

MATLAB® Compiler™

Web Apps



MATLAB®

R2018a



How to Contact MathWorks



Latest news: www.mathworks.com
Sales and services: www.mathworks.com/sales_and_services
User community: www.mathworks.com/matlabcentral
Technical support: www.mathworks.com/support/contact_us



Phone: 508-647-7000



The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

MATLAB® Compiler™ Web Apps

© COPYRIGHT 2018 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

Revision History

March 2018 Online only New for Version 6.6 (Release 2018a)

1 Install and Configure MATLAB Web App Server

Install MATLAB Web App Server	1-2
Configure MATLAB Web App Server	1-4
Supported Browsers and Platform Incompatibilities	1-9
Supported Browsers	1-9
Platform Incompatibilities	1-9

2 Create, Run, and Diagnose Web Apps

Create and Deploy a Web App	2-2
App Designer Prerequisite	2-2
Create a Web App	2-2
Deploy a Web App	2-3
Run a Web App	2-5
Diagnostics	2-6
Web App Session Log	2-8
Limitations and Unsupported Functionality	2-9

Security

3

Enabling HTTPS	3-2
MATLAB Web App Server Security	3-4
Potential Risks	3-6
Authoring Secure Web Apps	3-8
Securely Deploying Web Apps	3-10

Apps

4

Reference

5

Web App Examples

6

Simple Mortgage Calculator Web App	6-2
Prerequisites	6-2
Create Web App	6-2
Deploy Web App	6-3
Run Web App	6-4

Install and Configure MATLAB Web App Server

Install MATLAB Web App Server

Warning MATLAB Web App Server must be installed in a trusted intranet environment on dedicated hardware. The only purpose of the physical or virtual machine where the server is installed must be to host web apps that connect to the server. The server must never be exposed to the open Internet. For more information, see “MATLAB Web App Server Security” on page 3-4.

The MATLAB Web App Server hosts web apps packaged using the **Web App Compiler** app. For web apps to work the server must be installed and configured. The server mediates the HTTP/HTTPS communication between the client web browser and the packaged MATLAB web app. It has a home page listing all the available hosted web apps. The home page can be accessed from a browser using a URL.

To install and set up MATLAB Web App Server:

- 1 Navigate to the folder containing the MATLAB Web App Server installer.

Platform	Default Location of Installer
Windows®	C:\Program Files\MATLAB\R2018a\toolbox\compiler\deploy\win64
Linux®	/usr/local/MATLAB/R2018a/toolbox/compiler/deploy/glnxa64
macOS	/Applications/MATLAB_R2018a.app/toolbox/compiler/deploy/maci64

- 2 Run the installer to install the MATLAB Web App Server and MATLAB Runtime.

Platform	Steps
Windows (Administrator)	Right-click MATLABWebAppServerSetup.exe, and select Run as administrator .

Platform	Steps
Linux	At the shell prompt, type: <code>./MATLABWebAppServerSetup.install</code> Note On Debian® based Linux distributions, type: <code>gksudo ./MATLABWebAppServerSetup.install</code>
macOS	Double-click the app <code>MATLABWebAppServerSetup.app</code>

For information on how to start and configure the server, see “Configure MATLAB Web App Server” on page 1-4.

Note Make sure that MATLAB Runtime appears before any installed version of MATLAB in the Windows `Path` environment variable. For more information, see “MATLAB and MATLAB Runtime on Same Machine”.

See Also

More About

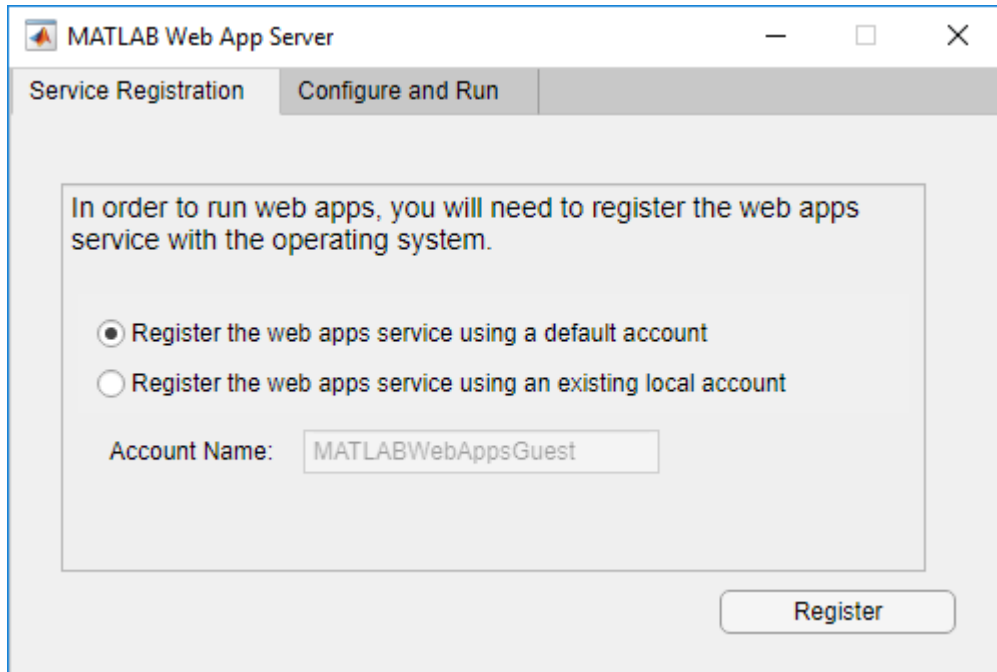
- “Configure MATLAB Web App Server” on page 1-4
- “MATLAB Web App Server Security” on page 3-4
- “Potential Risks” on page 3-6

Configure MATLAB Web App Server

To configure MATLAB Web App Server:

- 1 Navigate to the MATLAB Web App Server installation location and run the server application. For information about installing the server, see “Install MATLAB Web App Server” on page 1-2.

Platform	Steps
Windows <i>(Administrator)</i>	Default location of the server application is: C:\Program Files\MATLAB\MATLABWebAppServer\R2018a\application Right-click MATLABWebAppServer.exe and select Run as administrator .
Linux	Default location of the server application is: /usr/local/MATLAB/MATLABWebAppServer/R2018a/application At the shell prompt, type: run_MATLABWebAppServer.sh <MATLAB_Runtime_Location> <i>For example:</i> run_MATLABWebAppServer.sh /usr/local/MATLAB/MATLAB_Runtime/v94
macOS	Default location of the server application is: /Applications/MATLAB/MATLABWebAppServer/R2018a/application Double-click the app MATLABWebAppServer.app



2 Click the **Service Registration** tab, and select one of two options:

- **Register the web apps service using a default account**

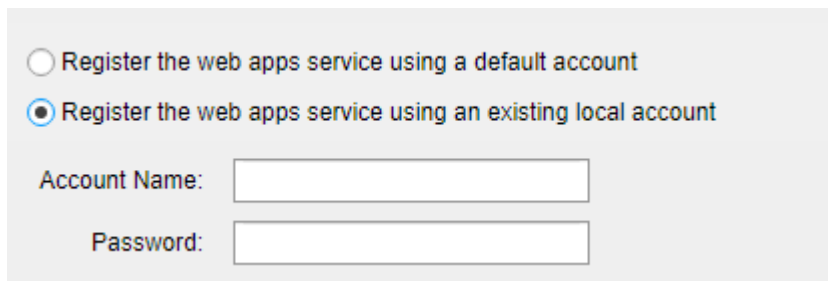
Selecting this option registers the web apps service using a default account name MATLABWebAppsGuest.

Platform	Account and Service Name
Windows	Account Name: MATLABWebAppsGuest Service Name: MATLAB Web Apps (R2018a)
Linux	Account Name: mwguest Service Name: mw-webapps-R2018a Service File: /etc/systemd/system/mw-webapps-R2018a.service

Platform	Account and Service Name
macOS	Account Name: _mwguest Service Name: mw-webapps-R2018a Service File: /Library/LaunchDaemons/ com.mathworks.mw-webapps-R2018a.plist

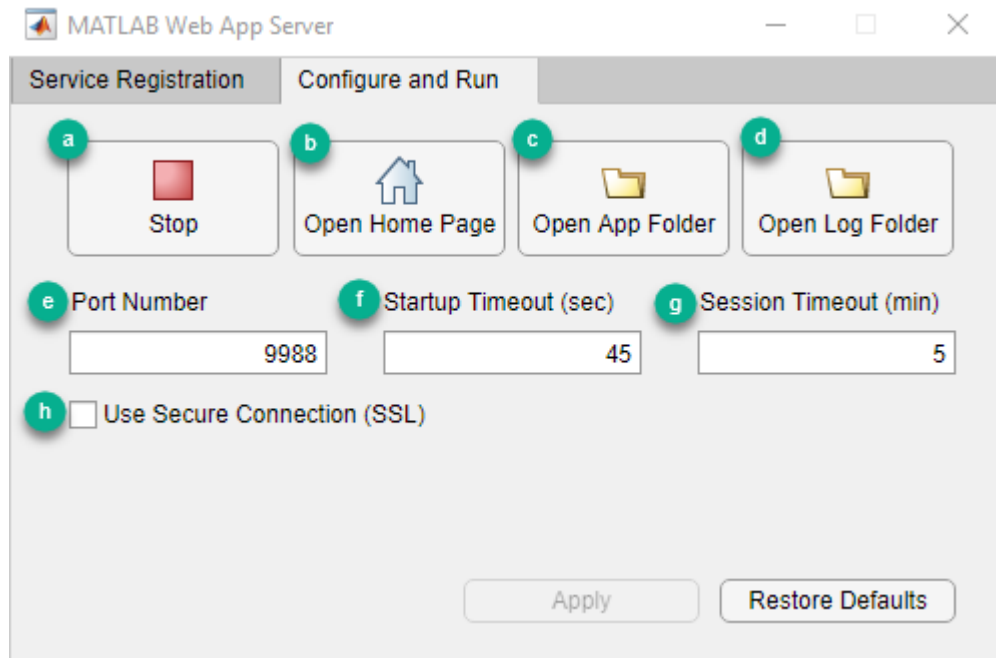
- **Register the web apps service using an existing local account**

Selecting this option registers the web apps service using an existing local account name and password.



The screenshot shows a configuration window with two radio button options. The first option, 'Register the web apps service using a default account', is unselected. The second option, 'Register the web apps service using an existing local account', is selected. Below the options are two text input fields: 'Account Name:' and 'Password:'.

- 3 You can start or stop the installed service from the the **Configure and Run** tab. The service is automatically started once the service is successfully registered. If the service does not start automatically, click **Start**.



- a** Start/Stop the server.
- b** Open the web apps home page.
- c** Open the app folder containing the web app archive (.ctf) files.
- d** Open the server log folder.
- e** Specify the port number. Default port is 9988.
- f** **Startup Timeout (sec)**: Defines the maximum time to prepare a new session for the requested app, in seconds. If server is under heavy load or hardware is not adequate, you may need to tweak this setting. Default value is 45 seconds.
- g** **Session Timeout (min)**: Defines the time interval in minutes after which session will be stopped on the server. When session is running and browser tab is opened, periodic heartbeat messages will let server know that user is still interested in the session. If session was abandoned for reasons like computer going to sleep or network disconnected, this interval will define for how long the session will be kept alive on the server. Default value is 5 minutes.
- h** Enable SSL and use HTTPS. For more information, see “Enabling HTTPS” on page 3-2.

- 4 Once you have installed and started the server, click the **Open App Folder** button to open the folder where compiled web apps are served from. Then serve your web apps by dropping their *appName*.ctf files in this folder. If the path specified for **App Path** does not exist, it is created after you start the service.
- 5 To open your web apps home page, click **Open Home Page**.

You can also configure the service by specifying:

- the port number used for your web apps
- timeout values for the session and for startup time

After making any changes, click **Apply** to save your configuration.

- 6 To stop the service, go to the **Configure and Run** tab, and click **Stop**.
- 7 To unregister the service, go to the **Service Registration** tab, and click **Unregister**.

Note The maximum number of sessions that the server supports is limited to 32.

See Also

More About

- “Install MATLAB Web App Server” on page 1-2
- “Enabling HTTPS” on page 3-2
- “MATLAB Web App Server Security” on page 3-4

Supported Browsers and Platform Incompatibilities

Supported Browsers

Web apps are compatible with Chrome™ web browsers running on Windows, macOS, and Linux desktop operating systems, and Internet Explorer® 11 on Windows desktops. For browsers such as Chrome that update automatically, the current stable version is supported.

For the best overall experience, we recommend using Google Chrome™ on a desktop. Web apps are not supported on smartphones or tablets.

Platform Incompatibilities

MATLAB Web App Server cannot be installed on Ubuntu® 14.04 LTS or Red Hat® Enterprise Linux 6.

See Also

More About

- “Install MATLAB Web App Server” on page 1-2
- “Configure MATLAB Web App Server” on page 1-4

Create, Run, and Diagnose Web Apps

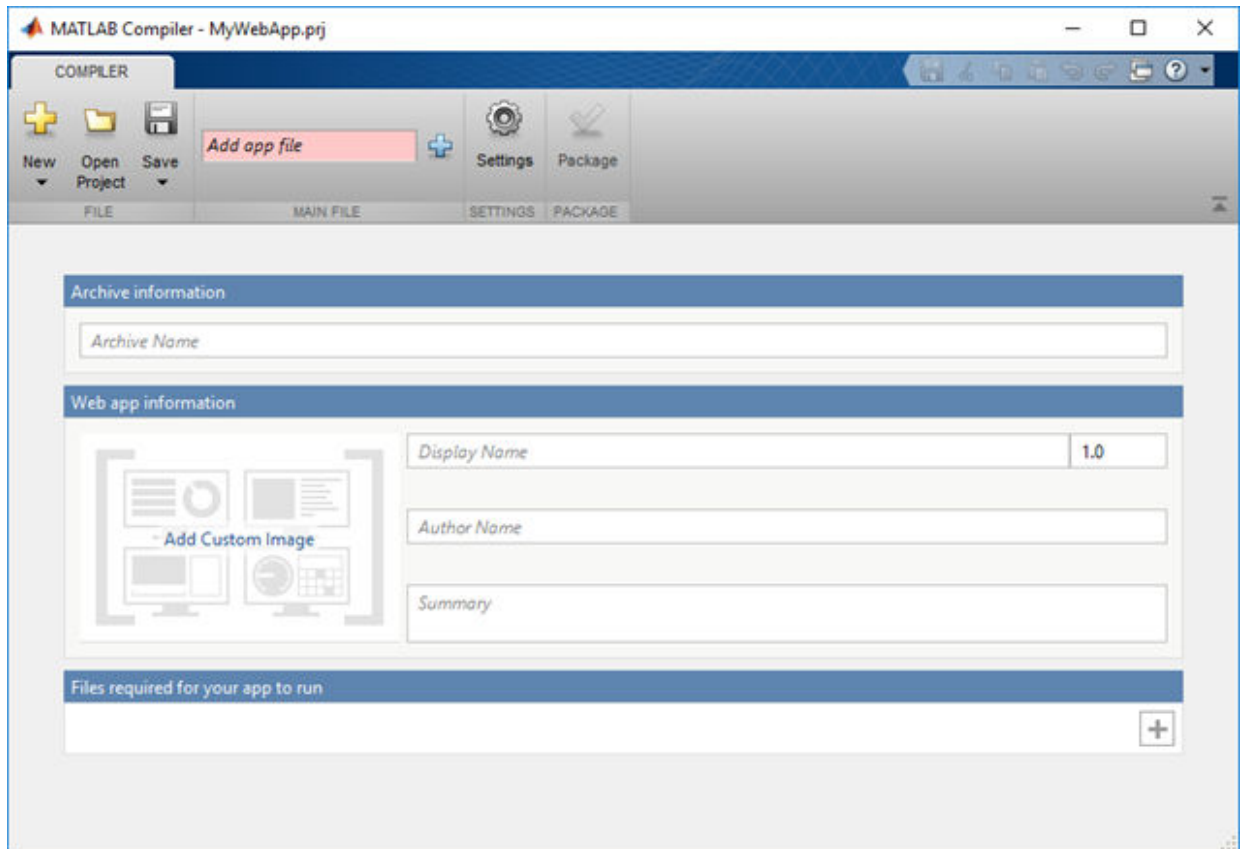
Create and Deploy a Web App


App Designer Prerequisite

Before you can deploy a web app, you need to create an app using MATLAB App Designer. For more information, see “Create a Simple App Using App Designer” (MATLAB).

Create a Web App

- 1 Run `deploytool` at the MATLAB command line, and choose **Web App Compiler**.



- 2 In the **MAIN FILE** section of the toolstrip, click the  button to add your App Designer `.mlapp` file to the project.
- 3 Edit web app information in the **Web App Information** section. Make sure to use a name that is easy to distinguish when your web app is deployed to the server. This name is visible on the web apps home page.
- 4 Change the version number for tracking purposes (optional). The version number is visible on the web apps home page.
- 5 Add a description for your web app in the **Summary** field (optional). This description is visible on the web apps home page.
- 6 Add supporting files, if any, in the **Files required for your app to run** section. This includes any MAT-files, images used by your web app, or user-written MATLAB functions not found by MATLAB Compiler™.
- 7 Click **Package** to package the app, and create a web app archive (`.ctf` file).

In the Save Project dialog box, specify the location to save the project.

- 8 In the **Package** dialog box, select **Open output folder when process completes**.

The **Web App Compiler** creates a log file `PackagingLog.txt` and two folders `for_redistribution` and `for_testing` that contain the web app archive file `webAppArchiveName.ctf`.

for_redistribution	for_testing
<code>webAppArchiveName.ctf</code>	<code>webAppArchiveName.ctf</code>
<code>readme.txt</code>	<code>mccExcludedFiles.log</code>
	<code>readme.txt</code>
	<code>requiredMCRProducts.txt</code>

You can use the log file `PackagingLog.txt` to see the exact `mcc` syntax used to package and create the web app.

- 9 To use the web app, deploy the web app archive file `webAppArchiveName.ctf`. For more information, see “Deploy a Web App” on page 2-3.

Deploy a Web App

- 1 Navigate to the `for_redistribution` folder generated by the **Web App Compiler** during the packing process.

- 2 Copy the file `webAppArchiveName.ctf` to the app folder specified by the MATLAB Web App Server.

You can access the app folder by clicking **Open App Folder** button in the MATLAB Web App Server utility.

Your web app is now deployed and can be accessed from the web apps home page. For more information, see “Run a Web App” on page 2-5.

See Also

More About

- “Run a Web App” on page 2-5
- “Simple Mortgage Calculator Web App” on page 6-2

Run a Web App

You can access web apps in a web browser by navigating to the web apps home page. You can obtain the location of the web apps home page from the MATLAB Web Apps Server. The format of the home page URL is:

```
http://webAppServer:PortNumber/webapps/home/index.html
```

Here, *webAppServer* is your web app server URL, and *PortNumber* is the port specified when configuring the server.

- 1 To run a web app, click the web app tile on the home page. By default, only valid web apps that are listed.
- 2 If some web apps are unavailable click the **Show Unavailable Apps** link in the top-right corner of the home page.
- 3 To see a list of all web apps with their status and diagnostic messages, click the **Diagnostics** link on the top-right corner of the home page.
- 4 To go back to the home page of your web apps, click the **Home** link.

See Also

More About

- “Create and Deploy a Web App” on page 2-2

Diagnostics

The web apps Diagnostics window shows the status of each web app and a diagnostic message.



App Name	Status	Message
BloodPressure	✓	OK
Fast Code Evaluator	✓	OK
DataViewer	✗	Corrupt CTF
ImportTool	✗	Not a Web App
LinuxDashboard	✗	Wrong Platform
Mortgage	✗	Wrong Runtime Version

The messages are:

- **OK** - The web app is available to run.
- **Corrupt CTF** - File cannot be read.
- **Wrong Runtime Version** - Web app uses different version of MATLAB Runtime than the one on the server.
- **Wrong Platform** - Web app cannot run on the server platform.
- **Not a Web App** - The deployed .ctf file is not a web apps .ctf file.

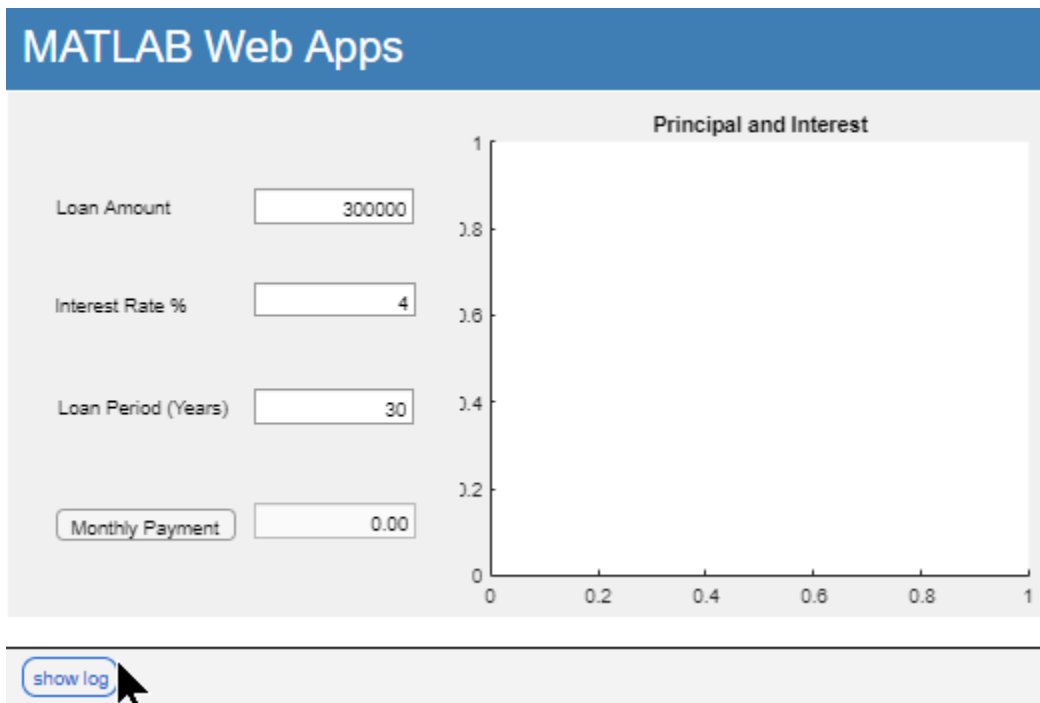
See Also

More About

- “Web App Session Log” on page 2-8
- “Run a Web App” on page 2-5

Web App Session Log

The web app session log captures all interaction associated with each web app. If you run into any issues, you can check the log for more information. You can pass this information to the server administrator for troubleshooting. You can access the log by clicking **show log** link at the bottom left corner of the browser.



See Also

More About

- “Diagnostics” on page 2-6

Limitations and Unsupported Functionality

When packaging a MATLAB app into a web app, there are functional limitations to consider. Using certain functions may result in an error or unexpected behavior. The list of limitations and unsupported functionality specific to web apps consists of:

- 1** Multiwindow apps
 - a** Multiple calls to `figure` or `uifigure` are not supported.
 - b** Dialog boxes: `dialog`, `msgbox`, `errordlg`, `warndlg`, `helpdlg`, `listdlg`, `questdlg`, `inputdlg`, `uicolor`, `uifont` are not supported. However `uialert`, `uiconfirm`, and `uiprogessdlg` are supported.
- 2** Opening files and folders on the client, or loading and saving files on the client via `uigetfile`, `uiputfile`, and `uigetdir` is not supported. As a workaround, MATLAB file I/O functions such as `readtable`, `load`, and `fread` will work if the file or folder is on a network location accessible by both the client and server.
- 3** Setting the `WindowState` property on a figure has no effect on a web app.
- 4** Printing: `print`, `printpreview`
- 5** Graphics root object properties: `MonitorPositions`, `PointerLocation`, `ScreenDepth`, `ScreenPixelsPerInch`, `ScreenSize`
- 6** Figure properties: `CloseRequestFcn`, `Position`, `InnerPosition`, `Resize`, `Visible`
- 7** System Commands: `computer`, `ispc`, `isunix`, `ismac`, and `listfonts`. While these function will execute, they will not return client-side state information. Only server-side state information is returned.

These limitations are in addition to App Designer graphics limitations. For more information, see “Graphics Support in App Designer” (MATLAB).

See Also

More About

- “Create and Deploy a Web App” on page 2-2
- “Authoring Secure Web Apps” on page 3-8

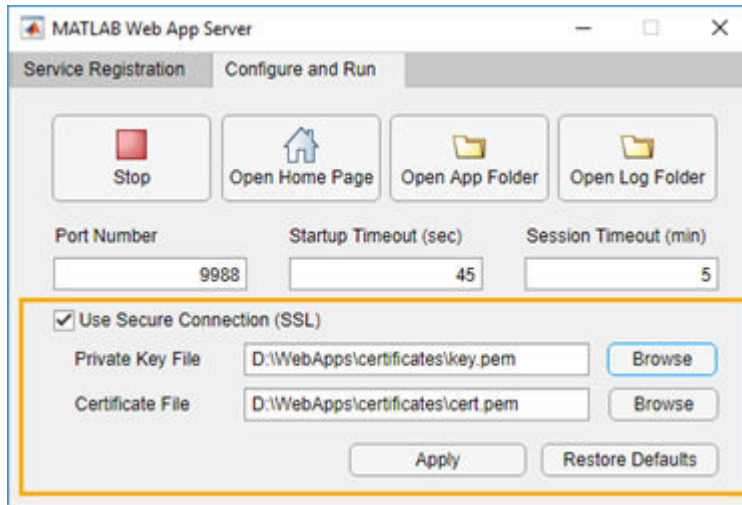
Security

Enabling HTTPS

- 1 Navigate to the MATLAB Web App Server installation location and run the server application.

Platform	Steps
Windows <i>(Administrator)</i>	<p>Default location of the application is:</p> <p>C:\Program Files\MATLAB\MATLABWebAppServer\R2018a\application</p> <p>Right-click MATLABWebAppServer.exe and select Run as administrator.</p>
Linux	<p>Default location of the application is:</p> <p>/usr/local/MATLAB/MATLABWebAppServer/R2018a/application</p> <p>At the shell prompt, type:</p> <p>run_MATLABWebAppServer.sh <MATLAB_Runtime_Location></p>
macOS	<p>Default location of the application is:</p> <p>/Applications/MATLAB/MATLABWebAppServer/R2018a/application</p> <p>Double-click the app MATLABWebAppServer.app</p>

- 2 Select the **Configure and Run** tab.



- 3 Check the option **Use Secure Connection (SSL)** in the MATLAB Web App Server.
- 4 Enter the location of the private key file and the certificate file. Click **Apply**.

The browser opens the web apps home page using HTTPS.

See Also

MATLAB Web App Server Security

Caution It is strongly recommended that you consult with your IT system administrator and discuss the security implications of installing the MATLAB Web App Server.

Installing and running the server on your network exposes your network and file system to risks. The machine running the server is most at risk from accidental or deliberate misuse of deployed web applications. Therefore, you must install the server software only on dedicated hardware. This can be a physical or virtual machine whose only purpose is to host web applications that connect to the server software. Using a physical or virtual machine limits the risk in the event that the machine is compromised.

The MATLAB Web App Server alters the security profile of the machine on which it is running. The installation process creates a server user account with low privileges. This new account has read-only permission to the app folder created during the installation of the server. However, through a process known as *privilege escalation*, attackers may be able to exploit bugs in the operating system or network to obtain the privileges of ordinary or even administrative users. They can then attempt to access files or other intellectual property without permission.

The server relies on the authentication and authorization scheme of its host machine and network. Other than supporting HTTPS, it does not contain any additional mechanisms for authenticating or authorizing web application users. For more information, see “Enabling HTTPS” on page 3-2.

You may be able to mitigate some of these risks by:

- *Restricting network access*: Only trusted users can access the server and its associated applications.
- *Executing only trusted applications*: Trust applications developed by only well-known, trusted, and authenticated sources.
- *Limiting application functionality*: Include in the application only those features of MATLAB required for the application to perform its function. For more information, see “Authoring Secure Web Apps” on page 3-8.

For a list of additional risks, see “Potential Risks” on page 3-6.

See Also

More About

- “Potential Risks” on page 3-6
- “Enabling HTTPS” on page 3-2
- “Authoring Secure Web Apps” on page 3-8
- “Securely Deploying Web Apps” on page 3-10

Potential Risks

- The MATLABWeb App Server has no specific mechanism to prevent HTTP request capture and replay.
- The server has no mechanism for authentication or authorization other than HTTPS.

Any user with access to the network can run any application created with this software and read any data the application is authorized to access.

- Both the server and applications run under the same low-privileged user account.

If multiple copies of the same application run simultaneously, they might interfere with each other. This situation happens if the application writes data to any shared resource, for example a file or a nonconcurrent database.

- Installation of the MATLAB Web App Server creates the low-privileged user account on the host machine.

This low-privileged account may inherit privileges given to all users. Care should be taken to restrict privileges given to all users.

- When deploying multiple applications to the server cookies are shared across sessions which may result in crosstalk between applications for a single user accessing more than one application.

This situation could allow unintentional crosstalk between multiple applications run by the same user.

- Deployed web applications are potentially vulnerable to *data* or *code injection* attacks whereby malicious or malformed inputs can be used to attempt to subvert the system. The server does not contain explicit protection against either type of injection attack. Certain MATLAB features, particularly the `eval()` function, can increase the risk of injection attacks. A common countermeasure is input sanitization or input whitelisting. MATLAB contains functions like `regexp()` and `regexprep()` that can assist in validating untrusted input.

- Your application may indirectly call `eval()`, potentially making it vulnerable to code-injection attacks.
- Other MATLAB functions may exhibit the same code injection vulnerabilities; any function that processes code-like input (XML, SQL, JSON, to name a few) is potentially vulnerable to code injection.
- Any application that accesses the operating system via MATLAB `system()`, `dos()`, or `unix()` commands might also be vulnerable to code injection.

Note This is a list of known risks and is not meant to be comprehensive.

See Also

More About

- “Enabling HTTPS” on page 3-2
- “Authoring Secure Web Apps” on page 3-8
- “Securely Deploying Web Apps” on page 3-10
- “MATLAB Web App Server Security” on page 3-4

Authoring Secure Web Apps

Most of the potential risk associated with deploying web apps comes from the code in each app. By limiting the features that your app uses and by following the secure coding practices listed here, you can reduce the potential risk to your applications.

- **Authentication and authorization:** If your app requires access to sensitive data or performs potentially dangerous actions, you can consider implementing your own authentication and authorization schemes. Consult your network security group for advice.
- **Do not call `eval()`:** The MATLAB `eval()` function turns text strings into commands. This powerful function allows users to execute arbitrary MATLAB code. This code can, in turn, enable execution of any installed program which is available to the low-privileged user or access to any file or data which the low-privileged users has access to. Applications created for web deployment and access must not contain calls to `eval()`. See “Alternatives to the `eval` Function” (MATLAB) for ways to eliminate `eval()` from your web app code. Relying on input sanitization (see below) can help mitigate the risk of any indirect calls to `eval()`.
- **Limit free-text user input:** Use menus, sliders, dials and buttons instead of editable text fields in your app user interface. In addition to providing a better user experience, this practice limits the types of input users can provide, and thus limits the risks such inputs might introduce.
- **Sanitize user input:** To a security expert, user supplied data is considered untrusted, since user input is a common attack vector for hackers. If your app must accept free-text input, the app must carefully examine the input for potential *code injection* attacks — text that contains special characters that coerce the app to interpret the input as commands rather than data.

In MATLAB, code injection attacks are most likely to be directed against XML, JSON, SQL and other similar types of structured input. If your app accepts structured input, consult your IT or security group for suggestions on how to sanitize that input. It is never a good idea to allow the user to directly enter any type of code (such as MATLAB, Java®, or JavaScript®) for immediate evaluation.

- **Sanitize data read from files:** Reading data from files exposes the app to the same types of risk as collecting interactive user input. The same countermeasures apply. Also, you can protect read-only data files from tampering by a cryptographically secure hashing algorithm to digitally fingerprint files.
- **Minimize file system write operations:** Limiting your application to read-only access greatly reduces the potential risk associated with your application. If you must

write to the file system, remember that the server runs multiple copies of your application simultaneously if multiple users access it at the same time. You will need to manage simultaneous writes to either through the use of runtime-generated unique filenames or the use of a database which can typically handle multiple simultaneous accesses. If simultaneous writes are not properly managed, data corruption may occur.

- **Verify trustworthiness of third-party code:** If your app includes MATLAB files, shared libraries, Java classes, or any other type of code developed by a third party, you must make sure that code is free of viruses, worms, Trojan horses, and other web-based attack and infiltration vectors. You can discuss this issue with the author of the code and your IT and security staff. In the case of binaries or Java classes, consider running a virus scanner or other security software on the code before including it in your deployed app.
- **Reduce exposure level:** One way to reduce exposure is to limit the time that the app is running to only those times when it is needed. For example, do not run it continuously from your desktop.

See Also

More About

- “Potential Risks” on page 3-6
- “Enabling HTTPS” on page 3-2
- “Securely Deploying Web Apps” on page 3-10
- “MATLAB Web App Server Security” on page 3-4

Securely Deploying Web Apps

- Install the MATLAB Web App Server on a dedicated physical or virtual machine, and do not use this machine for any other purpose.
- Run web apps behind your organization firewall. Do not allow access from the open Internet.
- Install web apps only from trusted and verified people and organizations.
- Limit the features and functionality you build into the web apps you develop.
 - Do not call the MATLAB function `eval()`.
 - Avoid free-text input where you can, and use menus, lists, buttons, and other affordances instead.
 - Sanitize input from the app user interface and data files.
 - Limit file, network, and other resource access to the minimum required by your app.
 - Verify the trustworthiness of any third-party code included in your app.
- If your application accesses sensitive data, use in-application authentication to limit access.
- Reduce exposure level by limiting the time that the app runs to only those times when it is needed. For example, do not run it 24x7 from your desktop.

See Also

More About

- “Authoring Secure Web Apps” on page 3-8
- “Potential Risks” on page 3-6
- “Enabling HTTPS” on page 3-2
- “MATLAB Web App Server Security” on page 3-4

Apps

Web App Compiler

Package App Designer apps for web deployment

Description

The **Web App Compiler** app lets you package MATLAB App Designer apps for web deployment. The app takes an App Designer `.mlapp` file and packages it as a web app archive `.ctf` file. You deploy the web app by copying the `.ctf` file to the apps folder in the MATLAB Web App Server. Once deployed, you can use the web app in a browser.

Open the Web App Compiler App

- MATLAB Toolstrip: On the **Apps** tab, under **Application Deployment**, click the Web App icon.
- MATLAB command prompt: Enter `webAppCompiler`.
- App Designer: In App Designer toolstrip, click the Share button, and select **Web App**.

Examples

- “Simple Mortgage Calculator Web App” on page 6-2

Programmatic Use

```
webAppCompiler
```

See Also

Apps

Functions

Topics

“Simple Mortgage Calculator Web App” on page 6-2

Introduced in R2018a

Reference

webAppCompiler

Package App Designer apps for web deployment

Syntax

```
webAppCompiler  
webAppCompiler project_name  
webAppCompiler -package project_name
```

Description

webAppCompiler opens the **Web App Compiler** for the creation of a new project. For more information on the app, see **Web App Compiler**.

webAppCompiler project_name opens the **Web App Compiler** with the project preloaded.

webAppCompiler -package project_name runs the **Web App Compiler** and packages the specified project.

Examples

Create a New Web App Project

Open the **Web App Compiler** to create a new project.

```
webAppCompiler
```

Open an Existing Web App Project

Open an existing web app project.


```
webAppCompiler myWebApp
```

Package a Web App using an Existing Project

Open the **Web App Compiler** to package an App Designer app as a web app using an existing project.

```
webAppCompiler -package myWebApp
```

Input Arguments

project_name — name of the project to be opened or packaged

character array or string

Specify the name of a previously saved web app project. The project must be on the current path.

See Also

deploytool | mcc

Introduced in R2018a

Web App Examples

Simple Mortgage Calculator Web App

This example shows how to create a simple mortgage amortization calculator web app. The example uses the simple calculator app from App Designer as a starting point. For information about the app and the numerical values you can enter, see “Use App Designer to Create a Simple Calculator App” (MATLAB). In the workflow, you:

- Package the simple calculator app from App Designer using the **Web App Compiler**. This creates a web app archive (.ctf) file.
- Deploy the web app by copying the web app archive (.ctf) file to the apps folder in the MATLAB Web App Server.
- Run the web app from the web apps home page.


Prerequisites

- 1 Install and configure the MATLAB Web App Server.
 - For information on installing the server, see “Install MATLAB Web App Server” on page 1-2.
 - For information on configuring the server, see “Configure MATLAB Web App Server” on page 1-4.
- 2 Copy the App Designer file `Mortgage.mlapp` to your current working directory.

Operating System	Default File Location
Windows	C:\Program Files\MATLAB\R2018a\examples\matlab_featured\Mortgage.mlapp
Linux	/usr/local/MATLAB/R2018a/examples/matlab_featured/Mortgage.mlapp
macOS	/Applications/MATLAB/R2018a.app/examples/matlab_featured/Mortgage.mlapp

Create Web App

- 1 Type `deploytool` at the MATLAB command line, and choose **Web App Compiler**.

- 2 In the **MAIN FILE** section of the toolstrip, click the  button to add the Mortgage.mlapp file to the project.
- 3 In the **Archive information** section, specify the archive name as myMortgageWebApp.
- 4 Edit web app information in the **Web app information** section. Make sure to use a display name that is easy to distinguish when your web app is deployed to the server. This name is visible on the web apps home page.
- 5 Change the version number for tracking purposes (optional). The version number is visible on the web apps home page.
- 6 Add a description for your web app in the **Summary** field (optional). This description is visible on the web apps home page.
- 7 Click **Package** to package the app and create a web app archive (.ctf) file.

In the Save Project dialog box, specify the location to save the project.

- 8 In the **Package** dialog box, select **Open output folder when process completes**.

The **Web App Compiler** app creates a log file PackagingLog.txt and two folders for_redistribution and for_testing that contain the web app archive file myMortgageWebApp.ctf.

for_redistribution	for_testing
myMortgageWebApp.ctf	myMortgageWebApp.ctf
readme.txt	mccExcludedFiles.log
	readme.txt
	requiredMCRProducts.txt

You can use the log file PackagingLog.txt to see the exact mcc syntax used to package and create the web app.

Deploy Web App

- 1 Copy the web app archive file myMortgageWebApp.ctf.
- 2 Open the MATLAB Web App Server application, and click the **Open App Folder** button.

Platform	Location
Windows <i>(Administrator)</i>	C:\Program Files\MATLAB\MATLABWebAppServer\R2018a\MATLABWebAppServer.exe
Linux	/usr/local/MATLAB/MATLABWebAppServer/R2018a/run_MATLABWebAppServer.sh <MATLAB_Runtime_Location>
macOS	/Applications/MATLAB/MATLABWebAppServer/R2018a/MATLABWebAppServer.app

- 3 Paste the web app archive file `myMortgageWebApp.ctf` in the folder that opens.
- 4 Click the **Open Home Page** button.

This action opens the web apps home page using your default web browser. You see a tile displaying the simple mortgage calculator web app. Your web app is now deployed.

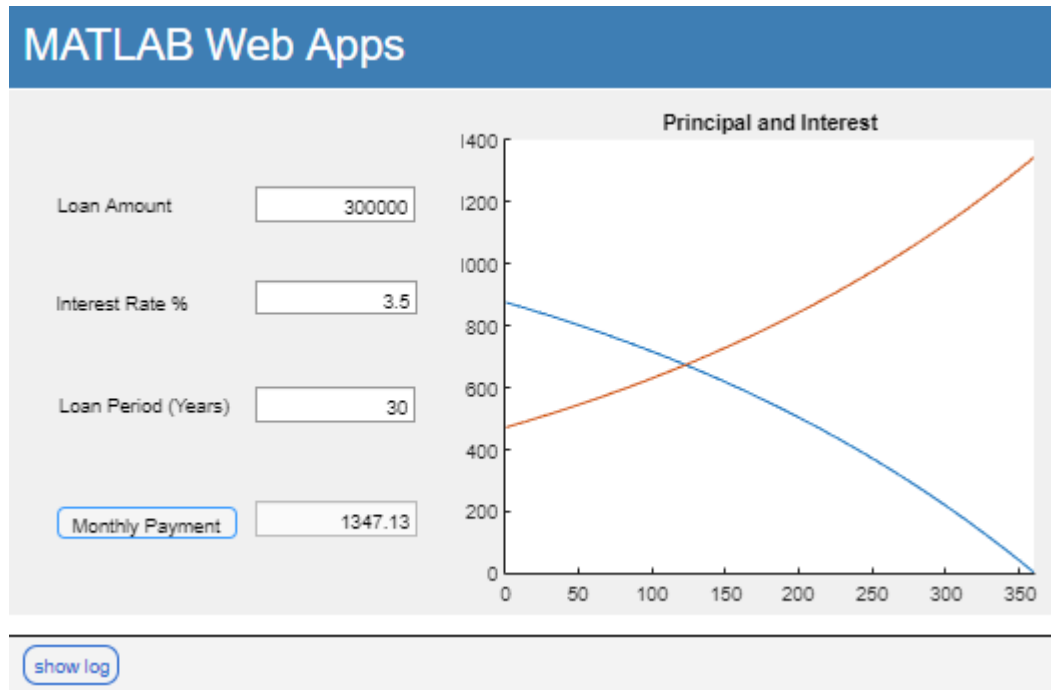
Run Web App

- 1 To run a web app, click the `myMortgageWebApp` tile.

The web app opens in a new tab.

- 2 Click the **Monthly Payment** button to get the monthly payment and the principal and interest graph.

You have successfully created, deployed, and run a web app.



See Also

More About

- “Create and Deploy a Web App” on page 2-2
- “Run a Web App” on page 2-5

