

Develop Office Add-ins with Visual Studio Code

Article • 08/20/2024

This article describes how to use [Visual Studio Code \(VS Code\)](#) to develop an Office Add-in.

ⓘ Note

For information about using Visual Studio to create an Office Add-in, see [Develop Office Add-ins with Visual Studio](#).

Prerequisites

- [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.
- 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 using the Yeoman generator

If you're using VS Code as your integrated development environment (IDE), you should create the Office Add-in project with the [Yeoman generator for Office Add-ins](#). The Yeoman generator creates a Node.js project that can be managed with VS Code or any other editor.

To create an Office Add-in with the Yeoman generator, follow instructions in the [5-minute quick start](#) that corresponds to the type of add-in you'd like to create.

Develop the add-in using VS Code

When the Yeoman generator finishes creating the add-in project, open the root folder of the project with VS Code.

Tip


On Windows, you can navigate to the root directory of the project via the command line and then enter `code .` to open that folder in VS Code. On Mac, you'll need to [add the code command to the path](#) before you can use that command to open the project folder in VS Code.

The Yeoman generator creates a basic add-in with limited functionality. You can customize the add-in by editing the [manifest](#), HTML, JavaScript or TypeScript, and CSS files in VS Code. For a high-level description of the project structure and files in the add-in project that the Yeoman generator creates, see the Yeoman generator guidance within the [5-minute quick start](#) that corresponds to the type of add-in you've created.

Create the add-in project using the Office Add-ins Development Kit (preview)

The [Office Add-ins Development Kit](#) is a Visual Studio Code extension that allows you to create new projects directly from VS Code. For information on installing the extension and creating projects from templates and samples, see [Create Office Add-in projects using Office Add-ins Development Kit for Visual Studio Code](#).

Important

The Office Add-ins Development Kit extension is currently in preview. It 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](#) .

Test and debug the add-in

Methods for testing, debugging, and troubleshooting Office Add-ins vary by platform. For more information, see [Test and debug Office Add-ins](#).

Publish the add-in

An Office Add-in consists of a web application and a manifest file. The web application defines the add-in's user interface and functionality, while the manifest specifies the location of the web application and defines settings and capabilities of the add-in.

While you're developing your add-in, you can run the add-in on your local web server (`localhost`), but when you're ready to publish it for other users to access, you'll need to deploy the web application to a web server or web hosting service (for example, Microsoft Azure) and update the manifest to specify the URL of the deployed application.

When your add-in is working as desired and you're ready to publish it for other users to access, complete the following steps.

1. From the command line, in the root directory of your add-in project, run the following command to prepare all files for production deployment.

command line

```
npm run build
```

When the build completes, the **dist** folder in the root directory of your add-in project will contain the files that you'll deploy in subsequent steps.

2. Upload the contents of the **dist** folder to the web server that'll host your add-in. You can use any type of web server or web hosting service to host your add-in.

3. In VS Code, open the add-in's manifest file, located in the root directory of the project (`manifest.xml`). Replace all occurrences of `https://localhost:3000` with the URL of the web application that you deployed to a web server in the previous step.
4. Choose the method you'd like to use to [deploy your Office Add-in](#), and follow the instructions to publish the manifest file.

See also

- [Core concepts for Office Add-ins](#)
- [Develop Office Add-ins](#)
- [Design Office Add-ins](#)
- [Test and debug Office Add-ins](#)
- [Publish Office Add-ins](#)

Develop Office Add-ins with Visual Studio

Article • 08/18/2023

This article describes how to use Visual Studio to develop an Office Add-in. If you've already created your add-in, you can skip ahead to the [Develop the add-in using Visual Studio](#) section.

ⓘ Note

As an alternative to using Visual Studio, you may choose to use the Yeoman generator for Office Add-ins and VS Code to create an Office Add-in. For more information about this choice, see [Creating an Office Add-in](#).

Create the add-in project using Visual Studio

Visual Studio can be used to create Office Add-ins for Excel, Outlook, PowerPoint, and Word. An Office Add-in project gets created as part of a Visual Studio solution and uses HTML, CSS, and JavaScript. To create an Office Add-in with Visual Studio, follow instructions in the quick start that corresponds to the add-in you'd like to create.

- [Excel quick start](#)
- [Outlook quick start](#)
- [PowerPoint quick start](#)
- [Word quick start](#)

Visual Studio doesn't support creating Office Add-ins for OneNote or Project. To create Office Add-ins for either of these applications, you'll need to use the Yeoman generator for Office Add-ins, as described in the [OneNote quick start](#) or the [Project quick start](#).

Develop the add-in using Visual Studio

Visual Studio creates a basic add-in with limited functionality. You can customize the add-in by editing the [manifest](#), HTML, JavaScript, and CSS files in Visual Studio. For a high-level description of the project structure and files in the add-in project that Visual Studio creates, see the Visual Studio guidance within the quick start that you completed to create your add-in.

Tip

Because an Office Add-in is a web application, you'll need at least basic web development skills to customize your add-in. If you're new to JavaScript, we recommend reviewing the [Mozilla JavaScript tutorial](#) [↗].

To customize your add-in, you'll need to understand concepts described in the [Core concepts > Develop](#) area of this documentation, as well as concepts described in the application-specific area of documentation that corresponds to the add-in you're building (for example, [Excel](#)).

Test and debug the add-in

Methods for testing, debugging, and troubleshooting Office Add-ins vary by platform. For more information, see [Debug Office Add-ins in Visual Studio](#) and [Test and debug Office Add-ins](#).

Publish the add-in

An Office Add-in consists of a web application and a manifest file. The web application defines the add-in's user interface and functionality, while the manifest specifies the location of the web application and defines settings and capabilities of the add-in.

While you're developing your add-in in Visual Studio, your add-in runs on your local web server (`localhost`). When your add-in is working as desired and you're ready to publish it for other users to access, you'll need to complete the following steps.

1. Deploy the web application to a web server or web hosting service (for example, Microsoft Azure).
2. Update the manifest to specify the URL of the deployed application.
3. Choose the method you'd like to use to [deploy your Office Add-in](#), and follow the instructions to publish the manifest file.

See also

- [Core concepts for Office Add-ins](#)
- [Develop Office Add-ins](#)
- [Design Office Add-ins](#)
- [Test and debug Office Add-ins](#)
- [Publish Office Add-ins](#)