hilfsfunktion der *PieChart*-Klasse eliminieren.

Dynamische Formatierung

Tabellenkalkulationen wie LibreOffice Calc oder Microsoft Excel bieten die Möglichkeit an, Diagramme und Tabellen durch wertbezogene Farbuweisungen übersichtlicher zu gestalten. Dieses Verhalten lässt sich in *MPAndroidChart* mit minimalem Aufwand nachrüsten, da der Entwickler auf das von Symbian und Co. bekannte Mixin-Konzept setzt.

Im Moment ist das diesbezügliche API noch nicht komplett implementiert, weshalb etwas Eigenarbeit erforderlich ist. In zukünftigen Versionen dürfte das Überschreiben von *getColor* nicht mehr notwendig sein, da das Mixin dann direkt in das *DataSet* wandert.

Farbinformationen werden über eine oder mehrere Klassen angeliefert, die das *ColorFormatter*-Interface implementieren. Das Framework versorgt sie während der Diagrammdarstellung mit einem Verweis auf das jeweilige Entityobjekt, welches im nächsten Schritt zur Ermittlung der jeweiligen Farbe herangezogen wird:

```
public interface ColorFormatter {
    int getColor(Entry e, int index);
}
```

Wir wollen mit der Erstellung eines normalen *ColorFormatters* beginnen. Seine Struktur präsentiert sich nach folgendem Schema:

```
public class NMGColorFormatter implements ColorFormatter
{
    @Override
    public int getColor(Entry e, int index)
    {
        if(e.getVal()>50)
```

Listing 4: Handler für das Liniendiagramm

```
public void onClick(View v) {
    if(v==CmdLine)
    {
        RelativeLayout r1 = (RelativeLayout)
        findViewById(R.id.chartContainer);
        r1.removeAllViewsInLayout();
        r1.addView(lineChart);
    }
    ...
}
```

```

        return Color.RED;
    if(e.getVal()>25)
        return Color.YELLOW;
    return Color.GRAY;
}
}

```

Farben werden in Java in Form von Integers dargestellt. Unser Formatter liefert je nach angeliefertem Entry einen bestimmten Farbwert zurück. Die von Google im *Colors*-Objekt vorangelegten Einträge ersparen uns das Hantieren mit RGB-Werten.

Wer die zu einer Farbe gehörenden RGB-Codes ermitteln möchte, findet unter https://en.wikipedia.org/wiki/Web_colors eine Tabelle der rund 200 als Webfarben bezeichneten Standardwerte.

Für die Einbindung des Moduls ist etwas Überlegung erforderlich. MPAndroidChart lagert die Zeichenintelligenz in eine eigene Unterklasse aus: PieCharts werden beispielsweise durch Instanzen von *PieChartRenderer* auf den Bildschirm gebracht.

Durch Reflexion lässt sich ermitteln, dass die Beschaffung der zum Rendern notwendigen Farbwerte durch Aufruf der in *dataSet* implementierten Methode *getColor* erfolgt. Diese wird mit einem nullbasierten Index versorgt, der das zu verwendende Entry beschreibt:

```

if ((Math.abs(e.getVal()) > 0.000001)) {
    if (!mChart.needsHighlight(e.getXIndex(),
        mChart.getData().getIndexOfDataSet(dataSet))) {
        mRenderPaint.setColor(dataSet.getColor(j));
    }
}

```

Als nächster Akt muss eine neue Variante von *PieDataSet* entstehen. Dank den Möglichkeiten der objektorientierten Programmierung ist an dieser Stelle keine komplette Neuprogrammierung erforderlich. Es genügt, wenn die betroffene Methode nach folgendem Schema überschrieben wird:

```

public class NMGPieDataSet extends PieDataSet {
    public NMGColorFormatter myFormatter;
    ...
    @Override
    public int getColor(int index) {
        return myFormatter.getColor
            (getEntryForXIndex(index),0);
    }
}

```

Im Rahmen des Konstruktors erstellen wir eine neue Instanz des *NMGColorFormatters*, der aus Performancegründen in eine Membervariable wandert. Das Neu-Erstellen von Klasseninstanzen in engen Schleifen oder häufig aufgerufenem Zeichencode ist ein klassisches Antipattern, das mitunter zu massiven Verlangsamungen führt:

```

public class NMGPieDataSet extends PieDataSet {
    public NMGColorFormatter myFormatter;

```

```

public NMGPieDataSet(List<Entry> yVals, String label)
{
    super(yVals, label);
    myFormatter=new NMGColorFormatter();
}
...
}

```

MainActivity erstellt das Diagramm im Moment anhand einer normalen *PieDataSet*-Instanz. Dies lässt sich durch eine kleine Anpassung in *generatePieChart()* beheben:

```

PieChart generatePieChart() {
    ArrayList<Entry> pie1 = new ArrayList<Entry>();
    pie1.add(new Entry(26f, 0));
    pie1.add(new Entry(51f, 1));
    pie1.add(new Entry(4f, 2));
    pie1.add(new Entry(11f, 3));
    pie1.add(new Entry(8f, 4));
    NMGPieDataSet set1 = new
    NMGPieDataSet(pie1, "Zeile 1");
}

```

Ab diesem Zeitpunkt beschafft der Renderer seine Farben aus dem von uns angelegten *ColorFormatter*. Änderungen am Diagrammdesign lassen sich nun an zentralisierter Stelle vornehmen. **Bild 5** zeigt, wie unser Beispiel zum momentanen Zeitpunkt aussieht.

Alternative Farbenspiele

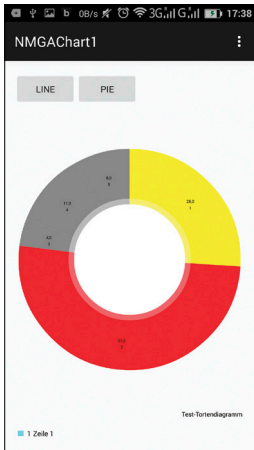
Bei kleinen Datenmengen ist es mitunter sinnvoller, die Farbwerte direkt einzuschreiben. Die meisten *DataSets* enthalten hierfür spezielle Methoden, die Farbarrays entgegennehmen. Zur Demonstration dieses Features soll unser Liniendiagramm erhalten, das mit etwas Farbe ausgestattet wird.

Listing 5: Deklaration der Liniendiagrammkasse

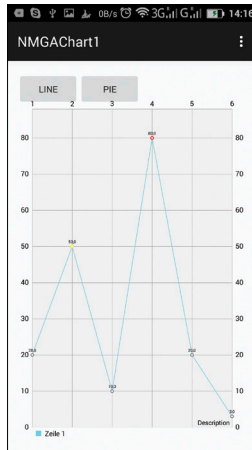
```

LineChart generateLineChart()
{
    ArrayList<Entry> line1 = new ArrayList<Entry>();
    line1.add(new Entry(20f, 0));
    ...
    LineDataSet set1 = new LineDataSet(line1,
    "Zeile 1");
    NMGColorFormatter myFormatter=new
    NMGColorFormatter();
    ArrayList<Integer> myColorList=
    new ArrayList<Integer>();
    for(int i=0;i<line1.size();i++)
    {
        myColorList.add(myFormatter.getColor
            (line1.get(i),0));
    }
    set1.setCircleColors(myColorList);
    ...
}

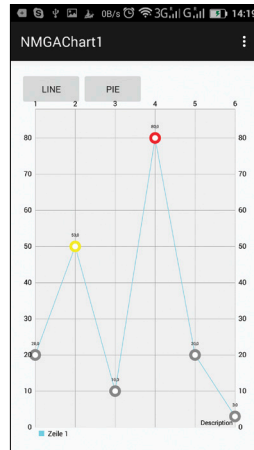
```



Tortendiagramm als mehrfarbige Variante (Bild 5)



Beim Liniendiagramm sind wir auf die Kringel-Farbe beschränkt (Bild 6)



Größere Kringel erhöhen die Sichtbarkeit des Diagramms (Bild 7)

Das Zeichnen von Linien mit weichen Farbübergängen stellt immense Ansprüche an die Rechenleistung der Hardware. Wer es von Hand implementieren möchte, sollte unbedingt performanceoptimierte Linienalgorithmen (Stichwort Bresenham) einsetzen.

Aufgrund dieser Probleme bietet MPAndroidChart keine Möglichkeit zum Zeichnen von Linien mit Farbübergängen. Als Ersatz hierfür erlaubt der Entwickler das Einfärben der zur Hervorhebung der einzelnen Datenpunkte verwendeten Kringel.

Da wir die Logik für die Farbbestimmung bereits in eine eigene Klasse ausgelagert haben, bietet es sich an, diese zur Ermittlung der Kringelfarben zu nutzen. Dazu ist der Code in Listing 5 notwendig, der in die Deklaration der Liniendiagrammklass wandert.

`LineChartRenderer` holt die an `setCircleColors` übergebenen Farbwerte zur Laufzeit automatisch aus dem Array. Bild 6 zeigt das Ergebnis der App-Ausführung.

Leider sind die Kreise im Moment vergleichsweise klein. Auf Telefonen und Tablets mit hoch auflösendem Bildschirm sind sie mit freiem Auge kaum zu sehen. Dieses Problem lässt sich mit folgendem Code lösen, der zum in Bild 7 dargestellten Diagramm führt:

```
LineChart generateLineChart() {
    for(int i=0;i<line1.size();i++) {
        myColorList.add(myFormatter.getColor(
            line1.get(i),0));
    }
    set1.setCircleColors(myColorList);
    set1.setCircleSize(25);
}
```

Das bulgarische Unternehmen Telerik ist für seine JavaScript-GUI-Stacks bekannt. Weniger Entwickler wissen, dass es auch

eine – kostenpflichtige – Diagrammbibliothek für Android aus demselben Haus gibt.

Diagramme à la Telerik

Telerik vertreibt den Chartrenderer als Teil des um 499 US-Dollar pro Entwickler erhältlichen Android UI-Pakets. Der Quellcode des Frameworks steht den Nutzern dabei vollumfänglich zur Verfügung, die Lizenz enthält zudem ein Jahr kostenlosen Support samt der Beantwortung von Anfragen.

Telerik UI funktioniert ab Gingerbread und unterstützt neben Android Studio auch Eclipse. Wir wollen hier – schon aus Gründen der Zukunftssicherheit – mit Android Studio arbeiten. Der erste Schritt zur Nutzung des Produkts ist das Herunterladen der unter www.telerik.com/download/android-ui bereitstehenden Demo-version. Wer noch kein Telerik-Konto hat, kann

sich direkt anmelden. Da der Download sofort nach der Eingabe der E-Mail-Adresse startet, bietet sich die Nutzung von Providern wie Suremail.info an. Extrahieren Sie die Datei `Chart-2015.3.1113-trial-release.aar` sodann an einen gut zugänglichen Ort im Dateisystem.

Öffnen Sie nach der Erstellung eines neuen Projektskeletts die Project-Leiste auf der linken Seite des Bildschirms und wechseln Sie durch Anklicken des in Bild 8 gezeigten Auswahl Fensters in die *Project*-Ansicht.

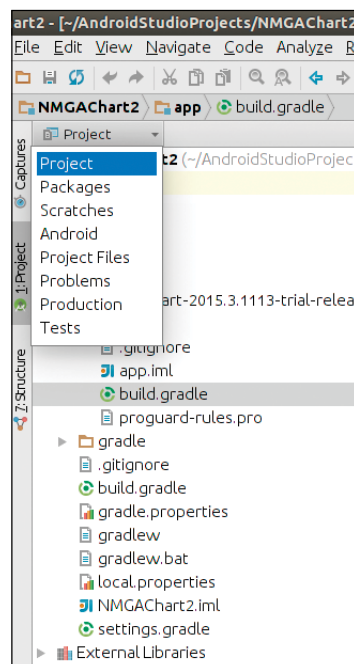
Öffnen Sie den Aufenthaltsort der Datei `Chart-2015.3.1113-trial-release.aar` in einem Dateimanager ihrer Wahl, wo Sie sie durch Drücken von [Strg C] in die Zwischenablage verfrachten. Klicken Sie den unter *app* befindlichen *libs*-Ordner rechts an und wählen Sie im daraufhin erscheinenden Kontextmenü die Option *Paste*. Verfahren Sie im nächsten Schritt analog mit den Dateien `Common-2015.3.1113-trial-release.aar` und `Primitives-2015.3.1113-trial-release.aar`.

Gradle kann AAR-Dateien – im Normalfall – nur aus Maven-Repositories herunterladen. Öffnen Sie die `build.gradle`-Datei des Projekts und ergänzen Sie den Header um folgende Passage:

```
apply plugin: 'com.android.application'
repositories {
    mavenCentral()
    flatDir {
        dirs 'libs'
    }
}
```

Im *dependencies*-Block wird die neue Bibliotheken-Gruppe eingebaut:

```
dependencies {
    compile fileTree(dir: 'libs',
```



Wenn der Project Explorer im Android-Modus ist, werden einige Ordner nicht angezeigt (Bild 8)

```
include: ['*.jar'])
testCompile 'junit:junit:4.12'
compile 'com.android.support:appcompat-v7:23.0.1'
compile 'com.android.support:design:23.0.1'
compile (name:
'Chart-2015.3.1113-trial-release',ext:'aar')
compile (name:
'Common-2015.3.1113-trial-release',ext:'aar')
compile (name:
'Primitives-2015.3.1113-trial-release',ext:'aar') }
```

Achten Sie darauf, nach dem Abarbeiten dieser Änderungen eine Gradle-Resynchronisation auszuführen. Android Studio bietet dazu einen Button an, der in einer auf der Bildschirmoberseite eingeblendeten gelben Zeile aufscheint.

Auf Telerik UI basierende Diagramme können nicht in XML-Dateien eingebunden werden. Ergänzen Sie die Layoutdatei der *MainActivity* stattdessen um den von *MPAndroidChart* bekannten Widgetcontainer, der im Rahmen von *onCreate()* nach dem Schema von Listing 6 mit einem Diagramm versehen wird.

Telerik Charts bietet Entwicklern zwei verschiedene *Chart-View*-Klassen an. *RadCartesianChartView* dient als Container für all jene Diagramme, die nicht rund sind. Tortendiagramme entstehen stattdessen in *PieChartView* – eine Unterscheidung, die bei den in folgenden Schritten besprochenen hybriden Diagrammen von eminentester Bedeutung ist.

Daten in Komplex

Wer das Programm im vorliegenden Zustand ausführt, blickt in ein leeres Diagrammfenster. Die Erzeugung eines Charts mit Telerik UI erfolgt per Data Binding und setzt das Vorhandensein einer dedizierten Datenklasse voraus – der Korpus von *NMGCDDataClass* sieht so aus:

```
public class NMGCDataClass {
    private String firmenname;
```

```
private double stuecke;
public NMGCDataClass(String name, double result) {
    this.setName(name);
    this.setStuecke(result);
}
...
```

Data Binding setzt die Erstellung von Properties voraus. Auch wenn die von Telerik bereitgestellte Engine theoretisch mit Feldern umgehen kann, wollen wir uns an das etablierte Designpattern halten. Telerik Charts erwartet bei der Nutzung von *PropertyNameDataPointBinding* das Vorhandensein von Akzessormethoden, deren Name nach dem gezeigten Schema *<operation><Feldname>* aufgebaut sein muss:

```
public double getStuecke() {
    return this.stuecke;
}
public void setStuecke(double value) {
    this.stuecke = value;
}
public String getName() {
    return this.firmenname;
}
public void setName(String value) {
    this.firmenname = value;
}
}
```

Als nächster Akt folgt die Erstellung eines Arrays, in dem die darzustellenden Informationen abgelegt werden. Die eigentliche Anzeige erfolgt über eine Instanz von *LineSeries*. Die Klasse ist unter anderem für das Herstellen der Bindungsbeziehungen zwischen den Datenfeldern und der Rendering-Engine zuständig (Listing 7).

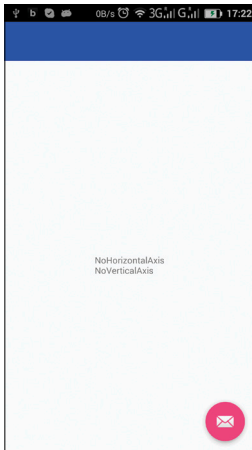
Das Programm kann an dieser Stelle zur Verifikation der Funktion angeworfen werden – das Chart präsentiert sich wie

Listing 6: Widgetcontainer

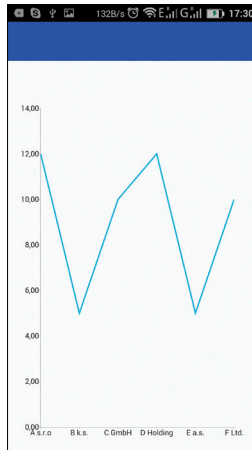
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    RadCartesianChartView chartView = new
    RadCartesianChartView(this);
    RelativeLayout rl=(RelativeLayout)
    findViewById(R.id.chartContainer);
    ViewGroup.LayoutParams myParams=new
    ViewGroup.LayoutParams(ViewGroup.LayoutParams.
    MATCH_PARENT,
    ViewGroup.LayoutParams.MATCH_PARENT);
    chartView.setLayoutParams(myParams);
    rl.addView(chartView);
}
```

Listing 7: Array mit Informationen

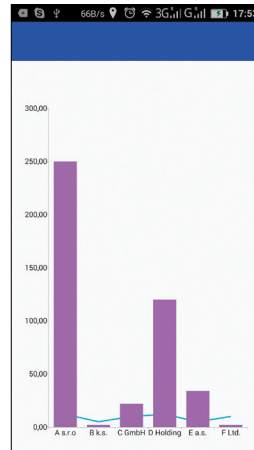
```
List<NMGCDataClass> firmResults = new
ArrayList<NMGCDataClass>();
firmResults.add(new NMGCDataClass("A s.r.o", 12));
firmResults.add(new NMGCDataClass("B k.s.", 5));
firmResults.add(new NMGCDataClass("C GmbH", 10));
firmResults.add(new NMGCDataClass("D Holding", 12));
firmResults.add(new NMGCDataClass("E a.s.", 5));
firmResults.add(new NMGCDataClass("F Ltd.", 10));
LineSeries lineSeries = new LineSeries();
lineSeries.setCategoryBinding(new
PropertyNameDataPointBinding("Name"));
lineSeries.setValueBinding(new
PropertyNameDataPointBinding("Stuecke"));
lineSeries.setData(firmResults);
chartView.getSeries().add(lineSeries)
```

Für die Darstellung von Daten sind Achsinformationen notwendig (Bild 9)



Das Liniendiagramm ist am richtigen Platz positioniert (Bild 10)



Balkendiagramm überdecken das Liniendiagramm (Bild 11)

in Bild 9 gezeigt. *NMGChart2* zeigt während des Starts zudem eine Nachricht an, die Nutzer und Entwickler auf die fehlende Lizenz der GUI-Komponenten hinweist.

Zuordnung von x und y

Der letzte Schritt zum funktionsfähigen Diagramm ist das Einführen von Achsen, die eine Beziehung zwischen den Daten und den x- und y-Koordinaten herstellen. Fürs Erste genügt uns hierbei folgender Code:

```
CategoricalAxis horizontalAxis = new CategoricalAxis();
chartView.setHorizontalAxis(horizontalAxis);
LinearAxis verticalAxis = new LinearAxis();
chartView.setVerticalAxis(verticalAxis);
```

LinearAxis-Klassen implementieren eine gewöhnliche Koordinatentransformation: Die Obergrenze wird dynamisch errechnet, während die Untergrenze normalerweise null ist. Die *CategoricalAxis* unterteilt Werte derweil in Gruppen – für unsere Firmendaten ist dies geradezu ideal (Bild 10).

Kombinatorische Darstellung

Telerik betont in Produktvorstellungen die Möglichkeit zur Erzeugung hybrider Diagramme. Es handelt sich dabei um Charts, die mehrere Darstellungsarten kombinieren. Ein interessantes Beispiel hierfür wäre ein Chart, das Preise und Handelsvolumen hinter- beziehungsweise übereinander darstellt.

Als ersten Schritt müssen wir *NMGCDatClass* um ein weiteres Informationsfeld erweitern. Die Erweiterung des Konstruktors wird hier aus Platzgründen nicht abgedruckt – achten Sie darauf, im Rahmen der Erstellung in *onCreate* beliebige Werte im Bereich von 0 bis 250 zu vergeben:

```
public class NMGCDataClass {
    private double gewinn;
    ...
    public double getGewinn() {
```

```
        return this.gewinn;
    }
    public void setGewinn(double value) {
        this.gewinn = value;
    }
}
```

Die Erzeugung eines hybriden Diagramms unterscheidet sich nur minimal von der normaler Charts. Das fertige *BarSeries*-Objekt wird im nächsten Schritt ebenfalls in das *Series*-Objekt des Diagramms eingeschrieben, um die Charting-Engine zur Berücksichtigung beider Datenelemente zu animieren:

```
LineSeries lineSeries = new LineSeries();
...
BarSeries mySeries=new BarSeries();
mySeries.setCategoryBinding(new
    PropertyNameDataPointBinding("Name"));
mySeries.setValueBinding(new
    PropertyNameDataPointBinding("Gewinn"));
mySeries.setData(firmResults);
chartView.getSeries().add(mySeries);
```

ChartView arbeitet den Inhalt des *getSeries*-Arrays linear ab. Wer die Linie vor den Bars in das Array einschreibt, muss mit Darstellungsfehlern leben (Bild 11). Dieses Problem lässt sich durch das Angeben eines Index beheben. Wer das Diagramm durch Übergeben eines nach dem Schema

```
chartView.getSeries().add(0,mySeries);
```

aufgebauten Index zu reparieren sucht, wird mit einem komplett einfarbigem Chart belohnt. Da das manuelle Festsetzen von Farben ebenfalls nicht funktioniert, bleibt nur die Umkehrung des Einpflegens der einzelnen Diagrammobjekte. Wer den Code folgendermaßen aufbaut, bekommt das korrekt aussehende Chart (Bild 12):

```
BarSeries mySeries=new BarSeries();
...
chartView.getSeries().add(mySeries);
LineSeries lineSeries = new LineSeries();
...
chartView.getSeries().add(lineSeries);
```

Angeichts der großen Unterschiede in den Wertebereichen der beiden Datenreihen ist die Übersichtlichkeit unseres kombinatorischen Diagramms nicht eben befriedigend: Die Stückzahl ist an den unteren Rand des Bildschirms gedrängt.

Dieses Problem lässt sich durch die Erstellung eines Diagramms mit zwei Achsen erzeugen. Der Entwickler weist jeder Datenserie eine eigene Achse zu, die für die Anzeige der jeweiligen Informationen ideal geeignet ist:

```
CategoricalAxis horizontalAxis = new CategoricalAxis();
chartView.setHorizontalAxis(horizontalAxis); ▶
```

```

LinearAxis profitAxis = new LinearAxis();
profitAxis.setHorizontalLocation
(AxisHorizontalLocation.RIGHT);
mySeries.setVerticalAxis(profitAxis);
LinearAxis verticalAxis = new LinearAxis();
chartView.setVerticalAxis(verticalAxis);

```

Unser Beispiel nutzt zwei Achsenarten. *CategoricalAxis* nutzt das ID-Attribut zur Unterteilung der einzelnen Datenelementen. *LinearAxis* ist eine gewöhnliche lineare Achse, die sich analog zu *MPAndroidChart* verhält.

Schon aus Platzgründen wollen wir unsere Besprechung von gewöhnlichen Diagrammen an dieser Stelle beenden. **Bild 13** zeigt den Lohn unserer Mühen – weitere Informationen finden sich in der unter <http://docs.telerik.com/devtools/android/controls/chart> abrufbaren Dokumentation.

Traders Liebling

John J. Murphys Klassiker zur technischen Analyse der Finanzmärkte listet Dutzende von als Indikator bezeichneten Funktionen auf, die Aktienhändlern zur Abschätzung der Marktbewegungen dienen. Rein theoretisch spricht nichts dagegen, diese von Hand nachzuprogrammieren und in *MPAndroidChart* zu implementieren. Wirklich ergiebig ist diese Arbeit indes nur dann, wenn Ihr Produkt mit seiner Chartdarstellung zu punkten sucht.

Für alle anderen Entwickler ist es sinnvoller, die Erstellung von Aktiendiagrammen an eine darauf spezialisierte Bibliothek auszulagern. Modulus Global ist für seine im Bankenbereich weit verbreiteten Finanzmanager bekannt – mit *StockChartX* bietet die Firma ein Modul an, mit dem Drittentwick-

ler die Chart-Engine mitnutzen können. Leider ist das Produkt nicht am freien Markt erhältlich. Wer die Bibliothek verwenden möchte, muss beim Unternehmen anrufen und um eine Teststellung bitten.

Wir wollen hier stattdessen die Erzeugung von Diagrammen auf Basis von Telerik Charts besprechen, die ebenfalls fertige Indikatoren anbieten.

NMGACHart3 ist vom Aufbau her mit *NMGACHart2* eng verwandt, unterscheidet sich aber im Rahmen der Informationsbeschaffung. Da Yahoo die Syntax seines zum Abrufen historischer Aktiendaten vorgesehenen CSV-API regelmäßig verändert, wollen wir uns – es geht hier ja um Diagramme, und nicht um das Parsen von Daten – auf die in den auf [Finance.yahoo.com](http://finance.yahoo.com) angezeigten Aktienlistings enthaltenen historischen Kursdaten einschießen.

Laden Sie die Informationen durch Anklicken von *Download to Spreadsheet* herunter und speichern Sie den Inhalt als neuen String in *values.xml*:

```

<resources>
    <string name="app_name">NMGACHart3</string>

```

Listing 8: NMGFinancialSeries

```

public class NMGFinancialSeries {
    float open;
    ...
    static int counter=0;
    public NMGFinancialSeries(String _myLine)
    {
        String myValues[]=_myLine.split(",");
        open=new Float(myValues[1]);
        high=new Float(myValues[2]);
        low=new Float(myValues[3]);
        close=new Float(myValues[4]);
        id=counter++;
    }
    public int getId() {
        return id;
    }
    public float getOpen() {
        return open;
    }
    ...
}

```

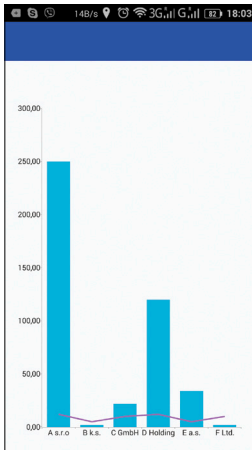
Listing 9: RadCartesianChartView

```

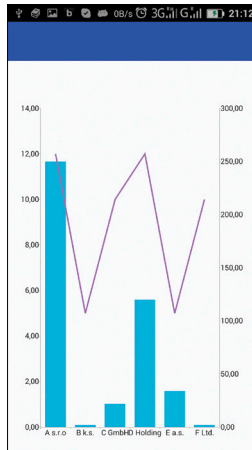
RadCartesianChartView chartView = new RadCartesian
ChartView(getApplicationContext());
CandlestickSeries series = new
CandlestickSeries();
series.setCategoryBinding(new
PropertyNameDataPointBinding("Id"));
series.setOpenBinding(new
PropertyNameDataPointBinding("Open"));
series.setHighBinding(new
PropertyNameDataPointBinding("High"));
series.setLowBinding(new
PropertyNameDataPointBinding("Low"));
series.setCloseBinding(new
PropertyNameDataPointBinding("Close"));
series.setData(mySeries);
chartView.getSeries().add(series);

CategoricalAxis horizontalAxis = new
CategoricalAxis();
chartView.setHorizontalAxis(horizontalAxis);
LinearAxis verticalAxis = new LinearAxis();
chartView.setVerticalAxis(verticalAxis);
RelativeLayout rl=(RelativeLayout)findViewById
(R.id.chartContainer);
ViewGroup.LayoutParams myParams=new
ViewGroup.LayoutParams
(ViewGroup.LayoutParams.MATCH_PARENT,
ViewGroup.LayoutParams.MATCH_PARENT);
chartView.setLayoutParams(myParams);
rl.addView(chartView);
}

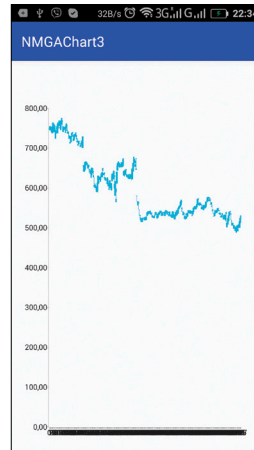
```



Gewinn- und Umsatzkurven teilen sich eine Diagrammfläche (Bild 12)



Eine zweite Achse erhöht die Übersichtlichkeit (Bild 13)



Die Aktieninfos erscheinen in Form eines Candlestick-Charts (Bild 14)

```
<string name="csvdata">
Date,Open,High,Low,Close,Volume,Adj Close
2015-12-24,749.549988,751.349976,746.619995,748.
400024,527200,748.400024
2015-12-23,753.469971,754.210022,744.00,750.
309998,1565900,750.309998
...</string>
</resources>
```

Unsere Datenklasse unterscheidet sich nur insofern von ihrem Vorgänger, als sie die zeilenweise angelieferten Werte im Rahmen des Konstruktors in ihre Einzelteile zerlegt. Stringbearbeitung hat beim eigentlichen Zeichnen von Diagrammen insofern keinen Platz, als es hier auf Performance ankommt (Listing 8).

NMGFinancialSeries muss die einzelnen Datenpunkte mit einer einzigartigen ID ausstatten. Der einfachste Weg zur Erzeugung einer laufenden Seriennummer ist die Nutzung einer statischen Membervariable, die bei jedem Aufruf des Konstruktors inkrementiert wird.

Damit können wir uns der eigentlichen Realisierung des Charts zuwenden, die abermals in *onCreate* erfolgt:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    String myString=getApplicationContext().getString(
        R.string.csvdata);
    String myField[]=myString.split("2015");
```

Da die Daten als *<string>* vorliegen, splitten wir über die Jahreszahl – es handelt sich hier ja nur um ein didaktisches Beispiel. Im nächsten Schritt nutzen wir die einzelnen Datenfragmente zur Bevölkung des *mySeries*-Arrays:

```
ArrayList<NMGFinancialSeries> mySeries=new ArrayList<
NMGFinancialSeries>();
```

```
for(int i=1;i<myField.length;i++)
{
    mySeries.add(new NMGFinancialSeries
        (myField[i]));
}
```

Die eigentliche Erstellung der Chartklasse unterscheidet sich insofern von den bisher vorgestellten linearen Diagrammen, als wir nun eine Instanz von *CandlestickSeries* anfordern und sie mit der vergleichsweise umfangreichen Datenklasse verdrahten (Listing 9). Führen Sie das Programm im vorliegenden Zustand aus, um das in Bild 14 gezeigte Diagramm zu erhalten.

Berechnete Kurven

Als letzte Aufgabe wollen wir unser Telerik-Aktiendiagramm um einen Indikator erweitern. Es handelt sich dabei um berechnete In-

formationen, die durch eine mathematische Transformation aus dem Kursverlauf des Wertpapiers entstehen.

Da die Datenbereitstellung im Backend von Telerik Charts erfolgt, sind nur minimale Änderungen im Aufbau des Diagramms erforderlich. Ein auf dem Prinzip des gleitenden Durchschnitts basierender Indikator erfordert neben der Zuweisung von zwei Datenpaaren nur Informationen über die Berechnungsperiode:

```
MovingAverageIndicator indicator = new
MovingAverageIndicator();
```

```
indicator.setPeriod(15);
indicator.setCategoryBinding(new
PropertyNameDataPointBinding("Id"));
indicator.setValueBinding(new
PropertyNameDataPointBinding("Close"));
indicator.setData(mySeries);
chartView.getSeries().add(indicator);
```

Damit sind wir auch an dieser Stelle fertig.

Fazit

Diagramme helfen Nutzern bei der Erfassung komplexer Sachverhalte. Jede der hier vorgestellten Bibliotheken hat ihre individuellen Stärken und Schwächen – wählen Sie die, die am besten zu Ihrer Situation passt. ■



Tam Hanna

ist Autor, Trainer und Berater mit den Arbeitsschwerpunkten Webentwicklung und Webtechnologien. Es lebt in der Slowakei und leitet dort die Firma Tamoggemon Holding k.s. Er bloggt sporadisch unter www.tamoggemon.com

XTEXT, XTEND: DOMAIN SPECIFIC LANGUAGE FÜR C++ / QT (TEIL 4)

Modelle der mobaDSL

Open-Source-DSL zur Erstellung von mobilen Business-Anwendungen.

In den ersten drei Folgen dieser Artikelserie habe ich bereits die Vorteile einer DSL herausgestellt und an einigen Beispielen gezeigt, wie die Entwicklung schneller umgesetzt wird und typische Copy-and-Paste-Fehler vermieden werden können. Als Basis für meine DSL diente eine bereits für ein anderes Projekt verfügbare DSL zur Beschreibung von Datentransferobjekten (DTO), die dann über Annotationen um einige Features angereichert wurde.

Während der letzten EclipseCON Europe habe ich mich mit Florian Pirchner, dem Entwickler der DTO-DSL, zusammengesetzt und ihm geschildert, was ich optimal von einer DSL erwarten würde, die bei der Entwicklung mobiler Business-Anwendungen unterstützt. Das Ergebnis: Es gibt eine neue DSL, die exakt diese Anforderungen umsetzt.

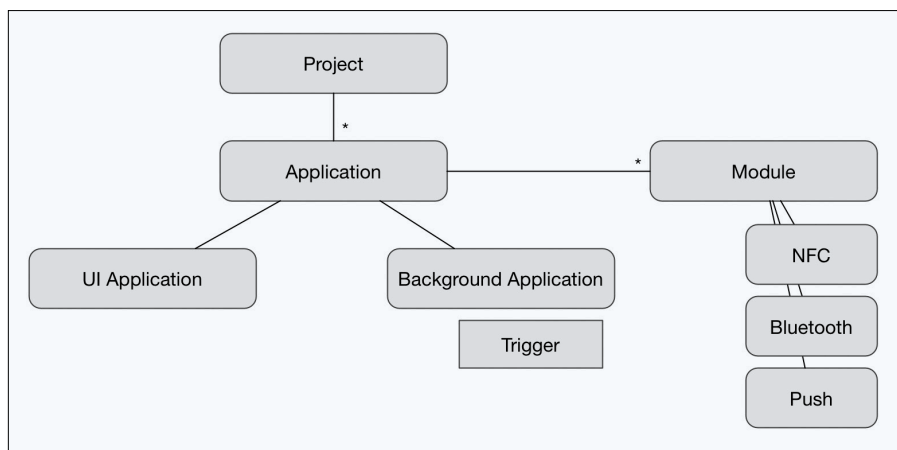
Florian Pirchner erstellt die Grammar der DSL anhand meiner Anforderungen mit Xtext und ich teste diese in realen Projekten und schreibe den Xtend-Generator dazu. Dieser Artikel erklärt die grundlegenden Anforderungen an die DSL, die weit über das hinausgehen, was bisher angedacht war. Wie die Modelle der mobaDSL aussehen, kann auch schon gezeigt werden.

Natürlich ist auch diese DSL Open Source und unter der Apache-Lizenz verfügbar.

de Projekte basierend auf älteren APIs nicht zu beeinflussen. Die UI-App und die Background-App werden als *application* modelliert:

```
application CustomerUiApp version = 1.0.1 {
    // ... }
application CustomerBackgroundApp version = 1.0.1 {
    trigger push }
```

Einer Background-App können mehrere Trigger zugeordnet werden: App-Start, App-Update, App-Installation, Push, SMS, Mail-Eingang und Geofence.



mobaDSL-Projekte: Das Diagramm zeigt den Aufbau der mobaDSL-Projekte (Bild 1)

Project und Application

Ein Projekt kann eine oder mehrere Applikationen haben – derzeit ist die UI-Applikation und die Background-Applikation umgesetzt.

Die UI-Applikation ist der Normalfall einer App mit einer Bedienoberfläche. In vielen Fällen wird auch noch eine Background-Applikation benötigt – unter Android wäre dies ein Service, bei BlackBerry 10 eine Headless App. Ein Projekt wird im Model so beschrieben:

```
project FooCustomer version = 0.7.3 {
    uiApp = CustomerUiApp
    backgroundApp = CustomerBackgroundApp
}
```

Wie wir sehen, unterstützt die DSL die Versionierung von Artefakten, um es zu erleichtern, neue Funktionen einzubauen oder Anpassungen vorzunehmen und gleichzeitig existieren-

Diese Trigger legen fest, wann die Background-Applikation aktiv wird. Allgemeiner Code zur Implementierung kann dann leicht dazugeneriert werden.

Die Modelle selbst werden in MOBA-Dateien beschrieben, wobei sich ein Projekt über beliebig viele MOBA-Dateien verteilen kann. Eingebunden werden diese Dateien über:

```
use Template MyTemplate
```

Es werden bereits viele Templates und Beispiele zur Verfügung gestellt. Diese Templates können direkt genutzt werden oder als Muster für eigene Templates dienen. Templates können beliebig tief verschachtelt sein.

Daten, die von den Applikationen genutzt werden, können als Entity oder DTO beschrieben werden. Entities werden gespeichert (gecacht), während DTOs für Server-Payloads oder Ähnliches zum Einsatz kommen. Eine spezielle Ausprägung

von Entities sind die Settings. Settings gibt es nur einmal für eine Applikation, und Settings werden immer sofort zurückgeschrieben und immer als JSON-Datei persistiert.

Server, REST-Services, Server-Queues und Workflows können mit der mobaDSL ebenfalls definiert werden. Eine Queue wird wie eine Entity persistiert, kann aber sowohl Payloads als auch Entities referenzieren und beinhaltet spezielle Logiken, um die Payloads aus der Warteschlange an den Server zu senden, sowie im Fehlerfall oder Offline-Modus richtig reagieren zu können. Workflows beschreiben eine Kette von Aktionen, die nacheinander ausgeführt werden müssen – es werden also mehrere asynchrone Serveraufrufe ähnlich einer Transaktion zusammengefasst. Zur Definition von Server und Services siehe weiter unten im Artikel. Bild 1 gibt einen groben Überblick über mobaDSL-Projekte, Bild 2 zeigt die Datenbeziehungen.

Module (Plug-ins)

Während der Entwicklung von mobilen Business-Anwendungen kommt es immer wieder vor, dass bestimmte Funktionalitäten programmiert werden müssen und man sich dann ähnlichen Code aus bereits umgesetzten Projekten sucht und diesen anpasst und integriert. Um das zu vereinfachen, kennt die mobaDSL Module, die einer Applikation zugewiesen werden können. Derzeit gibt es Bluetooth, NFC und Push Services. Hier ein Beispiel:

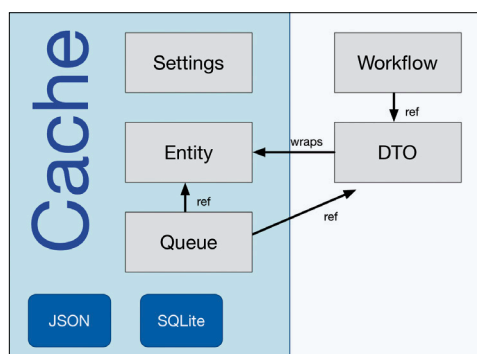
```
module bluetooth BTControllerLE type = lowEnergy
```

Beim Bluetooth-Modul kann definiert werden, ob es sich um BluetoothLE handelt oder Beacon oder SPP (Seriellles Profil). Der dazugehörige Controller (BTControllerLE.cpp/.hpp) und die entsprechenden Events (SIGNALS, SLOTS) werden vom Generator erzeugt.

Server und (REST-)Services

Einer Applikation können mehrere Server zugeordnet werden. Dies ermöglicht es, verschiedene Content-Provider zu integrieren oder beispielsweise Bilder-Uploads an einen anderen Server zu senden als den REST-API-Server.

Jedem Server können Gruppen von Services zugeordnet werden, die sich wiederum in RESTcustom- und RESTcrud-Services unterteilen. CRUD-Services ermöglichen *Create*, *Read*, *Update* und *Delete* gemäß den REST-Spezifikationen.



mobaDSL

Daten:

Das Diagramm gibt einen Überblick über die Datenbeziehungen (Bild 2)

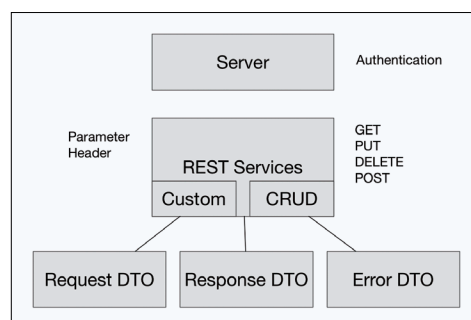
Custom-Services werden dagegen für sonstige *GET*- und *POST*-Aktionen genutzt.

Zu einem Service können Parameter, Header, HTTP-Methode, Request Payload (DTO), Response Payload (DTO), Error Payload (DTO) sowie die Authentication (User-Passwort, Token, OAUTH, SAP Mobile Platform, ...) beschrieben werden. Bild 3 gibt einen Überblick.

Caching

Das voreingestellte Caching-Verhalten wird an der Applikation hinterlegt. Folgende Varianten sind derzeit möglich:

- Caching-Strategie,
- Cachen, wenn Programm beendet,
- Jede Änderung in Cache schreiben,
- Zeitbezogenes Cachen (Intervall alle x Sekunden),
- Caching-Typ,
- Kein Cache,
- Read-only Cache,
- Read-Write-Cache,



Server und Services:

Das Diagramm zeigt, was zu einem Service gehört (Bild 3)

- Persistenz,
- JSON-Dateien,
- SQLite.

Ein Beispiel aus dem Modell:

```
cache(strategy = CACHE_STRATEGY_AT_EXIT type =
CACHE_TYPE_READ_WRITE)
```

Strategie, Typ und Persistenz sind über Konstanten frei definierbar, sodass sich jeder den Generator leicht anpassen kann.

An jeder Entity kann das Caching-Verhalten überschrieben werden. So ist es möglich, das Caching als Default in JSON-Dateien auszuführen, aber für eine spezielle Entity mit sehr großer Datenmenge in die SQLite zu schreiben. Außerdem könnte man normalerweise den Cache erst am Programm-Ende zurückzuschreiben, aber für eine Entity bereits nach jeder Änderung.

Entities und DTOs

Entities und DTOs können all das, was schon in den ersten drei Folgen dieser Artikelserie beschrieben wurde – nur ist vieles jetzt einfacher geworden, da nicht mehr mit ►

Listing 1: Beispiel-Modell (Teil 1)

```

project ekkesDSLsample version = 1.0.0 {
    uiApp = sampleUiApp }

application sampleUiApp version = 1.0.0 cache(strategy
= CACHE_STRATEGY_AT_EXIT type = CACHE_TYPE_READ_WRITE)
friends("DataConverter") {
    use template MobaDataTypeTemplates
    use template MobaCachingTemplate
    use template MobaAuthorizationTemplate

    serialization XML
    serialization JSON
    // MODULES
    module bluetooth BTControllerLE type = lowEnergy
    friends("CustomBTController")
    // SERVER
    server TheServer url =
    "http://www.mobadsl.org/server/api" {
        services = (CustomerService)
        authorization = USER_PWD }
    // QUEUE
    queue ServerQueue {
        ref CustomerRequestPayload customerRequest }
    // SERVICES
    rest CustomerService {
        path = "/customer"
        method = GET
        parameters {
            param String aString = "Something"
            param int aNumber = 42 }
        headers {
            ContentTypeHeader = CONTENTTYPE_APP_JSON
            RawHeader RAWHEADER_AUTHENTICATION =
                "Basic a2V0dGVycDprZXROZXJwMw=="
            RawHeader RAWHEADER_ACCEPT =
                CONTENTTYPE_APP_JSON
            RawHeader RAWHEADER_ACCEPT_LANGUAGE =
                LANGUAGE_EN }
        requestDto = CustomerRequestPayload
        errorDto = CommonErrorPayload
        responseDto = CustomerPayload isArray as JSON }
    // DATATYPES
    datatype OrderState enum {
        lit NEW = ("new", 00)
        lit CONFIRMED = ("confirmed", 01)
        lit DELIVERED = ("delivered", 02)
        lit CLOSED = ("closed", 03) }
    // SETTINGS
    settings SettingsData {
        var domainKey int ^id
        var bool isProductionEnvironment
        var bool isMock
        var int ^version
        var String defaultScanner
        refEntity cascading ServerSettings[1 ..*]
        ^server // }
    entity ServerSettings {
        var domainKey String uuid
        var String serverUrl
        var bool isProductionEnvironment }
    // ENTITY
    entity Order {

```

Annotationen, sondern direkt mit Schlüsselwörtern der DSL gearbeitet werden kann. Hier zwei Beispiele: Die Formatierung von Datumsfeldern und die Verwendung anderer Property-Namen auf Serverseite:

```

@DateFormatString("yyyy-MM-dd")
var Date[1] orderDate;
//
var Date[1] orderDate format(DF_YEAR_MONTH_DAY)

```

Das Format selbst ist in einer Konstanten definiert, die in einem der Templates beschrieben ist:

```

application MobaDateFormatTemplates version = 0.9.1 {
    const DF_TIME = "HH:mm:ss"
    const DF_TIME_STAMP = "yyyy.mm.dd HH:mm:ss"
    const DF_YEAR_SHORT = "yy"
    const DF_YEAR_LONG = "yyyy"
    const DF_MONTH = "mm"
    const DF_DAY = "dd"
    const DF_YEAR_MONTH_DAY = "yyyy-MM-dd"

```

```

const DF_YEAR_MONTH_DAY_HOUR_MIN_SEC =
    "yyyy-MM-dd hh:mm:ss"
}

```

Das Template wird durch diesen Aufruf eingebunden:

```
use template MobaDataTypeTemplates
```

Hier sieht man exemplarisch, wie leicht die mobaDSL an projektspezifische Eigenheiten angepasst werden kann: Einfach die Konstanten anpassen, ändern oder weitere hinzufügen und einer *Date*-, *Time*- oder *Timestamp*-Variablen über *format MY_CONSTANT* zuweisen.

Hier das Beispiel mit abweichenden Properties. Anstelle einer Annotation mit String ist das jetzt bei DTOs direkt in der Variablen-Definition möglich:

```
var String companyName1 alias(CompanyName1)
```

Man sieht auf einen Blick, was gemeint ist, und durch den Context-Assist ist die Eingabe im Editor ebenfalls einfacher.

Listing 1: Beispiel-Modell (Teil 2)

```

var domainKey int nr
var bool expressOrder
var String[1] title
var Date[1] orderDate format(DF_YEAR_MONTH_DAY)
var OrderState state
var transient int processingState
ref cascading Item[1 ..*] positions opposite order
ref lazy Customer[1] customerId
var String[0 ..*] domains }
entity Item {
    var domainKey String uuid
    var int[1] posNr
    var String[1] name
    var double[1] quantity
    ref Order order opposite positions
    ref cascading SubItem[0 ..*] subItems opposite
    item }
entity SubItem {
    var domainKey String uuid
    var int[1] subPosNr
    var String[1] description
    var String[1] barcode

    ref Item item opposite subItems }
entity Customer cache (strategy =
CACHE_STRATEGY_AT_EXIT type = CACHE_TYPE_READ_WRITE
using SQ_LITE) {
    var domainKey int ^id
    var String companyName
    var GeoCoordinate coordinate
    var GeoAddress geoAddress
    index companyNameINDEX(companyName) }
// DATA TRANSFER OBJECTS --- PAYLOADS
dto CustomerRequestPayload {
    var String aString
    var int aNumber }
dto CustomerPayload {
    var domainKey int ^id alias(CompanyID)
    var String companyName1 alias(Company_Name1)
    var String companyName2 alias(Company_Name2) }
dto CommonErrorPayload {
    var String message
    var int errorCode }
}

```

Listing 1 zeigt ein erstes komplettes Modell für unsere Beispielanwendung – jetzt erweitert um Module, Server und Services. Wer jetzt schon selbst etwas entwickeln und testen möchte: Das GitHub-Projekt von mobaDSL stellt die DSL, Templates, Sample, Dokumentation sowie Informationen aus dem Buildprozess mit Jenkins zur Verfügung.

Zielplattformen Mobile, Web und Server

mobaDSL ist in der Lage, aus einem Modell Code für verschiedene Zielplattformen zu generieren. Die erste Version ist wie bisher auf BlackBerry 10 (Qt 4.8, QML, C++) ausgelegt

– es folgen Android, iOS und Windows 10. Darüber hinaus werden wir aber auch eine Browservariante als Vaadin-Webapplikation, sowie den zugehörigen REST-Server generieren. Selbst wenn das im eigentlichen Projekt nicht gefordert ist und die REST-Services durch externe Server zur Verfügung gestellt werden, ist es zum Testen eine gute Sache, die Schnittstellen mit einem eigenen REST-Server zu testen und in einem Web-UI zu verifizieren.

Fazit

Die komplette Neuentwicklung der DSL für mobile Business-Anwendungen gibt uns viel mehr Möglichkeiten als die vorher genutzte DTO-DSL. In der nächsten Folge sehen wir dann, wie der Generator zum Modell aussieht.

Eine andere wichtige Neuerung durch die Verwendung von Xtext 2.9 ist die Möglichkeit, den Editor nicht nur in der Eclipse IDE, sondern auch im Browser ausführen zu können. Dies vereinfacht die Installation und die Akzeptanz für Entwickler, die nicht aus dem Eclipse-Umfeld kommen. ■

Links zum Thema

- mobaDSL bei GitHub
<http://Github.com/florianpirchner/mobadsl>
- mobaDSL-Templates bei GitHub
<http://github.com/florianpirchner/mobadsl-templates>
- mobaDSL-Dokumentation
<http://github.com/florianpirchner/mobadsl/tree/master/org.mobadsl.docu>
- mobaDSL Jenkins
<http://jenkins.lunifera.org:8083/job/mobaDSL>
- Xtext DSL im Browser
<http://spoenemann.blogspot.de/2015/05/editing-dsls-in-web-browser.html>



Ekkehard Gentz

ist Autor, Trainer und Speaker auf Konferenzen. Er entwickelt als Independent Software Architect mobile Anwendungen für internationale Kunden, ist BlackBerry Elite Member und bloggt unter:
<http://ekkes-corner.org>

FRAMEWORK: THE-M-PROJECT

Systemunabhängig

Das Framework The-M-Project vereinfacht die Cross-Plattform-Entwicklung.

Die native Anwendungsentwicklung für bestimmte mobile Geräte ist zeitaufwendig, und die anschließende Portierung von einem mobilen Betriebssystem, zum Beispiel iOS, auf ein anderes, zum Beispiel Android oder Windows Phone, ist aufgrund des damit verbundenen Zeit- und Kostenaufwands so gut wie unmöglich.

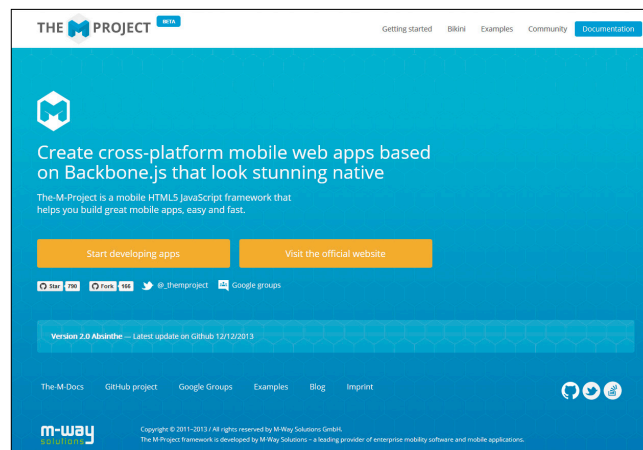
Abhilfe sollen Webanwendungen schaffen, die systemunabhängig für die Ausführung in Webbrowsern entwickelt werden. Frameworks, die Rahmengerüste für diese Webanwendungen definieren, vereinfachen und vereinheitlichen dabei die Programmentwicklung. Das hat allerdings zur Folge, dass Aufbau und Erscheinungsbild der mit diesen Frameworks entwickelten Anwendungen aufgrund vordefinierter Grundstrukturen vorgegeben sind.

Um den HTML5-Zugriff zu vereinheitlichen, gibt es bereits mehrere Framework-Ansätze. Das Framework The-M-Project zielt auf die vereinfachte Entwicklung von Webanwendungen ab (Bild 1).

Grundlagen

Um eine einfache Handhabung des Frameworks zu garantieren, setzt The-M-Project auf HTML5 und JavaScript. Die Skriptsprache JavaScript unterstützt die funktionale, prozedurale und objektorientierte Programmierung und stellt grundlegende Datentypen sowie leistungsfähige Kontrollstrukturen zur Verfügung.

The-M-Project ist ein Framework, das zum Bau von Webanwendungen entworfen wurde, die wahlweise den Online-, aber auch den Offline-Modus unterstützen. Die Bereitstellung der grundlegenden Offline-Funktionalität realisieren



Das Framework The-M-Project zielt auf die vereinfachte Entwicklung von Webanwendungen ab (Bild 1)

Entwickler mit HTML5 und der zugeordneten Speichertechnik WebStorage. Anwendungen lassen sich auch dann weiter ausführen, wenn die Online-Verbindung abreißt.

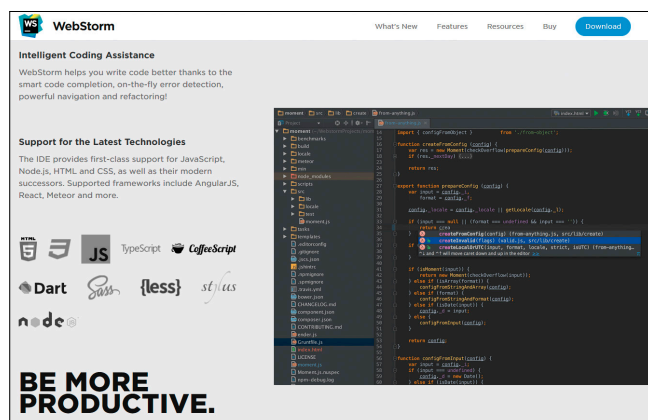
HTML5 bildet dabei die Grundlage für die Entwicklung von optimierten, mobilen Webanwendungen, die ihrerseits verbesserte Benutzerinteraktionen und attraktivere Programmoberflächen aufweisen. Die HTML5-Spezifikation bietet spezielle Funktionen für mobile Geräte an, die sich auch in The-M-Project wiederfinden.

The-M-Project ist geräte- und systemübergreifend nutzbar. Es ist keine zeitintensive Einarbeitung in die unterschiedlichen mobilen Betriebssysteme und plattformspezifische Programmiersprachen erforderlich. Entwickler müssen sich lediglich mit The-M-Project und JavaScript vertraut machen. Anwendungen lassen sich damit nicht nur einfacher und schneller entwickeln, sondern auch ohne zeitaufwendige und kostenintensive Programmportierungen auf unterschiedlichen mobilen Geräten einsetzen.

Als Open-Source-Anwendung unterliegt The-M-Project der MIT-Lizenz. Diese Lizenz erlaubt es, das Framework und die zugehörige Software und Quelltexte samt Dokumentation frei und uneingeschränkt zu nutzen und zu verteilen, unabhängig davon, ob der eigene Quelltext einsehbar ist oder nicht.

Grundlegende Komponenten

The-M-Project ist ein Open-Source-Framework, das selbst diverse Open-Source-Projekte als Grundlage verwendet, darunter jQuery Mobile sowie Espresso und Node.js. Hinter jQuery Mobile verbirgt sich eine Rendering-Engine. Sie sorgt



WebStorm ist eine Entwicklungsumgebung für Windows, Mac OS und auch Linux, die HTML5-, CSS- und JavaScript-Unterstützung bietet (Bild 2)

für spezifische Darstellungen auf unterschiedlichen mobilen Systemplattformen und passt die Oberfläche für die mobilen Geräte wie Smartphones und Tablets an, die für Berührungseingaben per Touchscreen optimiert sind.

Mit Espresso steht ein gesondertes Build-Werkzeug bereit, das auch die Programmverteilung erleichtert und selbst auf Node.js basiert. Darüber werden ein integrierter Testserver und ein Compiler angeboten. Espresso unterstützt den Entwickler bei der Strukturierung von Quelltext, beim Erstellen von Anwendungen und beim Ausführen der Apps.

Node.js ist eine JavaScript-Bibliothek, die ereignisbasierte Ein- und Ausgaben für die Erstellung skalierbarer Netzwerk-anwendungen bereitstellt und das Überwachen von Verbindungen basierend auf dem HTTP-Protokoll erlaubt.

Unter Windows arbeiten Programmierer außerdem mit Cygwin, um eine Linux-Umgebung zusammen mit einem untergeordneten API verfügbar zu machen.

Model-View-Controller-Framework

Auf diesen Komponenten setzt wiederum das HTML5-Framework The-M-Project auf. Es bietet die grundlegenden Funktionalitäten der Komponenten jQuery Mobile, Espresso und Node.js und stellt einen integrierten Datenprovider und ein Model-View-Controller-Framework (MVC) bereit. Die Besonderheit ist, dass bei Webanwendungen die MVC-Elemente zwischen Webserver und Browser verteilt werden.

Hinter dem Model verbirgt sich das Datenmodell. Dies ist für das mobile Gerät die HTML-Seite, die im Browser dargestellt wird. Die View ist die Präsentation, die über den Webbrowser clientseitig anhand der über den URL übergebenen Webseite aufbereitet wird. Die Webseite kann Tests, Bilder, Formulare oder auch eingebettete Objekte enthalten und wird ihrerseits nach Bedarf serverseitig aufbereitet.

Die Kommunikation zwischen Client und Server erfolgt über HTTP-Request und HTTP-Response. Der Controller ist für die Programmsteuerung verantwortlich und kann sowohl das Model als auch die einzelnen Views ändern. Dabei reagiert der Controller auf Ereignisse wie zum Beispiel Eingaben und wertet diese nach Bedarf aus.

Clientseitig versendet der Controller-Programmteil Anfragen an den Server. Dort werden per Controller die über HTTP empfangenen Anfragen ausgewertet und verarbeitet. Dann werden nach Bedarf neue Views generiert und das Datenmodell wird angepasst. Zur vereinfachten Aufbereitung der Views arbeitet die Software serverseitig mit Vorlagen, deren Platzhalter durch Daten des Models ersetzt werden.

Ein Beispiel für den Kommunikationsfluss in Anwendungen, die das Framework The-M-Project verwenden, finden Sie auf der Projekt-Homepage. Für die Aufbereitung von Views bietet The-M-Project unterschiedliche Objekte an, die Callback-Funktionalitäten für die interne Programmsteuerung bereitstellen.

Einstieg und Beispiele

Auf der Website von The-M-Project finden Sie Hinweise zur Framework-Einrichtung unter Mac OS X, Linux sowie Windows. Neben Node.js sollten Sie in jedem Fall Espresso ein-

Links zum Thema

- The-M-Project
<http://the-m-project.net>
- jQuery Mobile
<http://jquerymobile.com/gbs>
- Espresso-Sourcecode
<https://github.com/mwaylabs/Espresso>
- Node.js
<http://nodejs.org>
- Cygwin
www.cygwin.com
- WebStorm
www.jetbrains.com/webstorm
- Dokumentation
<http://panacodalabs.github.com/The-M-Docs/#home>

richten, das seinerseits das The-M-Project-Framework mitliefert.

Wie Sie das Framework für eine Hallo-Welt-Anwendung einsetzen, beschreibt ein kompaktes Tutorial. Prinzipiell legen Sie ein neues Projekt an und verwalten es über Espresso via Eingabeaufforderung sowie spezielle Tools und Befehle für die Kommandozeile.

Für die Projekt- und Quelltextbearbeitung benötigen Sie zusätzlich einen Editor, wobei die Entwickler die WebStorm-IDE empfehlen, und zwar nicht ohne Grund (**Bild 2**). Die Entwicklungsumgebung ist für Windows, Mac OS und auch Linux verfügbar.

Die IDE selbst ist up to date und bietet HTML5-, CSS- und JavaScript-Unterstützung sowie Syntax-Coloring, Autovervollständigung und integrierte, automatische Überprüfungs- und Korrekturfunktionen.

Während des Testlaufs werden alle Programme von Espresso im Speicher verwaltet. Espresso sorgt abschließend mit dem Build-Befehl über den integrierten Compiler für die physikalische Anlage im Dateisystem.

In der Dokumentation gibt es eine Einführung, Tutorials, Informationen zu den grundlegenden Konzepten, den mitgelieferten Utilities (Date, Environment, Internationalisierung, Logging und Mathematik), den Framework-Komponenten (Controller, ViewManager und Request) sowie Views. ■



Andreas Maslo

leitet das Ingenieurbüro IngES, das sich mit der Erstellung von EDV-Publikationen und der Software-Entwicklung befasst. Er ist als freier Journalist, Herausgeber, EDV-Berater, Fachbuchautor, Entwickler und Redakteur tätig.

AMAZON WEB SERVICES MIT PHP

Einfache Integration

Mit dem AWS PHP SDK lassen sich PHP-Anwendungen in AWS-Angebote integrieren.

Neben Microsoft Azure und der Google Cloud Platform zählt AWS zu den bedeutendsten internationalen Angeboten im Cloud Computing und ist zudem Pionier in diesem Bereich. Das Angebot der Amazon Web Services wird auch als Infrastructure as a Service (IaaS) bezeichnet, da es dem Endanwender eine einfach zu handhabende Cloud-Infrastruktur bietet (Bild 1).

Für den Zugriff auf die AWS gibt es zahlreiche SDKs – zum Beispiel Java, .NET, Python, Ruby, Go, C++ und eben auch PHP. Aktuell ist die Version 3 des SDK (genauer 3.12.1 am 7. Januar 2016). Das SDK ist auf GitHub unter der Adresse <https://github.com/aws/aws-sdk-php> erreichbar.

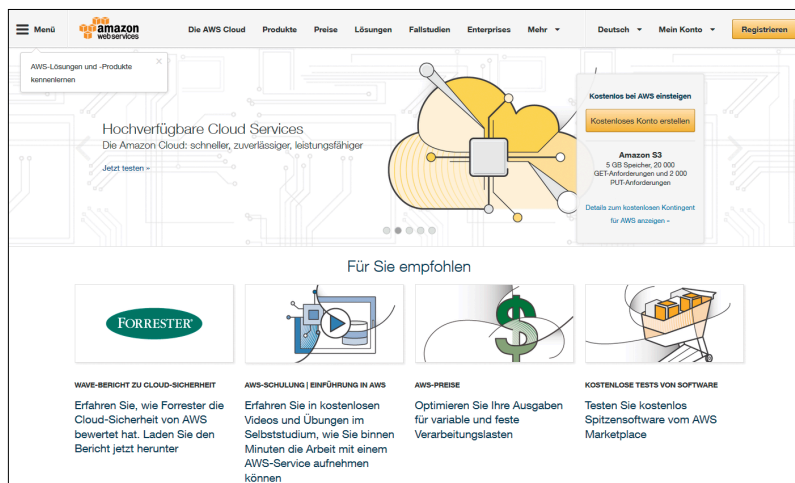
Auch wenn das SDK nahezu jedes verfügbare API abdeckt – und das sind fast 60 – gehen wir in dem vorliegenden Artikel näher auf den sogenannten Amazon S3 ein. Amazon Simple Storage Service (Amazon S3) bietet Entwicklern und IT-Teams sicheren, beständigen und hochgradig skalierbaren Objektspeicher. Damit können von überall im Internet beliebige Datenmengen gespeichert und abgerufen werden.

Systemanforderungen

Grundsätzlich benötigt das AWS PHP SDK eine PHP-Version ab 5.5.0. Optional sollte PHP über die OpenSSL-PHP-Extension verfügen, wenn man private Amazon CloudFront-URLs signieren möchte. Ferner müssen die PHP-Extensions *pcres*, *spl*, *json*, *dom*, *simplexml* und *zip* installiert sein.

Weitere PHP-Extensions sind optional, sollten aber installiert sein.

Auch wenn die Verwendung von cURL optional ist, so ist es doch unbedingt zu empfehlen. Der PHP Stream Wrapper (der anstelle von cURL verwendet wird, wenn jener fehlt) ist deutlich leistungsschwächer. Die Extension OPcache verbessert die Performance. Während der Entwicklung ist die Verwendung von Xdebug sinnvoll und gegebenenfalls auch notwendig. In produktiven Systemen sollte man zugunsten der Perfor-



AWS: Die Projektseite der Amazon Web Services (AWS) (Bild 1)

mance darauf verzichten. Unter dem URL <https://github.com/aws/aws-sdk-php/blob/master/compatibility-test.php> kann man einen Kompatibilitätstest herunterladen und durchführen. Zusätzlich zum Testen der minimalen Systemanforderungen prüft das SDK auch auf optionale Settings und macht Vorschläge, um die Performance des SDK zu erhöhen (Bild 2).

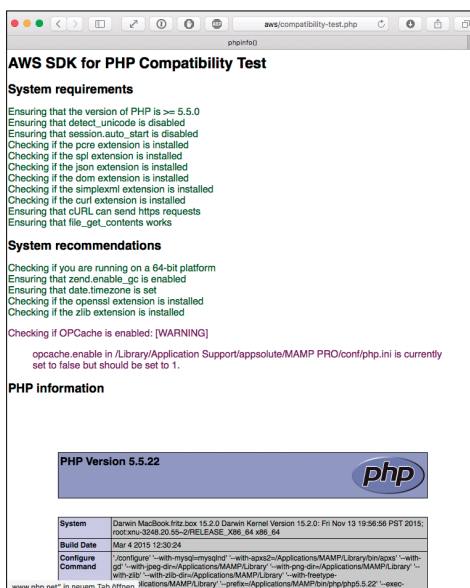
Der Kompatibilitätstest kann über der Browser, aber auch über die Kommandozeile ausgeführt werden. Im Browser werden die erfolgreichen Checks in Grün, die Warnungen in Purpur und die Fehler in Rot dargestellt. Die Kommandozeile wiederum verzichtet auf Farbangaben und stellt das Ergebnis textuell zeilenweise dar.

Installation per Composer

Composer ist das bevorzugte Tool, um das AWS SDK für PHP zu installieren, da Composer vorhandene Abhängigkeiten automatisch auflöst:

```
$ cd [DocumentRoot]
$ curl -sS https://getcomposer.org/installer | php
$ php composer.phar require aws/aws-sdk-php
```

Hier wird zuerst Composer als Datei vom Server geholt und in das Document Root (hier müssen Sie Ihr eige-



Kompatibilitätstest: So sieht die Ausgabe des Kompatibilitätstests aus (Bild 2)

nes Verzeichnis angeben) abgelegt. Nun wird per Composer-Befehl das SDK ermittelt; Abhängigkeiten werden aufgelöst und im Document Root im Verzeichnis *vendor* abgelegt. Um dann das SDK einzubinden, reicht es, den Composer-Autoloader anzusprechen:

```
require 'vendor/autoload.php';
```

Es ist zudem auch möglich, das SDK per Phar (PHP Archive) unter <http://docs.aws.amazon.com/aws-sdk-php/v3/download/aws.phar> herunterzuladen und zu referenzieren:

```
require '/path/to/aws/aws.phar';
```

Für die ZIP-Datei gehen Sie zur zugehörigen Download-Adresse unter <http://docs.aws.amazon.com/aws-sdk-php/v3/download/aws.zip>, entpacken die Datei im Document Root und laden den Autoloader:

```
require '/path/to/aws/aws-autoloader.zip';
```

Um mit dem AWS API zu arbeiten, benötigt man sogenannte Access Keys, die man erhält, sobald man sich bei Amazon registriert hat.

Access Keys werden hierzu an einen Account (Root-Account) gebunden und bestehen aus zwei Bestandteilen: Access Key ID und Secret Access Key.

Erstellung eines Accounts

Die Erstellung des Accounts ist grundsätzlich kostenlos, die Verwendung der Amazon Services für zwölf Monate ebenfalls. Anschließend fallen nutzungsabhängige Gebühren an. Im Detail sind die in **Tabelle 1** aufgeführten Nutzungen in den ersten zwölf Monaten kostenfrei. Gehen Sie zunächst auf die Adresse <https://portal.aws.amazon.com/gp/aws/developer/>

registration, um einen neuen AWS-Account anzulegen. Sollten Sie bereits über ein Amazon-Konto verfügen, so können Sie sich auch mit diesen Zugangsdaten einloggen.

Für spätere Abbuchungen muss nach der Eingabe der persönlichen Daten eine Kreditkarte hinterlegt werden. Zudem wird die Angabe einer Telefonnummer benötigt, damit das Amazon-System den User authentifizieren kann.

Wahl des Support-Levels

Dann können Sie das Support-Level wählen. Das Basis-Level ist kostenlos – weitere Level beginnen ab 49 US-Dollar pro Monat und beinhalten kurze Reaktionszeiten, direkten Chat, Rund-um-die-Uhr-Erreichbarkeit und mehr (**Bild 3**).

Das Erstellen der Access Keys und das Verwalten der Berechtigungen wird durch die AWS Identity and Access Management Konsole (IAM) erledigt. Gehen Sie hierzu auf den URL <https://console.aws.amazon.com/iam/home> und führen Sie dort die folgenden Schritte durch:

- **Gruppe anlegen:** Klicken Sie links auf *Groups* und dann oben auf *Create New Group*. Dann geben Sie einen Gruppennamen an (zum Beispiel *awstest*).
- **Policy:** Im nächsten Fenster *Attach Policy* suchen Sie im Filter-Feld nach *AmazonS3FullAccess* und wählen die Checkbox davor aus. Nun klicken Sie *Create Group* und bestätigen anschließend.
- **User anlegen:** Wählen Sie nun links *Users* aus und anschließend *Create New Users*. Dort wird nun ein User mittels *Create* angelegt (zum Beispiel *aws_demo*). Achten Sie dabei darauf, dass die Checkbox *Generate an access key for each user* angewählt ist.
- **Download Credentials:** Klicken Sie auf *Download Credentials*. Die Datei *credentials.csv* enthält nun Username, Access Key Id und Secret Access Key.
- **User zur Gruppe hinzufügen:** Um den User nun noch zur Gruppe hinzuzufügen, klicken Sie links auf *Groups*, wählen dann die Gruppe aus und fahren mit *Group Actions* fort. Dort wählen Sie die Option *Add Users to group* aus, selektieren einen User und klicken rechts unten auf *Add Users*.

Es gibt zahlreiche Möglichkeiten, die Credentials zu hinterlegen: über eine AWS-Credential-Datei, über Umgebungsvariablen, IAM-Rollen für EC2, über eine Configuration-Datei des Service Builders, mittels der Übergabe über eine Client-Factory-Methode oder über temporäre Credentials via AWS STS. Ausführliche Informationen hierzu finden Sie unter <http://docs.aws.amazon.com/aws-sdk-php/v2/guide/credentials.html>.

Wir entscheiden uns für die erstgenannte Methode und verwenden ►

Tabelle 1: Amazon Services

Services	Bedingungen
Amazon EC2 (Rechenkapazität)	750 Stunden monatliche Nutzung einer Linux-, RHEL- oder SLES t2.micro-Instance 750 Stunden monatliche Nutzung einer Windows t2.micro-Instance
Amazon EC2 Container Registry	500 MByte/Monat Speicher
Amazon S3 (Datenspeicherinfrastruktur)	5 GByte Standardspeicher 20.000 GET-Anforderungen 2000 PUT-Anforderungen
Amazon DynamoDB (NoSQL-Datenbank)	25 GByte Speicher 25 Schreibkapazitätseinheiten 25 Lesekapazitätseinheiten
IAWS IoT (Internet of Things)	250.000 Nachrichten (veröffentlicht und geliefert) pro Monat
Elastic Load Balancing (Load Balancer)	750 Stunden pro Monat 15 GByte Datenverarbeitung
Amazon CloudFront (Inhaltsbereitstellung)	50 GByte Übertragung ausgehender Daten 2.000.000 HTTP- oder HTTPS-Anforderungen

den eine Credentials-Datei mit dem Namen *credentials*, die wir im Ordner `~/.aws/` (unter Windows: `C:\Users\USER_NAME\.aws\`) mit dem folgenden Inhalt anlegen:

```
[default]
aws_access_key_id = YOUR_AWS_ACCESS_KEY_ID
aws_secret_access_key = YOUR_AWS_SECRET_ACCESS_KEY
```

In einer solchen Datei kann es verschiedene Abschnitte geben, die jeweils von einem String in eckigen Klammern eingeleitet werden. Auf diesen String referenzieren wir später. In obigen Beispiel gibt es den Abschnitt *default* mit den zugehörigen Daten.

Amazon Simple Storage Service (S3)

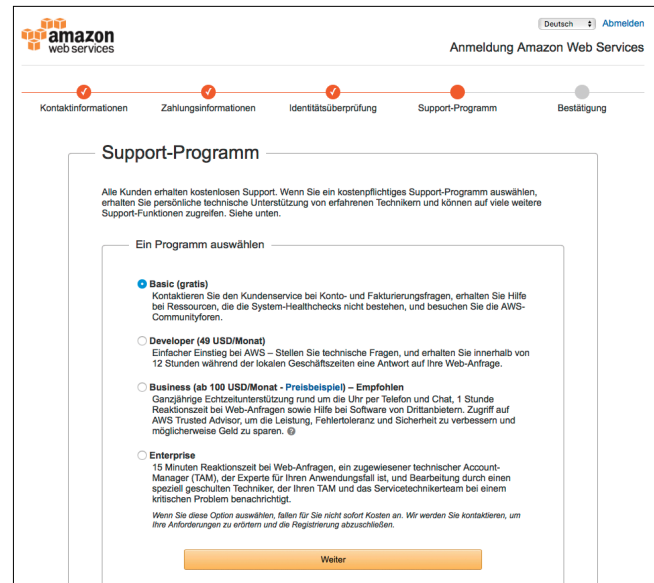
Als erstes Beispiel wollen wir den Amazon Simple Storage Service (S3) ansprechen. S3 ist ein Filehosting-Dienst, der theoretisch beliebig große Datenmengen speichern kann:

```
$s3 = new Aws\S3\S3Client([
    'version' => '2006-03-01',
    'region' => 'eu-central-1'
]);
```

Beim Erstellen des Clients können verschiedene Konfigurationsoptionen angegeben werden, die in **Tabelle 2** aufgelistet sind.

Buckets zur Verteilung und Optimierung von Daten

Bei der internen Entwicklung von Amazon S3 musste beachtet werden, dass die bei dem Speicherdienst hinterlegten Daten möglichst schnell weltweit verfügbar sein sollen und S3 auch bei unzähligen parallelen Zugriffen nicht zusammen-



Support-Level: Diese Support-Level bietet Amazon an (Bild 3)

bricht. Um eine einfache Verteilung und Optimierung gespeicherter Daten ermöglichen, hat Amazon daher die sogenannten Buckets erfunden. Ein Bucket ist eine Art Ordner mit einem – innerhalb von Amazon – eindeutigen Namen. Diesem wird ein eindeutiger URL `[bucketname].s3.amazonaws.com` zugeordnet. Wir legen daher zuerst selbst einen Bucket an:

```
$bucket = uniqid("php-sdk-sample-", true);
echo "Bucket mit dem Namen {$bucket} angelegt\n";
$s3->createBucket(['Bucket' => $bucket]);
$s3->waitUntil('BucketExists', ['Bucket' => $bucket]);
```

Der Methode *uniqid()* kann ein vorderer Namensbestandteil mitgegeben werden – hinten wird ein zufälliger und eindeutiger Hash generiert. Nun wird mittels *createBucket()* ein Bucket generiert und so lange gewartet, bis dieser zur Verfügung steht.

Objekte statt Dateien

Dateien innerhalb von Amazon S3 werden als *objects* (Objekte) bezeichnet und in Buckets gespeichert. Ein spezielles Objekt wird über seinen Schlüssel (Key) spezifiziert (zum Beispiel den Namen) und beinhaltet die Daten.

Wir wollen nun ein neues Objekt mit dem Namen *hello_world.txt* anlegen:

```
$key = 'hello_world.txt';
echo "Neues Objekt mit dem Schlüssel {$key} erstellt\n";
$s3->putObject([
```

Tabelle 2: Konfigurationsoptionen

Option	Erläuterung
<i>version (required)</i>	Hier muss die Version des API angegeben werden. Die letzte Version ist dabei <code>2006-03-01</code> . Es kann aber auch <i>latest</i> verwendet werden, um die wirklich letzte Version anzusprechen – dies wird aber in Produktivumgebungen nicht empfohlen
<i>region (required)</i>	Die geografische Region, in der die Daten gespeichert werden. Die möglichen Regionen sind hier zu finden: http://docs.aws.amazon.com/general/latest/gr/rande.html
<i>credentials</i>	Mit den Untereigenschaften <i>key</i> und <i>secret</i> kann das Credential auch hier angegeben werden
<i>debug</i>	Setzt man den Wert auf <i>true</i> , so werden Debug-Informationen ausgegeben
<i>delay</i>	Verzögerung in Millisekunden, bevor der Request gesendet wird
<i>progress</i>	Angabe einer Funktion, die den Fortschritt verarbeiten kann
<i>synchronous</i>	Angabe (in Boolean), ob der Request synchron ausgeführt werden soll
<i>timeout</i>	Timeout-Angabe in Sekunden
<i>retries</i>	Angabe, wie viele Verbindungsversuche durchgeführt werden sollen
<i>scheme</i>	Schema (http oder https)


```
'Bucket' => $bucket,
'Key'    => $key,
'Body'   => "Hello World!"
]);
```

Die `putObject()`-Methode hat noch zahlreiche weitere Parameter, die mit angegeben werden können. So kann zum Beispiel ein Source-File direkt angegeben werden, oder Meta-Daten, der Content-Type sowie Berechtigungen.

So kann beispielsweise über das Attribut `ACL` die Zugriffsberechtigung auf die Datei angegeben werden – zum Beispiel `public-read`, wenn diese öffentlich zugänglich sein soll. Legt man die Ressource mittels

```
$result = $s3->putObject([...])
```

an, dann kann man auf den URL per

```
$url = $result->get('ObjectUrl')
```

zugreifen. Um nun wieder an die Daten zu kommen, kann die Methode `getObject()` bemüht werden:

```
echo "Download des eben gespeicherten Objekts:\n";
$result = $s3->getObject([
    'Bucket' => $bucket,
    'Key'    => $key
]);
echo "\n---BEGIN---\n";
echo $result['Body'];
echo "\n---END---\n\n";
```

Der Amazon S3 Stream Wrapper erlaubt es, Daten zu speichern und darauf zuzugreifen, indem PHP-Funktionen wie `file_get_contents`, `fopen`, `copy`, `rename`, `unlink`, `mkdir`, `rm` ►

Links zum Thema

- AWS SDK for PHP – GitHub
<https://github.com/aws/aws-sdk-php>
- AWS SDK for PHP – Website
<http://aws.amazon.com/de/sdk-for-php>
- AWS-API-Dokumentation
<http://docs.aws.amazon.com/aws-sdk-php/v3/api>
- Buckets
<http://docs.aws.amazon.com/AmazonS3/latest/dev/BucketRestrictions.html>
- PutObjects-Methode
<http://docs.aws.amazon.com/aws-sdk-php/v3/api/api-s3-2006-03-01.html#putobject>
- Commands
<http://docs.aws.amazon.com/aws-sdk-php/v2/guide/feature-commands.html>

devbooks in 2016

.NET WPF
Moderne Desktop-
Anwendungen mit .NET

Neu!
02/
2016

ASP.NET
Microsofts neuer Weg
der Webentwicklung

Neu!
03/
2016

Data Science
Intelligente Verarbeitung
großer Datenmengen

Neu!
04/
2016

.NET-Architektur
So wird aus Anforderungen
eine perfekte Anwendung

Neu!
05/
2016

**Flexible Projektstrukturen
für .NET**

Neu!
06/
2016

Angular 2
Modulares Frontend
für den Browser

Neu!
07/
2016

**Digitale Revolution
in der IT**

Neu!
09/
2016

dir et cetera verwendet werden. In diesem Fall kann auf die Objekte über den URL `s3://[Bucket]/[Key]` zugegriffen werden.

Stream Wrapper

Das vorherige Beispiel, mittels Stream Wrapper realisiert, sieht dann wie folgt aus:

```
$s3->registerStreamWrapper();

$key2 = 'hello_again_world.txt';
echo "Erstellung eines zweiten
Objekts via Stream Wrapper mit
dem Schlüssel {$key2}\n";
file_put_contents
("s3://{$bucket}/{$key2}",
'Hello Again!');
```

```
echo "Download des eben
gespeicherten Objekts:\n";
$data = file_get_contents
("s3://{$bucket}/{$key2}");
```

```
echo "\n---BEGIN---\n";
echo $data;
echo "\n----END----\n\n";
```

Die Verwendung des Stream-Wrappers muss registriert werden. Nun kann mittels `file_put_contents` und `file_get_contents` gearbeitet werden.

Daten löschen

Um nun die Inhalte eines Buckets zu löschen, kann ein sogenannter Batch-Befehl abgesetzt werden, der die Inhalte löscht. Wenn der Bucket dann leer ist, kann dieser selbst mittels `deleteBucket()` gelöscht werden (Bild 4):

```
echo "Löschen aller Objekte im Bucket {$bucket}\n";
$batch = Aws\S3\BatchDelete::fromListObjects($s3,
['Bucket' => $bucket]);
$batch->delete();

echo "Löschen des Buckets {$bucket}\n";
$s3->deleteBucket(['Bucket' => $bucket]);
```

Die Fehlerbehandlung erfolgt über einen Try-Catch-Mechanismus, der die Fehlermeldung – sofern vorhanden – in ein Objekt vom Typ `S3-Exception` überführt:

```
try {
    $s3->createBucket(['Bucket' => $bucket]);
    $s3->waitUntil('BucketExists', ['Bucket' => $bucket]);
}
catch (S3Exception $e) {
```

Tabelle 3: Operationen (Auswahl)

Operation	Beschreibung
Commands	Operationen (wie <code>putObject</code> oder <code>getObject</code>) können über einen allgemeineren Befehl <code>getCommand</code> ausgeführt werden: <code>\$command = \$client->getCommand('GetObject', array(...))</code> . Dann ist es einseits möglich, per <code>\$command->set(Key,Value)</code> Attribute zu setzen. Mit <code>\$command->getResult()</code> wird der Befehl ausgeführt. Hat man mehrere Commands erstellt, so können diese über <code>\$s3->execute(\$commands)</code> ausgeführt werden
Objekte auflisten	Über <code>\$s3->listObjects([...])</code> können Objekte in einem Bucket aufgelistet werden. Will man alle Objekte ermitteln, so verwendet man die Methode <code>getListsObjectsIterator([...])</code> und kann dann in einer For-Each-Schleife darüber iterieren
Upload Directory	Es ist möglich, ein Verzeichnis in einen Bucket zu laden – der dafür notwendige Befehl lautet <code>\$s3->uploadDirectory(\$dir,\$bucket)</code> . Dabei muss das Verzeichnis noch nicht einmal lokal existieren, sondern kann auch ein Bucket sein, der mittels <code>s3://bucket_src</code> referenziert wird
Bucket herunterladen	Über den Befehl <code>\$s3->downloadBucket(\$dir,\$bucket)</code> kann man genau das Gegenteil realisieren – den Bucket in ein lokales Verzeichnis herunterzuladen

```
1 Bucket mit dem Namen php-sdk-sample-56900c0d1826d5.28544121 angelegt!
2 Neues Objekt mit dem Schlüssel hello_world.txt erstellt
3 Download des eben gespeicherten Objekts:
4
5 ---BEGIN---
6 Hello World!
7 ---END---
8
9 Erstellung eines zweiten Objekts via Stream Wrapper mit dem Schlüssel hello_again_world.txt
10 Download des eben gespeicherten Objekts:
11
12 ---BEGIN---
13 Hello Again!
14 ---END---
15
16 Löschen aller Objekte im Bucket php-sdk-sample-56900c0d1826d5.28544121
17 Löschen des Buckets php-sdk-sample-56900c0d1826d5.28544121
18
```

Output: Diese Ausgabe produziert ein Skript zum Löschen von Buckets (Bild 4)

```
echo $e->getMessage();
}
```

Es gibt noch zahlreiche weitere Operationen (Tabelle 3).

Fazit

Die Sammlung von Diensten, die Amazon unter den Namen Amazon Web Services (AWS) anbietet, ist sehr leistungsfähig. Sie zählen neben Microsoft Azure und der Google Cloud Plattform zu den Platzhirschen in Sachen Cloud Computing. Dass Amazon für den Zugriff auch ein PHP-API anbietet, ist angesichts der Verbreitung dieser Sprache konsequent. ■



Patrick Lobacher

ist Digital Native, Entwickler, Berater, Trainer, Coach und Autor zahlreicher Fachbücher und Fachartikel. Er ist Vorstandsvorsitzender der Pluswerk AG, die digitale Kommunikationslösungen konzipiert, umsetzt und betreut.



SMART DATA

Developer Conference

Big Data Analytics

- 18.04.2016 – Konferenz
 - 19.04.2016 – Workshops
- Novotel München City



Die SMART DATA Developer Conference macht Softwareentwickler mit den Herausforderungen von Big Data vertraut.

- In den Konferenz-Sessions erlangen Sie Wissen zu Speicherung, Analyse, Plattformen und Tools.
- Die Workshops bieten intensives Training in den Technologien Multi-Model-Database mit CQRS und OrientDB, Microsoft Azure und Entity Framework sowie Smart Development mit Enapso.



Kostenfreie
Webinare



Blog



Konferenz 2015
als Video

smart-data-developer.de #smartddc



SMARTDATADeveloperConference

Präsentiert von:

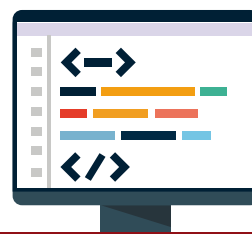


web & mobile
DEVELOPER

Veranstalter:



Neue
Mediengesellschaft
Ulm mbH



Konferenz-Programm: 18. April 2016

09.00 Uhr

Clouds in the Wind – Big und Smart Data in der Cloud und trotzdem beweglich • Michael Nolting

Smart und Big Data sind die Trendthemen heutzutage. Aber welche Technologie und welcher Anbieter sind für den speziellen Anwendungsfall am besten geeignet? Die entwickelte Entscheidungsmatrix soll den Zuhörer mündig machen, die im Markt verfügbaren Technologie-Stacks schnell zu vergleichen und die für den eigenen Anwendungsfall am besten geeigneten zu finden.

10.00 Uhr

Kaffeepause

10.30 Uhr

Smart
Data mit
.NET

Performance trotz Entity Framework

• André Krämer

Ist das Entity Framework wirklich langsam? JA! In dieser Session werden wir uns ansehen, was das Entity Framework langsam macht und was wir dagegen unternehmen können!

Best
Practices

Datenqualität für Entwickler

• Werner Keil

Big Data ohne Datenqualität wird chaotisch und bedeutungslos. Die Session bietet einen Überblick bekannter Standards, sowie deren Unterstützung durch Sprachen, APIs etc.

11.30 Uhr

Smart
Data mit
.NET

Datenpipeline-Komponenten für die SSIS

• Thomas Worm

Blog

Die SQL Server Integration Services ermöglichen die Integration von Datenbanken in vorhandene IT-Landschaften. Die Session behandelt Framework und Eigenheiten der SSIS.

Best
Practices

Daten entlang der Seidenstraße

• Jan Fellien

Die Session stellt die Multi-Model-Graphen-Datenbank OrientDB mit node.js vor. Sie kann in Documents wie in Graphen speichern, was die Anwendung im Alltag sehr vereinfacht.

12.30 Uhr

Mittagspause

13.30 Uhr

Smart
Data mit
.NET

Datenquellen mit F# Data und Deedle

• Stefan Lieser

Mit diesen beiden Open-Source-Bibliotheken ist der Zugriff auf Datenquellen sowie ihre Analyse und Aggregation auf einfache Weise möglich.

Best
Practices

Das Analytics Framework BRAIN Reflex

• Rupert Steffner und Jan Lendholt

Fokus der Session ist das Streaming Analytics Framework und dessen containerized Micro-Service-Architektur. Gezeigt werden Design Patterns und Use Cases.

14.30 Uhr

Smart
Data mit
.NET

Interaktive Analysen mit Apache Spark

• Olivia Klose und Sascha Dittmann

Bei interaktiven Datenabfragen schwächeln die meisten Big Data Technologien. Diese Session zeigt die wichtigsten Grundlagen der In-Memory Technologie anhand von Codebeispielen.

Best
Practices

Mehr als SQL Server in the Cloud

• Constantin Klein

Gewinnen Sie in dieser Session einen Überblick über die Microsoft Cloud Data Platform und die Möglichkeiten beider Bereiche „Data & Storage“ und „Analytics“.

15.30 Uhr

Kaffeepause

16.00 Uhr

Smart
Data mit
.NET

Looking for the meaning to our data

• Diego Poza

This talk is going to get deep into real world scenarios using Azure Machine Learning. Build from scratch a predictive model, comparing algorithms and adding custom modules.

Mensch &
Maschine

The UX of Data: Responsive Visualisierung

• Peter Rozek

Big Data ist nicht gleich Smart Data. Interaktive und skalierbare Datenvisualisierung ist der nächste Schritt, um Daten für den Nutzer nutzerfreundlich aufzubereiten.

17.00 Uhr
bis
18.00 Uhr

Smart
Data mit
.NET

Hey Cortana - let's talk about Analytics!

• Olivia Klose und Marcel Tilly

Ein Tiefflug über den Cortana Analytics Stack: Wie spielen die einzelnen Dienste, wie EventHub, Stream Analytics, Azure Data Lake, Azure Machine Learning und PowerBI zusammen?

Mensch &
Maschine

Ergonomische Datendarstellung

• Daniel Greitens

Daten sind nur so gut, wie dem Betrachter verständlich. Dieser Vortrag liefert psychologische Hintergründe und einen strukturierten Weg, wie Daten optimal aufbereitet werden können.



101101011100110110
1011010111001101101101



Workshops: 19. April 2016

- ! **Tipp:** Vorbereitendes kostenfreies Webinar zu jedem Workshop-Thema mit dem Trainer.
- Weitere Infos auf der Webseite: smart-data-developer.de

Workshop 1

CQRS und Multi-Model-DB – ein Herz und eine Seele

Jan Fellien

Uhrzeit: 09.00 – 18.00 Uhr

Lange habe ich nach einer guten Lösung für eine Datenbank gesucht, die sich ins CQRS Umfeld gut einfügt. Mit der OrientDB scheint die Suche ein Ende zu haben. Hiermit können Graphen und Documents gleichermaßen gespeichert (sprich ein Event Store) und die Read Models passgenau zu den Anforderungen abgelegt werden.

Der Workshop führt Sie zunächst in das CQRS-Paradigma ein und leitet Sie behutsam hinüber in die Datenspeicherung mit node.js und OrientDB.

Workshop 2

Smart Development mit Enapso

Alexander Schulze

Uhrzeit: 09.00 – 18.00 Uhr

Big-Data-Analysen erzeugen Wissen als Basis für smarte Maschinen, für Vorhersagen, für Empfehlungs- und Assistenz-Systeme. Doch smarte Technologien verbessern nicht nur Software als Ergebnis, sondern auch deren Entwicklung.

Dieser Workshop stellt Modelle, Abfragen und Regeln semantischer Ontologien mit OWL2, SPARQL und SWRL vor. Wir erläutern Machine-Learning, BPMN mit Micro-Services in der Cloud und wie intelligente Agenten als Assistenten zur Softwareentwicklung in Enapso Anforderungen und Wissen verbinden.

Workshop 3

Smart Data with Microsoft Azure

Sascha Dittmann, Olivia Klose Uhrzeit: 09.00 – 18.00 Uhr

In diesem Workshop werden wir anhand eines anschaulichen Szenarios die Analytics-Dienste der Azure-Datenplattform vorstellen und selber aufbauen, mit denen aus der Datenflut hilfreiche Erkenntnisse gewonnen werden können:

Azure Data Factory, HDInsight (Hadoop auf Azure), Stream Analytics und Machine Learning.

Beispielsweise wird erläutert, wie Erkenntnisse zum Einkaufsverhalten sowohl aus historischen Daten als auch in Echtzeit gewonnen und darauf basierend dem Kunden mithilfe von Machine Learning personalisierte Empfehlungen gegeben werden können.

Workshop 4

Enterpriseanwendungen mit Entity Framework 6

Christian Giesswein

Uhrzeit: 09.00 – 18.00 Uhr

Entity Framework gilt als DIE Lösung im Zusammenhang mit dem .NET-Framework und relationalen Datenbanken. Doch kaum geht eine Anwendung über das beliebte „Hello World“-Projekt hinaus, tauchen die wirklichen Fragen und Probleme erst auf. Plötzlich wird über die Performance gemeckert, oder den Komplexitätsgrad der Abfragen. Denn das Entity Framework ist bei falscher Verwendung alles andere als dankbar.

Anhand von „Best Practices“ verschiedener Projekte erfahren Sie, wo die Stolpersteine des Entity Framework liegen, wenn es um die Einbindung in komplexe Enterprise-Anwendungen geht, und erhalten wertvolle Tipps und Anleitungen, um diese zu überwinden.

Die Referenten der SMART DATA Developer Conference (u.a.):



Sascha Dittmann
Microsoft Deutschland GmbH



Jan Fellien,
devCrowd GmbH



Christian Giesswein
Giesswein Software-Solutions



Constantin Klein,
Freudenberg IT GmbH & Co. KG



Olivia Klose,
Microsoft Deutschland GmbH



Michael Nolting,
Sevenval Technologies GmbH



Jan-Hendrik Lendholt,
Otto (GmbH & Co KG)



Diego Poza,
Auth0 Inc.



Alexander Schulze,
Innotrade GmbH



Marcel Tilly,
Microsoft Research



Foto: Shutterstock / locrifa

ENTWICKLUNG VON MODULEN IN MAGENTO 2

Innovative Architektur

Magento 2 zeichnet sich durch moderne Entwicklungsprinzipien aus.

Mit der Unterstützung von PHP 5.5 beziehungsweise bald PHP 7, dem Datenbanksystem MySQL 5.6, der neuesten Webserver-Versionen von Nginx und Apache sowie dem Einsatz von Varnish und Redis Cache bereits in der CE-Version ist Magento 2.0 sicher eines der modernsten Shop-Software-Systeme.

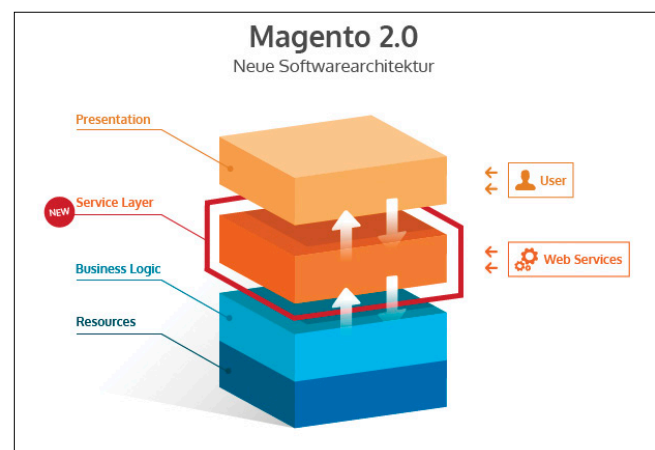
Magento 2 zeichnet sich aber nicht nur durch die unterstützten Technologien aus, sondern auch und vor allem durch moderne Entwicklungsprinzipien. Was hat sich für Entwickler geändert und welchen Nutzen versprechen die innovative Software-Architektur und die neuen Entwicklungsmuster?

Module in Magento 2

Eines vorweg: Magento 2 ist sicherlich komplexer als sein Vorgänger. Hat man sich aber erstmal an die veränderte Benennung der Magento-Module und die neue Verzeichnisstruktur gewöhnt, eröffnet Magento 2 eine Vielzahl von Vorteilen, die die Entwicklungsarbeit, aber auch die Pflege des Systems wesentlich vereinfachen.

Von einer etwas höheren Abstraktionsebene aus betrachtet ist Magento 2 noch stärker einem mehrschichtigen Soft-

waremodell verpflichtet als Magento 1.x und wurde um einen Service Layer erweitert (Bild 1). Etwas konkreter geht es um den Einsatz von Service Contracts.



Softwaremodell: Magento 2 wurde um einen Service Layer erweitert (Bild 1)

Das sind PHP-Interfaces nach einem bestimmten Muster oder Design Pattern. Sie regeln die Art und Weise der Kommunikation zwischen Modulen. Der Austausch soll ausschließlich über diese vordefinierten Schnittstellen (Data und Service Interfaces) erfolgen, somit die Datenintegrität schützen sowie die Businesslogik der Module voneinander getrennt halten.

Die Service Contracts stellen dazu anderen Modulen und Extensions genau jene Funktionalitäten beziehungsweise Methoden zur Verfügung, die sie für den definierten Fall auch benötigen.

Durch die Einführung der Service Contracts steigt die Stabilität in der Kommunikation und der Zusammenarbeit von Modulen untereinander. Außerdem verbessert sich dadurch die Updatefähigkeit und Flexibilität des Gesamtsystems (Bild 2). Vorausgesetzt natürlich, man hält sich an das Prinzip.

Konfiguration von Modulen

Im Bereich der Modul-Entwicklung gibt es etwa mit der Erweiterung der XML-Struktur eine deutliche Verbesserung. Anpassungen erfolgen nicht mehr nur über die globalen Konfigurationsdateien, sondern es lassen sich für jedes Modul eigene XML-Dateien erstellen.

Damit steigt zwar deren Anzahl, andererseits ist es so möglich, ganz gezielt für jeden Einsatzzweck XML-Strukturen zu erstellen und über eigene XML-Schema-Definitionen (XSD) auf Korrektheit zu validieren. Die modulspezifischen XML-Dateien sind ein weiterer Baustein in der Modularisierung von Magento und sorgen für eine einfachere Pflege des Shop-Systems.

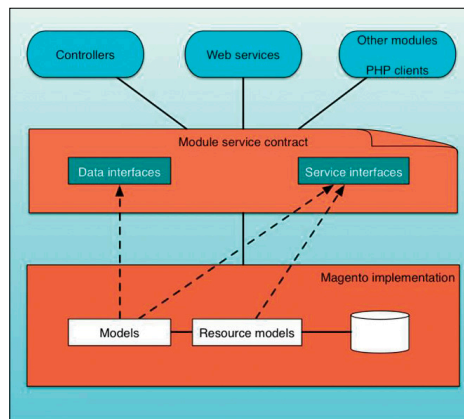
Portierung zu Magento 2

Magento-1.x-Module lassen sich nicht eins zu eins auf Magento 2 migrieren. Dankbarerweise liefert Magento mit einem automatischen Modul-Migrator aber ein brauchbares Tool, das den Entwicklern einen Teil ihrer Arbeit abnimmt. Aber eben nur einen Teil – Nachbesserungen sind dennoch nötig.

Wer allerdings bei der Magento-1-Entwicklung auf eine hohe Codequalität und einen sauberen Aufbau seiner Module geachtet hat, wird nur mit einem minimalen Mehraufwand rechnen müssen. Manche Befürchtungen der ersten Stunde, die Portierung von Modulen werde zum finanziellen Risiko, haben sich ganz sicher nicht bewahrheitet.

Hat man bereits in den bestehenden Magento-Modulen Dependency-Injection als Muster verwendet, wird die Migration nochmals einfacher.

Auch bei Dependency Injection handelt es sich um ein Design Pattern. Prinzipiell geht es darum, die Abhängigkeiten (Dependencies) zwischen Modulen oder Klassen von Modulen nicht als einen Bestandteil der Module selbst zu definie-



Service Contracts verbessern die Updatefähigkeit und Flexibilität des Gesamtsystems (Bild 2)

ren, sondern an eine zentrale Stelle zu delegieren.

Damit werden die Abhängigkeiten zwischen den einzelnen Modulen quasi flexibel steuerbar, beziehungsweise die Module werden unabhängiger. Dieser Mechanismus wurde nun auch in Magento 2 umgesetzt, und die Steuerung erfolgt über eine modulspezifische XML-Datei (*di.xml*).

Die durchgängige Einführung von Dependency Injections in Magento 2 war ein weiterer wichtiger und notwendiger Schritt für die Optimierung der Magento-Codebasis. Neben der Codequalität verbessern sich zudem die Testbarkeit und die Erweiterbarkeit einer Magento-Plattform. Letzteres, da sich Abhängigkeiten einfach austauschen lassen.

Die Umsetzung des Mechanismus ist recht gut gelungen, wenn er auch noch kleinere Schwächen aufweist. Noch werden die Definitionen der Plug-ins und der Abhängigkeiten in der *di.xml* vermischt.

Plug-ins in Magento 2

Mit Plug-ins, oder noch genauer Interceptoren, erhält man in Magento 2 die Möglichkeit, gezielt einzelne öffentliche Methoden einer Klasse zu überschreiben, ohne dass es zu den von Magento 1 bekannten Rewrite-Konflikten kommt. In Magento 1 konnte bisher immer nur ein Modul eine Methode überschreiben.

Über Plug-ins können nun mehrere Module die gleiche Methode erweitern und sich dabei vor, nach oder um die betroffene Methode schalten. Allerdings werden die Plug-in-Definitionen ebenfalls in der *di.xml* vorgenommen. Bei umfangreichen Modulen kann dies dazu führen, dass eine *di.xml* sehr viele Definitionen umfasst und die Trennung zwischen den Dependency-Injections und den Plug-in-Definitionen schwer fällt.

Fazit

Diese Kleinigkeiten ändern aber nichts mehr am äußerst positiven Gesamteindruck der Software. Vollzieht man den Paradigmenwechsel erfolgreich nach, fällt die Programmierung in Magento 2 wirklich leicht und macht dann noch mehr Spaß als in Magento 1. ■



Matthias Walter

betreut als Senior Lead Magento-Entwickler zahlreiche Projekte bei Netz98 und ist einer der wenigen Magento-2-Spezialisten in Europa.
www.netz98.de



Foto: Shutterstock / Macrovector, Jemastock

MOBILITY, CLOUD COMPUTING, SOCIAL BUSINESS

Arbeiten im Team

Menschen kommunizieren und arbeiten auf ganz neue Weise miteinander.

Die Art und Weise, wie wir in Zukunft arbeiten, wird sich massiv ändern. Darin sind sich die über 170 Quellen einig, die das Institut für Arbeitsdesign und Zukunftstechnologien (ifaz) für seine Metastudie »Wie wir morgen arbeiten ...« ausgewertet hat. Dabei ändert sich nicht nur die Arbeitswelt des Einzelnen, sondern zwangsläufig auch, wie er mit anderen in Teams, Projekten, Abteilungen und Organisationen zusammenarbeitet.

»Der Mitarbeiter wird vom Informationsempfänger zum aktiven Teilnehmer am Wissensmanagement«, so das Fazit einer Trendstudie der Hochschule Mainz (www.trends-in-der-it.de). »Im Mittelpunkt der Veränderung von Büroarbeit stehen die Menschen, die flexibles, selbstbestimmtes und weitgehend hierarchiefreies Arbeiten fordern«, ergänzt Diana Heinrichs, Communications Manager Digital Workstyle bei Microsoft Deutschland.

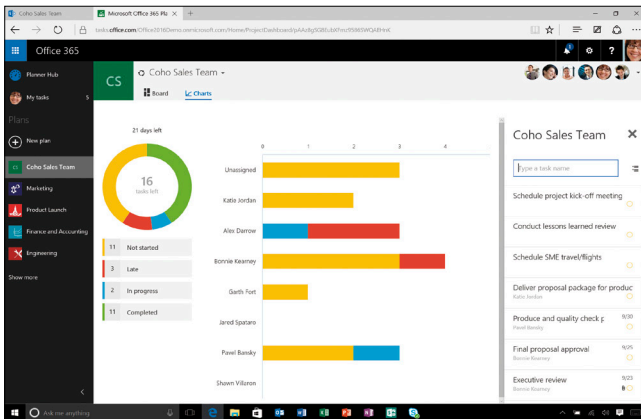
Selbst die Unternehmensgrenzen lösen sich auf, meint Stefan Waldhauser, Director Product Marketing beim ECM-/BPM-Spezialisten Alfresco, der mit Alfresco One eine Lösung

zur sicheren Zusammenarbeit mit externen Partnern bietet: »Unternehmen sind keine autonomen Einheiten mehr, sondern mehr und mehr Teil eines Netzwerks, zu dem weitere Unternehmen gehören, interne Mitarbeiter ebenso wie externe Partner, Lieferanten, Händler und Kunden.«

»Neue Formen der Zusammenarbeit eröffnen Unternehmen neue Möglichkeiten in vielen Bereichen«, findet auch Thomas Gmeiner, Senior Marketing Specialist T-Systems International. »Mit Hilfe der neuen Collaboration-Tools können leichter Experten außerhalb des Unternehmens in die Produktentwicklung einbezogen werden.«

Zunehmende Digitalisierung

Ein Hauptgrund für diesen Wandel ist laut der ifaz-Analyse die zunehmende Digitalisierung, die alle Lebensbereiche durchdringt. Insbesondere die gestiegene Mobilität durch Smartphones und Tablets, aber auch durch Cloud-Anwendungen hat die Zusammenarbeit von den Fesseln der physischen Präsenz befreit.



Office 365: Microsoft hat in seine Office-Cloud viele Collaboration-Funktionen integriert.

»War Arbeit ehemals ein Ort, an den wir gehen – ist es heute etwas, das wir machen, unabhängig von Zeit, Ort und Endgerät«, sagt Sascha Hirschhoff, Director Systems Engineers Central Europe bei Polycom. Der Manager glaubt fest an das Ende der Präsenzpflcht: »In Deutschland arbeiten laut einer Studie im Auftrag von Polycom schon heute 34 Prozent der Mitarbeiter mindestens einmal pro Woche von zu Hause aus. Darüber hinaus werden bis zum Jahr 2022 etwa 60 Prozent der Mitarbeiter, die bisher regelmäßig vom Büro aus gearbeitet haben, ihren Arbeitsplatz nach Hause verlegen.«

Technologische Herausforderungen

Hersteller und Diensteanbieter sehen hier vor allem die technologischen Herausforderungen: »Die zunehmende Mobilität von Mitarbeitern verlangt, dass insbesondere die gut ausgebildeten Fachkräfte mit Collaboration-Tools und -Plattformen ausgestattet werden müssen, damit sie trotz ihrer Mobilität in Prozesse der Zusammenarbeit vollumfänglich und unterbrechungsfrei eingebunden werden können«, sagt beispielsweise T-Systems-Manager Gmeiner.

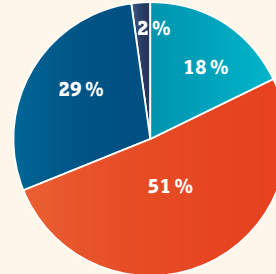
»Eine moderne Collaboration-Lösung muss die Teilnahme an Meetings wie Online- oder Videokonferenzen von überall und mit jedem üblichen Gerät ermöglichen«, pflichtet ihm Anton Michael Döschl bei, Leiter Collaboration Architektur bei Cisco Deutschland.

Ganz so einfach sei die Sache allerdings nicht, sagt Frank Heuer, Senior Advisor bei der Experton Group: »Der Mensch ist ein soziales Wesen. Für den alltäglichen Austausch ist die technisch gestützte Kommunikation sehr hilfreich, für

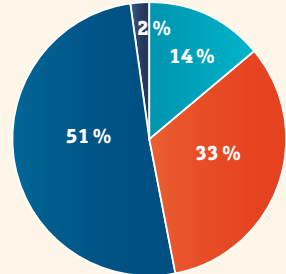
Collaboration wird wichtiger

In den Fachabteilungen wächst die Einsicht, dass Collaboration für den Geschäftserfolg an Bedeutung gewonnen hat.

Produktivität der Mitarbeiter



Flexibilität und Work-Life-Balance



● Stark verbessert ● Leicht verbessert
● Nicht verändert ● Leicht verschlechtert

Quelle: Citrix/Techvalidate

das emotionale Einbinden in ein Team oder ein Unternehmen ist die persönliche Präsenz dagegen unverzichtbar.« Besonders bei Diskussionen zu schwierigen oder kontroversen Themen und beim sozialen Networking seien direkte Kontakte wichtig, so Heuer weiter. »Das kann Technologie nicht ersetzen.«

Der Tod der E-Mail

Neben dem Trend zu mehr Unabhängigkeit prägen vor allem Entwicklungen, die aus dem privaten Bereich in die Unternehmen schwappen, die Zusammenarbeit der Zukunft. Blogs, Wikis, Chats und soziale Plattformen sollen die Zusammenarbeit revolutionieren, von Enterprise 2.0 und Social Business ist die Rede. Viele propagieren den Tod der E-Mail, die in erster Linie bei den sogenannten Digital Natives in der Mitarbeiterschaft als gestrig gilt.

»Digital Natives prägen mit ihren Präferenzen und Erfahrungen auch die Erwartungshaltung bezüglich der Zusammenarbeit in Unternehmen«, sagt T-Systems-Manager Gmeiner. »Die jüngeren Generationen sind mit Facebook und Twitter groß geworden und wollen auch im Businessumfeld solche Tools nutzen«, gibt ihm Tobias Loewenhofer, Enterprise Mobility Consultant beim IT-Dienstleister CEMA, recht. Die E-Mail-Flut nerve in der Tat, findet Anton Michael Döschl von Cisco »vor allem aufgrund der Schwierigkeit, in E-Mails Konversationszusammenhänge und Kommunikationsflüsse zu erkennen.«

Experton-Analyst Heuer ist hingegen skeptisch, was die Prophezeiungen vom baldigen Ende der E-Mail angeht: »Wir sehen das nicht so schwarz-weiß.« Beide, die alten und die neuen Kommunikationsformen, hätten Vor- und Nachteile und würden nebeneinander Bestand haben. »Wenn man eine direkte Antwort haben möchte, greift man zum Telefon oder zur E-Mail, in großen Teams sind dage- ►

»Die Unternehmenskultur ist ein ganz wichtiger Faktor, wenn es darum geht, neue Mechanismen der Zusammenarbeit einzuführen.«

Frank Heuer
Senior Advisor
www.experton-group.de



gen soziale Plattformen zur Organisation und Kommunikation sicher sinnvoller, als E-Mail-Schlachten mit Riesenverteilern zu führen.«

Wie schwierig der Abschied von der E-Mail ist, musste der IT-Dienstleister Atos erfahren, der 2011 eine Zero-E-Mail-Initiative startete. Bis 2013 wollte das Unternehmen in der internen Kommunikation komplett auf elektronische Post verzichten. Die Investitionen in das Lieblingsprojekt von Atos-Chef Thierry Breton waren enorm. Nach Schätzungen von Gartner gab der Dienstleister 500-mal mehr für sein Collaboration-Programm aus als der Durchschnitt, zehn Prozent der Manager-Boni wurden mit der Zielerreichung verknüpft. Dennoch verfehlte das Unternehmen die Vorgaben, konnte aber immerhin bis Ende 2013 das interne E-Mail-Aufkommen um 60 Prozent reduzieren.

E-Mail oder Social Media?

Frank Heuer glaubt, dass sich die Frage E-Mail oder Social Media ohnehin erledigen wird: »Wir sehen hier einen Trend zur Konvergenz, und das spiegelt der Markt ja auch wider.« Der Analyst nennt als Beispiel Verse von IBM. Das Tool, das derzeit nur als Cloud-Anwendung sowie als App für iOS und Android verfügbar ist, fasst unterschiedliche Kommunikationswege unter einer Oberfläche zusammen.

Verse soll mit der Zeit aus dem Verhalten des Anwenders lernen, welche Kontakte und Konversationen besonders wichtig für ihn sind, und diese prominent platzieren. Klickt man in Verse auf einen Kontakt, zeigt die Oberfläche alle damit verbundenen Aktionen wie Meetings, Aufgaben, geteilte Dateien und eben auch E-Mails an.

Nachrichten, die nicht direkt beantwortet werden können, lassen sich mit einer Erinnerung zur Wiedervorlage versehen. Umgekehrt kann für eine versendete E-Mail ein Termin definiert werden, bis wann man eine Antwort erwartet.



»Eine moderne Collaboration-Lösung muss die Teilnahme an Online- und Videokonferenzen von überall und mit jedem üblichen Gerät ermöglichen.«

Anton Michael Döschl

Leiter Collaboration Architektur
www.cisco.de

Foto: Cisco

Standardmäßig teamfähig

Neben Neuentwicklungen wie IBM Verse treibt vor allem die Standard-Bürosoftware den Team-Gedanken vorwärts, allen voran Microsoft Office in seiner aktuellen Abo- und Cloud-Form. »Office 365 – insbesondere mit Yammer und Skype for Business – macht produktives Arbeiten an jedem Ort und zu jeder Zeit möglich«, verspricht Microsoft-Managerin Diana Heinrichs. »Bei Microsoft Office 365 können mehrere Personen gleichzeitig an einem Dokument arbeiten. Durch die Versionierung ist nachvollziehbar, welcher Mitarbeiter wann was geändert hat«, ergänzt CEMA-Manager Loewenhofer.

Je nach Abo-Plan sind im Lieferumfang bereits einige Tools zur Zusammenarbeit enthalten, beispielsweise der mit einem TByte pro Nutzer üppig bemessene Online-Speicher OneDrive, der das Teilen von Dokumenten intern und mit externen Partnern erlaubt, eine Plattform für HD-Videokonferenzen mit bis zu 250 Teilnehmern, Teamfunktionen für Kalender sowie die App Skype for Business, die Chats erlaubt und Präsenzinformationen zur Verfügung stellt.

Für Oliver Kretzschmar, Leiter des Competence Centers Applications & Endpoint bei Orbit IT-Solutions, ist die Office-365-Komponente OneNote, mit der sich auf verschiedenen Endgeräten Notizen erfassen lassen, mittlerweile zum Lieblings-Collaboration-Tool geworden: »Ist das Notizbuch in

Collaboration-Lösungen (Auswahl)

Produkt	Internet	Lizenzmodell
Asana	http://asana.com	Benutzerbasiert; kostenlos bis 15 Benutzer; Premium-Version ab 21 US-Dollar pro Monat für 5 Benutzer
Basecamp	http://basecamp.com	Projektbasiert; 20 US-Dollar pro Monat bei 10 aktiven Projekten
Bitrix24	www.bitrix24.de	Benutzerbasiert; kostenlos bis 12 Benutzer; bis 24 Benutzer 39 Euro pro Monat
Atlassian HipChat	www.hipchat.com	Benutzerbasiert; Basisversion kostenlos; Plus-Version 2 US-Dollar pro Monat und Benutzer
Jive	www.jivesoftware.com/de	Paket- und benutzerbasiert; ab 2 US-Dollar pro Monat und Benutzer
Podio	http://podio.com	Benutzerbasiert; ab 9 US-Dollar pro Monat und Benutzer
Redbooth	http://redbooth.com/de	Benutzerbasiert; Professional-Variante 5 US-Dollar pro Monat und Benutzer; Business-Variante 15 Dollar pro Monat und Benutzer
Slack	http://slack.com	Benutzerbasiert; Basisversion kostenlos; Standardversion 6,67 US-Dollar pro Monat und Benutzer bei jährlicher Bezahlung.
Stackfield	www.stackfield.com/de	Unternehmensbasiert; 89 Euro pro Monat
Trello	http://trello.com	Benutzerbasiert; Basisversion kostenlos; Business-Class-Version 8,33 US-Dollar bei jährlicher Bezahlung

SharePoint gespeichert, können sehr leicht andere Kollegen an den Notizen teilhaben. Einmal daran gewöhnt, ist es schwer, sich ein Arbeiten ohne OneNote vorzustellen.«

SharePoint, Microsofts klassische Collaboration-Plattform ist allerdings nur in den Enterprise-Plänen von Office 365 enthalten.

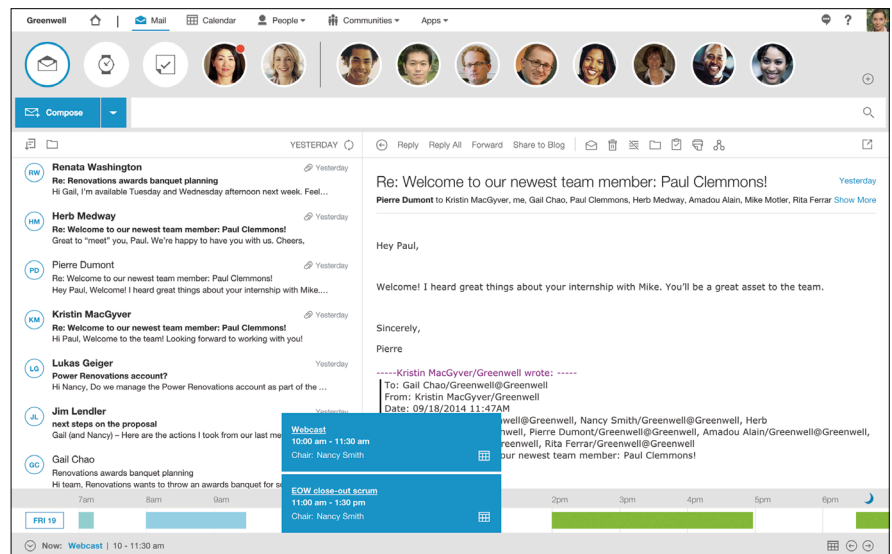
Das soziale Netzwerk Yammer, das Microsoft 2012 für 1,2 Milliarden Dollar gekauft hatte, ist dagegen auch Bestandteil der Office-365-Versionen Business Essentials und Business Premium, die für kleine und mittelständische Unternehmen mit maximal 300 Benutzern gedacht sind. Yammer ist eine Art Facebook für Unternehmen.

Kommunikation in Teams

Die Kommunikation wird in Gruppen organisiert, in denen Teams zusammenarbeiten, kommunizieren und Dokumente teilen können. Infos, Dateien oder Umfragen lassen sich posten, persönliche Nachrichten und Unterhaltungen sind ebenso möglich wie die Kommunikation mit externen Partnern. Teams können gemeinsam an Inhalten arbeiten, wobei Änderungen in Echtzeit bei allen Beteiligten zu sehen sind. Für eine Übersicht lassen sich die Informationsströme zwischen Teamkollegen im Kontext von Konversationen und geteilten Dokumenten grafisch darstellen.

Eine ähnliche Funktion namens Delve gibt es im Übrigen Office-365-weit. Sie basiert auf Office Graph, das kontinuierlich im Hintergrund die Aktivitäten des Nutzers und seiner Kollegen analysiert. Wenn Mitarbeiter beispielsweise das gleiche Dokument ändern oder anzeigen, ist dies für Office Graph ein Zeichen, dass sie wahrscheinlich zusammenarbeiten. Wer mit wem per E-Mail kommuniziert, für wen Dokumente freigegeben werden und wer denselben Vorgesetzten hat, sind andere Hinweise, auf denen Office Graph ein Beziehungsgeflecht aufbaut.

Auch Adobes Cloud-Angebote integrieren immer mehr Team-Funktionen. In Photoshop und Illustrator lassen sich Stile und grafische Elemente in Bibliotheken teilen sowie Designs gemeinsam entwerfen und verändern. Bis zu 32 Collaboration-Projekte sind gleichzeitig möglich. Jedes Team-Projekt kann bis zu 100 Mitwirkende umfassen.



Verse: Mit der Lösung von IBM soll sich die E-Mail-Flut besser beherrschen lassen.

Im Bereich Videoproduktion bietet der Hersteller mit Adobe Anywhere eine Plattform für die Zusammenarbeit, die in Premiere Pro, Prelude und Adobe After Effects integriert ist. Sie ermöglicht es Teams, gemeinsam Filmprojekte zu entwerfen, zu erstellen und zu editieren. Der Zugriff auf die zentral gehosteten Dateien soll über Standardnetzwerke möglich sein und keine zusätzliche Hard- oder Software erfordern. Selbst per Mobilfunk sollen sich Filmdateien prüfen und editieren lassen, wenn auch nur in niedriger Auflösung.

Die Lösungen der Spezialisten

Trotz der umfassenden Angebote der Standardsoftware-Hersteller, allen voran Microsoft, tummeln sich auch eine Menge Spezialisten auf dem Collaboration-Markt. Vor allem Hersteller aus dem Telekommunikationssegment haben ihre Suiten schon seit Längerem um Funktionen für die Zusammenarbeit erweitert und bieten diese als Unified Communications & Collaboration (UCC) an. Zu nennen wären etwa Alcatel-Lucent, Avaya, Cisco oder Unify.

Cisco hat zum Beispiel im März dieses Jahres mit Spark eine cloudbasierte Lösung für Chats, Videoanrufe und Screen-sharing auf den Markt gebracht, bietet mit WebEx einen allseits bekannten Webkonferenzdienst und bindet mit Jabber mobile Endgeräte unter iOS, Android, BlackBerry oder Windows Phone in eine Kommunikationsinfrastruktur ein.

Die Software-as-a-Service-Lösung Circuit von Unify ist angetreten, um die Grenzen zwischen verschiedenen Kanälen, Endgeräten und Anwendungen aufzuheben und auf einer einheitlichen Bedienoberfläche zusammenzuführen. Als browserbasierte Anwendung setzt Circuit – wie auch Spark – auf WebRTC, einen offenen Standard, der eine sichere Audio- und Videokommunikation ermöglicht und bereits in die Browser Chrome, Firefox und Opera integriert ist.

»Mit WebRTC hat sich in den letzten Monaten ein neuer Standard etabliert, der völlig ►

»War Arbeit ehemals ein Ort, an den wir gehen – ist es heute etwas, das wir machen, unabhängig von Zeit, Ort und Endgerät.«

Sascha Hirschhoff

Director Systems Engineers Central Europe
www.polycom.de



»Unternehmen müssen sich komplett ändern«

web & mobile developer: Sie haben in Ihrer Metastudie »Wie wir morgen arbeiten ...« über 170 Quellen ausgewertet, die sich mit der Zukunft der Arbeit beschäftigen. Was verändert aktuell die Zusammenarbeit in Unternehmen besonders stark?

Stefan Birk: Es haben sich vier Megatrends herauskristallisiert, die Digitalisierung, der demografische Wandel, die Globalisierung und der Wertewandel. Derzeit prägt besonders die Digitalisierung die Arbeitswelt. Sie stellt die gesamten Prozesse in Unternehmen infrage und verändert die Mobilität, zeitlich und räumlich. Neben diesen Megatrends haben wir acht Trends definiert, die mit den Megatrends zusammenhängen, wenn oft auch nur sehr indirekt. Zwei davon sind aus unserer Sicht aktuell besonders bedeutsam: Da sind zum einen neue Technologien wie elektronische Assistenten oder Robotik, die durchaus auch hochqualifizierte Wissensarbeiter durch Maschinen ersetzen können. Das zweite ist das Thema Arbeitskultur. Sie gerät immer stärker in den Fokus der Unternehmen und ist sehr wesentlich für alle modernen Formen der Arbeit.

web & mobile developer: Viele sind von den modernen Möglichkeiten der Zusammenarbeit auch überfordert. Die Always-on-Mentalität führt oft zu Überlastung bis hin zum Burn-out. Wie muss eine Arbeitskultur beschaffen sein, um dem entgegenzuwirken?

Birk: Die Frage ist doch: Sind die Unternehmen überhaupt dafür zuständig, ihre Angestellten auf diesem Gebiet zu schulen? Auf der einen Seite wollen Mitarbeiter immer ernst genommen und in ihren Freiheiten nicht beschränkt werden, auf der anderen soll sich der Arbeitgeber darum kümmern, wenn seine Angestellten abends ihre Mobiltelefone nicht weglegen können. Wenn Unternehmen aber tatsächlich dafür zuständig sind, die Mitarbeiter vor dieser Überforderung zu schützen, stellt sich gleich eine weitere Frage: Wie sollen sie das tun? Zwangsmaßnahmen wie das Abschalten der Mail-Server um 18 Uhr helfen wenig und beleidigen eher die Intelligenz der Mitarbeiter. Die sind ja nicht dämlich und umgehen solche Sperren locker, was die Unternehmen womöglich implizit sogar von ihnen erwarten. In anderen Ländern, etwa in Skandinavien, gibt es diese seltsamen Diskussionen gar nicht. Dort ist es völlig normal, wenn man mal früher nach Hause geht. Und genauso normal ist es, abends um 21 Uhr von zu Hause aus noch einmal die Mails zu checken. Wenn Sie aber in einer Zwangskultur leben, in der man den Arbeitsplatz nicht vor 20 Uhr verlassen darf, wenn man Karriere machen will, und um 4:30 Uhr die Mails des Chefs beantworten muss, weil der eben gern früh aufsteht, dann haben Sie diesen Druck der ständigen Erreichbarkeit. Der kommt aber aus einer anderen Ecke, die Technologie ist nicht Schuld daran. Sie können mit Technologieschulungen nichts erreichen, wenn gleichzeitig das Dogma existiert: Wer 16 Stunden arbeitet, hat gewonnen.

web & mobile developer: Die Verantwortung liegt also beim Einzelnen?

Birk: Ja und nein. Natürlich ist jeder für sein Verhalten selbst verantwortlich. Auf der anderen Seite müssen Unternehmen tiefgreifende Veränderungen in der Kultur vornehmen und die häufig noch vorherrschenden superhierarchischen Strukturen aufbrechen, wenn sie tatsächlich ihre oft und gern herausgestellten sozialen und kulturellen Ziele erreichen wollen. Es hilft beispielsweise wenig, offiziell die Elternzeit für Väter zu propagieren, wenn das in den Abteilungen ein Karrierekiller ist. Unsere heutige Arbeitskultur ist zu sehr im Alten verhaftet. Die neuen Möglichkeiten und die alte Welt passen nicht zusammen, was zwangsläufig zum Missbrauch des technologisch Möglichen führt. Wenn wir also noch einmal die Frage aufgreifen, wie Unternehmen ihre Mitarbeiter vor Überforderung schützen können, dann lautet die kurze Antwort: Ihr müsst euch komplett ändern, wenn ihr diese Effekte vermeiden wollt.

web & mobile developer: Wir haben über die negativen Aspekte der neuen Technologien gesprochen. Welche Vorteile bringen sie denn dem Einzelnen und dem Unternehmen?

Birk: Das ist natürlich unterschiedlich. Die Individualisierung des Arbeitsumfelds ist durch die neuen Technologien so groß

»Für die Betroffenen muss etwas herausspringen.«

wie nie zuvor. Man kann sich aussuchen, wann, wo und mit wem man arbeitet. Man kann viel an Autonomie gewinnen, wenn man die neuen Arbeitsmittel richtig einsetzt. Auf der anderen Seite stehen die Unternehmen, die vor allem gern Prozesse optimieren und die Effizienz verbessern. Das kann durchaus mit den Wünschen des Einzelnen kollidieren.

web & mobile developer: Und wer gewinnt dieses Spiel?

Birk: Die Antwort darauf ist längst nicht mehr so einfach wie noch vor ein paar Jahren. Da war ja alles besprochen – die Arbeitgeber bestimmen und die Arbeitnehmer fügen sich. Heute stellen Bewerber Fragen nach Arbeitszeit und Flexibilität, das hätte ich mich am Anfang meiner Karriere nie getraut. Diese Machtverschiebung lässt die Unternehmen nervös werden. Sie müssen sich plötzlich fragen: Wie kann ich fähige, kreative und motivierte Mitarbeiter bekommen und halten? Das betrifft vielleicht nicht unbedingt die BMWs und Googles dieser Welt, aber das mittelständische Unternehmen, das seinen Firmensitz womöglich noch fern einer Großstadt hat, wird bei der Suche nach hochqualifiziertem Personal zunehmend Probleme bekommen. Auf der anderen Seite sehen wir aber den gegenläufigen Trend, dass viele qualifizierte Arbeiten technologisch unterstützt oder sogar automatisiert durchgeführt werden können. Die Sicher-

heit, ich bin in einem hochqualifizierten Wissensjob, scheint zu schwinden.

web & mobile developer:

Worauf sollten Unternehmen bei Auswahl und Einführung von Collaboration-Lösungen achten?

Birk: Mir kommt da das Beispiel eines mittelständischen Unternehmens in den Sinn. Die Beteiligten haben sich an zwei Punkten orientiert, die gar nichts mit technologischer Finesse zu tun haben. Erstens: Die Collaboration-Lösung sollte nicht so weit weg sein von der Zusammenarbeit, die Menschen seit jeher pflegen. Das sollte sich nicht nur in den Prozessen widerspiegeln, sondern auch im Aussehen der Bedienoberfläche. Zweitens: Der Business Case muss für alle klar sein. Die Unternehmen müssen hundertprozentig daran glauben und sie müssen es an harten Fakten festmachen können. Die neue Lösung sollte am besten quantitativ messbare Verbesserungen bringen, aber auch jenen das Leben erleichtern, die mit der Software Tag für Tag umgehen müssen. Es hat zum Beispiel keinen Sinn, ein Wissensmanagement einzuführen, wenn die Mitarbeiter das Gefühl haben »Ich investiere Zeit und Know-how und die anderen haben die Vorteile.« Ob Sie eine einfache Software einführen oder ein komplexes System – für die Betroffenen muss dabei etwas herauspringen.

web & mobile developer: *Auf welches Collaboration-Tool würden Sie persönlich auf keinen Fall mehr verzichten wollen?*

Birk: Trello ist recht nützlich, das ist relativ nah dran an dem, wie Leute arbeiten. Auch Yammer finde ich ganz hilfreich.

web & mobile developer: *Was nervt am meisten?*

Birk: Wenn jede Abteilung ihr eigenes System einsetzt. Viele Lösungen sind ja recht preiswert oder sogar kostenlos, daher sind die Hürden niedrig, mal etwas Neues auszuprobieren. Dann haben die Entwickler ihr eigenes Tracking-Tool, der Vertrieb sein CRM und jeder glaubt, die anderen schauen da auch rein, was sie aber nicht tun. Wenn man als IT-Abteilung oder Geschäftsführung nicht massiv darauf drängt, dass sich alle auf ein System einigen, dann hat man dieselben Probleme in der Kommunikation und im Wissenstransfer wie vorher, sie sind nur besser versteckt.



Stefan Birk

untersucht am Institut für Arbeitsdesign und Zukunftstechnologien (ifaz), wie wir zukünftig arbeiten werden. Im Interview mit **web & mobile developer** erklärt er, was sich ändern muss, damit die Vision vom grenzenlosen und flexiblen Zusammenarbeiten nicht zum Albtraum wird.

neue Ansätze ermöglicht«, sagt Klaus Stöckert, General Manager Central Europe und Geschäftsführer bei Unify. Circuit speichert und verwaltet Informationen in Form von Assoziationen und Konversationen. Über kontextbezogene Such- und Filterfunktionen erlaubt die Plattform den Zugriff auf Dateien und Informationen, die im Lauf der Konversationen geteilt wurden.

Neben den UCC-Angeboten gibt es eine Vielzahl reiner Collaboration-Lösungen, die meist aus der Cloud heraus angeboten werden und die vor allem den Social-Business-Gedanken voranbringen wollen. Zu nennen wären etwa Asana, Basecamp, Bitrix24, HipChat, Jive, Podio, Redbooth, Stackfield, Trello oder auch Slack.

Selbst Facebook will mit Facebook at Work in die interne Unternehmenskommunikation einsteigen. Der Dienst, der seit Jahresbeginn als Test läuft, bringt typische Kommunikationsformen des sozialen Netzwerks ins Unternehmen. Bis Jahresende soll Facebook at Work nach Angaben des Magazins Fortune offiziell als Freemium-Angebot verfügbar sein. Basisfunktionen lassen sich demnach kostenlos nutzen; wer mehr will, muss entsprechende Lizenzen erwerben. Wie das Bezahlmodell genau aussehen wird, ist noch nicht bekannt.

Das Problem mit der Cloud

Was momentan genauso wenig feststeht, ist allerdings die Zukunft cloudbasierter Dienste US-amerikanischer Unternehmen. Seit der Europäische Gerichtshof das Safe-Harbor-

»Nur wer ein Tool gern nutzt, wird es auch regelmäßig nutzen.«

Frank Roth
CEO

www.appsphere.com



Foto: AppSphere

Abkommen gekippt hat, lassen sich personenbezogene Daten von EU-Bürgern auf Basis von Safe Harbor nicht mehr rechtskonform auf US-Servern speichern.

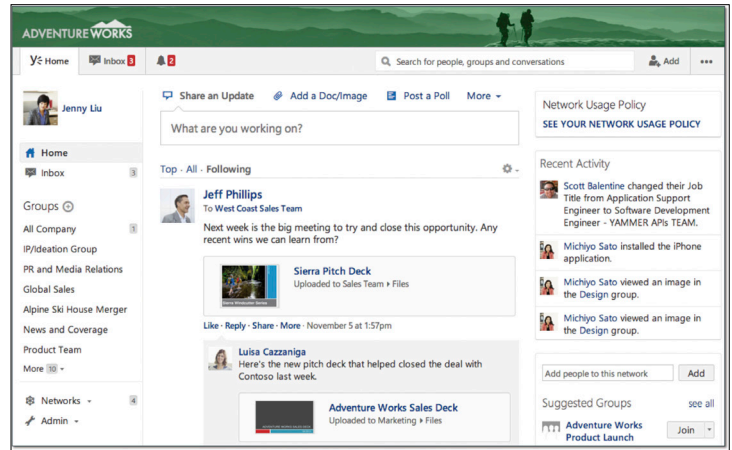
»Die Beendigung des Safe-Harbor-Abkommens durch den Europäischen Gerichtshof ist für alle deutschen Unternehmen, die personenbezogene Daten in die USA übermitteln, ein großes Problem«, sagt AppSphere-CEO Frank Roth. »Die Unternehmen müssen ab sofort überprüfen, ob von entsprechenden Transfers in die USA abzusehen ist, oder aber der Gebrauch anderer Instrumente, wie EU-Standardverträge oder Binding Corporate Rules (BCR), in Betracht kommt«, empfiehlt die Bundesbeauftragte für den Datenschutz und die Informationsfreiheit Andrea Voßhoff.

Auf die EU-Standardverträge, oft auch Model Clauses (Modellklauseln) genannt, setzen nach Angabe von Stefan Waldhauser ohnehin bereits viele Anbieter: »Alfresco teilt die Ansichten der meisten großen IT-Konzerne wie Ama- ►

zon und Salesforce.com, dass die EU-Modellklauseln als alternative Compliance-Methode zum Transfer von Daten in die USA gelten.«

AppSphere-Manager Roth glaubt, dass die meisten Unternehmen, die Collaboration-Services aus amerikanischen Rechenzentren nutzen, sich von dem Urteil nicht beeindrucken lassen und auf ein neues Safe-Harbor-Abkommen warten werden. »Wenn der Nutzen und Mehrwert einer Lösung sehr hoch ist, treten Datenschutzbedenken meist in den Hintergrund. Das ist nicht gut, aber leider die Realität.«

Auf eine rasche Lösung setzt Cisco: »Wir hoffen, dass sich die USA und die EU zügig auf ein neues Abkommen einigen werden, denn gerade für kleinere Firmen ist eine solche Vereinbarung sehr wichtig«, sagt Anton Michael Döschl. Orbit-Manager Kretzschmar kann dem Ganzen positive Seiten abgewinnen: »Durch eine Neuregelung stehen die Chancen nicht schlecht, dass es höhere Standards als bisher geben wird. So könnten Collaboration-Services der amerikanischen Anbieter auch in Deutschland eine höhere Relevanz bekommen.«



Yammer: Das soziale Netzwerk ist eine Art Facebook für Unternehmen – und Bestandteil von Office 365.

Experton-Analyst Heuer findet die Aufregung ein wenig befremdlich: »Was im Safe-Harbor-Urteil ausgesagt wurde, ist im Grunde nichts Neues, wenngleich die rechtlichen Kon-

Collaboration-Lösungen einführen

Zehn Punkte, die Sie bei der Einführung von Collaboration-Lösungen beachten sollten.

1. Planen Sie ganzheitlich

Berücksichtigen Sie alle Collaboration-Funktionen in einem Konzept, um Medienbrüche zu vermeiden. Prüfen Sie alle drei Ebenen der Zusammenarbeit (im Team, mit externen Partnern, mit Kunden). Denken Sie an die Anbindung mobiler Endgeräte.

2. Wählen Sie einen Best-of-Breed-Ansatz

Kombinationen aus Lösungen mehrerer Hersteller machen Ihr Unternehmen weniger abhängig, erfordern allerdings einen erfahrenen Integrator.

3. Integrieren Sie die neuen Lösungen in Ihre Workflows

Die Integration von Collaboration-Lösungen in die Wertschöpfungskette erfordert neben technischem Know-how umfassende Prozesskenntnisse und zumeist auch Branchenkompetenzen.

4. Entwickeln Sie Ihre Geschäftsprozesse

Nutzen Sie die Chance, mit der Einführung der Collaboration-Lösung bisherige Kooperationsformen zu hinterfragen und neue Abläufe und Interaktionen zu entwerfen. Machen Sie nicht nur das Alte besser.

5. Schätzen Sie Social Media realistisch ein

Begegnen Sie dem Web-2.0-Hype weder mit Euphorie noch mit genereller Ablehnung. Soziale Netzwerke einzusetzen bietet Chancen für den Kundenkontakt und die interne Zusammenarbeit, erfordert aber Fingerspitzengefühl.

6. Setzen Sie ein übergreifendes Sicherheitskonzept auf

Zusammenarbeit setzt Vertrauen voraus und stellt hohe Anforderungen an den Schutz geistigen Eigentums. Ein ausgefeiltes Berechtigungssystem sorgt für die notwendige Daten- und Informationssicherheit.

7. Nehmen Sie kulturelle Aspekte ernst

Bedenken Sie nicht nur die technische Integration, sondern beachten Sie auch Ihre Unternehmenskultur. Eine Collaboration-Lösung wird keinen Erfolg haben, wenn die Mitarbeiter sie nicht akzeptieren. Binden Sie Anwender und Betriebsrat frühzeitig mit ein, schaffen Sie Multiplikatoren durch technikaffine Vorreiter.

8. Erstellen Sie ein Migrationskonzept

Nutzen Sie die Instrumente des Projektmanagements für die Organisation der technischen und kulturellen Umsetzung und bilden Sie ein Team aus Vertretern aller beteiligten Interessensgruppen.

9. Gehen Sie die IT- und TK-Perspektive integriert an

In einer effektiven Collaboration-Lösung müssen die Anwendungen und die Kommunikationsinstrumente nahtlos zusammenwirken. Das schließt mobile Anwendungen ein. End-to-end-SLAs (Service Level Agreements) können über die gesamte Infrastruktur hinweg die Verfügbarkeit der Lösung sicherstellen.

10. Stellen Sie Leistung und Verfügbarkeit sicher

Collaboration-Plattformen, die über Unternehmensgrenzen hinaus arbeiten sollen, erfordern skalierbare Infrastrukturressourcen auf höchstem Niveau. Prüfen Sie die Möglichkeiten einer cloudbasierten Collaboration-Implementierung.

sequenzen natürlich tiefgreifend sind.« Zudem sei der Einsatz von Collaboration-Lösungen aus der Cloud datenschutzrechtlich meist weniger kritisch als bei ERP- oder CRM-Systemen, so Heuer weiter. »Nicht umsonst gehört Collaboration zu den am meisten als Cloud-Service genutzten Anwendungsgebieten.«



»Mit Hilfe der neuen Collaboration-Tools können leichter Experten außerhalb des Unternehmens in die Produktentwicklung einbezogen werden.«

Thomas Gmeiner

Senior Marketing Specialist
www.t-systems.com

Die richtige Strategie finden

Das riesige Angebot und das Safe-Harbor-Problem machen die Suche nach der optimalen Collaboration-Lösung nicht einfacher. Alfresco-Manager Waldhauser empfiehlt, sich vor der Entscheidung folgende Fragen zu stellen:

- Sind unsere Sicherheits- und Compliance-Anforderungen erfüllt?
- Ist das System offen genug, um sich in die bestehende Systemlandschaft zu integrieren? Ist es offen für zukünftige Entwicklungen?
- Besteht die Möglichkeit, externe Partner einzubinden, wie das im modernen verteilten Unternehmen notwendig ist?

Die Collaboration-Lösung müsse ins IT-Gesamtkonzept des Unternehmens passen, findet auch Oliver Kretzschmar von Orbit IT-Solutions. Vor allem aber müsse der Funktionsumfang immer klar definiert sein: »Die neue Collaboration-Lösung soll eine ganz bestimmte Aufgabe übernehmen, und diese Aufgabe sollte dann von keiner weiteren Plattform angeboten werden. Wird das nicht berücksichtigt, sind Insellösungen vorprogrammiert.« Oft bildeten sich in Unternehmen zwei Lager, die unterschiedliche Systeme favorisierten. »Das führt über kurz oder lang immer zu Unzufriedenheit bei den Benutzern beider Plattformen.«

Prozent ein technisches Problem.« Wer das nicht berücksichtige, werde scheitern: »Man kann nicht einfach von oben sagen, wir sind jetzt ein Social Unternehmen und machen Social Business«. Entscheidend sei, die Akzeptanz derjenigen zu gewinnen, die mit den Tools arbeiten sollen: »Ansonsten wird die schöne neue Lösung, die teuer eingekauft und mit hohem Aufwand implementiert wurde, nicht genutzt.«

Die Akzeptanz der Mitarbeiter für eine neue Collaboration-Plattform hängt auch wesentlich von deren Bedienbarkeit ab. »Lösungen, die kaum oder keine Schulungen benötigen, sind in vielen Fällen die beste Entscheidung«, sagt Polycom-Manager Hirschhoff. »Nur wer ein Tool gern nutzt, wird es auch regelmäßig nutzen«, so AppSphere-CEO Roth. Klaus Stöckert von Unify kann das nur bestätigen: »Für mich sind intuitive Bedienung und ansprechendes Design entscheidende Kriterien, die für oder gegen ein Tool sprechen.«

Fazit

Die Zusammenarbeit mobiler, einfacher und effizienter machen, die Informationsflut bändigen, Wissen managen und den Zusammenhalt stärken – das sind nur einige der Versprechen aktueller Collaboration-Lösungen. Wie weit sie sich in der Realität umsetzen lassen, hängt dabei nicht so sehr von der Nutzerfreundlichkeit, Integrationsfähigkeit und Innovationsstärke der Plattformen ab, sondern vor allem von der Kultur im Unternehmen. In »superhierarchischen Organisationen« wie wir sie laut Stefan Birk vom ifaz heute noch häufig finden, wird der tollste Social-Business-Ansatz wirkungslos verpuffen. Wer wirklich von den neuen Möglichkeiten profitieren will, muss die Unternehmenskultur radikal um- und echtes Vertrauen in seine Mitarbeiter aufbauen. Erst dann sollte er die angebotenen Lösungen technisch evaluieren. ■

»Unternehmen sind keine autonomen Einheiten mehr, sondern mehr und mehr Teil eines Netzwerks, zu dem weitere Unternehmen gehören.«

Stefan Waldhauser
Director Product Marketing
www.alfresco.com/de



Bei der Auswahl sollte man nicht vorschnell vorgehen und vor allem nicht nur auf den Preis schauen, warnt Tobias Loewenhofer von CEMA: »Da kann man richtig viel Geld verbrennen, etwa weil sich das Produkt mit der günstigen Investition nur mit viel Aufwand anpassen lässt oder aufwendig im Betrieb ist.«

Auswahl und technische Integration sind aber nur der kleinere und einfachere Part bei der Einführung einer Social-Collaboration-Lösung, sagt Experton-Analyst Heuer: »Die Unternehmenskultur ist ein ganz wichtiger Faktor, wenn es darum geht, neue Mechanismen der Zusammenarbeit einzuführen. Das ist zu 80 Prozent ein kulturelles und nur zu 20



Dr. Thomas Hafen

ist seit mehr als 15 Jahren als Redakteur und Journalist tätig, unter anderem für die IT-Fachzeitschriften NetworkWorld Germany und ChannelPartner sowie die Fotoseiten Seen.by und Digitalkamera.de.

<http://thomas-hafen.de>

AMAZON PRODUCT ADVERTISING API MIT PHP NUTZEN

Zugriff leicht gemacht

Der weltgrößte Online-Händler betreibt eine riesige Produktdatenbank.

Mit dem Associate Program von Amazon können Sie als Website-Betreiber Geld verdienen. Das geschieht dadurch, dass Sie auf Ihren Seiten ein Produkt von Amazon platzieren, wobei der dazu benutzte Link Ihre Kennung (Associate Tag) enthält. Kauft der Besucher das Produkt, fließt eine Provision auf Ihr Konto.

Wollen Sie einfach nur gezielt bestimmte einzelne Produkte bewerben, genügt es, den weiterführenden URL von Hand mit dem Associate-Tag auszustatten.

Soll die Kopplung aber intelligenter und dynamischer erfolgen, etwa über die Verbindung einer Produktkategorie, dann muss Ihre Website mit der Produktdatenbank von Amazon in Kontakt treten, wozu das Product Advertising API oder kurz PA API dient (Bild 1).

Haben Sie zum Beispiel eine Veröffentlichung zum Thema Haustiere im Internet vorgenommen, dann können Sie dieses API nutzen, um in diesen Beitrag die Top Ten der zu diesem Thema passenden Produkte einzubetten. Fühlt sich ein Leser von einem der Produkte angesprochen und verwendet er einen Ihrer Links, um es zu kaufen, dann profitieren Sie davon.

Hauptsächlich dient das API zur Abfrage von Produkten oder Produktkategorien. Sie könnten sich zum Beispiel die Liste der am meisten verkauften Bücher einer bestimmten

Kategorie geben lassen, ähnliche Artikel zu einem Produkt finden oder einfach einen bestimmten Artikel suchen. Zusätzlich bietet das API aber auch die Möglichkeit, einen Warenkorb im Kundenauftrag zu pflegen, also mit bestimmten Produkten zu füllen oder einen existierenden Warenkorb zum Beispiel durch Ergänzungen zu verändern.

Als Anwendungsfall stellen Sie sich etwa eine Website mit Basteltipps vor, die dem Kunden hilft, alles Notwendige für ein bestimmtes Projekt zusammenzustellen. Die Site legt dem Kunden also alles Notwendige in seinen Korb und er muss den Kauf nur noch auf der Amazon-Website abschließen.

So funktioniert die Kommunikation

Die Anfragen an das API erfolgen im REST-Stil, also als HTTP-Request mit *GET*-Parametern. Die Antworten erhalten Sie in der Form einer XML-Struktur.

Einige der *GET*-Parameter bleiben bei jedem Abruf aus Ihrer Anwendung heraus gleich, beispielsweise Ihr Associate-Tag oder der öffentliche Zugangsschlüssel. Andere sind dynamisch, gehören aber lediglich zur Absicherung der Anfrage, wie der Timestamp oder die Signatur, die unter Zuhilfenahme Ihres geheimen Schlüssels einen Hash über alle Parameter der Anfrage bildet (Tabelle 1).

Tabelle 1: Die wichtigsten Funktionen des Amazon API

Operation	Notwendige Parameter	Zweck
<i>ItemSearch</i>	Benötigt die Angabe eines Produktbereichs, wie <i>SearchIndex=Automotive</i> oder <i>SearchIndex=All</i> (dann wird über alle Bereiche gesucht)	Durchsucht Datenbank nach Produkten, gibt bis zu 10 Produkte zurück. Zusatzparameter wie <i>Keywords=Windjacke</i> grenzen die Auswahl ein
<i>BrowseNodeLookup</i>	<i>BrowseNodeId</i> mit der Nummer der gewünschten Kategorie, <i>ResponseGroup=BrowseNodeInfo</i>	Gibt allgemeine Informationen zur angegebenen Kategorie zurück, wie die Auflistung aller Unterkategorien oder die übergeordnete Kategorie, um per API navigieren zu können
<i>BrowseNodeLookup</i>	<i>BrowseNodeId</i> mit der Nummer der gewünschten Kategorie, <i>ResponseGroup=TopSellers</i>	Liefert die Bestseller der Kategorie, es werden aber nur die grundlegenden Daten dieser Produkte (ASIN, Name) geliefert. Mehr Infos erfordern eine weitere Abfrage, etwa per <i>ItemLookup</i>
<i>BrowseNodeLookup</i>	<i>BrowseNodeId</i> mit der Nummer der gewünschten Kategorie, <i>ResponseGroup=NewReleases</i>	Gibt die neuesten Produkte dieser Kategorie zurück
<i>ItemLookup</i>	Art der Identifizierung wird über <i>IdType=(ASIN EAN ISBN)</i> gewählt, der Parameter <i>ItemId</i> muss dann diese eindeutige Kennung oder eine kommagetrennte Liste von Kennungen enthalten	Liefert Informationen zu einem oder mehreren eindeutig über die gewählte ID bezeichneten Produkt zurück. Die gewünschten Informationen sind über eine Vielzahl möglicher Werte für <i>ResponseGroup</i> steuerbar.
<i>SimilarityLookup</i>	Benötigt genauso eindeutige Identifizierung wie <i>ItemLookup</i> , also beispielsweise <i>IdType=ASIN&ItemId=1234,5678,4711</i>	Bringt ähnliche Artikel wie das angegebene Produkt. Geben Sie eine Liste von IDs an, dann erhalten Sie Ergebnisse, die zu allen genannten ähnlich sind. Wollen Sie stattdessen Produkte, die zum einem oder einem anderen der genannten Artikel passen, dann müssen Sie zusätzlich angeben: <i>SimilarityType=Random</i>

Operation ist der zentrale Parameter eines Funktionsaufrufs. Er drückt die Methode aus, die Sie aufrufen möchten.

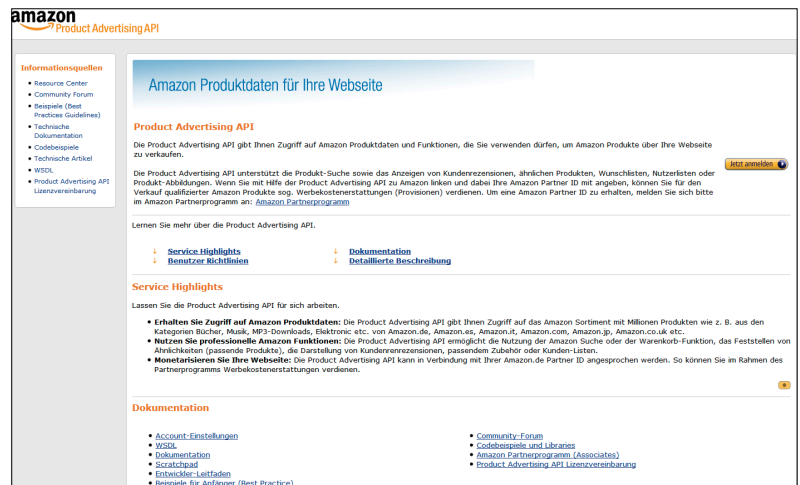
Über die Angaben beim Parameter *ResponseGroup* bestimmen Sie, welche Art von Daten das API als Antwort zurückliefern soll. Was hier möglich ist, hängt von der Art der *Operation* ab. Je nach *Operation* müssen Sie eventuell noch weitere Parameter angeben, wie etwa *ItemId*. Damit spezifizieren Sie den eindeutigen Produktcode von Amazon (ASIN) oder eine andere Kennung, wie etwa einen EAN-Code, wenn Sie Infos zu einem bestimmten Produkt erhalten möchten. Ein anderer wichtiger Parameter ist die *BrowseNodeId*, wenn es um eine Kategorie-Abfrage geht.

Das API ist übrigens Case-sensitiv. Sie müssen also auch bei den Namen der Parameter die korrekte Groß- und Kleinschreibung beachten.

Weil durch die vielen Parameter und die Notwendigkeit der Signaturbildung die Erzeugung des korrekten Abruf-URL nicht ganz trivial ist, verweist die Dokumentation von Amazon auf die PHP-Funktion `aws_signed_request()`, die das übernimmt. Sie geben ihr eine Reihe von Parametern mit und erhalten den URL zurück. Alternativ bietet das Scratchpad zu einer konkreten Abfrage den dazu passenden PHP-Code inklusive der Signierung.

Eine einfache Beispielabfrage

Die API soll Ihnen Informationen zu dem Produkt mit der ASIN *B00NGOCJ74* liefern – das ist ein iPhone 6. Als Opera-



Über das Product Advertising API können Produkte aus dem Amazon-Katalog intelligent und dynamisch beworben werden (Bild 1)

tion verwenden Sie dazu *ItemLookup*, womit ein eindeutig identifiziertes Produkt gesucht wird. Die ASIN packen Sie in den Parameter *ItemId*. Als Antwort möchten wir nur die grundlegenden Informationen wie Namen oder Produkt-URL erhalten und wählen darum als *ResponseGroup* den Wert *Small*. Die Funktion würde daraufhin für den deutschen Marktplatz sinngemäß den folgenden URL auswerfen:

```
http://ecs.amazonaws.de/onca/xml?
AWSAccessKeyId=<Ihr Key>&
AssociateTag=<Ihr Tag>&
IdType=ASIN&
```

So finden Sie einen passenden Browse Node

Die Nummer eines Browse Nodes drückt beim Amazon-API eine Produktkategorie aus, die viele der Funktionen als Eingabeparameter verarbeiten können.

Sie sehen diese Kennung beispielsweise auch als Teil des URL beim Stöbern in Amazon. Viel einfacher klappt die Ermittlung eines Browse Node aber über die Website <http://findbrowsenodes.com>. Hier geben Sie Stichwörter ein, erhalten eine gute Übersicht aller in Frage kommender Kategorien und übernehmen die ID durch Anklicken des gewünschten Labels. Zusätzlich zeigt die Website auch den gesamten Pfad für die jeweilige Kategorie an, also nacheinander alle übergeordneten Kategorien.

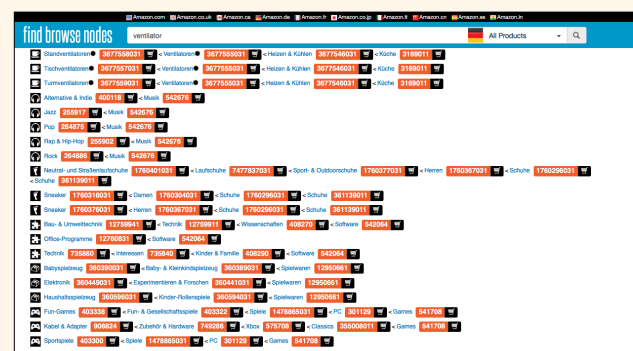
So können Sie sich zum Beispiel für eine allgemeinere Kategorie entscheiden und herausfinden, ob es auch wirklich die von Ihnen gewünschte Ecke im Warenangebot des Online-Riesen bezeichnet.

Aber Achtung: Manchmal findet der Browser eine Kategorie nicht. Suchen Sie etwa nach Baumarkt, also einer Kategorie, die Amazon selbst anbietet, dann erscheint diese nicht im Finder. In diesem Fall müssen Sie doch in Amazon selbst suchen und sich aus dem URL den korrekten Browse Node extrahieren. In diesem

Fall ist das zum Beispiel der folgende URL:

```
http://www.amazon.de/baumarkt-werkzeug-heimwerken/b/
ref=topnav_storetab_diy?ie=UTF8&node=8008403
```

Der BrowseNode für Baumarkt-Artikel lautet also *8008403*.



Statt sich die gewünschte Kennzeichnung einer Produktkategorie bei Amazon aus deren Website zu holen, können Sie diese bequem bei Findbrowsenodes.com recherchieren

```

ItemId=B00NGOCJ74&
Operation=ItemLookup&
ResponseGroup=Small&
Service=AWSECommerceService&
Timestamp=2016-01-
04T04%3A36%3A46Z&
Version=2013-08-01&
Signature=<individuelle
Kennung>

```

Wenn Sie nun diesen URL im Browser eingeben, erhalten Sie als Antwort eine XML-Datenstruktur wie in **Bild 2**.

Jede Rückmeldung des API enthält unter *Items – Request* noch einmal alle wichtigen Parameter Ihrer Anfrage. Der im Screenshot geöffnete Zweig *Item* zeigt dann die eigentlichen Ergebnisse der Anfrage, beispielsweise einen Link auf die Detailseite zum Produkt oder den URL, um es auf die Wunschliste zu nehmen, oder Informationen zum Produkt selbst, wie Name und Hersteller.

Aufbereitete Antwort erleichtert die Nutzung

Wenn Sie eine Anfrage stellen, die mehrere Antworten liefert, dann gibt es unter *Items* den Item-Block mehrmals. Bei anderen Produktarten wird der Aufbau zum Teil abweichen. So finden Sie etwa für ein Buch unter *ItemAttributes* zwar noch *Title* und *Manufacturer* (hier ist das der Verlag), aber zusätzlich auch einen Eintrag, der den Autor angibt.



Die Antworten auf eine Anfrage an Amazons Product Advertising API erhalten Sie in der Form einer XML-Struktur (**Bild 2**)

An der Codestelle in Ihrer Anwendung, an denen die von Amazon gelieferten Daten in Ihrer Website verwendet werden sollen, möchten Sie wahrscheinlich nicht unbedingt mit XML hantieren, um an die für Sie relevanten Informationen zu gelangen.

Darum haben wir eine Klasse gebaut, die Ihnen den Einsatz des PA API möglichst bequem macht. Der Nutzer dieser Klasse gibt beim Instanzieren die drei Zugangsparameter an. Die Verwendung einer bestimmten Funktion des API wird dann über eine individuelle Methode erledigt. Diese hat in ihrer Parameterliste dann wirklich nur die relevanten Informationen.

Das API live austesten

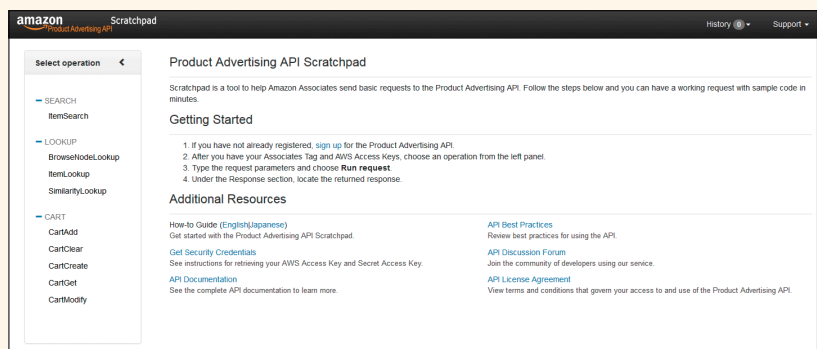
Zum reinen Herumprobieren mit den Möglichkeiten des API eignet sich das Scratchpad von Amazon ganz prima.

Es bietet Ihnen die wichtigsten Funktionsaufrufe wie *ItemLookup*, *ItemSearch* oder *BrowseNodeLookup* und ermöglicht es Ihnen, mit dem Wert für *ResponseGroup* und Zusatzparametern herumzuexperimentieren. Dabei erscheint bei ungültigen Kombinationen eine aussagekräftige Fehlermeldung.

Richtig klasse ist, dass das Scratchpad die Antwort als nicht nur als XML, sondern auch als fertig gerendertes HTML liefert. So können Sie sowohl überprüfen, ob die angezeigten Produkte Ihrem Wunsch entsprechen, als auch nachvollziehen, ob die Daten alles Nötige beinhalten, um ihre eigenen HTML-Schnipsel daraus zu generieren.

Praktisch ist auch, dass das Scratchpad Ihnen Codebeispiele zur Generierung der aktuellen Anfrage in PHP, Ruby und Java liefert,

die Sie auch abspeichern können. Zwar ist es nicht sinnvoll, den auf diese Weise erzeugten Code direkt in ein eigenes Programm einzubauen, weil er ja eine ganz spezifische Anfrage erledigt, aber Sie können die enthaltene Parameterliste gut weiterverwenden, um eine eigene Methode der hier vorgestellten Klasse daraus zu generieren.



Mit dem Scratchpad können Sie die Funktionen von Amazons PA API ausprobieren und erhalten den kompletten Aufruf auch gleich als PHP-Code

```
array (size=5)
  0 => array (size=5)
    'ASIN' => string 'B00U2ILPOU' (length=10)
    'Title' => string 'Rowenta Standventilator ESSENTIAL/Durchmesser 40 cm/sehr leise/hohe Luftumwälzung, schwarz, VU4110' (length=99)
    'DetailPageURL' => string 'http://www.amazon.de/Rowenta-Standventilator-ESSENTIAL-Durchmesser-Luftumw%C3%A4lzung/dp/B00U2ILPOU%3FSub'
    'Images' => array (size=3)
      'SmallImage' => string 'http://ecx.images-amazon.com/images/I/315E6Twy%2B0L._SL75_.jpg' (length=62)
      'MediumImage' => string 'http://ecx.images-amazon.com/images/I/315E6Twy%2B0L._SL160_.jpg' (length=63)
      'LargeImage' => string 'http://ecx.images-amazon.com/images/I/315E6Twy%2B0L.jpg' (length=55)
    'Offers' => array (size=2)
      'LowestNewPrice' => string 'EUR 84,99' (length=9)
      'LowestUsedPrice' => string '' (length=0)
  1 => array (size=5)
    'ASIN' => string 'B004H1U2SK' (length=10)
    'Title' => string 'Rowenta VU5540 Standventilator Turbo Silence' (length=44)
    'DetailPageURL' => string 'http://www.amazon.de/Rowenta-VU5540-Standventilator-Turbo-Silence/dp/B004H1U2SK%3FSubscriptionId%3DAKIAJ3'
    'Images' => array (size=3)
      'SmallImage' => string 'http://ecx.images-amazon.com/images/I/41jPd2-YTfL._SL75_.jpg' (length=60)
      'MediumImage' => string 'http://ecx.images-amazon.com/images/I/41jPd2-YTfL._SL160_.jpg' (length=61)
      'LargeImage' => string 'http://ecx.images-amazon.com/images/I/41jPd2-YTfL.jpg' (length=53)
    'Offers' => array (size=2)
      'LowestNewPrice' => string 'EUR 166,00' (length=10)
      'LowestUsedPrice' => string 'EUR 249,00' (length=10)
  2 => array (size=5)
    'ASIN' => string 'B008592Q5K' (length=10)
    'Title' => string '16 Zoll (40cm) Pedestal Boden Fan Oszillierende' (length=48)
    'DetailPageURL' => string 'http://www.amazon.de/16-Zoll-40cm-Pedestal-Boden-Oszillierende/dp/B008592Q5K%3FSubscriptionId%3DAKIAJ3QAKKZT'
    'Images' => array (size=3)
      'SmallImage' => string 'http://ecx.images-amazon.com/images/I/211RdUMGO0L._SL75_.jpg' (length=60)
      'MediumImage' => string 'http://ecx.images-amazon.com/images/I/211RdUMGO0L._SL160_.jpg' (length=61)
      'LargeImage' => string 'http://ecx.images-amazon.com/images/I/211RdUMGO0L.jpg' (length=53)
    'Offers' =>
```

Dies ist der Rückgabewert des Methodenaufrufs `getTopsellers("3677558031")`, was die Bestseller der Produktkategorie *Standventilatoren* bei Amazon liefert (Bild 3)

Nach Erhalt der Antwort parst die Methode das zurückgelieferte XML und gibt die benötigten Informationen als PHP-Array zurück. Der Aufrufer kommt also mit dem XML gar nicht in Kontakt.

Obiges Beispiel, das Infos zu einem bestimmten Produkt liefert, würde dann zum Beispiel so aussehen:

```
$api = new amzPaApi(<key>,<secret Key>,<tag>);
$produktInfo = $api->getItemInfo(' B00NGOCJ74');
```

Die Variable `$produktInfo` enthält dann zum Beispiel folgende Struktur:

```
array(
  'ASIN' => 'B00U2ILPOU',
  'Title' => 'RowentaStandventilator',
  'DetailPageURL' => 'http://www.amaz...'
)
```

Die abgedruckte Klasse `amzPaApi.php` sollten Sie nicht als schlüsselfertige Lösung sehen, sondern eher als Vorlage für Ihre eigenen Zwecke. Denn es sind nur skizzenhaft ein paar Abfrage-Methoden umgesetzt. Dass die Methoden nicht genauso heißen wie die entsprechende API-Operation, ist sinnvoll. Denn die Wirkung einer Operation unterscheidet sich zum Teil recht deutlich, je nachdem, welche ResponseGroup Sie angeben.

Reine Stichwortsuche ist fehleranfällig

So gibt die Abfrage einer Kategorie über `BrowseNodeLookup` mit der ResponseGroup `BrowseNodeInfo` Informationen zur angegebenen Kategorie aus, mit `TopSellers` aber die Hitliste der Produkte aus dieser Kategorie. Weil hier eine Ausarbeitung der Methoden für alle denkbaren Anwendungszwecke schwierig ist, schneiden Sie die Methoden am besten auf Ihre eigenen Anforderungen zu und benennen sie auch entsprechend.

Sie können zur Abfrage passender Produkte beispielsweise ein Stichwort verwenden, das auf Ihrer jeweiligen Website vorkommt, und erhalten eine Liste gefundener Produkte. Dabei ist aber die Fehlerquote relativ groß. Angenommen, Sie suchen zum Beispiel nach Ventilatoren. Die gibt es im Shop als Standgeräte zur Klimaverbesserung eines Raumes, aber auch als Einbaumodule für Computergehäuse oder als Autoersatzteil. Wird der Kunde durch solche falsche Treffer mit einem unpassenden Produkt konfrontiert, wird er es wahrscheinlich nicht kaufen und empfindet dies auch als uner-

wünscht. Punktgenau dagegen können Sie arbeiten, wenn Sie mit Amazon-Kategorien arbeiten.

Eine sinnvolle Verknüpfung eines Beitrags auf Ihrer Website mit den passenden Produktlinks ermöglicht beispielsweise die Abfrage der Topseller, also der Produkte einer Kategorie, die sich derzeit am besten verkaufen. Dabei ist der ►

Zugang erhalten

Wollen Sie das API nutzen, brauchen Sie insgesamt drei Kennungen.

Der eigentliche Zugang wird durch den 20-stelligen Access Key und den 40-stelligen Secret Access Key gesichert. Der erste wird direkt sichtbar im URL verwendet, der zweite dient nur zum Signieren einer Abfrage. Der Sinn dieser Zweiteilung ist, dass jemand, der Ihren URL abfängt, nicht einfach in Ihrem Namen mit veränderten Parametern herumexperimentieren kann. Denn dann passt die Signatur nicht zu den Parametern und das API wird die Abfrage zurückweisen.

Zusätzlich brauchen Sie noch ein Associate Tag. Das ist der Name des Kontos, dem bei erfolgreichen Käufen die Provisionszahlungen gutgeschrieben werden. Hier müssen Sie besonders Acht geben. Denn das API prüft bei den von Ihnen angestoßenen Aktionen nicht, ob dieses Konto existiert oder tatsächlich mit Ihnen zu tun hat. Stattdessen reicht es diese Kennung nur durch und baut sie in den erzeugten Links ein. Sie erkennen das Associate Tag am Parameter `tag` innerhalb eines Amazon-URL. Am besten prüfen Sie einfach über die erzeugten URLs, ob wirklich Ihr korrektes Tag enthalten ist. Denn ansonsten gehen Sie bei einem Kauf leer aus, obwohl er von Ihrer Website generiert wurde.

Wenn Sie die Kennungen noch nicht haben, finden Sie die URLs zur Anmeldung in der Online-Dokumentation von Amazon (siehe Links).

Aufwand für Sie als Betreiber gering, weil Sie nur die Ermittlung der Kategorie durchführen müssen. Die können Sie entweder fix jedem Beitrag zuordnen oder über ein vorhandenes Tagging-System lösen.

Angenommen, Sie betreiben etwa ein Mode-Blog für Frauen. Dann könnten Sie alle Beiträge, die bei Ihnen das Tag *Schuhe* haben, mit Links aus der Amazon-Kategorie *Damenschuhe* ausstatten.

Die Kategorisierung funktioniert über sogenannte Browse Nodes. Das sind IDs, die Amazon den Knoten seiner Kategorie-Bäume gibt. Sie können den Browse Node beim Stöbern im Amazon-Shop ablesen. Es ist der Teil des URL. Aber das ist etwas mühsam. Wie es einfacher geht, zeigt Ihnen der Kasten »So finden Sie einen passenden Browse Node«.

Um die Bestseller einer Kategorie zu finden, nutzen Sie die Operation *BrowseNodeLookup* in Kombination mit der *ResponseGroup TopSellers*. Die eindeutige Kennung für die Kategorie erwartet das API hier im Parameter *BrowseNodeId*.

Allerdings ist die Ermittlung der Topseller wenig informativ. In der Antwort sind nur die notwendigsten Infos enthalten, Preise und Bilder fehlen und können auch nicht durch eine zusammengesetzte *ResponseGroup* wie *Topsellers*, *Images*, *Orders* erreicht werden. Diese Kombination erlaubt das API nicht. Also ist die Lösung, diese Daten durch eine weitere Anfrage zu erhalten. Im ersten Ansatz würden Sie vielleicht für jedes erhaltene Produkt eine Abfrage vom Typ *ItemLookup* stellen. Das ist aber erstens wenig effektiv, und bei zu

Dokumentation: Während Sie bei Amazon zum Teil veraltete PDF-Dokumentationen zum API finden, bietet die HTML-Variante die aktuellsten Infos und eine gute Referenz (Bild 4)

vielen Abfragen in kurzer Zeit stellt sich auch das API quer und bringt einen Fehler wegen zu hoher Dichte von Anfragen. Darum arbeitet unsere Methode *getTopsellers()* so, dass Sie im zweiten Schritt eine Mehrfach-Abfrage stellt, die alle ASINs aus der Antwort der ersten Abfrage übermittelt. Daraus extrahiert sie die gewünschten Informationen und baut sie in das im ersten Schritt gewonnene PHP-Array zusätzlich ein, also die Zweige *Images* und *Offers* (Bild 3).

Möchten Sie die Klasse um weitere Aufrufe des API erweitern, gehen Sie am besten so vor, dass Sie über die Dokumentation von Amazon eine für Ihren Fall geeignete Kombination aus *Operation* und *ResponseGroup* identifizieren und ausprobieren (Bild 4). Dann machen Sie – am einfachsten mit dem Scratchpad – einen beispielhaften Aufruf mit den benötigten Parametern und sehen sich das zurückgelieferte XML an.

Holen Sie sich daraus die Infos, die Sie wirklich benötigen, und bauen sich daraus eine PHP-Datenstruktur, die die Methode zurückliefert. Eventuell müssen Sie dazu zwei oder mehr Aufrufe an das API richten und die Ergebnisse zusammenbauen, wie das im Beispiel mit den Bildern und Preisen für die Topseller geschah. ■

Links zum Thema

- Online-Doku zum PA API von Amazon
<http://docs.aws.amazon.com/AWSECommerceService/latest/DG>
- Scratchpad zum Ausprobieren der API-Aufrufe
<http://webservices.amazon.com/scratchpad>
- Tool der SEO-Beratung Sistrix, die beim Suchen nach erfolgversprechenden Amazon-Stichwörtern hilft
<http://amz.sistrix.com/keyword-search>
- Website zum Downloaden der Funktion *aws_signed_request()* von Ulrich Mierendorff
www.ulrichmierendorff.com/software/aws_hmac_signer.html



Markus Schraudolph

ist Journalist und Programmierer. Er schreibt seit 16 Jahren Bücher und Artikel für Fachzeitschriften. Seine Schwerpunktgebiete als Programmierer sind die Webprogrammierung und Datenbanken.

Jetzt Aussteller werden!



developer-week.de



DeveloperWeek

Aussteller & Sponsoren 2015/2016:



Veranstalter:



Neue
Mediengesellschaft
Ulm mbH

Präsentiert von:



web & mobile
DEVELOPER

GRAFIK FÜR ENTWICKLER: PHOTOSHOP, ILLUSTRATOR UND INDESIGN

Zusammenspiel

Photoshop, Illustrator und InDesign arbeiten optimal zusammen.

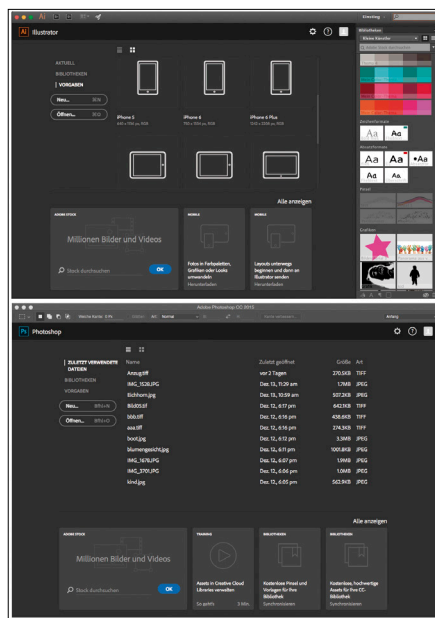
Photoshop dient hauptsächlich zur Bearbeitung von Pixelbildern, Illustrator sorgt für Vektorgrafiken. InDesign bietet alle Werkzeuge und Funktionen zum Erstellen von mehrseitigen Print-Projekten oder benutzerfreundlichen Oberflächen für digitale Anwendungen.

Gab es früher noch eine strikte Trennung der einzelnen Aufgabenbereiche, ist heute das Zusammenspiel der verschiedenen Software-Lösungen angesagt – nicht zuletzt durch die Möglichkeiten, die die Adobe Creative Cloud bietet.

Viele Neuerungen

Gerade die jüngsten Updates Ende des vergangenen Jahres haben viele Neuerungen gebracht, die für eine verbesserte Zusammenarbeit sorgen und auch für Web- und Mobile-Entwickler sehr interessant sein dürften. Hier also eine Sammlung von Tipps und Tricks.

Mit dem im November 2015 erschienenen Update hat Adobe seinem Bildbearbeitungsprogramm Photoshop wie auch InDesign und Illustrator den neuen Arbeitsbereich *Einstieg* spendiert. Dieser bietet im Überblick der zuletzt geöffneten Dateien auch weitere Informationen, etwa die Dateigröße, wann die Datei zuletzt geöffnet und in welchem Format sie gespeichert wurde.



Photoshop wie auch Illustrator bieten seit dem jüngsten Update die Arbeitsfläche *Einstieg* (Bild 1)

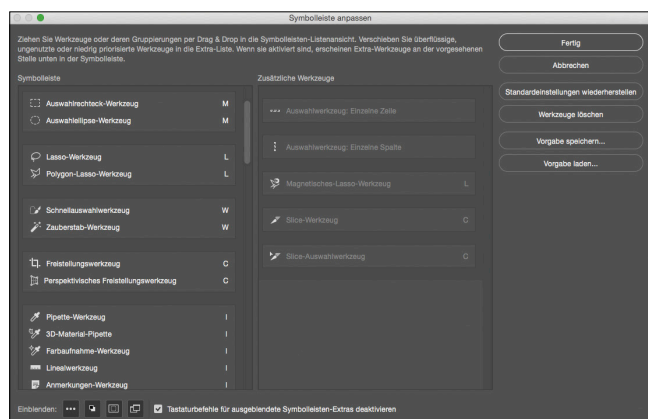
Wer auf Elemente aus einer Adobe-Cloud-Bibliothek zugreifen möchte, kann bereits zu Beginn der Arbeit auf alle Bibliotheken zugreifen, deren Inhalte dann am rechten Rand der Arbeitsfläche erscheinen. Ein direkter Link zu Adobe Stock soll dem Anwender die Suche nach geeignetem Material erleichtern.

Wie bereits auf dem Startbildschirm der früheren Versionen stehen auch verschiedene Trainingsvideos bereit. Zudem gewährt Adobe in der neuen Version einen direkten Zugriff auf verschiedene Vorgaben, die sinnvolle Einstellungen für App- und Webentwickler bieten: Bereits erstellte Dokumente wie auch die Vorgaben erscheinen im neuen Arbeitsbereich wahlweise als Listenansicht oder als Thumbnail-Vorschau.

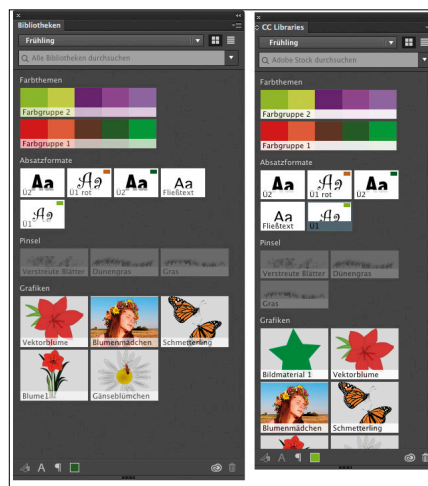
Auch bei geöffnetem Dokument hat der Anwender Zugriff auf den Arbeitsbereich *Einstieg* – zumindest in In-

Design: Hier hat er sich bei den bereits vorhandenen Arbeitsbereichen angesiedelt.

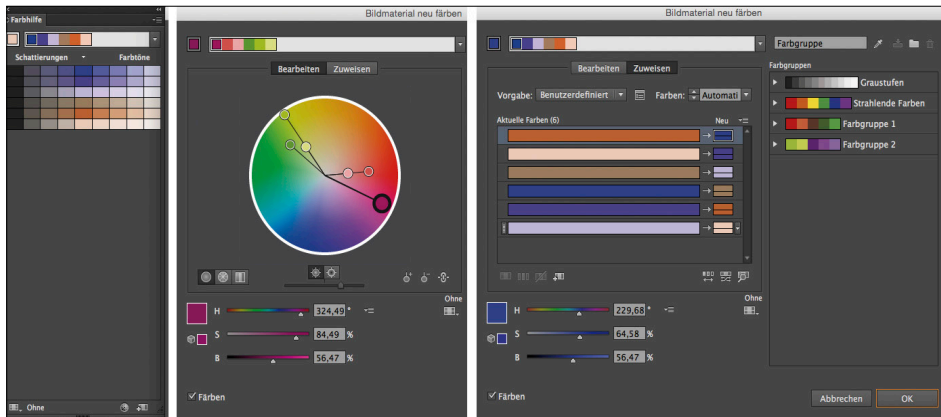
Illustrator hingegen bietet den neuen Arbeitsbereich *Zuletzt verwendete Dateien*. Dieser ist zwar auch in Photoshop verfügbar, muss jedoch zunächst in den *Voreinstellungen* unter *Allgemein* aktiviert werden. An dieser Stelle kann auch der Arbeitsbereich *Einstieg* deaktiviert werden (Bild 1).



In der aktuellen Version von Photoshop kann die Werkzeugleiste individuell angepasst werden (Bild 2)



In einer Bibliothek können Grafiken, Formate und Farben zum Austausch gespeichert werden (Bild 3)



In **Illustrator** erstrahlen Vektorgrafiken über den Dialog *Bildmaterial neu färben* in den aktualisierten Farben (Bild 4)

Symbolleiste, Menüs und Tastaturbefehle anpassen

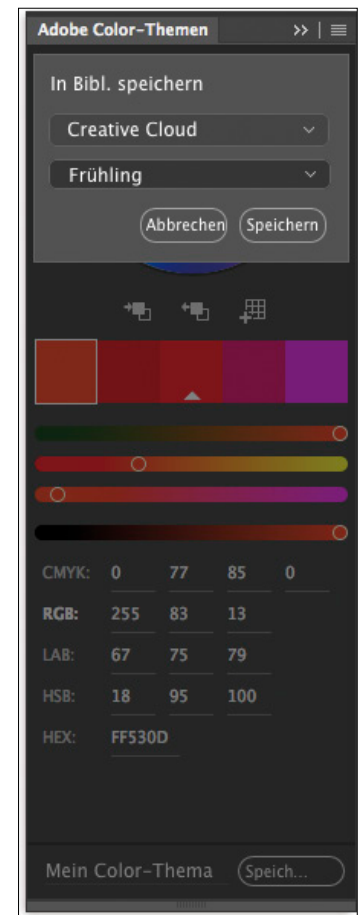
Gerade ein Programm, das eine jahrelange Entwicklung hinter sich hat, liefert oft eine Vielfalt an Funktionen und Werkzeugen, von denen nur ein Bruchteil bei der täglichen Arbeit benötigt wird. So ist beispielsweise die Werkzeugleiste in Photoshop sehr wohlwollend bestückt. Um diese etwas zu entrümpeln, hat Adobe den neuen Dialog *Symbolleiste anpassen* in das *Bearbeiten*-Menü gepackt: Hier können Tools, die kaum Verwendung finden, als *Zusätzliche Werkzeuge* ausgesiedelt werden, die dann an unterster Stelle gesammelt bereitstehen. Außerdem lassen sich über den neuen Dialog die einzelnen Werkzeuggruppen verschieben und so individuell anpassen (Bild 2).

Leider steht diese Option in Illustrator oder InDesign noch nicht bereit. Tastaturbefehle lassen sich jedoch in allen drei Programmen über das *Bearbeiten*-Menü neu belegen. So ist es beispielsweise praktisch, dem Befehl *Platzieren* in Illustrator und InDesign ein einheitliches Tastenkürzel zuzuweisen.

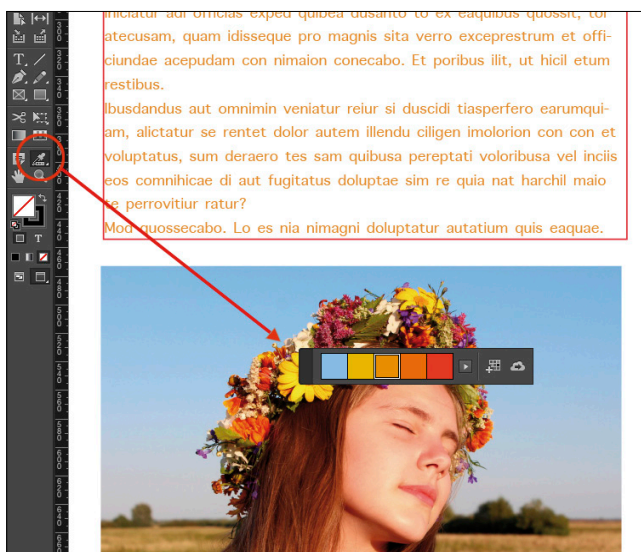
Änderungen der Anwendungs-, Kontext- und Bedienfeldmenüs sind in InDesign ebenfalls möglich. Eigene Einstellungen können unter einem neuen Namen gespeichert werden.

Bibliotheken

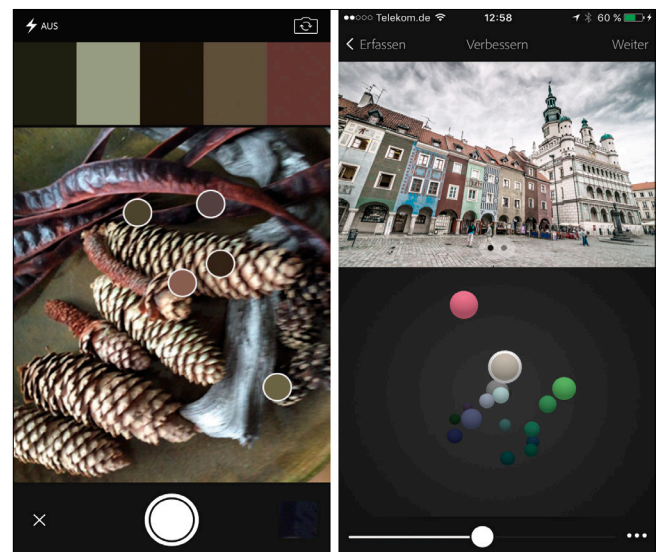
Meist kommen bei der Gestaltung einer Oberfläche mehrere Programme zum Einsatz. So werden die Bilder zunächst in Photoshop korrigiert, bevor sie in InDesign in das Layout integriert oder in Illustrator mit verschiedenen Vektorzeichnungen versehen werden. Bei der programmübergreifenden Arbeit sind die Bibliothe-



Das neue Bedienfeld Adobe Color-Themen gibt es in InDesign und in Photoshop (Bild 5)



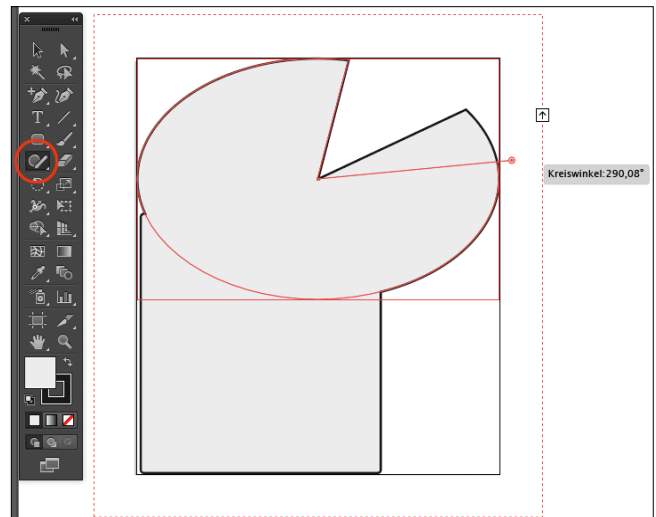
Das neue Werkzeug *Farbeinstellung* in InDesign kreiert Farbgruppen und Texte und Rahmen mit einer neuen Farbe (Bild 6)



Die App AdobeCapture erstellt anhand eines Fotos neue Farbgruppen (links) oder einen Look für Videos (rechts) (Bild 7)



Auf der Adobe-Color-CC-Website können eigene Farbgruppen erstellt und über die Adobe Cloud gespeichert werden (Bild 8)



Das Shaper-Werkzeug in Illustrator dient zum intuitiven Erstellen und Bearbeiten von Vektorformen (Bild 9)

ken besonders hilfreich: Diese können in Photoshop, Illustrator, InDesign, Premiere Pro, in After Effects und in Dreamweaver sowie in verschiedenen mobilen Apps angelegt und gefüllt werden. Eine Sammlung dieser Art kann beispielsweise aus individuell angelegten Farbgruppen bestehen. Zeichnen und Absatzformate aus InDesign oder Illustrator sowie Pinselspitzen und Grafiken finden ebenfalls in einer Bibliothek Platz und können so programmübergreifend angewandt werden. Bibliotheken sind in der Adobe Creative Cloud gespeichert und stehen somit auch Mitarbeitern zur Verfügung, die an ein und demselben Projekt arbeiten.

Dank des jüngsten Updates auf Photoshop CC 2015.1.1 lassen sich nun alle Elemente, die in einem Dokument enthalten sind, zugleich in einer Bibliothek ablegen oder direkt aus dem Bedienfeld auf das aktuelle Dokument ziehen (Bild 3).

Farbwelten

Gerade bei der Planung einer Webseite oder einer App spielen die passenden Farben eine große Rolle. Hierzu gibt es nicht nur die einfachen Farbwähler, die alle genannten Programme per Doppelklick auf die Flächen- beziehungsweise Vordergrundfarbe öffnet.

Bislang haben alle drei Programme das Kuler-Bedienfeld für Farbkombinationen bereitgestellt. Mit der CC liefert Adobe nun weitere Feinheiten zur Farbgruppenerstellung – unter anderen Benennungen:

Hier sei zunächst die den alten Hasen vertraute *Farbhilfe* aus Illustrator genannt. Darin kann nach vorgege-

benen Harmonieregeln eine individuelle Farbgruppe erstellt werden, die auf der gewählten Flächenfarbe basiert. Per Klick auf einen kleinen Schalter im unteren Bedienfeldrand öffnet sich der Dialog *Bildmaterial neu färben*. Dort lässt sich die zuvor erstellte Farbgruppe zunächst über ein Farbrad weiter bearbeiten. Sind die Grafiken mit Farben aus der Farbhilfe-Farbgruppe versehen, können sie über die *Einstellungen* im Register *Zuweisen* die neuen Farben annehmen (Bild 4).

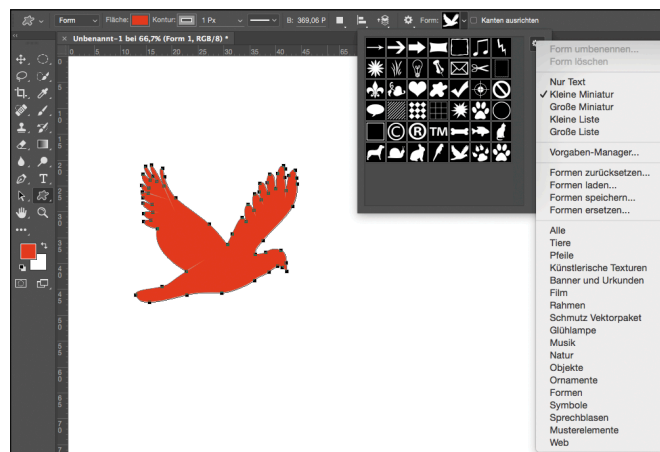
Zudem liefert die aktuelle Version von Illustrator als Kuler-Ersatz das Bedienfeld Color-Schemata. Hier erscheinen alle Farbgruppen, die über die Adobe-ID auf der Color-Website erstellt und synchronisiert wurden.

Photoshop und InDesign bieten das Bedienfeld Adobe Color-Themen. Ähnlich wie in Illustrator lassen sich hier verschiedene Farbgruppen erstellen, über das Farbrad ändern und in einer Bibliothek speichern (Bild 5).

Zudem liefert InDesign ein spezielles Werkzeug zur Erstellung von Farbgruppen: Über dem Pipette-Werkzeug liegt nun das Werkzeug *Farbeinstellung*. Per Klick auf ein im Lay-

out platziertes Bild generiert InDesign eine neue Farbgruppe. Zieht der Anwender einen Auswahlrahmen über einen Teil des Layouts, werden nur die auf diese Art gewählten Farben für die neue Farbgruppe verwendet. Ebenfalls mit der Farbeinstellung lassen sich Rahmen oder Text mit einer Farbe belegen, die zuvor in einer neuen Farbgruppe angelegt wurde.

Für die Farbgruppe gibt es unterschiedliche Designs, also Regeln, die die Farben weiter verändern. Dazu ge-



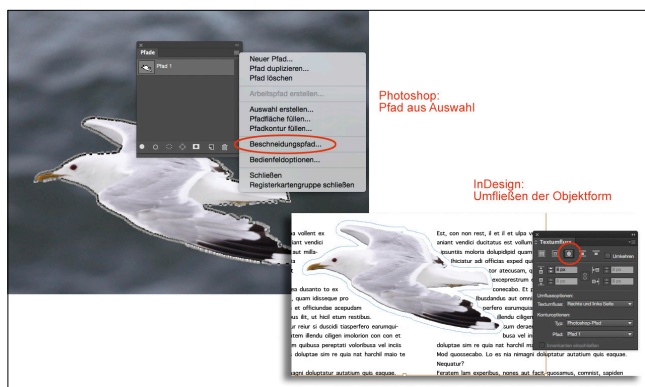
Mit dem Eigene-Form-Werkzeug entstehen Vektorformen in Photoshop (Bild 10)

hören Bunt, Hell, Dunkel oder Gedeckt. Farbgruppen – oder einzelne Farben daraus – lassen sich als Farbfelder speichern oder direkt in einer Bibliothek ablegen (Bild 6).

Wer schnell eine Farbgruppe von unterwegs erstellen möchte, sollte sich die App Capture CC ansehen. Ähnlich wie mit dem InDesign-Werkzeug *Farbeinstellung* wählt die App in einem Foto einzelne Farben; die vorgeschlagenen Farben lassen sich jedoch mit dem Finger gezielt an eine andere Stelle im Bild verschieben, um so gezielter die Farbkombinationen zu bestimmen. Soll ein Video in eine eigene Farbwelt getaucht werden, kann mit dieser App auch ein neuer Look festgelegt werden. Dabei sucht die Anwendung ebenfalls Farben und Lichter aus einer neuen Aufnahme oder verwendet als Basis ein in der Creative Cloud gespeichertes Bild.

Zusätzlich bietet die Anwendung zwei nette Spielereien: Neben eigenen Farben und einem Look verwendet die App einen Bildausschnitt zum Erstellen von eigenen Formen oder neuen Pinselspitzen (Bild 7).

Farbgruppen, Looks, Formen oder Pinsel können zudem über die Cloud in einer Bibliothek gespeichert werden und



Ein **Beschneidungspfad** aus Photoshop (links) sorgt für das freigestellte und mit Text umflossene Motiv in InDesign (rechts) (Bild 11)

stehen so zum Austausch bereit. Ebenfalls im Web bietet Adobe unter Color.adobe.com die Möglichkeit, neue Farbzusammenstellungen zu testen. Neben Farbrad und -Balken gibt es dazu verschiedene Farbgeregeln. Zudem stehen hier unter *Entdecken* die gespeicherten Farbgruppen anderer Anwender bereit. Per Anmeldung mit der Adobe-ID lassen sich die neuen Farbgruppen in einer Bibliothek speichern (Bild 8).

Pfade

Illustrator und InDesign sind die beiden Programme, in denen hauptsächlich Pfade jeglicher Art zum Einsatz kommen: In InDesign bilden Pfade Rahmen für die unterschiedlichen Inhalte, in Illustrator entstehen aus einer Kombination verschiedener Formen mehr oder weniger komplexe Vektorgrafiken. Präsent sind in beiden Programmen ähnliche Pfadwerkzeuge: Beiden gemein sind beispielsweise Tools wie das Zeichenstiftwerkzeug zum Erstellen neuer Pfade, das Auswahl- und das Direktauswahlwerkzeug zum Bearbeiten bereits erstellter Rahmen oder Pfade. ►

Top-Trainings für Entwickler

Domain Driven Design mit PHP

Trainer: Stefan Pribsch
2 Tage, München, Termin n. V.



Node.js, io.js & Co. – Entwickeln für die Cloud

Trainer: Golo Roden
3 Tage, Termin & Ort n. V.



Effiziente Software- architekturen mit PHP

Trainer: Stefan Pribsch
2 Tage, München, Termin n. V.



Codequalität mit JavaScript

Trainer: Golo Roden
3 Tage, Termin & Ort
nach Vereinbarung



Testgetriebene PHP-Entwicklung

Trainer: Sebastian Bergmann
2 Tage, Köln, Termin n. V.



Usability verstehen und verbessern

Trainer: Armin Reuter
10.-12.03.2016, Köln



IT-Recht für Entwickler

Trainerin: Antje Kilián
04.-05.04.2016, Köln



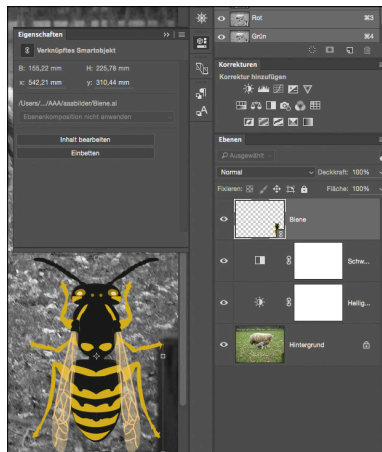
Dabei zeigt sich Illustrator immer mehr als wahrer Pfadakrobat: Neben den altbekannten Funktionen und Werkzeugen ermöglicht nunmehr in der aktuellen Version das neue *Shaper*-Werkzeug eine wesentlich intuitivere Arbeit mit Pfaden, als es mit den herkömmlichen Tools möglich ist. Besonders eignet sich dieses Werkzeug in Verbindung mit einem Zeichentablett. Durch freie Gesten entstehen Linien und geometrische Formen wie Rechtecke, Ellipsen und Polygone. Mittels Wischgeste über übereinanderliegende Bereiche zweier Formen werden diese miteinander verbunden, bleiben jedoch einzeln bearbeitbar. Per Doppelklick auf die neue Form erscheinen die sogenannten *Widgets*, kleine Steuerpunkte innerhalb oder außerhalb der Kontur, mit deren Hilfe die Form weiter modifiziert werden kann. So lassen sich Ecken abrunden, oder Segmente aus Ellipsen schneiden (Bild 9).

Doch auch das Bildbearbeitungsprogramm Photoshop kennt Pfade und liefert in dieser Hinsicht für die Zusammenarbeit mit Illustrator und Photoshop sinnvolle Werkzeuge und Optionen. Wohl jeder Grafiker kennt die Vorgehensweise, eine Auswahl in einen Arbeitspfad umzuwandeln, um diesen dann als Beschneidungspfad zu speichern. Einen solchen Pfad wiederum erkennt InDesign und blendet automatisch alle Bereiche, die außerhalb des Pfades liegen, aus. Zudem kann mit Hilfe des Beschneidungspfades Text das platzierte Objekt umfließen (Bild 10).

Eine weitere Möglichkeit, in Photoshop mit Pfaden zu arbeiten, bietet das *Eigene Form*-Werkzeug, das die anderen beiden Programme leider nicht bieten. Verschiedene Formen sind über die Optionsleiste in Bibliotheken abrufbar und können mit dem Tool als Pixelfläche, Form oder Pfad aufgezogen werden. Form oder Pfad lassen sich problemlos für eine weitere Bearbeitung in Illustrator platzieren. Leider nicht in InDesign; hier muss die Vektorform zunächst in Illustrator eingefügt werden und erscheint dann per Copy and Paste in InDesign (Bild 11).

Verknüpfte Smartobjekte

Smartobjekte in Photoshop behalten die Quelldaten der platzierten Pixel- oder Vektordateien. Somit ist der Import von Vektordaten als Smartobjekt sinnvoll, da sie andernfalls gerastert werden würden, keine scharfen Kanten mehr zeigen und auch nicht mehr frei skalierbar sind. Zudem sollte der Import einer Grafik in ein oder auch mehreren Photoshop-Dokumenten nicht über die Zwischenablage erfolgen. Es ist wesentlich besser, die Vektorgrafik über das *Datei*-Menü per *Platzieren und verknüpfen* zu übernehmen: Wird die Grafik in Illustrator modifiziert und gespeichert, übernimmt Photoshop umgehend die Änderungen. Ebenfalls direkt über die Photoshop-Datei lässt sich das verknüpfte Element bearbeiten: Hierzu genügt ein Doppelklick auf das Verknüpfungssymbol der entsprechenden Ebene. Alternativ dazu bietet



Smartobjekte lassen sich in Photoshop als verknüpftes Element platzieren (Bild 12)

das Bedienfeld *Eigenschaften* die Schaltfläche *Inhalt bearbeiten*, über die die Grafik in Illustrator geöffnet wird. Nicht nur Illustrator-Grafiken lassen sich so verknüpfen. Auch ein Bildelement, das ebenfalls in Photoshop gespeichert wurde, kann als verknüpfte Datei in mehreren Bildern platziert werden, um es dann nur einmal ändern zu müssen (Bild 12).

Symbole

Symbole aus Illustrator haben die Eigenschaft, verschiedene Änderungen am originalen Symbol an die in einem oder mehreren Dokumenten platzierten Instanzen weiterzugeben. Wird ein Symbol bearbeitet, übernehmen alle Instanzen die Änderungen – es sei denn, beim Anlegen des

Symbols wurde die Option *Dynamisch* gewählt: Verschiedene Eigenschaften der Instanzen dieser Symbolart können geändert werden, ohne alle weiteren Instanzen zu ändern. Dennoch bleibt die Verknüpfung zum Original bestehen. So lassen sich die Instanzen beispielsweise einzeln skalieren oder in ihrer Farbe ändern; wird danach die eigentliche Form des Master-Symbols modifiziert, behalten alle Instanzen ihre individuell zugewiesene Farbe und Größe.

Symbole können per Drag and Drop oder direkt über die Zwischenablage in ein InDesign-Layout eingefügt werden. Hier erscheinen sie jedoch nur als »normale« Vektorgrafik; Änderungen am Original in Illustrator werden nicht automatisch von den in InDesign platzierten Instanzen übernommen. In Photoshop eingefügte Symbole werden wie jedes andere Vektorobjekt behandelt und erscheinen auf der Arbeitsfläche nach Wahl als Pfad oder als Smartobjekt.

Fazit

Die Creative Cloud und besonders die darin abgelegten Bibliotheken erleichtern eine programmübergreifende Arbeit deutlich. Gerade wenn es um einheitliche Farben geht, gibt es hier viele neue Funktionen und Werkzeuge, die in enger Verbindung dazu stehen. Auch sonst hat sich so manches in puncto Zusammenarbeit getan. Besonders hervorzuheben seien hier die verknüpften Smartobjekte. In manchen Bereichen gibt es jedoch auch noch einiges zu tun: So wäre es etwa wünschenswert, die Vorzüge der Illustrator-Symbole auch in Photoshop oder InDesign zur Verfügung zu haben. ■



Katharina Sckommodau

arbeitet als freiberufliche Autorin, Grafikerin und Dozentin, unter anderem für die Akademie der Bayerischen Presse und für Macromedia. Sie veröffentlicht regelmäßig Beiträge in renommierten Fachzeitschriften.

Impressum

Verlag

Neue Mediengesellschaft Ulm mbH
Bayerstraße 16a,
80335 München
Telefon: (089) 741 17-0,
Fax: (089) 741 17-101
(ist zugleich Anschrift aller
Verantwortlichen)

Herausgeber

Dr. Günter Götz

Chefredakteur

Max Bold
– verantwortlich für
den redaktionellen Teil –
E-Mail: redaktion@webundmobile.de

Schlussredaktion

Ernst Altmannshofer

Redaktionelle Mitarbeit

Philip Ackermann, Tobias Amon,
Daniel Basler, Christian Bleske,
Ekkehard Gentz, Thomas Hafen,
Tam Hanna, Johannes Hoppe,
Anna Kobylinska, Bernhard Lauer,
Patrick Lobacher, Filipe Martins,
Andreas Maslo, Florence Maurice,
Michael Röhrlich, Michael Schams,
Markus Schraudolph,
Katharina Sckommodau, Frank Simon,
Matthias Walter, Gregor Woiwode

Art Directorin

Maria-Luise Sailer

Grafik & Bildredaktion

Alfred Agatz, Dagmar Breitenbach,
Catharina Burmester, Hedi Hefele,
Manuela Keller, Simone Köhnke,
Cornelia Pflanzner, Petra Reichenspurner,
Ilka Rüther, Christian Schumacher,
Nicole Üblacker, Mathias Vietmeier

Anzeigenberatung

Jens Schmidtman, Anzeigenleiter
Klaus Ahlering, Senior Sales Manager
Telefon: (089) 741 17-125
Fax: (089) 741 17-269
E-Mail Anzeigenberatung: sales@nmg.de

Anzeigendisposition

Dr. Jürgen Bossmann
Telefon: (089) 741 17-281
Fax: (089) 741 17-269
E-Mail: sales@nmg.de

Leitung Herstellung/Vertrieb

Thomas Heydn
Telefon: (089) 741 17-111
E-Mail: thomas.heydn@nmg.de

Leserservice

Hotline: (089) 741 17-205
Fax: (089) 741 17-101
E-Mail: leserservice@nmg.de

Kooperationen

Denis Motzko
Telefon: (089) 741 17-116
E-Mail: kooperationen@nmg.de

Druck

L.N. Schaffrath Druckmedien
Marktweg 42-50
47608 Geldern

CD-Produktion

Stroemung GmbH

Vertrieb

Axel Springer Vertriebsservice GmbH
Objektvertriebsleitung Lothar Kosbü
Süderstraße 77
20097 Hamburg
Telefon: (040) 34724857

Bezugspreise

web & mobile developer ist das
Profi-Magazin für Web- und
Mobile-Entwickler und erscheint
zwölfmal im Jahr. Der Bezugszeitraum
für Abonnenten ist jeweils ein Jahr.
Der Bezugspreis im Abonnement
beträgt 76,20 Euro inklusive Versand
und Mehrwertsteuer im Halbjahr, der
Preis für ein Einzelheft 14,95 Euro.
Der Jahresbezugspreis beträgt damit
152,40 Euro.

In Österreich sowie im übrigen Ausland
kostet das Abonnement 83,70 Euro im
Halbjahr. Der Jahresbezugspreis beträgt
somit 167,40 Euro. In der Schweiz kostet
das Abonnement 152,00 Franken im
Halbjahr. Der Jahresbezugspreis in der
Schweiz beträgt 304,00 Franken.

Das Abonnement verlängert sich
automatisch um ein Jahr, wenn es
nicht sechs Wochen vor Ablauf der
Bezugszeit schriftlich beim Verlag
gekündigt wird.

Studenten erhalten bei Vorlage eines
Nachweises einen Rabatt von 50 Prozent.

ISSN: 2194-4105

© 2016 Neue Mediengesellschaft Ulm mbH

**Jetzt Ihre
web & mobile developer
auf dem iPad lesen**



**Jetzt online
weiterbilden!**

„Wer an
Weiterbildung spart,
spart langfristig auch
am Erfolg.“

David Tielke
Softwareentwickler,
Berater, Trainer



developer-media.de/webinare



Foto: Shutterstock / Anteromite

MÖGLICHKEITEN ZUR AUSSCHLIESSUNG VON SUCHMASCHINEN

Search Engine Blocking

Wie Webentwickler Websites vor neugierigen Suchmaschinen schützen können.

Damit Websites bei gängigen Suchmaschinen in den Ergebnislisten auftauchen und bei bestimmten Suchbegriffen eine möglichst gute Platzierung im Ranking erzielen, ist in den meisten Fällen eine gezielte Optimierung notwendig.

Die Suchmaschinenoptimierung (SEO, Search Engine Optimization) wird seit Mitte der 1990er Jahre betrieben und gehört heutzutage in der Regel zu jedem professionellen Webprojekt.

Was bei dem Großteil von Websites völlig normal ist – und wofür nicht selten viel Geld investiert wird – kann allerdings in einigen Fällen nachteilig sein: Unter gewissen Umständen sollte eine Internetseite gar nicht (oder noch nicht) in den Ergebnislisten von Suchmaschinen auftauchen. Typischerweise zählen hierzu Websites, die sich noch in der Entwicklung befinden oder lediglich als Platzhalter für eine Internetpräsenz dienen, die in Kürze veröffentlicht werden soll.

Blindtexte, wie beispielsweise *Lorem ipsum dolor sit amet* ..., die keine inhaltliche Bedeutung haben, sondern lediglich einen Eindruck von dem Layout vermitteln sollen, sind eher schlecht platziert, wenn sie bei Google erscheinen.

Auch Inhalte, die nur eine kurze Gültigkeitsdauer besitzen und bei denen es daher nicht sinnvoll ist, diese im Internet zu archivieren, zählen zu Ressourcen, die man unter Umständen von Suchmaschinen gerne ausschließen möchte.

Websites, die Inhalte enthalten, die auf keinen Fall von Suchmaschinen gefunden und als Suchergebnisse dargestellt werden sollten, stellen eine weitere Kategorie dar. Das betrifft insbesondere Intranets und Extranets, auf die nur autorisierte Personengruppen Zugriff haben.

Mitte 2015 haben die Journalisten Marvin Oppong und Vaclav Demling aufgezeigt, dass es viele Firmen und Organisationen mit der Abschottung ihres Extranets nicht so genau nehmen. Mit speziellen Parametern bei der Google-Suche förderten sie mitunter vertrauliche Daten zutage und bekamen einen Einblick in interne Dokumente einer Klinik und einer SPD-nahen Stiftung. Außerdem konnten sie mit relativ wenig Aufwand Geburtsdaten, Rufnummern und familiäre Details aufdecken.

Die bisher genannten Beispiele zeigen deutlich, dass Webentwickler sich nicht nur mit der Suchmaschinenoptimie-

rung, sondern auch mit der Blockierung auskennen sollten. Im Folgenden werden vier etablierte Techniken und deren Vor- und Nachteile beschrieben, und es wird erläutert, in welchen Anwendungsfällen sie einsetzbar sind.

HTML-Meta-Tags

Das Meta-Tag im Header eines HTML-Dokuments wird verwendet, um Metadaten einer Seite zu notieren. Das können beispielsweise eine Beschreibung der Seite, eine Auflistung von Schlüsselwörtern, Informationen über den Autor des Dokuments, Copyright-Hinweise et cetera sein.

Meta-Tags spielen heutzutage bei der Optimierung von Websites nur noch eine untergeordnete Rolle. Allerdings respektieren die meisten Suchmaschinen das folgende Meta-Tag und nehmen dadurch die entsprechende Seite nicht in ihren Suchindex auf:

```
<meta name="robots" content="noindex">
```

Um nur Googles Webcrawler auszuschließen, kann statt *robots* das Schlüsselwort *googlebot* eingesetzt werden. Trifft Google beim nächsten Besuch der Website auf dieses Meta-Tag, wird die Seite vollständig aus deren Suchergebnissen entfernt – unabhängig davon, ob andere Websites auf die Seite verlinken.

Natürlich gelten Meta-Tags in HTML-Dokumenten nicht nur für Suchmaschinen, sondern für alle Dienste, die eine Internetseite auslesen. Daher spricht man genau genommen von Webcrawlern.

Laut der HTML-Version-4.01-Spezifikation (Anhang B.4.1) sind folgende vier Werte als *content*=“...” zulässig: *all*, *index*, *nofollow* und *noindex*. Häufig findet man auch *follow* (was allerdings nicht in der offiziellen Spezifikation notiert ist), das sich zusammen mit *nofollow* auf Links bezieht. Das Schlüsselwort *all* ist identisch mit der Kombination *index,follow* und erlaubt Crawlern, alle Seiten, die im aktuellen Dokument verlinkt werden, sowie die aktuelle Seite auszulesen.

Die Datei robots.txt

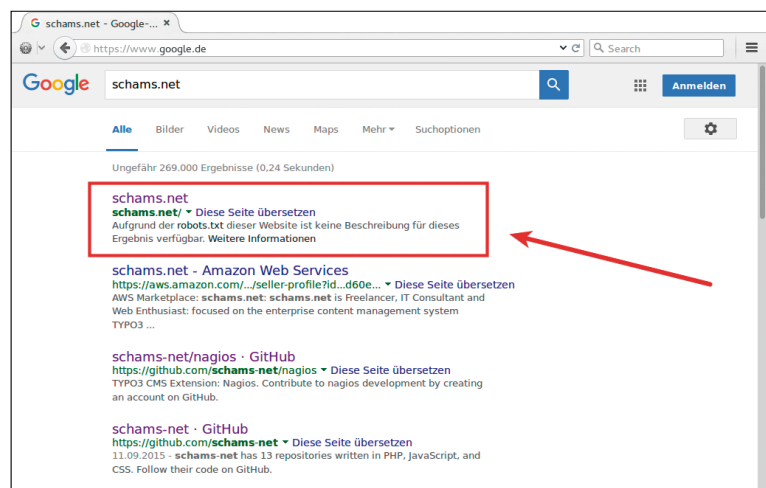
Jedem Webentwickler sollte die *robots.txt*-Datei mehr oder weniger geläufig sein. Auch wenn es sich bei der Datei lediglich um einen De-facto-Standard handelt, ist sie seit Jahr-

zehnten etabliert und wird von allen seriösen Web-Robots (Suchmaschinen, Webcrawlern et cetera) berücksichtigt.

Vor über 20 Jahren diskutierte man bereits auf Mailinglisten über einen Standard for Robot Exclusion, und 1996/1997 wurde ein diesbezüglicher Entwurf veröffentlicht. Dies wurde ebenfalls in der zuvor erwähnten HTML-Version-4.01-Spezifikation dokumentiert (Anhang B.4.1).

Ganz einfach ausgedrückt beinhaltet die Datei *robots.txt* – die sich im Document Root, also dem Hauptverzeichnis des Webserver, befinden muss – Anweisungen für bestimmte Webcrawler, welche URLs diese Suchroboter nicht aufrufen dürfen.

Die Identität der Webcrawler wird durch den User-Agent-String definiert, gefolgt von einer Liste der ausgeschlossenen Ressourcen (*disallow* = verbieten). Folgende Zeile in der Da-



Google listet Websites trotz *robots.txt*-Dateien auf (Bild 1)

tei *robots.txt* weist beispielsweise Google an, keine Seiten unterhalb */private* zu analysieren:

```
user-agent: googlebot disallow: /private
```

Um generell alle Webcrawler zu informieren, dass sie keine Ressourcen einer Website auslesen sollen, kann folgendes Konstrukt verwendet werden:

```
user-agent: * disallow: /
```

Wer mehrere Domains auf einer Webinstanz konfiguriert hat, sollte unbedingt sicherstellen, dass die *robots.txt*-Datei auf allen Instanzen und Protokollen (HTTP und HTTPS) zugreifbar ist. Die Anweisungen in einer *robots.txt*, die nur unter *http://example.com/robots.txt* erreichbar ist, gilt nämlich nicht automatisch auch für *https://example.com* und auch nicht für Subdomains wie zum Beispiel *www.example.com*.

Grundsätzlich sind folgende Schlüsselwörter innerhalb der *robots.txt*-Datei gültig:

- *user-agent*: identifiziert den zugreifenden Dienst, beispielsweise *googlebot*,

Meta-Tags und robots.txt

Von dem gleichzeitigen Einsatz des *noindex*-Meta-Tags und der *robots.txt*-Datei ist abzuraten.

Suchmaschinen, die durch *robots.txt* ausgeschlossen werden, beginnen erst gar nicht, die einzelnen Seiten einer Website zu analysieren, und treffen somit niemals auf Meta-Tags. Das kann zur Folge haben, dass Seiten trotzdem in ihren Ergebnislisten auftauchen, zum Beispiel wenn andere Seiten auf sie verweisen.

- *disallow*: listet auf, welche Ressourcen nicht indiziert werden sollen,
- *allow*: listet auf, welche Ressourcen indiziert werden können,
- *sitemap*: definiert den URL einer Sitemap, die zum Beispiel im TXT- oder XML-Format vorliegen kann.

Ein weiterer wichtiger Punkt, der beim Einsatz der *robots.txt*-Datei zu beachten ist, ist die Tatsache, dass die Website in der Ergebnisliste von Google auftaucht, obwohl Google angewiesen wurde, die komplette Website zu ignorieren.

Anstatt die gewohnte Kurzfassung der Inhalte auszugeben, weist Google darauf hin, dass aufgrund einer *robots.txt*-Datei keine Beschreibung verfügbar ist. Bild 1 zeigt exemplarisch einen solchen Fall für die Domain Schams.net.

Soziale Netzwerke, wie zum Beispiel Facebook und Google+, sind gegenüber der *robots.txt*-Datei sogar noch ignoranter: Wird bei Facebook ein Link geteilt, versucht das soziale Netzwerk, darauf zuzugreifen und den Inhalt auszulesen – völlig unabhängig von den Deklarationen in der *robots.txt*-Datei. Als Argument dafür wird angeführt, dass die Website-Betreiber mit dem Einsatz der Datei nur die automatische Indexierung beeinflussen, nicht aber gezielte Zugriffsversuche.

HTTP-Header X-Robots-Tag

Die beiden zuvor genannten Lösungen lassen sich in der Regel relativ einfach implementieren. Webentwickler haben in den meisten Fällen Einfluss auf HTML-Tags oder können Dateien wie zum Beispiel *robots.txt* auf dem Server erstellen und bearbeiten.

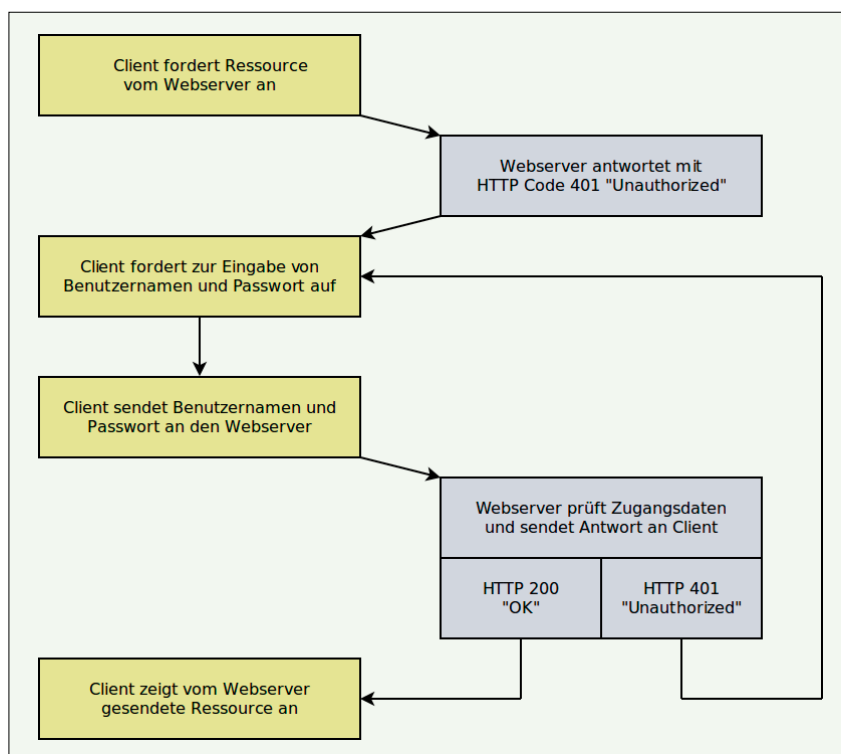
Die dritte Methode ist allerdings nicht immer möglich: Sie erfordert einen Zugriff auf die Konfiguration des Webserver oder ausreichende Berechtigungen in der Applikation, um Zeilen dem HTTP-Header hinzuzufügen.

Letzteres lässt sich beispielsweise bei TYPO3 CMS mit dem TypoScript `config.additionalHeaders` realisieren, oder bei WordPress mit der Funktion `add_header_xua()`.

Generell wird dem HTTP-Response vom Server folgende Zeile hinzugefügt: *X-Robots-Tag*. Der darauffolgende Wert beziehungsweise die Werte weisen die Webcrawler an, bestimmte Ressourcen nicht zu indexieren. Hierbei gelten dieselben Schlüsselwörter wie bei den zuvor beschriebenen Meta-Tags:

- **X-Robots-Tag noindex**: indexiert die Seite nicht,
- **X-Robots-Tag googlebot: noindex**: weist nur Google an, die Seite nicht in dem Suchindex zu notieren.

Die meisten seriösen Suchmaschinen halten sich auch an diesen Standard: Google, Bing und Yahoo Search zum Beispiel. Die Betreiber von DuckDuckGo geben an, dass man die ini-



Vereinfachte Darstellung einer HTTP-Authentifikation (Bild 2)

tialen Inhalte von »Hundert von Quellen« erhält, und man daher nicht genau feststellen kann, welche Anweisungen dabei berücksichtigt werden. Offiziell unterstützt der DuckDuckGoBot zwar die *robots.txt*-Datei, die Spezifikation schweigt sich aber über jegliche andere Methoden aus. Bei Google taucht eine mit dem X-Robots-Tag-Header konfigurierte Website gar nicht in der Ergebnisliste auf.

Wer Zugriff auf die Konfiguration des Webserver hat, kann die entsprechende Einstellung natürlich auch auf dieser Ebene vornehmen. Beim Apache-Webserver lauten das Schlüsselwort und der exemplarische Wert:

```
Header set X-Robots-Tag "noindex"
```

Das aktivierte Apache-Modul `mod_headers` wird hierbei natürlich vorausgesetzt.

Authentifikation

Die drei zuvor beschriebenen Techniken haben alle einen gravierenden Nachteil: Sie weisen Suchmaschinen an, etwas zu tun, halten sie aber grundsätzlich nicht zwingend davon ab. Man ist darauf angewiesen, dass sich Suchmaschinen an die Anweisungen halten. Das ist bei den etablierten, namhaften Suchmaschinen wie beispielsweise Google, Bing, Yahoo Search und DuckDuckGo auch kein Problem – trotzdem bleibt das, was man schützen möchte, zugreifbar.

Besonders die *robots.txt*-Datei stellt ein besonderes Problem dar. Sie weist eindeutig auf Ressourcen hin, die nicht abgerufen werden sollen. Das lädt zwielichtige Dienste natürlich geradezu ein, sich jene genauer anzuschauen beziehungsweise zu kopieren.

Eine sichere Maßnahme, um diesem entgegenzuwirken, ist das Erzwingen einer Authentifikation. Das kann auf mehreren Wegen erfolgen. Eine einfache Lösung ist das Abfragen eines Benutzernamens und Passworts mit Hilfe einer HTTP-Authentifikation.

Hierbei wird bei dem Versuch, auf eine geschützte Ressource zuzugreifen, der HTTP-Code 401 (Unauthorized) vom Server an den Client geschickt. Jeder normal konfigurierte Browser zeigt in diesem Fall ein Popup-Fenster an und fordert den Benutzer auf, die Zugangsdaten einzugeben, die dann in einem nächsten Request an den Server geschickt werden (Bild 2). Ohne eine korrekte Authentifikation werden keine Inhalte vom Server ausgeliefert.

Andere mögliche Methoden einer Authentifikation sind beispielsweise Client-Server-SSL-Zertifikate oder die Authentifikation auf Applikationsebene. Letzteres bedarf keiner ausführlichen Erklärungen: Auch hier wird in der Regel die Kombination aus Benutzername und Passwort abgefragt (heutzutage immer häufiger mit noch einem zusätzlichen Faktor, zum Beispiel einem Code auf dem Mobiltelefon), allerdings wird die Authentifikation durch die Webanwendung selbst realisiert, anstatt vom Webserver.

Mit SSL gesicherter Webserver

Ein mit SSL gesicherter Webserver kann so konfiguriert werden, dass sich nur Clients verbinden dürfen, die über ein entsprechendes Zertifikat verfügen. Ein typisches Beispiel ist ein Zertifikat, das mit dem SSL-Schlüssel des Servers signiert wurde. Ohne ein gültiges Zertifikat ist der Zugriff auf den Server nicht möglich.

Es ist klar, dass diese Methoden Suchmaschinen und Webcrawler garantiert ausschließen – unabhängig davon, ob sie sich an Standards halten oder nicht. Ohne gültige Authentifikation haben sie einfach keinen Zugriff auf bestimmte Ressourcen.

Leider trifft man immer wieder auf Fälle, in denen sich Benutzer anmelden müssen, um bestimmte Dateien herunterzuladen zu können, jedoch werden diese Dateien in öffentlich zugänglichen Verzeichnissen abgelegt. Ist der entsprechende URL der Datei einmal bekannt, erraten oder ermittelt, ist eine Anmeldung hinfällig und bietet keinen wirklichen Schutz mehr. Hier kommen häufig die zuvor erwähnten speziellen Parameter ins Spiel, die man einer Google-Suche mitgeben kann (Google Dorks genannt). Grundlegend sollte folgender Grundsatz befolgt werden: Dateien und Dokumente, die nicht für die Öffentlichkeit bestimmt sind, haben weder im Document Root (zum Beispiel Verzeichnis *htdocs* eines Apache-Webserver), noch in einem öffentlich zugänglichen Unterverzeichnis etwas zu suchen. Daher gehören sie aus Sicherheitsgründen außerhalb des Document Roots, und es ist eine zusätzliche Funktion notwendig, die die Zugriffsrechte auf die Ressource überprüft und erst im Erfolgsfall die Datei an den Client ausliefert.

Fazit

Die in diesem Artikel beschriebenen vier Schutztechniken zeigen, dass es unterschiedliche Stufen gibt, um das Verhal-

Links zum Thema

- Artikel »Offene Intranets verraten zu viel«
<http://heise.de/~2680058>
- W3C Standard, Abschnitt »Search Robots«
www.w3.org/TR/html401/appendix/notes.html#h-B.4.1
- Google Dorks
https://en.wikipedia.org/wiki/Google_hacking
- Googles Richtlinien für Webmaster
<https://support.google.com/webmasters/answer/35769?hl=de>

ten von Suchmaschinen zu beeinflussen. Wenn es sich bei den zu schützenden Daten tatsächlich um sensible Informationen handelt, sollte man sich stets im Klaren darüber sein, dass nur eine wirkliche Authentifikation einen realen Schutz darstellt.

Grundsätzlich sollte man sich – beziehungsweise dem Kunden – die in solchen Fällen berechnete Frage stellen, ob diese Daten überhaupt in einem vom Internet zugreifbaren System gespeichert werden müssen.

Sofern ein solches Verfahren die einzige mögliche Option darstellt, ist ein passwortgeschützter Mitarbeiter- oder Kundenbereich gegenüber öffentlich zugreifbaren Seiten die bessere und sicherere Lösung. Viele webbasierte Systeme legen hochgeladene Dateien allerdings standardmäßig im Web-Verzeichnis ab.

Wer über die Kenntnis des URL verfügt, unter dem die zu schützenden Daten liegen (zum Beispiel *http://example.com/uploads/streng-geheim.pdf*), kann unter Umständen die Authentifikation umgehen und als Folge davon jederzeit auf diese Dateien zugreifen. Schlimmer noch, wenn der Webserver falsch konfiguriert ist und bei einem Zugriff auf das Verzeichnis *upload* dessen Inhalt auflistet.

Bei Suchmaschinen finden sich zahlreiche Beispiele in den Ergebnissen, wenn man die richtigen Suchparameter (Google Dorks) verwendet.

Um ihre Kunden kompetent beraten zu können, ist es für Webentwickler nicht ausreichend, sich nur mit der Optimierung von Websites auszukennen. Der Schutz von Daten ist heutzutage mindestens genauso wichtig, und je nach Sensibilität der Informationen reicht ein Meta-Tag oder eine *robots.txt*-Datei nicht aus. ■



Michael Schams

lebt seit 2008 in Melbourne/Australien und arbeitet als Projekt Manager und Consultant. Er ist zertifizierter TYPO3 Integrator und Project Leader des offiziellen TYPO3 Security Guides.
@MichaelSchams

ACCELERATED MOBILE PAGES

Googles Antwort

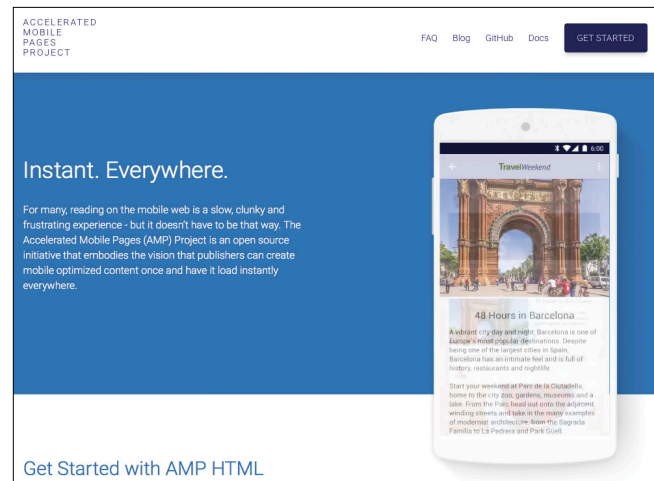
Accelerated Mobile Pages sind Googles Antwort auf Instant Articles von Facebook.

Der nächste große Trend im Medienbereich wird sicher die plattformunabhängige Veröffentlichung von Inhalten werden. Während Facebook seit Mai 2015 seine Instant Articles gestartet hat, arbeitet Google ebenfalls an einem ähnlichen Format mit dem Namen Accelerated Mobile Pages oder kurz AMP.

Dieses Format wird zusammen mit Twitter als Open-Source-Lösung entwickelt und soll Anfang 2016 reif für den produktiven Einsatz sein. Vorgestellt wurde es von Google am 7. Oktober 2015 in einem Blogbeitrag auf Googles eigenem Blog Blogspot (Bild 1).

Eine Vielzahl großer Publisher haben ihre Unterstützung angekündigt und setzen bereits heute auf das neue, schlanke Artikelformat: BBC, BuzzFeed, The Economist, El País, Financial Times, Frankfurter Allgemeine, The Guardian, The Huffington Post, The New York Times, The Telegraph, Time, The Wall Street Journal, Zeit Online – um nur einige der mittlerweile weit über 1000 zu nennen.

Aber nicht nur Publisher, sondern auch Technologiefirmen setzen auf das neue Format: Auch Adobe, Google, LinkedIn, Pinterest, Twitter und Wordpress zählen zu den Unterstützern. Nachdem das Internet mittlerweile zunehmend mobil genutzt wird, hat sich auch die Mediennutzung verändert. Anwender wollen die Inhalte genau dort konsumieren, wo sie sie gerade vorfinden. Manchmal direkt von der Website eines Publishers, oftmals aber auch direkt aus Aggregatoren heraus oder in den Suchmaschinen.



Startseite des AMP-Projekts (Bild 1)

Bedenkt man, wie frustrierend die Experience im mobilen Web sein kann, suchen viele Content-Distribution-Plattformen nach Alternativen. In vielen Fällen sind diese Alternativen allerdings plattformspezifisch und belasten damit die Verlage, indem die Fragmentierung der Formate unterstützt und diese somit vom Ökosystem abgeschottet werden.

Das Projekt Accelerated Mobile Pages ist eine Initiative, um das mobile Web und damit das Ökosystem der Distribution zu verbessern. Wenn Inhalte schnell, flexibel und ansprechend

Listing 1: Boilerplate

```
<!doctype html>
<html amp lang="en">
  <head>
    <meta charset="utf-8">
    <title>Hello, AMPs</title>
    <link rel="canonical"
      href="http://www.lobacher.de/amp/listing01.html" />
    <meta name="viewport" content=
      "width=device-width,minimum-scale=1,
      initial-scale=1">
    <script type="application/ld+json">
      {
        "@context": "http://schema.org",
        "@type": "NewsArticle",
        "headline": "Open-source framework for
          publishing content",
        "datePublished": "2015-12-12T12:33:41Z",
        "image": [
          "logo.jpg"
        ]
      }
    </script>
    <style>body {opacity: 0}</style><noscript>
    <style>body {opacity: 1}</style></noscript>
    <script async src=
      "https://cdn.ampproject.org/v0.js"></script>
  </head>
  <body>
    <h1>Willkommen zur ersten AMP-Seite</h1>
  </body>
</html>
```


realisiert sind und überzeugende sowie effektive Anzeigen beinhalten, können wir das offene Webpublishing-Modell und die Einnahmequellen bewahren, die so wichtig für die Nachhaltigkeit des qualitativen Publizierens sind.

AMP HTML ist ein neuer Weg, um Webseiten auf mobilen Geräten sofort zu laden, wenn diese speziell optimiert wurden. In der Entwicklung waren vor allem smartes Caching, vorhersagbare Performance und moderner, ansprechender mobiler Inhalt wichtig.

Da AMP HTML auf bestehenden Webtechnologien aufsetzt und kein Template-basiertes System darstellt, können Verlage weiterhin ihre eigenen Inhalte selbst wie gewohnt hosten, diese aufgrund ihrer eigenen User-Experience weiterentwickeln und beliebig Werbung integrieren – und dies alles innerhalb einer technischen Architektur, die in Bezug auf Geschwindigkeit und Leistung optimiert wurde.

Warum ist AMP schneller?

Damit AMP deutlich schneller lädt, wurden einige Optimierungen vorgenommen:

- Skripts werden ausschließlich asynchron geladen und eigene Skripts sind ausschließlich in Iframes möglich.
- Die Größe aller Ressourcen wird immer festgelegt. Zum Aufbau der Seite wird nur ein HTTP-Request benötigt.
- Extensions (wie Lightboxes, Instagram Embeds, Tweets) blocken nicht das Rendering.
- CSS muss inline angegeben werden und dieses wird auf 50 KByte begrenzt.
- Erst finden alle DOM-Leseoperationen statt und anschließend alle DOM-Schreiboperationen.
- Animationen werden ausschließlich GPU-beschleunigt durchgeführt.
- Es werden nur die Ressourcen geladen, die auch direkt benötigt werden. Beispielsweise nur diejenigen, die above the fold liegen – andere (wie Videos) per Lazy Loading.
- Seiten werden über Preconnect (www.w3.org/TR/resource-hints/#dfn-preconnect) vorab geladen.

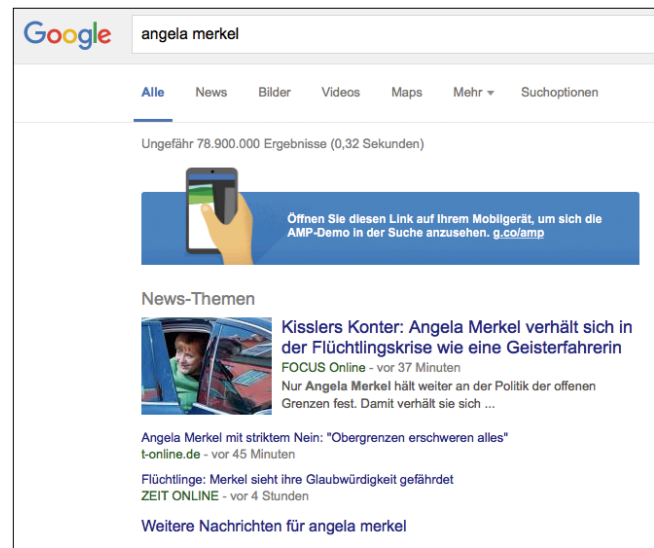
Listing 2: Bilder einfügen

```
<amp-img src="responsive.jpg" width=527 height=193
layout="responsive" ></amp-img>

<amp-img id="img1" src="fixed.jpg" width=264
height=96></amp-img>

<amp-img id="img2" src="hidden.jpg" width=527
height=193 layout="nodisplay"></amp-img>

<amp-img
  src="wide.jpg"
  srcset="wide.jpg" 640w,
    "narrow.jpg" 320w
  sizes="(min-width: 650px) 50vw, 100vw" >
</amp-img>
```



AMP Google Search mit einem Desktop-Browser (Bild 2)

Um das Ergebnis einer mit AMP angereicherten Suche bereits jetzt schon (vor Ende Februar 2016) auszuprobieren, muss man Google auf einem Mobilgerät mit einigen Parametern aufrufen. Google hat aber dafür einen Shortlink eingerichtet: g.co/ampdemo.

Ruft man die Suche auf dem Desktop auf, bekommt man direkt eine entsprechende Meldung, die darauf hinweist, dass die Suche derzeit nur auf Mobile möglich ist (Bild 2).

Dort gibt man zuerst den Shortlink und in der dann geladenen Suche ein Thema der Tagespresse ein, zum Beispiel *Angela Merkel*. Scrollt man etwas nach unten, sieht man in einem Karussell die neuesten Nachrichten von Frankfurter Allgemeine sowie der Zeit Online in einem anderen Format: Bild, Publisher, Text und Datum. Sobald man auf eine der Nachrichten tippt, wird man sofort – das heißt ohne merkliche Verzögerung – zum Artikel geleitet (Bild 3).

Google plant, dass dieses Verhalten überall dort eingesetzt werden kann, wo man schnell weitere Informationen benötigt: Social Media, Apps, Feed-Reader, Suche und viele weitere.

Die erste AMP-Seite

Das folgende Markup kann als Startpunkt oder Boilerplate verwendet werden. Speichern Sie es in einer Datei mit der Endung *.html* (Listing 1). Der Code im Body ist dabei selbsterklärend – ganz im Gegenteil zum Head, den wir uns genauer ansehen wollen. Dabei muss ein AMP-HTML-Dokument Folgendes zwingend umsetzen:

- Das Dokument muss mit dem Doctype `<!doctype html>` starten.
- Es muss ein `<html ...>`-Tag existieren (`<html amp>` ist ebenfalls erlaubt).
- Die Tags `<head>` und `<body>` müssen verwendet werden.
- Es muss ein `<link rel="canonical" href="$HTML_URL" />`-Tag im Head platziert werden, welches auf die reguläre HTML-Version zeigt, oder auf sich selbst, wenn es eine solche nicht gibt.

- Direkt nach dem `<head>`-Tag muss das Tag `<meta charset="utf-8">` folgen.
- Der Quelltext muss ein `<meta name="viewport" content="width=device-width,minimum-scale=1">` im Head enthalten. Es wird empfohlen, zusätzlich noch das Attribut `initial-scale=1` zu verwenden.
- Als letztes Element im Head muss die Angabe `<script async src="https://cdn.ampproject.org/v0.js"></script>` platziert werden (dies lädt die AMP-JavaScript-Bibliothek).
- Es muss die Angabe `<style>body {opacity: 0}</style><noscript><style>body {opacity: 1}</style></noscript>` im Head enthalten sein.

Die Schema-Angabe ist optional, aber empfehlenswert, wenn man dieses distribuieren will (zum Beispiel im News-Karussell). Hier wurde `NewsArticle` verwendet – die Definition dazu findet sich unter <https://schema.org/NewsArticle>.

Erlaubtes Markup

Die meisten HTML-Tags können innerhalb von AMP HTML unverändert verwendet werden, aber einige davon sind verboten oder werden durch eigene ersetzt. Verbotene Tags sind `script` (es sei denn, der Typ ist `application/ld+json`), `base`, `frame`, `frameset`, `object`, `param`, `applet`, `embed`, `form` sowie die `input`-Elemente `input`, `textarea`, `select` und `option`. Erlaubt ist aber `button`.

Folgende Tags werden ersetzt, nämlich `` durch `<amp-img>`, `<video>` durch `<amp-video>`, `<audio>` durch `<amp-audio>` und `<iframe>` durch `<amp-iframe>`.

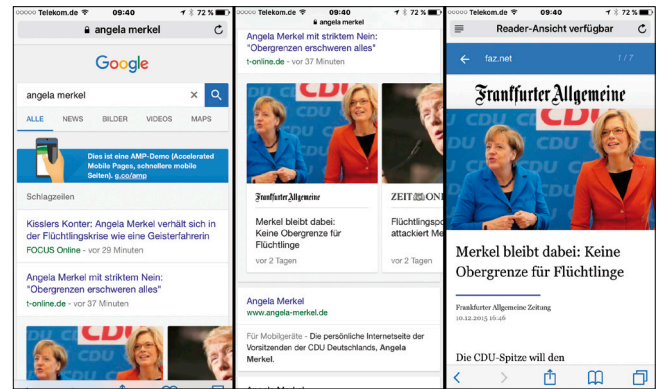
Medien hinzufügen

Zur Einbindung von Bildern kann einer der in Listing 2 aufgeführten Codes verwendet werden.

Breite und Höhe müssen zwingend mit angegeben werden. Das Attribut `layout` bestimmt dann, wie das Bild dargestellt wird. Verwendet man beispielsweise `responsive` als Wert, so wird das Bild im Verhältnis der Größenangaben in den zur Verfügung stehenden Platz hineinskaliert. Der Wert `nodis-`

Listing 3: Media Queries

```
<amp-img
  media="(min-width: 650px)"
  src="wide.jpg"
  width=466
  height=355
  layout="responsive" >
</amp-img>
<amp-img
  media="(max-width: 649px)"
  src="narrow.jpg"
  width=527
  height=193
  layout="responsive" >
</amp-img>
```



AMP Google Search mit einem Mobile-Browser (Bild 3)

`play` sorgt dafür, dass das Bild zunächst nicht angezeigt wird (sondern erst, wenn man dieses explizit darstellen will).

Mögliche Attribute sind: `width`, `height`, `id`, `figure`, `figcaption`, `attribution` (ein String zur Zuordnung), `src`, `srcset`, `sizes` sowie alle Standard-HTML5-Attribute wie zum Beispiel `alt`. Animierte Bilder (GIF-Animationen) werden über das `<amp-anim>`-Element dargestellt. Hierfür muss aber zuerst im Head ein entsprechendes JavaScript geladen werden. Anschließend kann das Animations-Tag direkt verwendet werden:

```
<script async custom-element="amp-anim" src="https://
cdn.ampproject.org/v0/amp-anim-0.1.js"></script>
...
<amp-anim width=400 height=300 src="my-gif.gif">
  <amp-img placeholder width=400 height=300
  src="my-gif-screencap.jpg">
</amp-img>
</amp-anim>
```

Über die Angabe des Platzhalters kann die Zeit während des Ladens der Animation überbrückt werden.

Videos werden über das Tag `<amp-video>` realisiert und dann wie folgt eingebunden:

```
<amp-video width=400 height=300
src="https://yourhost.com/videos/myvideo.mp4"
poster="myvideo-poster.jpg">
<div fallback>
  <p>Der Browser unterstützt kein HTML5</p>
</div>
<source type="video/mp4" src="foo.mp4">
<source type="video/webm" src="foo.webm">
</amp-video>
```

Dabei ist zu beachten, dass das Video lazy geladen wird, also erst, wenn es im oder in der Nähe des Viewports erscheint. Die Einbindung von Audio funktioniert analog:

```
<amp-audio width=400 height=300
src="https://yourhost.com/audios/myaudio.mp3">
<div fallback>
  <p>Der Browser unterstützt kein HTML5</p>
```

```

</div>
<source type=
"audio/mpeg" src="foo.mp3">
<source type="audio/ogg"
src="foo.ogg">
</amp-audio>

```

Auf alle AMP-Elemente kann das *media*-Attribut angewendet werden, um damit Media Queries zu realisieren (Listing 3). Hier wird nur das Bild geladen, das zur Breitenangabe passt.

Responsiver Text

Über das Element `<amp-fit-text>` kann Text realisiert werden, der seine Schriftgröße verändert – je nach dem zur Verfügung stehenden Platz:

```

<script async custom-element=
"amp-fit-text" src="https://
cdn.ampproject.org/v0/amp-fit-text-
0.1.js"></script>
...
<amp-fit-text width="300" height="200" layout=
"responsive"
  max-font-size="52">
  Lorem ipsum dolor sit amet, has nisl nihil
  convenire et, vim at aequa inermis reprehendunt.
</amp-fit-text>

```

Hier kann sowohl die maximale Schriftgröße (*max-font-size*) als auch die minimale (*min-font-size*) angegeben werden.

Iframes integrieren

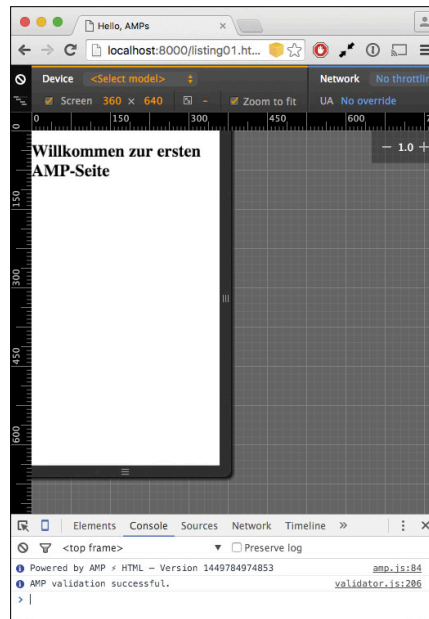
Um Iframes zu integrieren, verwendet man das `<amp-iframe>`-Tag. Dafür gibt es allerdings die Einschränkung, dass der Iframe entweder mindestens 600px oder 75 Prozent von oben in Bezug auf den ersten Viewport integriert werden

Listing 4: Iframe einbinden

```

<script async custom-element="amp-iframe"
src="https://cdn.ampproject.org/v0/
amp-iframe-0.1.js"></script>
...
<amp-iframe width=300 height=300
  sandbox="allow-scripts allow-same-origin
  allow-popups allow-popups-to-escape-sandbox"
  layout="responsive"
  frameborder="0"
  src="https://www.google.com/maps/embed/v1/place?
key=AizaSyDG9YXInKBhqclZizcSzJ0R0iE0qgVfwzI&q=
Alameda,%20CA">
</amp-iframe>

```



Syntax-Check ohne Fehler (Bild 4)

muss. Ressourcen darin müssen zudem per HTTPS angefordert werden.

Um Iframes zu laden, muss zudem zuerst ein zugehöriges JavaScript eingebunden werden – anschließend kann der Iframe integriert werden (Listing 4).

YouTube-Integration

Die Integration von YouTube-Videos ist möglich, benötigt aber ebenfalls ein eigenes JavaScript zum Betrieb. Über das Attribut *data-video-id* wird schlicht die YouTube-ID übermittelt:

```

<script async
custom-element="amp-youtube"
src="https://cdn.ampproject.org/v0/
amp-youtube-0.1.js"></script>
...
<amp-youtube
  data-video-id="mGENRkrdogY"
  layout="responsive"
width="480" height="270"></amp-youtube>

```

Um Seitenaufrufe zu zählen, kann über das Tag `<amp-pixel>` ein Tracking-Pixel-ähnliches Verhalten simuliert werden – es wird ein *GET*-Request zum angegebenen URL ausgeführt:

```

<amp-pixel src="https://foo.com/pixel?RANDOM">
</amp-pixel>

```

Als Parameter hinter dem URL können die folgenden verwendet werden: ▶

Listing 5: Ad-Integration

```

<amp-ad width=300 height=250
  type="a9"
  data-aax_size="300x250"
  data-aax_pubname="test123"
  data-aax_src="302">
</amp-ad>

<amp-ad width=320 height=50
  type="doubleclick"
  json="{...}">
</amp-ad>

<amp-ad width=300 height=250
  type="adtech" src="https://adserver.adtechus.com/ad-
dyn/3.0/5280.1/2274008/0/-1/ADTECH;size=300x250;
key=plumber;alias=careerbear-ros-
middle1;loc=300;;target=_blank;grp=27980912;
misc=3767074">
</amp-ad>

```

- **RANDOM**: Erzeugt eine zufällige Zahl.
- **CANONICAL_URL**: Canonical-URL (Host + Pfad) des aktuellen Dokuments.
- **CANONICAL_HOST**: Canonical-Host des aktuellen Dokuments.
- **CANONICAL_PATH**: Canonical-Pfad des aktuellen Dokuments.
- **DOCUMENT_REFERRER**: Aktueller Referrer.
- **TITLE**: Inhalt des Title-Tags.
- **AMPDOC_URL**: URL des aktuellen Dokuments.
- **AMPDOC_HOST**: Host des aktuellen Dokuments.
- **PAGE_VIEW_ID**: String, der zufällig aus User, URL und Tag errechnet wird.

Ads werden über das `<amp-ad>`-Tag eingebracht. Über das Attribut `type` wird der Ad-Anbieter spezifiziert. Folgende Anbieter werden zurzeit unterstützt: A9, AdReactor, AdSense, Adtech und DoubleClick (Listing 5).

Twitter-Tweets anzeigen

Über die Extension `amp-twitter` können auch Tweets angezeigt werden. Dafür sind sowohl ein entsprechendes JavaScript als auch das zugehörige Tag notwendig:

```
<script async custom-element="amp-twitter" src="https://cdn.ampproject.org/v0/amp-twitter-0.1.js"></script>
...
<amp-twitter width=486 height=256 layout="responsive"
data-tweetID="585110598171631616" data-cards="hidden">
</amp-twitter>
```

In AMP wurden zudem einige UI-Elemente wie Lightbox (`<amp-lightbox>`) und Carousel (`<amp-carousel>`) integriert (Listing 6).

Styling im Head-Bereich

Es ist nur ein Stylesheet erlaubt, das zudem auch noch inline im Code (genauer im Head) platziert werden muss. Dafür ist zudem das Attribut `amp-custom` zu verwenden:

```
<head>
  <style amp-custom>
    body {
```

Listing 6: Lightbox & Carousel

```
<script async custom-element="amp-lightbox"
src="https://cdn.ampproject.org/v0/amp-lightbox-0.1.js"></script>
<script async custom-element="amp-lightbox"
src="https://cdn.ampproject.org/v0/amp-carousel-0.1.js"></script>
...

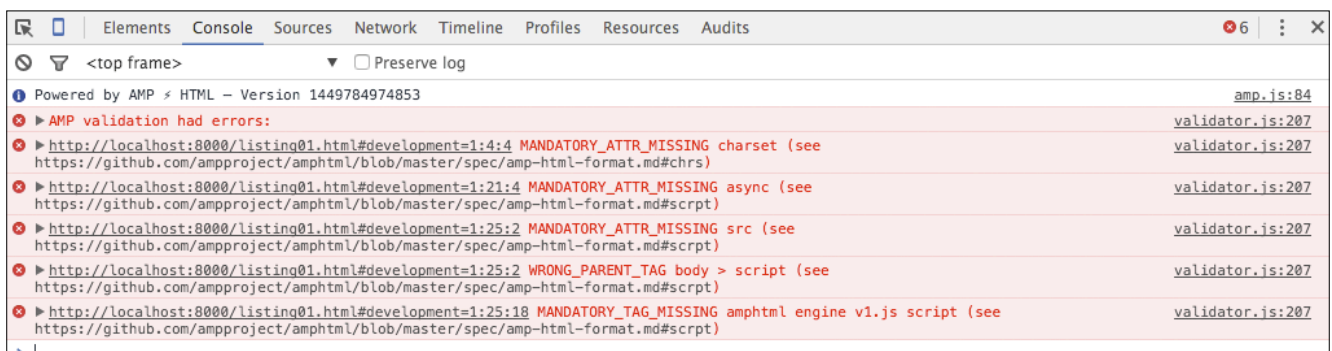
<button on="tap:my-lightbox">Lightbox öffnen
</button>

<amp-lightbox id="my-lightbox" layout="nodisplay">
  <div class="lightbox">
    <amp-img src="full-image.jpg" width=300
      height=800 on="tap:my-lightbox.close">
    </div>
  </amp-lightbox>

<amp-carousel width=300 height=400>
  <amp-img src="img1.png" width=300 height=400>
</amp-img>
  <amp-img src="img2.png" width=300 height=400>
</amp-img>
  <amp-img src="img3.png" width=300 height=400>
</amp-img>
</amp-carousel>
```

```
background-color: white;
}
amp-img {
  background-color: gray;
  border: 1px solid black;
}
</style>
</head>
```

Grundsätzlich verhält sich eine AMP-HTML-Seite wie jede andere HTML-Seite und kann daher ganz normal angezeigt



Syntax-Check mit Fehlern (Bild 5)

werden. Man kann sie entweder direkt in einen Browser ziehen – dann werden allerdings keine AJAX-Requests wie das Laden von Videos ausgeführt – oder auf einen Webserver laden und per Browser ansehen. Der schnellste Weg hierzu ist, im Verzeichnis, in dem sich die Quelldateien befinden, den folgenden Befehl auf der Kommandozeile auszuführen: `python -m SimpleHTTPServer`. Der Aufruf erfolgt dann via `localhost:8000`.

Für die Validierung müssen Sie Chrome als Browser verwenden, dort in die Developer-Tools gehen und den Tab *Console* auswählen. Nun hängen Sie den Parameter `#development=1` an den URL – also zum Beispiel `http://localhost:8000/listing01.html#development=1`. Eventuelle Parsing-Fehler werden nun direkt in der Konsole angezeigt (Bild 4). Die Lösung für eventuelle Fehler wird zudem ebenfalls in der Konsole ausgegeben (Bild 5):

- **MANDATORY_ATTR_MISSING** `<attr>`: Das Attribut `<attr>` wurde vergessen.
- **WRONG_PARENT_TAG** `<tag1> > <tag2>`: `<tag2>` ist innerhalb von `<tag1>` nicht erlaubt.
- **MANDATORY_TAG_MISSING** `<tag>`: Das Tag `<tag>` muss angegeben werden.
- **DISALLOWED_TAG** `<tag>`: Das Tag `<tag>` ist nicht erlaubt.
- **INVALID_ATTR_VALUE** `<attr>`: Das Attribut `<attr>` ist nicht gültig.

Oftmals hat man zwei Versionen einer Seite – eine AMP- und eine Nicht-AMP-Version für einen News-Artikel (Bild 6). Wenn nun Google die Nicht-AMP-Version entdeckt – wie kann man der Suchmaschine dann mitteilen, dass es noch eine AMP-Version gibt? Dies wird dadurch gelöst, dass man in beide Versionen des News-Artikels ein entsprechendes

Links zum Thema

- AMP-Projekt
<https://www.ampproject.org>
- AMP-Überblick
<https://www.youtube.com/watch?v=SOx1XfOjPI>
- Google AMP
<http://g.co/amp>
- AMP-Anatomie
<https://www.youtube.com/watch?v=hVRkG1CQScA>
- AMP-Dokumentation
https://www.ampproject.org/docs/get_started/about-amp.html
- AMP auf GitHub
<https://github.com/ampproject/amphtml>
- AMP-Beispiele
<https://github.com/ampproject/amphtml/tree/master/examples>
- Stackoverflow (Tag AMP)
<https://stackoverflow.com/questions/tagged/amp-html>

`<link>`-Tag mit der Referenz auf die jeweils andere Version einbaut, zum Beispiel so:

```
<link rel="amphtml" href="https://www.example.com/url/to/amp/document.html">
<link rel="canonical" href="https://www.example.com/url/to/full/document.html">
```

Das erste Tag wird in eine nicht-AMP Version eingebaut und das zweite in die AMP-Version. Hat man nur eine Version, dann verwendet man dort die zweite Variante, die auf sich selbst zeigt.

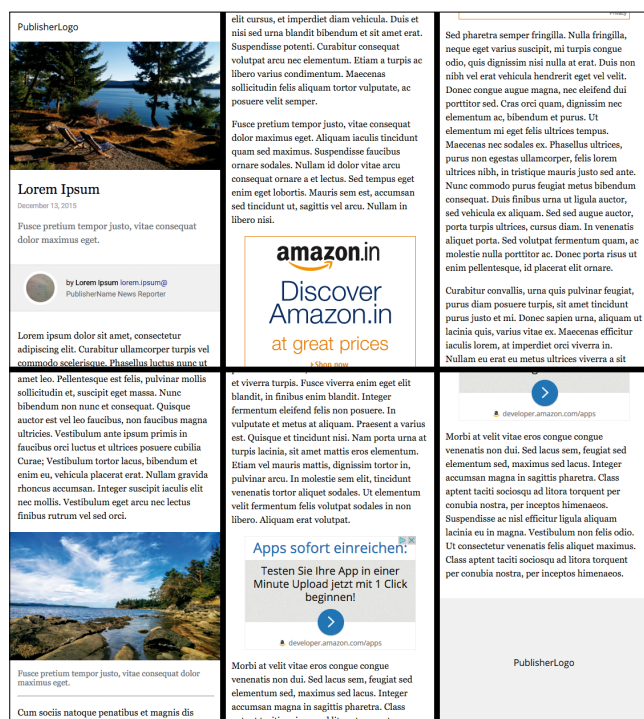
Fazit

Einige der Extensions, wie Analytics (zum Beispiel zur Einbindung von Google Analytics), Brightcove, Mustache, Pinterest, Slides, Vine und Ähnliche sind noch in der Entwicklung. Zudem ist damit zu rechnen, dass diese Liste noch deutlich länger wird. Man kann aber jetzt schon sagen, dass der Ansatz, den Google mit AMP geht, sicherlich der richtige ist. ■



Patrick Lobacher

ist Digital Native, Entwickler, Berater, Trainer, Coach und Autor zahlreicher Fachbücher und Fachartikel. Er ist Vorstandsvorsitzender der Pluswerk AG, die digitale Kommunikationslösungen konzipiert, umsetzt und betreut.



Darstellung eines Artikels (Bild 6)

UNTERLASSUNGSANSPRÜCHE

Fremde Rechtspositionen

Voraussetzungen für Unterlassungsansprüche von Geschädigten.

Rechtsverstoß, Abmahnung, Unterlassung, Schadensersatz – diese Aufzählung beschreibt stichwortartig die gängige Abfolge, wenn es zu einem Eingriff in fremde Rechtspositionen gekommen ist und dieser auch nicht unentdeckt geblieben ist. Um als Geschädigter seine Ansprüche auf Unterlassung, Auskunft oder auch Schadensersatz geltend machen zu können, gibt es neben den jeweiligen spezialgesetzlichen Grundlagen zwei wesentliche Instrumente: die Abmahnung sowie das Gesetz über Unterlassungsklagen bei Verbraucherrechts- und anderen Verstößen oder kurz: Unterlassungsklagengesetz (UKlaG).

Abmahnung

Eine Abmahnung ist eine Aufforderung, eine tatsächliche oder eine vermutete Rechtsverletzung zu unterlassen. Sie ist sozusagen die gelbe Karte für den Rechtsverletzer – der Gang vor Gericht wäre in dieser Hinsicht dann also die rote Karte. Generell gibt es mehrere Bereiche, in denen eine Abmahnung erfolgen kann. Dazu gehören zum Beispiel:

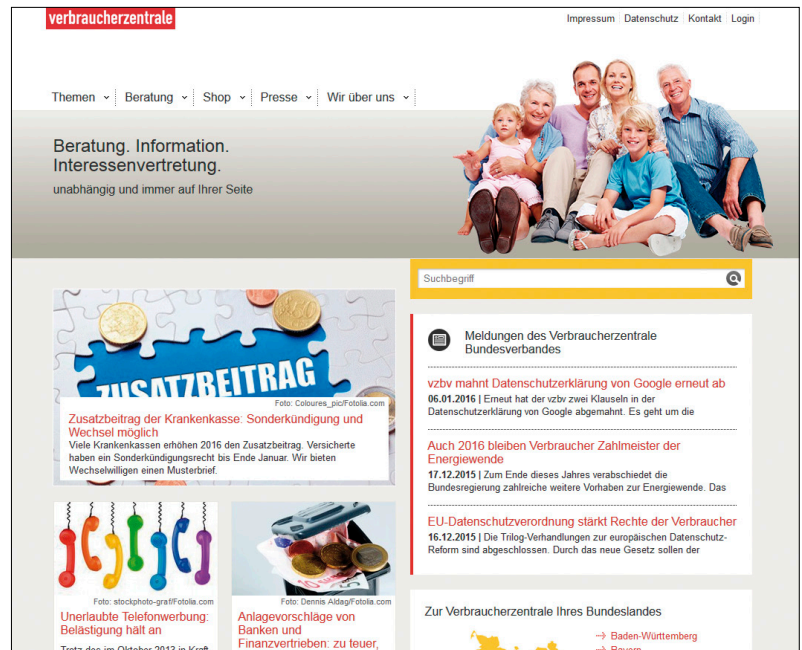
- Urheberrechtsverletzungen,
- Verletzung von Markenrechten,
- Wettbewerbsrechtsverstöße,
- rechtswidrige Äußerungen oder auch
- Verstöße gegen arbeitsvertragliche Pflichten.

Eine Abmahnung wird in aller Regel schriftlich gegenüber dem (potenziellen) Schädiger ausgesprochen. Allerdings kann eine Abmahnung zum Beispiel auch via E-Mail oder mündlich erfolgen, die Form ändert nichts an ihrer Wirksamkeit. Der Abmahnende wird jedoch schon aus Nachweisgründen in aller Regel die Schriftform wählen. Außerdem besteht kein Anwaltszwang, sodass jeder prinzipiell selbst eine Abmahnung aussprechen kann. Aufgrund der teilweise sehr komplexen Rechtslage ist davon jedoch abzuraten, da schon beim formellen Rahmen unterschiedliche Fehler passieren können.

Die Frage, wer überhaupt abmahnberechtigt ist, hängt entscheidend vom konkreten Rechtsverstoß beziehungsweise vom jeweiligen Rechtsgebiet ab.

Berechtigung

Entgegen der weit verbreiteten Ansicht kann nicht jeder einfach so eine Abmahnung aussprechen, insbesondere sind auch Rechtsanwälte nicht dazu berechtigt. So wird zwar die



Auch Verbraucherzentralen dürfen Abmahnungen nach Wettbewerbsverstößen aussprechen (Bild 1)

überwiegende Anzahl von Abmahnungen durch Anwälte verschickt, sie tun dies jedoch nicht aus eigenem Antrieb, sondern im Auftrag ihrer Mandantschaft. Bei Urheberrechtsverletzungen ist es der Urheber beziehungsweise Rechteinhaber, der abmahnen darf. Im Rahmen eines Arbeitsverhältnisses kann der Arbeitgeber seinen Arbeitnehmer abmahnen, sofern dieser dazu einen entsprechenden Anlass gegeben hat, indem er gegen seine arbeitsvertraglichen Pflichten verstoßen hat.

Auf dem Sektor des Wettbewerbsrechts ist die Sache nicht ganz so eindeutig. Bei Verstößen gegen das Gesetz gegen den unlauteren Wettbewerb (UWG) kommen grundsätzlich verschiedene Abmahnberechtigten in Betracht.

Dabei handelt es sich einerseits um ein Konkurrenzunternehmen und andererseits um bestimmte Verbraucherorganisationen. Konkurrenz bedeutet hier: Das abmahnende Unternehmen muss in der gleichen Branche tätig, somit also auch durch den Rechtsverstoß des abgemahnten Unternehmens in seinen Rechten betroffen sein.

Im UKlaG ist festgelegt, dass das Bundesamt für Justiz eine Liste mit qualifizierten Einrichtungen führt (https://www.bundesjustizamt.de/DE/SharedDocs/Publikationen/Verbraucherschutz/Liste_qualifizierter_Einrichtungen.pdf?__blob=publicationFile&v=29). Diese Liste enthält all diejenigen Ver-

braucherorganisationen, die nach Maßgabe des UKlaG Abmahnungen nach Wettbewerbsverstößen aussprechen dürfen. Dort finden sich unter anderem der ADAC, der Bund der Versicherten, BUND e.V., die Deutsche Umwelthilfe, der Deutsche Mieterbund, die Schutzvereinigung für Anleger oder auch die Verbraucherzentralen der einzelnen Bundesländer (Bild 1).

Im Falle einer Abmahnung von einem Verbraucherschutzverein et cetera sollte also ein kurzer Blick in die Liste der qualifizierten Einrichtungen gemäß UKlaG geworfen werden.

Rechtsmissbrauch

Nicht jede Abmahnung ist auch gleichsam zulässig und wirksam. Neben anderen Voraussetzungen darf in der Abmahnung kein rechtsmissbräuchliches Verhalten liegen. § 8 Abs. 4 UWG sagt dazu Folgendes:

»Die Geltendmachung der in Abs. 1 bezeichneten Ansprüche ist unzulässig, wenn sie unter Berücksichtigung der gesamten Umstände missbräuchlich ist, insbesondere wenn sie vorwiegend dazu dient, gegen den Zuwiderhandelnden einen Anspruch auf Ersatz von Aufwendungen oder Kosten der Rechtsverfolgung entstehen zu lassen.«

Leider hilft der Blick ins Gesetz nur bedingt weiter, denn die Rechtsprechung hat verschiedene Kriterien herausgearbeitet, die sich nicht im Gesetz finden, aber dennoch für das Vorliegen eines Rechtsmissbrauchs sprechen. Dazu gehören unter anderem folgende:

- vollkommen überzogener Streitwert,
- Forderung von nicht erstattungsfähigen Anwaltsgebühren,
- nach Hamburger Brauch formulierte Unterlassungserklärung wird nicht akzeptiert,
- Bestimmung des gleichen Tages für die Frist der Abgabe der Unterlassungserklärung und für die Erstattung der Abmahnkosten,
- Übermittlung einer Vollmachtskopie anstelle des Originals,
- Anzahl gleicher oder ähnlicher Abmahnungen,
- Verwendung von Textbausteinen beziehungsweise Mustertexten im Abmahnschreiben,
- ungewöhnlich gestaltete Gebührenvereinbarung zwischen Abmahnendem und seinem Anwalt,
- räumliche Entfernung zwischen Abmahnendem und seinem Anwalt,
- entsprechende Spezialisierung des abmahnenden Anwalts,
- Abzielen auf reines Gebührenerzielungsinteresse,
- Missverhältnis zwischen tatsächlichen Umsätzen des Abmahnenden und seinem Kostenrisiko aufgrund der Abmahnungen,
- die Vernachlässigung des Ladengeschäfts des Abmahnenden,
- fragliches beziehungsweise weit ausgelegtes Wettbewerbsverhältnis.

Liegen einzelne der genannten Punkte vor, bedeutet das nicht automatisch, dass eine unzulässige Abmahnung vorliegt. Je mehr hier allerdings bejaht werden kann, desto eher kann von einem Rechtsmissbrauch ausgegangen werden. Allerdings ist die Auflistung natürlich nicht abschließend. Das

Links zum Thema

- Video-Trainings des Autors
www.video2brain.com/de/trainer/michael-rohrlich
- Blog des Autors zum Thema Online-Recht für Webmaster
<http://webmaster-onlinerecht.de>
- Blog des Autors zum Verbraucherrecht online
<http://verbraucherrechte-online.de>
- One-Pager des BMJV
www.bmjbv.de/SharedDocs/Downloads/DE/PDF/Verbraucherschutz/OnePager/11192915_OnePager-Datenschutzhinweise.pdf?__blob=publicationFile&v=2

Oberlandesgericht Hamm hat beispielsweise das Versenden von 43 Abmahnung innerhalb von einer Woche als rechtsmissbräuchlich eingestuft (Urteil vom 15. September 2015, Aktenzeichen: 4 U 105/15).

Datenschutz

Am 17. Dezember 2015 hat das Plenum des Deutschen Bundestages ein Gesetz zur Änderung des UKlaG beschlossen. Künftig können von den dort genannten qualifizierten Einrichtungen auch Verstöße gegen Vorschriften abgemahnt werden:

»welche die Zulässigkeit regeln

- a) der Erhebung personenbezogener Daten eines Verbrauchers durch einen Unternehmer oder
- b) der Verarbeitung oder der Nutzung personenbezogener Daten, die über einen Verbraucher erhoben wurden, durch einen Unternehmer,

wenn die Daten zu Zwecken der Werbung, der Markt- und Meinungsforschung, des Betreibens einer Auskunft, des Erstellens von Persönlichkeits- und Nutzungsprofilen, des Adresshandels, des sonstigen Datenhandels oder zu vergleichbaren kommerziellen Zwecken erhoben, verarbeitet oder genutzt werden.«

Bei Lichte betrachtet bedeutet dies, dass nach dem Inkrafttreten dieser Gesetzesnovelle auch Verstöße gegen das Datenschutzrecht zum Beispiel von den Verbraucherzentralen abgemahnt werden können. In bestimmten Fällen ist dies bereits jetzt der Fall, wenn man nämlich Datenschutzerklärungen als Allgemeine Geschäftsbedingungen einstuft und diese Fehler aufweisen. In Zukunft können aber auch interne Datenverarbeitungsprozesse abgemahnt werden. ■



Michael Rohrlisch

ist Rechtsanwalt und Fachautor aus Würselen. Seine beruflichen Schwerpunkte liegen auf dem Gebiet des Online-Rechts und des gewerblichen Rechtsschutzes.

www.rechtssicher.info

ARBEITSMARKT

TRENDS UND JOBS FÜR ENTWICKLER

Monatliches Ranking

Gleiche Arbeit, mehr Geld

Innerhalb von Deutschland gibt es ein beträchtliches Gehaltsgefälle – selbst dann, wenn man nur die Großstädte in die Berechnung mit einbezieht, wie das die Arbeitsmarktstudie 2015 von Robert Half getan hat. Genau im Durchschnitt (100 Prozent) werden demnach Angestellte in Düsseldorf bezahlt. Besonders mager fallen die Gehälter in Berlin aus, wo lediglich 84 Prozent des durchschnittlichen Gehalts überwiesen werden. Am besten bezahlt wird in Köln, Frankfurt und München, wo 104 Prozent des Durchschnittsgehalts aufs Konto fließen. Wie **Bild 1** zeigt, kann sich ein Umzug für einen Entwickler lohnen, der in Berlin arbeitet und dort 3780 Euro monatlich verdient – zumindest aus dieser Perspektive. Er könnte in den am besten bezahlenden Städten 900 Euro monatlich mehr bekommen. Was aufgrund der

höheren Lebenshaltungskosten dann am Monatsende wirklich übrigbleibt, steht auf einem anderen Blatt.

Die aktuelle Auswertung der in der Datenbank der Meta-Jobsuchmaschine Jobkralle.de verzeichneten Stellenangebote hat ergeben, dass die meisten Jobs für Webentwickler derzeit in Bayern angeboten werden (**Bild 2**). 3276 Treffer lieferte die Suche für das Bundesland, wobei die parallel durchgeführte Suche für Großstädte zeigte, dass knapp mehr als 2000 dieser Treffer sich auf für München ausgeschriebene Jobs beziehen und weitere 336 Nürnberg betreffen. Auf den Plätzen zwei und drei im Länderranking liegen Baden-Württemberg und NRW dicht beieinander. Berlin folgt dem Spitzentrio mit gehörigem Abstand.

Jobs für Mobile-Entwickler

Die Verteilung der Jobs für Entwickler mobiler Apps zeigt ein sehr ähnliches Bild: Vorne liegt Bayern mit 19,2 Prozent der

Treffer, gefolgt von NRW (17,2 Prozent), Baden-Württemberg (15,3 Prozent) und Berlin (12,9 Prozent). Im Städteranking liegt München vor Berlin, Hamburg und Köln.

Technologien

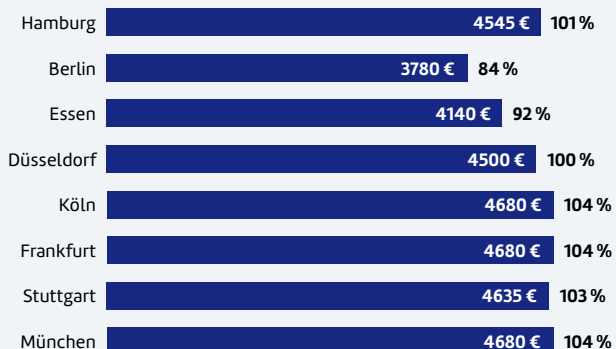
Im monatlichen Technologien-Ranking konnte sich HTML5 diesmal auf den dritten Platz verbessern. Ebenfalls höher notiert sind die Mobilgeräte-Betriebssysteme Android (jetzt auf Platz 5) und iOS auf Rang 8. Deutlich abwärts ging die Trefferzahl dagegen für das Suchwort Responsive Web. Die bei Jobkralle.de erreichten 1933 Treffer genügten nur noch für Rang 9 in der Tabelle. Auch das Suchwort Windows 10 kommt in Stellenanzeigen nicht mehr so häufig vor wie noch im Vormonat. Die Cloud-Technologie Azure hat diesmal den Sprung in die Top 16 geschafft, nachdem es im Vormonat nur für Rang 17 reichte. Herausgefallen ist dafür Microsofts Webtechnologie ASP.NET.

Tabelle 1

Rang	Technologie	Anteil *
1	Cloud	15,8 %
2	MySQL	12,7 %
3	HTML5	9,9 %
4	SharePoint	9,4 %
5	Android	7,0 %
6	Big Data	6,6 %
7	Microsoft SQL Server	6,4 %
8	iOS	6,4 %
9	Responsive Web	5,6 %
10	CSS3	4,5 %
11	Angular.js	4,4 %
12	Windows 10	3,1 %
13	WPF	2,7 %
14	NoSQL	2,5 %
15	WCF	1,7 %
16	Azure	1,3 %

* Prozentualer Anteil der Treffer

Gehaltsgefälle in Deutschland

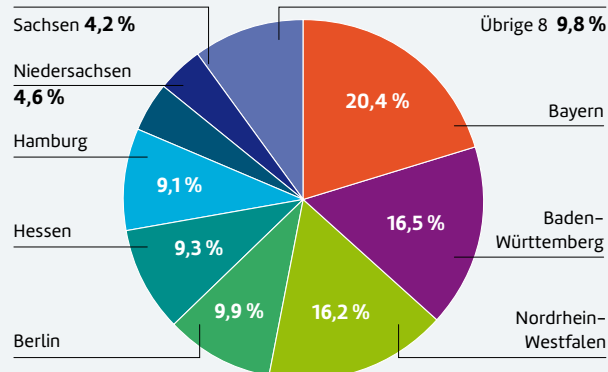


Wer in Berlin 3.780 Euro monatlich bekommt, könnte in Köln, Frankfurt oder München 900 Euro mehr erhalten (**Bild 1**)

web & mobile developer 3/2016

Quelle: Arbeitsmarktstudie von Robert Half

Jobs für Webentwickler



In Bayern, Baden-Württemberg und NRW werden derzeit die meisten Jobs für Webentwickler angeboten (**Bild 2**)

web & mobile developer 3/2016

Datenquelle: Jobkralle.de

Zahl des Monats

In den Münchner HighLight Towers sollen demnächst rund **1.000** Entwickler, Berater, Forscher und Designer tätig sein. IBM macht München damit zum weltweiten IoT-Hauptquartier. Es ist die größte Investition von IBM in Europa seit mehr als 20 Jahren.

Ein Job bei Apple

Die Welt verändern

Apple-Chef Tim Cook hat Charlie Rose vom US-Sender CBS verraten, wie ein Bewerber bei Apple einen Job erhalten könne. Kurz zusammengefasst sucht der Konzern Menschen, die nicht damit zufrieden sind, wie bestimmte Prozesse ablaufen, sondern die die Welt verändern wollen. Man sucht nach Mitarbeitern, die mit Leidenschaft und Idealismus zur Sache gehen, die sich mit einem Problem auseinandersetzen und sich darauf konzentrieren würden, bis sie eine Antwort darauf fänden.

W3B-Studie

Einfach abschalten

Der Alltag wäre angenehmer, würden mehr Leute ihr Smartphone ab und zu ausschalten – das denken der aktuellen W3B-Studie zufolge rund drei Viertel der deutschen Internetnutzer. Unter den Besitzern von Smart-

phones sind es mehr als 70 Prozent, die es begrüßen würden, wenn Smartphones im Alltag weniger präsent wären. Unter Nicht-Smartphone-Besitzern beträgt dieser Anteil 82 Prozent. Die Mehrheit der Befragten sagt, dass ihnen das Smartphone den Alltag erleichtert und auch Zeit spart. Selbst die ständige Erreichbarkeit wird von etwas mehr als einem Drittel der Befragten als Vorteil wahrgenommen.

Wunscharbeitgeber

Flexible Arbeitszeit und Kinderbetreuung zählen

Studenten in Deutschland wählen ihren Wunscharbeitgeber nach sozialen Faktoren aus. Für 78 Prozent der Studenten sind flexible Arbeitszeiten sehr wichtig oder wichtig (Bild 3), gefolgt von dem Wunsch nach Unterstützung bei der Kinderbetreuung und einer betrieblichen Gesundheitsförderung.

Auch die Betreuung Angehöriger ist schon in Studentenzeiten ein Thema. Rund die Hälfte der befragten Studenten bewertete die Unterstützung des Arbeitgebers in diesem Bereich mit sehr wichtig und wichtig.

Das Interesse an CSR-Angeboten (Corporate Social Responsibility) von Unternehmen ist sehr hoch. So möchten die Befragten bereits während der Schul- und Studienzeit auf interessante Arbeitgeber aufmerksam gemacht und über sie informiert werden. Fast 90 Prozent der Befragten informieren sich durch Magazine und Websites über spannende Arbeitgeber. Lediglich zehn Prozent nutzen Apps des Arbeitgebers.

Wie wichtig eine positiv besetzte Arbeitgebermarke für Schüler und Studenten ist, zeigt auch das Umfrageergebnis zum Standort der Unternehmen: Für 53,3 Prozent der Befragten ist die Arbeitgebermarke wichtiger als die Attraktivität des Standorts. 89 Prozent der Studenten bevorzugen eine Bewerbung

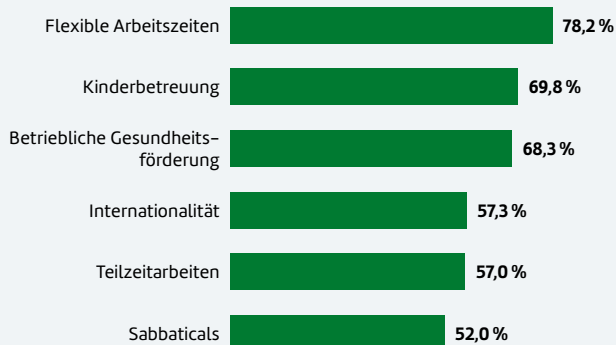
per E-Mail (Bild 4). 32 Prozent würden sich gerne mit der klassischen Bewerbungsmappe per Post bewerben, und lediglich sieben Prozent bevorzugen eine Arbeitgeber-App.

Youzu Interactive

Spiele-Entwickler gesucht

Der chinesische Mobile- und Online-Spiele-Publisher Youzu Interactive will sein Spiele-Portfolio weiter ausbauen und sucht dafür Entwickler-Teams. Der Fokus sollte dabei auf hochwertigen Games für mobile Plattformen und PC liegen, die sich an Core-Spieler richten. Mit seinen über 1500 Angestellten hat Youzu Interactive bisher mehr als 20 Spiele veröffentlicht und zählt rund 350 Millionen Spieler weltweit. Der mobile Fantasy-Titel League of Angels ist in Deutschland bereits erhältlich und hat seit seinem Start bereits mehr als 100 Millionen US-Dollar umgesetzt.

Wichtig oder sehr wichtig

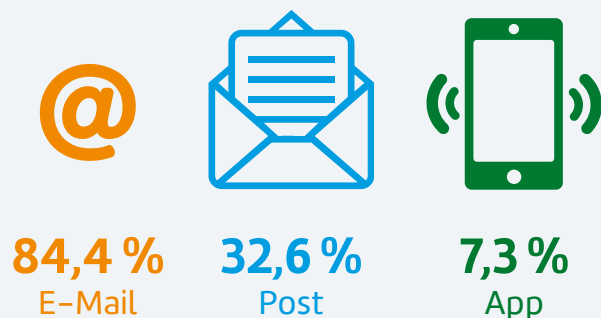


Studenten erwarten von ihrem zukünftigen Arbeitgeber vor allem flexible Arbeitszeiten sowie Kinderbetreuung (Bild 3)

web & mobile developer 3/2016

Quelle: www.csr-jobs.de, audimax

Klassisch wird bevorzugt



Die meisten Studenten möchten sich per E-Mail bewerben oder gar eine Mappe per Post verschicken (Bild 4)

web & mobile developer 3/2016

Quelle: www.csr-jobs.de, audimax

Jetzt kostenlos testen!



Das Fachmagazin für IT-Entscheider

2 Ausgaben kostenlos testen. Mit exklusivem Zugang zu unseren Digitalausgaben. Business-Newsletter inklusive.

www.com-magazin.de/gratis

Anbieterverzeichnis

für Deutschland, Schweiz und Österreich.

Consulting / Dienstleister



ANEXIA Internetdienstleistungs GmbH

Feldkirchner Straße 140
9020 Klagenfurt / AUSTRIA
T +43-50-556
F +43-50-556-500
info@anexia-it.com

ANEXIA wurde im Juni 2006 von Alexander Windbichler als klassischer Internet Service Provider gegründet. In den letzten Jahren hat sich ANEXIA zu einem stabilen, erfolgreichen und international tätigen Unternehmen entwickelt, das namhafte Kunden rund um den Globus mit Standorten in Wien, Klagenfurt, München, Köln und New York City betreut. ANEXIA bietet ihren Kunden hochwertige und individuelle Lösungen im Bereich Web- und Managed Hosting, sowie Individualsoftware und App Entwicklung.



prodot GmbH

Schifferstraße 196
47059 Duisburg
T: 0203 - 346945 - 0
F: 0203 - 346945 - 20
info@prodot.de
https://prodot.de

Intelligente Software für internationale Konzerne und mittelständische Unternehmen: prodot stärkt Kunden im weltweiten Wettbewerb – mit effizienten, stabilen und kostensenkenden Lösungen. Durch das Zusammenspiel aus Know-how, Kreativität und Qualitätsmanagement leisten wir einen Beitrag zum langfristigen Erfolg unserer Auftraggeber. Seit über 15 Jahren vertrauen uns deshalb Marktführer wie Aldi Süd, Microsoft und Siemens. prodot – People. Passion. Performance..

eCommerce / Payment



Payone GmbH & Co. KG

Fraunhoferstraße 2-4
24118 Kiel
T: +49 431 25968-400
F: +49 431 25968-1400
sales@payone.de
www.payone.de

PAYONE ist einer der führenden Payment Service Provider und bietet modulare Lösungen zur ganzheitlichen Abwicklung aller Zahlungsprozesse im E-Commerce. Das Leistungsspektrum umfasst die Zahlungsabwicklung von allen relevanten Zahlarten mit integriertem Risikomanagement zur Minimierung von Zahlungsausfällen und Betrug. Standardisierte Schnittstellen und SDKs erlauben eine einfache Integration in bestehende IT- und mobile Systemumgebungen. Über Extensions können auch E-Commerce-Systeme wie Magento, OXID eSales, Demandware, Shopware, plentymarkets und viele weitere unkompliziert angebunden werden.

Web- / Mobile-Entwicklung & Content Management



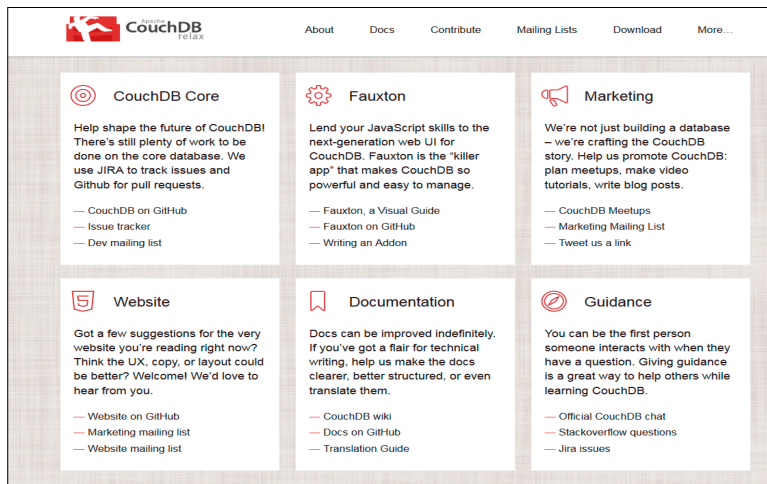
digitalmobil GmbH & Co. KG

Bayerstraße 16a, 80335 München, T: +49 (0) 89 7 41 17 760, info@digitalmobil.com, www.digitalmobil.com

In allen Fragen rund um das Dienstleisterverzeichnis berät Sie Frau Roschke gerne persönlich!
Juliane Roschke ■ 089 / 7 41 17 - 283 ■ juliane.roschke@nmg.de

Die Ausgabe 4/2016 erscheint am 10. März 2016

NoSQL als Alternative zu relationalen Datenbanksystemen



In den letzten Jahren führten neue Anwendungstypen aus den Bereichen Online Analytical Processing (OLAP), Data Mining, Big Data und vor allem aus dem mobilen Web/Internet zu einem enormen Wachstum an Datenbeständen und Anzahl von Benutzern. Web-/Internet- und Mobile-Anwendungen wie Amazon, Facebook, Google, Twitter, Instagram, YouTube, Telegram oder WhatsApp verdeutlichen diesen Sachverhalt. Für Realisierung und effizienten Betrieb derartiger Anwendungen stellten sich gängige Datenbank-Management-Systeme und die mit ihnen verbundenen Datenbanktechniken als ungeeignet heraus. Anwendungstypen mit massivem Datenaufkommen und ständig steigender Benutzeranzahl erfordern vielmehr den Einsatz gänzlich neuer Typen von Datenbanken.

Router von Angular 2

Die in den vergangenen Ausgaben der **web & mobile developer** entwickelte Angular-2-Anwendung hat mittlerweile viele Funktionen bekommen. Es wird eng auf dem zur Verfügung stehenden Bildschirm. In der nächsten Ausgabe wird es um den Router von Angular 2 gehen. In einem Artikel wird gezeigt, wie man zwischen einzelnen Komponenten navigieren kann und dabei anschauliche und für Suchmaschinen optimierte URLs erhält.

Hack Lang mit HHVM vs. PHP 7

Mit Hack, einer quelloffenen Sprache, möchte Facebook viele der Unzulänglichkeiten von PHP behoben und einen Beitrag zur Entstehung robuster Webapplikationen beigesteuert haben. Inzwischen liegt PHP in der Version 7.0 vor und die PHP-Gemeinde hat Hack klar im Visier. Der Bericht nimmt beide Sprachen, PHP 7 und Hack, unter die Lupe und beleuchtet, welche Vorteile sie jeweils haben und für welche Einsatzszenarien sie sich eignen.

Webseiten für Touchscreens

Beim responsiven Webdesign kümmert man sich hauptsächlich um die verschiedenen Größen der Bildschirme und verliert leicht aus dem Blick, dass es einen entscheidenden Unterschied macht, ob eine Webseite per Maus oder mit einem Finger bedient wird. Der Artikel zeigt praktische Techniken und neuere Entwicklung im Umgang mit Touchscreens und erläutert, welche Punkte Webdesigner dabei besonders beachten müssen.

dotnetpro



Ausgabe 3/2016 ab 18. Februar am Kiosk

Der Schwerpunkt der dotnetpro 3/2016 zeigt, wie Sie Plug-ins für Visual Studio bauen. Lernen Sie zum Beispiel, wie Sie ein spezielles Add-on entwickeln, das mit Hilfe von Roslyn dem Konstruktor übergebene Parameter automatisch auf null prüft.
www.dotnetpro.de

Unsere digitalen Angebote



Wöchentlicher Newsletter

www.webundmobile.de/newsletter-1022034.html



Shop

<https://shop.webundmobile.de>



YouTube

www.youtube.com/user/developermedia



Facebook

www.facebook.com/webundmobile



Google +

[gplus.to/webundmobile](https://plus.to/webundmobile)



Twitter

twitter.com/webundmobile

Stellenmarkt

dotnetpro + web & mobile Developer

○ 25.800 Exemplare Gesamtauflage

○ 25.300 Newsletter-Empfänger

○ 66.600 PI'S



○.NET ○Architektur ○HTML5/JavaScript ○iOS/Android ○

Kontakt:

Jens Schmidtman, Klaus Ahlering • Tel. 089/74117-125 • sales@nmg.de

Ihr Partner für mehr Online-Wachstum

Wir planen, entwickeln und steuern
Websites, Apps und Kampagnen.

Unsere Ziele sind Ihre Ziele:

- ⊕ **Mehr Sichtbarkeit**
- ⊕ **Mehr Traffic**
- ⊕ **Mehr Leads**
- ⊕ **Mehr Conversions**

.....

➔ **Mehr Kunden**

Besuchen Sie uns unter
www.digitalmobil.com

