How to Use Github Copilot to Generate a Postman Collection File Using Visual Studio Code

A tutorial for beginners who want to create and test APIs with Postman and Github Copilot

# Introduction

Postman is a popular tool for developing and testing APIs. It allows you to create requests, organize them into collections, and run them with different environments and variables. Postman also supports importing and exporting collections in different formats, such as JSON, YAML, and HTML.

Github Copilot is a new AI-powered code assistant that helps you write code faster and better. It works as an extension for Visual Studio Code, and it can suggest code completions, snippets, and entire functions based on your context and comments. Github Copilot can also generate code for different languages and frameworks, including JavaScript, Python, React, and more.

In this tutorial, you will learn how to use Github Copilot to generate a Postman collection file using Visual Studio Code. You will also learn how to import the collection file into Postman and test it with different parameters. This tutorial assumes that you have some basic knowledge of APIs, Visual Studio Code, and Postman.

# Prerequisites

* A Github account and a Github Copilot subscription. You can sign up for a free trial here: https://copilot.github.com/
* A Visual Studio Code editor with the Github Copilot extension installed. You can download it here: https://code.visualstudio.com/
* A Postman desktop app or web browser. You can download it here: https://www.postman.com/downloads/

# Steps

1. Create a new JavaScript file in Visual Studio Code and name it **sample-api.js**.
2. In the first line of the file, write a comment that describes the API you want to create. For example, you can write**: // A simple API for a todo list app**
3. Press **Ctrl+Enter** to activate Github Copilot and wait for it to suggest some code. You should see something like this:
4. Review the code and make any changes if needed. You can also add more comments, variables, or tests to the collection. For example, you can write: // {{todo}} is a JSON object with title and completed properties
5. Create a new JSON file in Visual Studio Code and name it postman-collection.json
6. Press **Ctrl+I** to activate Github Copilot and prompt it to create a Postman collection based on the API. For example, you can write: **Generate a JSON schema that I can import into Postman for my todo list API**
7. You should see something list this:  
     
   A screenshot of a computer program

   Description automatically generated
8. Click **Accept** to accept the code suggestion.
9. Open Postman and click on the **Import** button. Choose the JSON file you just created and click on **Import**.
10. You should see the Todo List API collection in the left sidebar. You can expand it and see the four requests: Get todo, Create todo, Update todo, and Delete todo.  
      
    A screenshot of a computer

    Description automatically generated
11. To test the API, you need to provide some values for the variables and parameters. You can do this by clicking on the **Variables** tab in the collection and adding some values. For example, you can add: todoId: 1  
      
    A screenshot of a computer

    Description automatically generated
12. Now you can select any request and click on the Send button. You should see the response in the right panel. You can also check the status code, headers, and body of the response.
13. You can also run the entire collection as a test suite by clicking on the Run button. You can configure the settings, such as the number of iterations, the delay, and the environment. You can also see the results and the logs of the test run.

# Conclusion

In this tutorial, you learned how to use Github Copilot to generate a Postman collection file using Visual Studio Code. You also learned how to import the collection file into Postman and test it with different parameters. You can use this technique to create and test any API you want with Postman and Github Copilot.

If you want to learn more about Postman and Github Copilot, you can check out these resources:

* Postman documentation: https://www.postman.com/docs/
* Github Copilot documentation: https://docs.github.com/en/codespaces/codespaces-reference/github-copilot-in-codespaces
* Postman learning center: https://learning.postman.com/
* Github Copilot blog: https://github.blog/category/copilot/