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

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

# Introduction

Postman is a popular tool for designing, testing, and documenting APIs. It allows you to create requests, organize them into collections, and run them with different environments and variables. Postman also provides a command-line tool called Newman that can execute your collections and generate reports. Newman is useful for integrating your API tests with continuous integration and delivery pipelines, or for running them in different environments and platforms.

However, writing tests for your Postman requests can be tedious and time-consuming, especially if you have a large number of requests and scenarios to cover. Fortunately, there is a way to automate the process of generating tests for your Postman collections using Github Copilot, an AI-powered code assistant that helps you write code faster and better. Github Copilot can suggest tests for your Postman requests based on the request name, URL, method, parameters, headers, body, and response. You can then edit, customize, or accept the suggested tests and save them to your Postman collection file.

In this tutorial, you will learn how to use Github Copilot to generate a Postman Collection and Newman tests in Visual Studio Code based on a Postman collection file. You will need the following tools and prerequisites to follow along:

# Prerequisites

* A Github account and a Github Copilot subscription.
* A Visual Studio Code editor with the Github Copilot extension installed.
* A Postman desktop app or web browser.
* A Node.js environment with Newman installed.

# Request Your API Key

1. In this lab, we will be testing thecatapi located at <https://thecatapi.com/>.
2. Open a web browser and navigate to <https://thecatapi.com>.   
   A screenshot of a website

   Description automatically generated
3. We will need an API key in order to call the public API. Click “**Get Your API Key**”.
4. Opt for the “**Free**” tier and fill out your email and app description and click **Submit.** A key will be emailed to you, make sure to check your junk folder.  
   A screenshot of a computer

   Description automatically generated

# Use Copilot to Generate a Postman Collection File

1. Open Visual Studio Code.
2. On the left pane of VS Code, open Copilot Chat.
3. In the chat box, prompt Copilot to create a Postman collection based on thecatapi located at https://thecatapi.com/. For example, you can write: **Generate a postman collection JSON schema that I can import into Postman in order to test thecatapi located at https://thecatapi.com/.**   
     
   A black screen with white text

   Description automatically generated
4. Press **Submit.** You should see something like this:  
     
   A screenshot of a computer screen

   Description automatically generated  
   A screenshot of a computer

   Description automatically generated
5. Hover over the suggested file and click “**Insert into New File**” to accept the code suggestion.
6. Copilot called out that we must replace “YOUR\_API\_KEY” with your actual API key from thecatapi.com. **Copy** the API key that was emailed to you and replace the value in the “x-api-key” header.
7. Click **Ctrl+S** to save the file and name it **postman-collection.json**.
8. Open Postman and click on the **Import** button. Choose the JSON file you just created and click on **Import**.
9. You should see the TheCatAPI Collection in the left sidebar. You can expand it and see the request: *List Breeds*  
     
   A screenshot of a phone

   Description automatically generated
10. Expand **List Breeds**. We should see something like this:  
      
    A screenshot of a computer program

    Description automatically generated
11. Now we can test the call by hitting the **Send** button. You should see the response in the bottom panel. You can also check the status code, headers, and body of the response.  
      
    A screenshot of a computer

    Description automatically generated

# Use Copilot to Generate Newman Tests

1. Now that we have successfully tested that we can make the API call in Postman, let’s use the Newman CLI to test it from the command line.
2. Navigate back to VS Code.
3. Open a new terminal. If you do not have newman installed, run: **npm install newman**
4. Once you have newman installed, run **newman run postman-collection.json.** This command will execute all the requests and tests in the Postman collection file and generate a report in the terminal. You can also specify different options and formats for the report, such as HTML, JSON, or JUnit. You can learn more about the Newman command-line options here: https://learning.postman.com/docs/running-collections/using-newman-cli/command-line-integration-with-newman/

1. We should see something like this:  
     
   A screenshot of a computer

   Description automatically generated
2. The next step is to use Github Copilot to generate tests for our requests. To do so, we need to add a test property for each request object in the JSON file.
3. Open Copilot Chat and prompt it to add tests to your postman-collection.json file. You can say something like: **Add test properties to my List Breeds request in my postman-collection.json for thecatapi.**  
     
   A black background with white text

   Description automatically generated
4. Copilot will suggest something like this:  
     
    A screenshot of a computer program

   Description automatically generated  
   A screenshot of a computer program

   Description automatically generated
5. Copilot suggested adding three tests to check that the status code of the response is 200, check that the response time is less than 500ms, and check that the “Content-Type” header is present in the response.
6. **Copy** the contents of the “**event**” block and paste them into your **postman-collection.json** file. Click **Save**.
7. Now let’s run our tests by running **newman run postman-collection.json**
8. We can see that in addition to the List Breeds GET request being executed, we now have an additional test-script and three assertions that have been executed and are passing.  
     
   A screenshot of a computer

   Description automatically generated

# Conclusion

In this tutorial, you learned how to use Github Copilot to generate a Postman collection file and Newman tests using Visual Studio Code. You learned how to open the Postman collection file in VS Code, how to use Github Copilot to suggest tests for your requests, and how to save the file and run it with Newman. You can use this technique to automate the process of generating tests for your Postman collections and improve the quality and coverage of your API tests.