


Create Office Add-in projects using Office Add-ins Development Kit for Visual Studio Code





Article • 12/19/2024

The Office Add-ins Development Kit helps set up your environment, create Office Add-ins, and debug your code in a streamlined experience.

Important

The Office Add-ins Development Kit extension only supports creating add-ins that use the [add-in only manifest](#). It also only creates Excel, PowerPoint, and Word add-ins at this time. Support for Outlook is under development, as are additional samples and other improvements. We welcome any feedback you have on the tool. Issues and features requests should be submitted through [GitHub issues on the extension's repo](#) .

Prerequisites

- Download and install [Visual Studio Code](#) .
- Node.js (the latest LTS version). Visit the [Node.js site](#)  to download and install the right version for your operating system. To verify if you've already installed these tools, run the commands `node -v` and `npm -v` in your terminal.
- Office connected to a Microsoft 365 subscription. You might qualify for a Microsoft 365 E5 developer subscription through the [Microsoft 365 Developer Program](#), see [FAQ](#) for details. Alternatively, you can [sign up for a 1-month free trial](#)  or [purchase a Microsoft 365 plan](#) .

Install the development kit

You can install Office Add-ins Development Kit using **Extensions** in Visual Studio Code or install it from the Visual Studio Code Marketplace.

Visual Studio Code

1. Launch Visual Studio Code.

2. Select **View > Extensions** or **Ctrl+Shift+X**. You can also open extensions by selecting the **Extensions** tab from the Visual Studio Code activity bar.
3. Enter **Office Add-ins Development Kit** in the search box.
4. Select **Office Add-ins Development Kit**. From the Office Add-ins Development Kit extension page that appears in the main pane, select **Install**. After successfully installing, the Office Add-ins Development Kit icon will appear in the Visual Studio Code activity bar.

Create an add-in project

The Office Add-ins Development Kit offers two ways to get started with a new project: templates and samples. Templates let you choose the Office application, programming language, and starting feature. Samples are more complete projects that demonstrate a realistic scenario.

Create an add-in from a template

Templates offer a basic starting point for your add-in. They add a minimal amount of functionality so you can get started without changing much of the existing code. The following instructions describe how to make a new project from a template using the development kit.

1. Open Visual Studio Code and select the Office Add-ins Development Kit icon in the **Activity Bar**.
2. Select **Create a New Add-in** in the extension task pane.
3. In the now-active Quick Pick menu, select the Office application for your add-in.
4. Select an add-in template from the list of available templates.
5. Select "JavaScript" or "TypeScript" as the programming language.
6. In the **Workspace folder** dialog that opens, select the folder where you want to create the project.
7. Give a name to the project (with no spaces) when prompted. Office Add-ins Development Kit will create the project with basic files and scaffolding. It then opens the project in a *second* Visual Studio Code window. You can freely close the original Visual Studio Code window.

Create an add-in from a sample

Samples show a working add-in that solves an end-to-end scenario. Samples are most useful as learning tools to understand how features of the Office Add-ins framework

work together. You can also expand a sample to fit your particular needs.

1. Open Visual Studio Code and select the Office Add-ins Development Kit icon in the **Activity Bar**.
2. Select **View Samples**.
3. Select the sample you would like to view.
4. Select the **Create** button above the now-open sample readme.
5. In the **Workspace folder** dialog that opens, select the folder where you want to create the project. The extension copies a version of the sample to that folder. It then opens the project in a *second* Visual Studio Code window. You can freely close the original Visual Studio Code window.

Test your add-in

To understand how the add-in will work in an Office application, use the Office Add-ins Development Kit to run and debug your Office add-in in your local development environment.

ⓘ Note

These steps are the same as the ones listed in Visual Studio Code by the extension when you create a new project.

1. Open the extension by selecting the Office Add-ins Development Kit icon in the **Activity Bar**.
2. Select **Preview Your Office Add-in (F5)**
3. In the Quick Pick menu, select the option **{Office Application} Desktop (Edge Chromium)**, where '{Office Application}' is the appropriate application, such as "Excel" or "Word". This will launch the add-in and debug the code.

The development kit checks that the prerequisites are met before debugging starts. Check the terminal for detailed information if there are issues with your environment. After this process, the Office desktop application launches and sideloads the add-in. Please note that the first time you run a project, it may take a few minutes to install the dependencies. You'll need to install the certificate when prompted.

Stop testing your Office Add-in

Once you are finished testing and debugging the add-in, *always* close the add-in by following these steps. (Closing the Office application or web server window doesn't

reliably deregister the add-in.)

1. Open the extension by selecting the Office Add-ins Development Kit icon in the **Activity Bar**.
2. Select **Stop Previewing Your Office Add-in**. This closes the web server and removes the add-in from the registry and cache.
3. Close the Office application window.

Troubleshooting

If you have problems running the add-in, take these steps.

- Close any open instances of Office.
- Close the previous web server started for the add-in with the **Stop Previewing Your Office Add-in** Office Add-ins Development Kit extension option.

The article [Troubleshoot development errors with Office Add-ins](#) contains solutions to common problems. If you're still having issues, [create a GitHub issue](#) and we'll help you.

For information on running the add-in on Office on the web, see [Sideload Office Add-ins to Office on the web](#).

For information on debugging on older versions of Office, see [Debug add-ins using developer tools in Microsoft Edge Legacy](#).