Leverage Codespaces with VS Code for Copilot

Duration: 20 minutes

GitHub Copilot is an AI pair programmer designed to make writing code easier and faster. It draws context from comments and code to suggest individual lines and whole functions instantly. GitHub Copilot is powered by OpenAI Codex, a generative pre-trained language model created by OpenAI.

**Copilot is compatible with a wide range of code editors, such as Neovim, JetBrains IDE, Visual Studio, and VS Code.**

Additionally, GitHub Copilot is trained in every language that can be found in public repositories. The amount and variety of training data for each language may have an impact on the quality of recommendations you obtain.

**GitHub Codespace** is a development environment that's hosted in the cloud. You can customize your project for GitHub Codespaces by committing configuration files to your repository (often known as Configuration-as-Code), which creates a repeatable codespace configuration for all users of your project.

Using Copilot inside a Codespace shows just how easy it is to get up and running with GitHub's suite of [Collaborative Coding](https://github.com/features#features-collaboration) tools.

In this exercise, you will create a development container and define specific extensions or configurations that will be used or installed in your codespace. You will create this development container and add Copilot to the list of extensions.

Start this exercise by creating a blank Github Repo, right in the Github UI. If you have never worked this way, we will go through this in class and will create a blank repo. The rest of the lab resumes at this point.

Task 1: Enable Copilot inside a Codespace

1. Navigate back to the home page of your repository, click on **Code** **(1)** tab of your repository, click the **Add file** **(2)** drop-down button, and then select Create new file **(3)**.

A screenshot of a computer

Description automatically generated

1. Type or paste the following in the empty text field prompt to name your file **(1)**.
2. .devcontainer/devcontainer.json

.devcontainer/devcontainer.json

1. In the body of the new **.devcontainer/devcontainer.json** file, add the following content **(2)** and click on **Commit changes** **(3)**:
2. {
3. // Name this configuration
4. "name": "Codespace for Skills!",
5. "customizations": {
6. "vscode": {
7. "extensions": [
8. "GitHub.copilot"
9. ]
10. }
11. }
12. }
13. {
14. // Name this configuration
15. "name": "Codespace for Skills!",
16. "customizations": {
17. "vscode": {
18. "extensions": [
19. "GitHub.copilot"
20. ]
21. }
22. }

}

A screenshot of a computer

Description automatically generated

1. Select the option to **Commit directly to the main branch**, and then click the **Commit changes** button.

A screenshot of a computer

Description automatically generated

1. Navigate back to the home page of your repository by clicking the **Code** **(1)** tab located at the top left of the screen. Click the **Code** **(2)** button located in the middle of the page.

A screenshot of a computer

Description automatically generated

1. Click the **Codespaces (1)** tab on the box that pops up and then click the **Create codespace on main (2)** button.

A screenshot of a computer

Description automatically generated

**Note**: If in case pop-up prompt doesn't appear in the browser to open Visual Studio code, manually launch Visual Studio code from the desktop and close it. Next, return to the browser, refresh the page and launch the codespace that was previously created.

1. You will encounter a pop-up prompt. Click **Open** to proceed. Subsequently, another pop-up window will appear within Visual Studio Code (VS Code), where you should once again select **Install and Open** to continue.

A screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

1. At the bottom right corner, you will get a prompt to sign in to GitHub.

A screenshot of a computer

Description automatically generated

1. Next, once you get the popup, click on **Allow**

A screenshot of a computer

Description automatically generated

**Note**: Wait about 2 minutes for the codespace to spin itself up.

1. Click **Authorize Visual-Studio-Code** once the Authorize GitHub for VS code tab appears in the browser.
2. Verify your codespace is running. Make sure the VS code looks as shown below:

A screenshot of a computer

Description automatically generated

1. Click on **Extensions** **(1)** from the left menu, and the **GitHub Copilot** **(2)** extension should show up in the VS Code extension list. Click the Copilot extension and verify its installation as shown below:

A screenshot of a computer

Description automatically generated

**Note**: If the GitHub Copilot extension is not installed, click on Install.

Summary

In this exercise, you have created a development container and added Copilot to the list of extensions.