

Build your first Outlook add-in

Article • 04/30/2025

In this article, you'll walk through the process of building an Outlook task pane add-in using Yo Office that displays at least one property of a selected message.

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](#).

- Outlook on the web, [new Outlook on Windows](#), Outlook 2016 or later on Windows (connected to a Microsoft 365 account), or Outlook on Mac.

Create the add-in project

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.

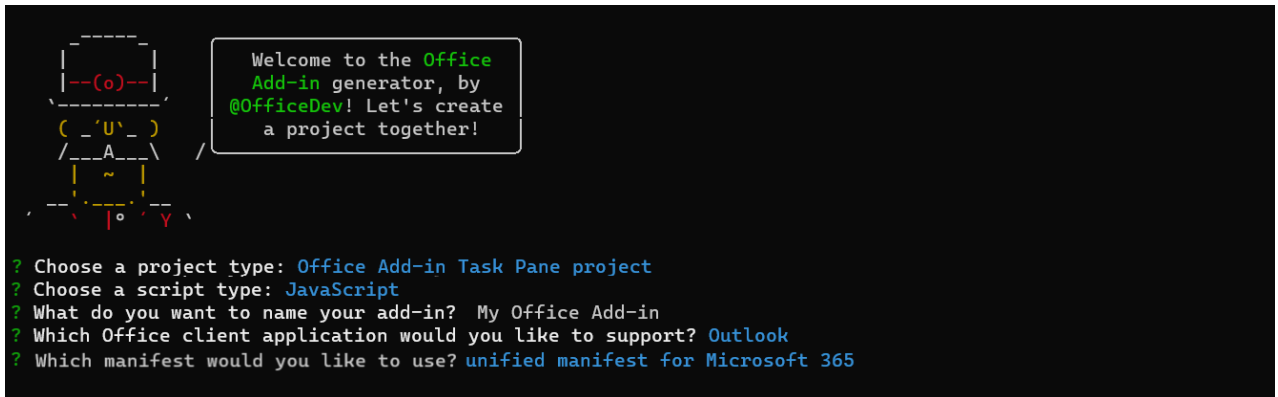
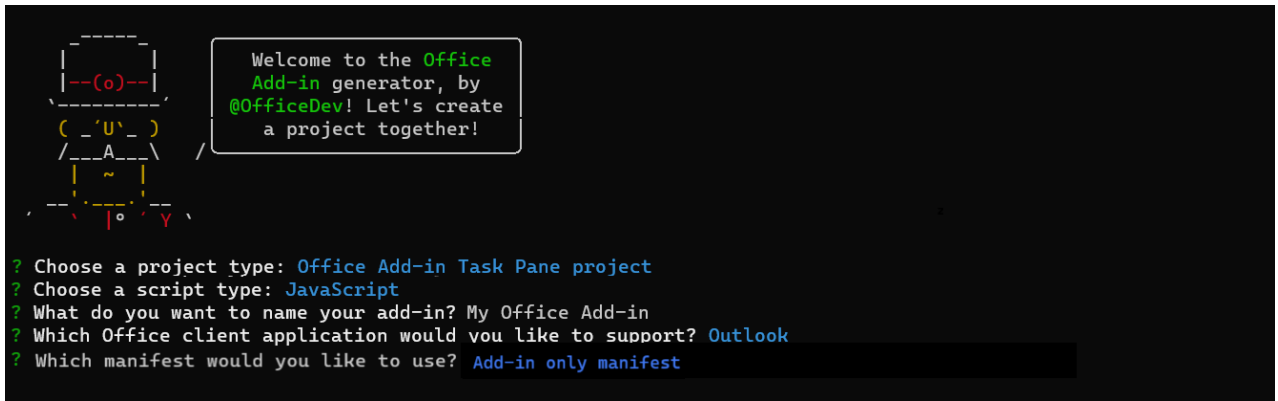
- **Choose a project type** - `Office Add-in Task Pane project`
- **Choose a script type** - `JavaScript`
- **What do you want to name your add-in?** - `My Office Add-in`
- **Which Office client application would you like to support?** - `Outlook`
- **Which manifest would you like to use?** - Choose either `Unified manifest for Microsoft 365` or `Add-in only manifest`

ⓘ Note

The unified manifest for Microsoft 365 enables you to combine an Outlook Add-in with a Teams app as a single unit of development and deployment. We're working to extend support for the unified manifest to Excel, PowerPoint, Word, custom Copilot development, and other extensions of Microsoft 365. For more about it, see [Office Add-ins with the unified manifest](#). For a sample of a combined Teams app and Outlook Add-in, see [Discount Offers](#) [↗].

We love to get your feedback about the unified manifest. If you have any suggestions, please create an issue in the repo for the [Office JavaScript Library](#) [↗].

Depending on your choice of manifest, the prompts and answers should look like one of the following:



After you complete the wizard, the generator will create the project and install supporting Node components.

2. Navigate to the root folder of the web application project.

command line

```
cd "My Office Add-in"
```

Explore the project

The Yeoman generator creates a project in a folder with the project name that you chose. The project contains sample code for a very basic task pane add-in. The following are the most important files.

- The `./manifest.json` or `./manifest.xml` file in the root directory of the project defines the settings and capabilities of the add-in.
- The `./src/taskpane/taskpane.html` file contains the HTML markup for the task pane.
- The `./src/taskpane/taskpane.css` file contains the CSS that's applied to content in the task pane.
- The `./src/taskpane/taskpane.js` file contains the Office JavaScript API code that facilitates interaction between the task pane and Outlook.

Try it out

ⓘ 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
```

- When you first use Yeoman generator to develop an Office Add-in, your default browser opens a window where you'll be prompted to sign in to your Microsoft 365 account. If a sign-in window doesn't appear and you encounter a sideloading or login timeout error, run `teamsapp auth login m365`.

1. Run the following command in the root directory of your project. When you run this command, the local web server starts and your add-in is sideloaded.

command line

```
npm start
```

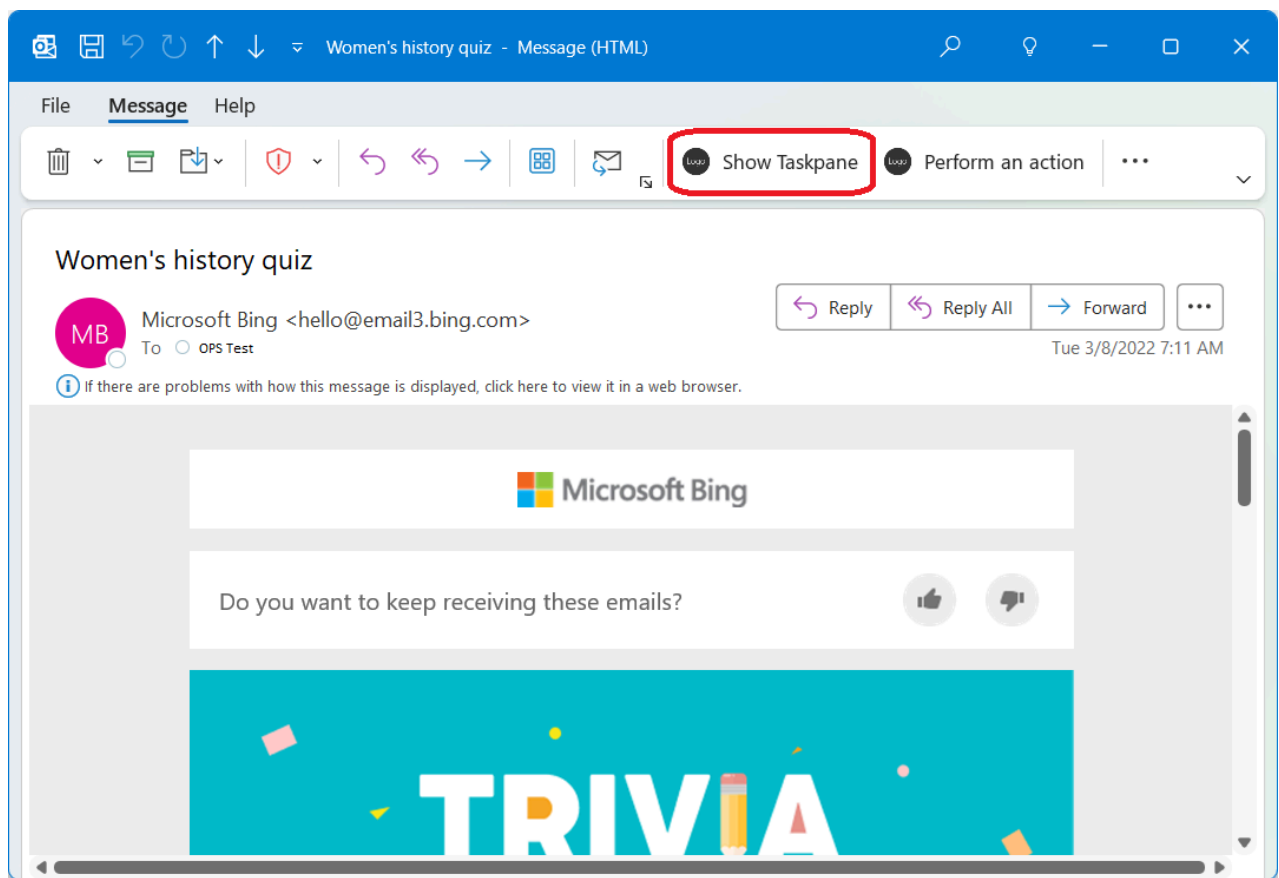
ⓘ Note

- When you first use Yeoman generator to develop an Office Add-in, your default browser opens a window where you'll be prompted to sign in to your Microsoft 365 account. If a sign-in window doesn't appear and you encounter a sideloading or login timeout error, run `teamsapp auth login m365` before running `npm start` again.

If your add-in wasn't automatically sideloaded, follow the instructions in [Sideload Outlook add-ins for testing](#) to manually sideload the add-in in Outlook.

2. In Outlook, view a message in the [Reading Pane](#), or open the message in its own window.
3. Select the **Show Taskpane** button to open the add-in task pane. The location of the add-in button varies depending on the Outlook client you're using.
 - **Outlook on the web** and **new Outlook on Windows**: From the action bar of the message, select **Apps**. Then, select **My Office Add-in > Show Taskpane**.
 - **Classic Outlook on Windows**: Select the **Home** tab (or the **Message** tab if you opened the message in a new window). Then, select **Show Taskpane** from the ribbon.
 - **Outlook on Mac**: Select **My Office Add-in** from the ribbon, then select **Show Taskpane**. You may need to select the ellipsis button (**...**) from the ribbon to view your add-ins.

The following screenshots show how the add-in appears in classic Outlook on Windows.

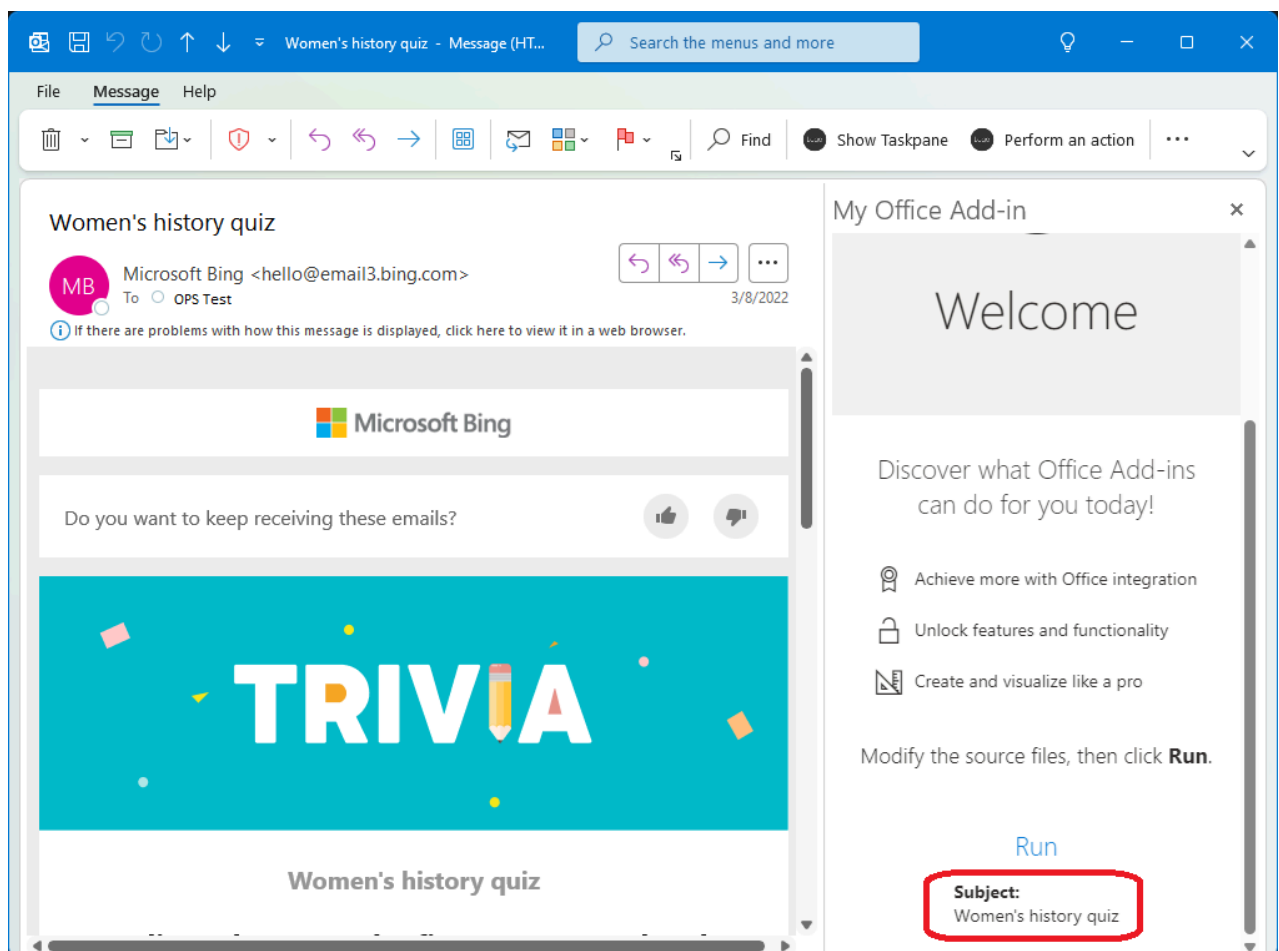
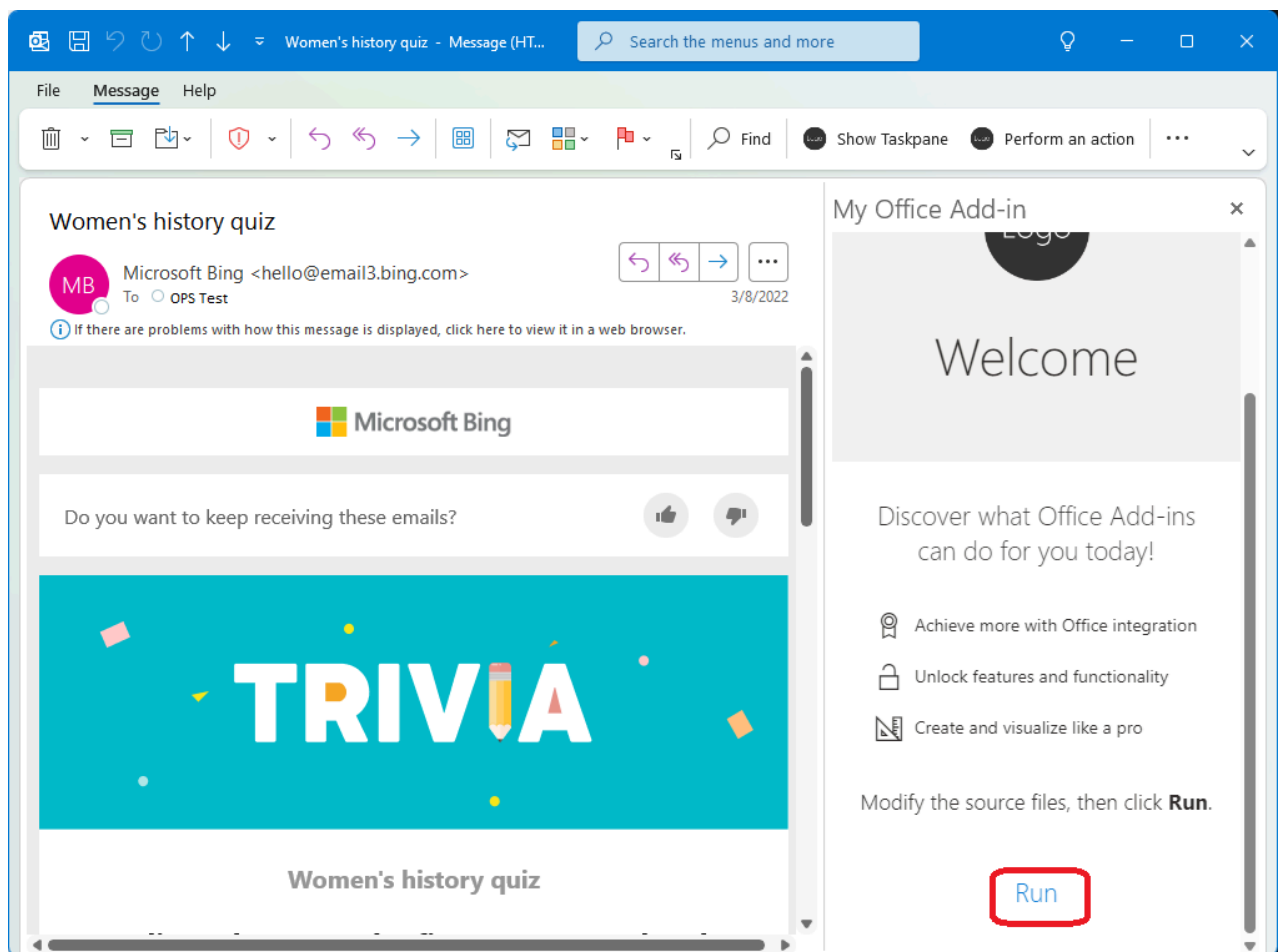


4. When prompted with the **WebView Stop On Load** dialog box, select **OK**.

ⓘ **Note**

If you select **Cancel**, the dialog won't be shown again while this instance of the add-in is running. However, if you restart your add-in, you'll see the dialog again.

5. Scroll to the bottom of the task pane and choose the **Run** link to write the message subject to the task pane.



6. 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 should also uninstall 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 your first Outlook task pane add-in! Next, explore more capabilities of an Outlook add-in by following along with the [Outlook add-in tutorial](#). In the tutorial, you'll build a more complex add-in that incorporates a task pane, which you've learned about in the quick start. Additionally, you'll create a button that invokes a UI-less function.

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](#).
- If you receive the error "We can't open this add-in from localhost" in the task pane, follow the steps outlined in the [troubleshooting article](#).

Code samples

- [Outlook "Hello world" add-in](#) [↗]: Learn how to build a simple Office Add-in with only a manifest, HTML web page, and a logo.

See also

- [Office Add-ins with the add-in only manifest](#)
- [Using Visual Studio Code to publish](#)