

Get started developing Excel custom functions

Article • 01/07/2025

With custom functions, developers can add new functions to Excel by defining them in JavaScript or TypeScript as part of an add-in. Excel users can access custom functions just as they would any native function in Excel, such as `SUM()`.

Prerequisites

- Node.js (the latest LTS version). Visit the [Node.js site](#) to download and install the right version for your operating system.
- The latest version of Yeoman and the Yeoman generator for Office Add-ins. To install these tools globally, run the following command via the command prompt.

command line

```
npm install -g yo generator-office
```

ⓘ Note

Even if you've previously installed the Yeoman generator, we recommend you update your package to the latest version from npm.

- Office connected to a Microsoft 365 subscription (including Office on the web).

ⓘ Note

If you don't already have Office, you might qualify for a Microsoft 365 E5 developer subscription through the [Microsoft 365 Developer Program](#); for details, see the [FAQ](#). Alternatively, you can [sign up for a 1-month free trial](#) or [purchase a Microsoft 365 plan](#).

Build your first custom functions project

To start, you'll use the Yeoman generator to create the custom functions project. This will set up your project with the correct folder structure, source files, and dependencies

to begin coding your custom functions.

1. Run the following command to create an add-in project using the Yeoman generator. A folder that contains the project will be added to the current directory.

command line

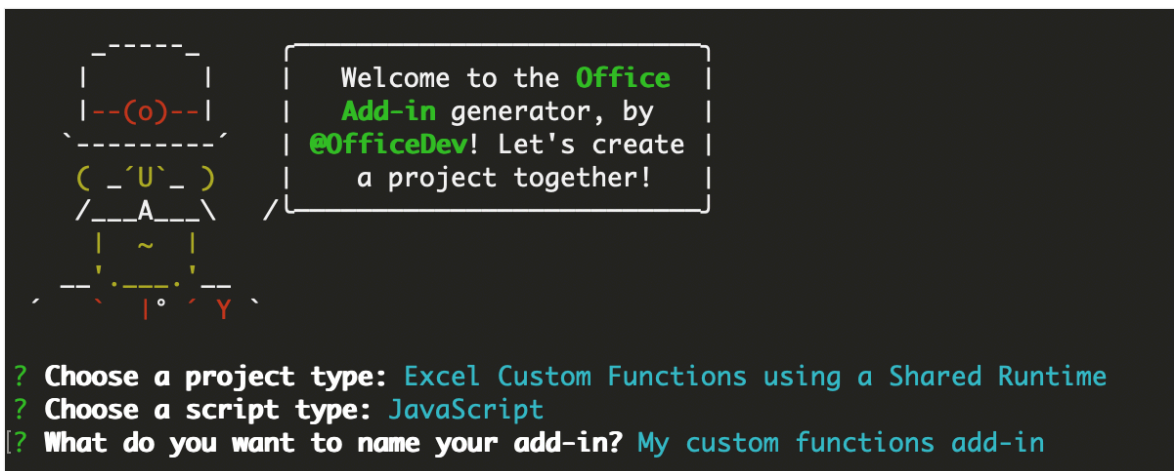
```
yo office
```

⚠ Note

When you run the `yo office` command, you may receive prompts about the data collection policies of Yeoman and the Office Add-in CLI tools. Use the information that's provided to respond to the prompts as you see fit.

When prompted, provide the following information to create your add-in project.

- **Choose a project type:** Excel Custom Functions using a Shared Runtime
- **Choose a script type:** JavaScript
- **What do you want to name your add-in?** My custom functions add-in



The Yeoman generator will create the project files and install supporting Node components.

2. The Yeoman generator will give you some instructions in your command line about what to do with the project, but ignore them and continue to follow our instructions. Navigate to the root folder of the project.

command line

```
cd "My custom functions add-in"
```

3. Build the project.

command line

```
npm run build
```

4. Start the local web server, which runs in Node.js. You can try out the custom function add-in in Excel. You may be prompted to open the add-in's task pane, although this is optional. You can still run your custom functions without opening your add-in's task pane.

Excel on the web

To test your add-in in Excel on the web, run the following command. When you run this command, the local web server will start. Replace "{url}" with the URL of an Excel document on your OneDrive or a SharePoint library to which you have permissions.

ⓘ Note

If you are developing on a Mac, enclose the {url} in single quotation marks. Do *not* do this on Windows.

command line

```
npm run start -- web --document {url}
```

The following are examples.

- ```
npm run start -- web --document https://contoso.sharepoint.com/:t:/g/EZGxP7ksiE5DuxvY638G798BpuhwluxCMfF1WZQj3VYhYQ?e=F4QM1R
```
- ```
npm run start -- web --document https://1drv.ms/x/s!jkCH7spkM4EGgcZUgqthk4IK3N0ypVw?e=Z6G1qp
```
- ```
npm run start -- web --document https://contoso-my.sharepoint-
df.com/:t:/p/user/EQda453DNTpFn1bFPh0VR0BwlrzetbXvnaRYii2lDr_oQ?
e=RSccmNP
```

If your add-in doesn't sideload in the document, manually sideload it by following the instructions in [Manually sideload add-ins to Office on the web](#).

### ⓘ Note

- Office Add-ins should use HTTPS, not HTTP, even while you're developing. If you're prompted to install a certificate after you run one of the following commands, accept the prompt to install the certificate that the Yeoman generator provides. You may also have to run your command prompt or terminal as an administrator for the changes to be made.
- If this is your first time developing an Office Add-in on your machine, you may be prompted in the command line to grant Microsoft Edge WebView a loopback exemption ("Allow localhost loopback for Microsoft Edge WebView?"). When prompted, enter **Y** to allow the exemption. Note that you'll need administrator privileges to allow the exemption. Once allowed, you shouldn't be prompted for an exemption when you sideload Office Add-ins in the future (unless you remove the exemption from your machine). To learn more, see ["We can't open this add-in from localhost" when loading an Office Add-in or using Fiddler.](#)

```
> office-addin-taskpane-js@0.0.1 start
> office-addin-debugging start manifest.xml

Debugging is being started...
App type: desktop
? Allow localhost loopback for Microsoft Edge WebView? (Y/n) Y
```

## Try out a prebuilt custom function

The custom functions project that you created by using the Yeoman generator contains some prebuilt custom functions, defined within the `./src/functions/functions.js` file. The `./manifest.xml` file in the root directory of the project specifies that all custom functions belong to the `CONTOSO` namespace.

In your Excel workbook, try out the `ADD` custom function by completing the following steps.

1. Select a cell and type `=CONTOSO`. Notice that the autocomplete menu shows the list of all functions in the `CONTOSO` namespace.

2. Run the `CONTOSO.ADD` function, using numbers `10` and `200` as input parameters, by typing the value `=CONTOSO.ADD(10,200)` in the cell and pressing `Enter`.

The `ADD` custom function computes the sum of the two numbers that you specify as input parameters. Typing `=CONTOSO.ADD(10,200)` should produce the result **210** in the cell after you press `Enter`.

If the `CONTOSO` namespace isn't available in the autocomplete menu, take the following steps to register the add-in in Excel.

#### Excel on the web

1. Select **Home** > **Add-ins**, then select **More Settings**.
2. On the **Office Add-ins** dialog, select **Upload My Add-in**.
3. Choose **Browse...** and navigate to the root directory of the project that the Yeoman generator created.
4. Select the file **manifest.xml** and choose **Open**, then choose **Upload**.
5. Try out the new function. In cell **B1**, type the text `=CONTOSO.GETSTARCOUNT("OfficeDev", "Excel-Custom-Functions")` and press `Enter`. You should see that the result in cell **B1** is the current number of stars given to the [Excel-Custom-Functions Github repository](#).

When you want to stop the local web server and uninstall the add-in, follow the applicable instructions:

- To stop the server, run the following command. If you used `npm start`, the following command also uninstalls the add-in.

command line

```
npm stop
```

- If you manually sideloaded the add-in, see [Remove a sideloaded add-in](#).

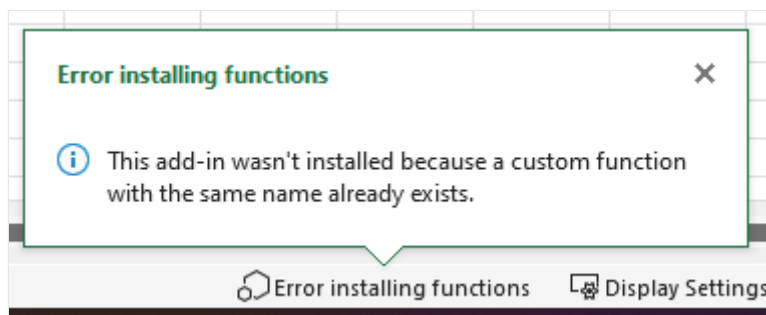
## Next steps

Congratulations, you've successfully created a custom function in an Excel add-in! Next, build a more complex add-in with streaming data capability. The following link takes you through the next steps in the Excel add-in with custom functions tutorial.

Excel custom functions add-in tutorial

## Troubleshooting

- Ensure your environment is ready for Office development by following the instructions in [Set up your development environment](#).
- Some of the sample code uses ES6 JavaScript. This isn't compatible with [older versions of Office that use the Trident \(Internet Explorer 11\) browser engine](#). For information on how to support those platforms in your add-in, see [Support older Microsoft webviews and Office versions](#). If you don't already have a Microsoft 365 subscription to use for development, you might qualify for a Microsoft 365 E5 developer subscription through the [Microsoft 365 Developer Program](#); for details, see the [FAQ](#). Alternatively, you can [sign up for a 1-month free trial](#) or [purchase a Microsoft 365 plan](#).
- The automatic `npm install` step Yo Office performs may fail. If you see errors when trying to run `npm start`, navigate to the newly created project folder in a command prompt and manually run `npm install`. For more information about Yo Office, see [Create Office Add-in projects using the Yeoman Generator](#).
- You may encounter issues if you run the quick start multiple times. If the Office cache already has an instance of a function with the same name, your add-in gets an error when it sideloads. You can prevent this by [clearing the Office cache](#) before running `npm run start` and making sure to run `npm stop` before restarting the add-in.



## See also

- [Custom functions overview](#)

- Custom functions metadata
- Runtime for Excel custom functions
- Using Visual Studio Code to publish