

# Build your first PowerPoint task pane add-in

In this article, you'll walk through the process of building a PowerPoint task pane add-in.

## 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](#).

## Create the add-in project

Decide the type of manifest that you'd like to use, either the [unified manifest for Microsoft 365](#) or the [add-in only manifest](#). To learn more about them, see [Office Add-ins manifest](#).

Unified manifest for Microsoft 365 (preview)

### ⓘ Note

Using the unified manifest for Microsoft 365 with PowerPoint add-ins is in public developer preview. The unified manifest for Microsoft 365 shouldn't be used in production PowerPoint add-ins. We invite you to try it out in test or development environments. For more information, see the [Microsoft 365 app manifest schema reference](#).

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, PowerPoint, and/or Word Task Pane with unified manifest for Microsoft 365 (preview)
- **What do you want to name your add-in?** My Office Add-in
- **Which Office client application would you like to support?** PowerPoint

```
>yo office

  _--(o)--_
  \_-'U`_ )
   /_A\_ \
    | ~ |
    | . |
    \_--'-'_` .
      | o |
      Y

  Welcome to the Office
  Add-in generator, by
  @OfficeDev! Let's create
  a project together!

✓Choose a project type: Excel, PowerPoint, and/or Word Task Pane with unified manifest for Microsoft 365 (preview)
✓What do you want to name your add-in? My Office Add-in
✓Which Office client application would you like to support? Powerpoint
```

After you complete the wizard, the generator creates the project and installs supporting Node components.

# Explore the project

The add-in project that you've created with the Yeoman generator contains sample code for a basic task pane add-in. If you'd like to explore the components of your add-in project, open the project in your code editor and review the files listed below. When you're ready to try out your add-in, proceed to the next section.

- 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.ts` file contains the Office JavaScript API code that facilitates interaction between the task pane and the Office client application.

## Try it out

1. Navigate to the root folder of the project.

```
command line
```

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

2. Complete the following steps to start the local web server and sideload your add-in.

 **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 `atk auth login m365`.

### 💡 Tip

If you're testing your add-in on Mac, run the following command before proceeding. When you run this command, the local web server starts.

command line

```
npm run dev-server
```

- To test your add-in in PowerPoint, run the following command in the root directory of your project. This starts the local web server (if it's not already running) and opens PowerPoint with your add-in loaded.

command line

```
npm start
```

- To test your add-in in PowerPoint on a browser, run the following command in the root directory of your project. When you run this command, the local web server starts. Replace "{url}" with the URL of a PowerPoint 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

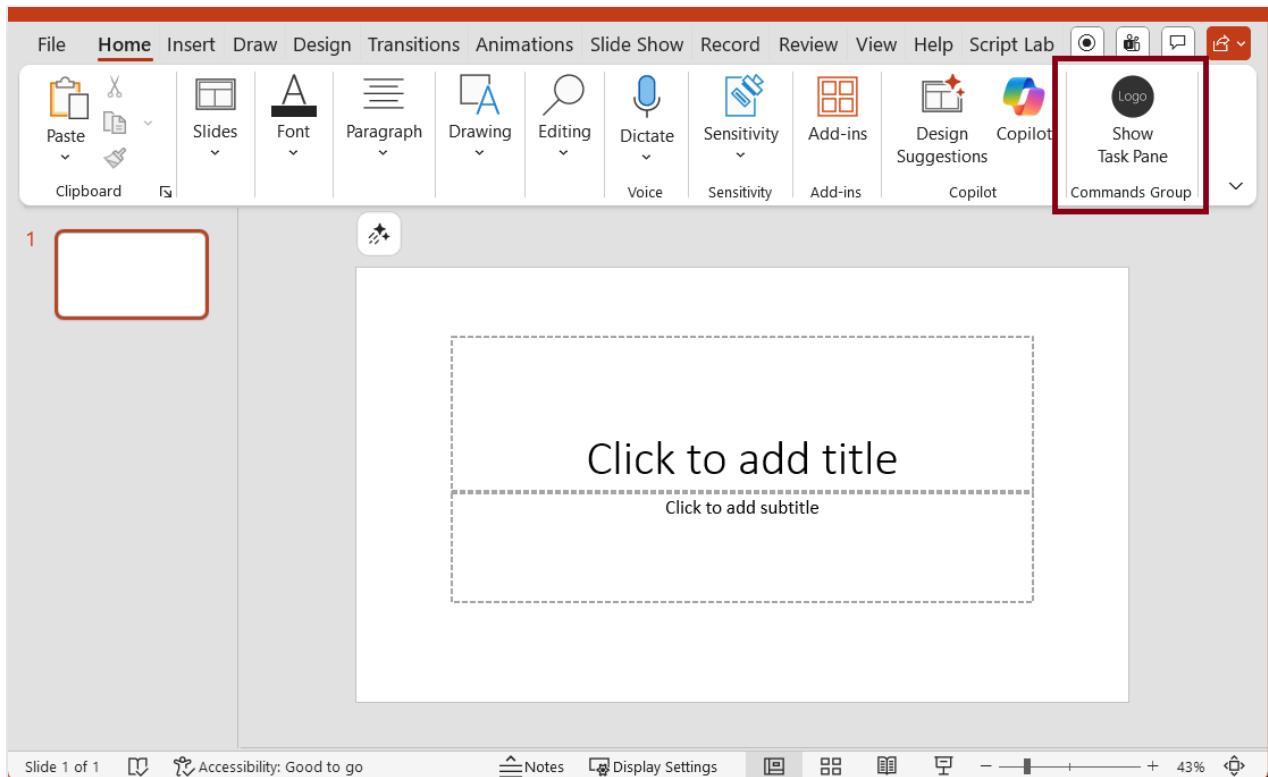
```
npm run start -- web --document {url}
```

The following are examples.

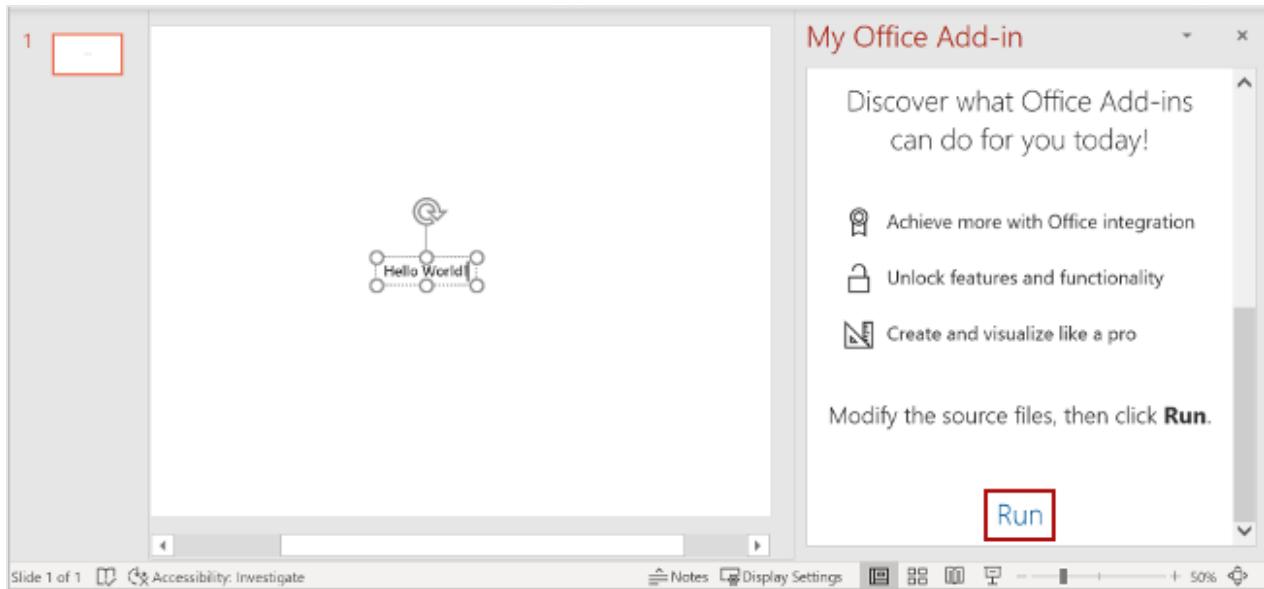
- o `npm run start -- web --document`  
`https://contoso.sharepoint.com/:t/g/EZGxP7ksiE5DuxvY638G798BpuhwluxCMfF1WZQj3VYhYQ?e=F4QM1R`
- o `npm run start -- web --document`  
`https://1drv.ms/x/s!jkch7spkM4EGgcZUgqthk4IK3N0ypVw?e=Z6G1qp`
- o `npm run start -- web --document https://contoso-my.sharepoint-df.com/:t/p/user/Eqda453DNTpFn1bFPh0VR0BwlrzetbXvnaRYii2lDr_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](#).

3. In PowerPoint, insert a new blank slide, choose the **Home** tab, and then choose the **Show Task Pane** button on the ribbon to open the add-in task pane.



4. At the bottom of the task pane, choose the **Run** link to insert the text "Hello World" into the current slide.



5. 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 PowerPoint task pane add-in! Next, learn more about the capabilities of a PowerPoint add-in and build a more complex add-in by following along with the [PowerPoint add-in tutorial](#).

## Troubleshooting

- Ensure your environment is ready for Office development by following the instructions in [Set up your development environment](#).
- 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 see warnings generated when running `npm install` for either Yeoman generator or the project. In most cases, you can safely ignore these warnings. Sometimes, dependencies become deprecated and their replacements aren't supported by other packages on which the project depends. If you would like to resolve these warnings, use the `npm-check-updates` tool.
  - In the command prompt while in the root project directory, run `npm i -g npm-check-updates`. This installs the tool globally.
  - Run `ncu -u`. This provides a report of all packages and to what versions they will be updated.
  - Run `npm install` to update all the packages.

For more information about warnings when running `npm install`, see [Warnings and dependencies in the Node.js and npm world](#).

- If you don't see the add-in button on the ribbon (e.g., **Show Task Pane**), select the **Add-ins** button from the **Home** ribbon tab, then choose the test add-in.

## Code samples

- [PowerPoint "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 platform overview](#)
- [Develop Office Add-ins](#)
- [Using Visual Studio Code to publish](#)

---

Last updated on 12/12/2025