1.Demonstrate how to set up Postman.

Postman Setup

This section will guide you to understand:

* Postman
* Importance of Postman
* Installation of Postman

This guide has two subsections, namely:

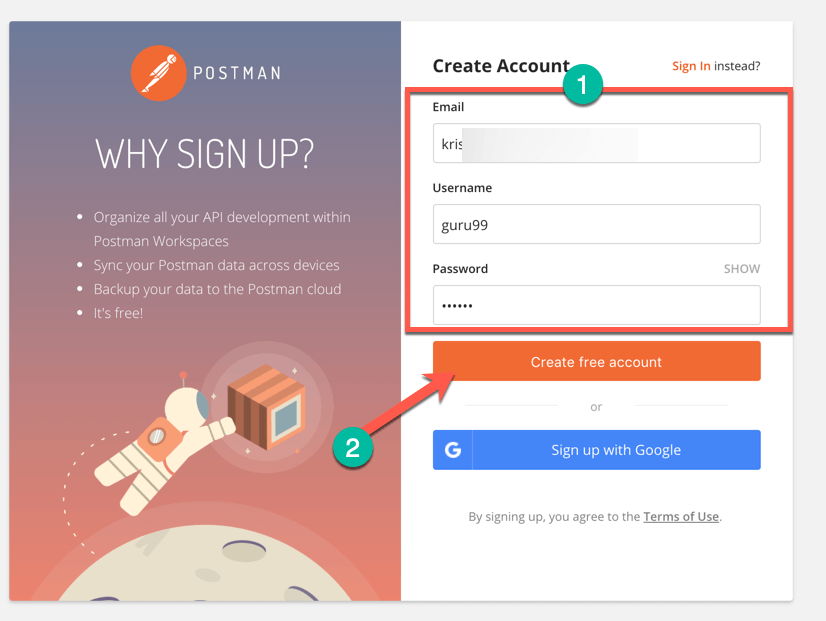
* + 1. Installing Postman
    2. Setting up Postman

**Step 2.1.1: Installing Postman**

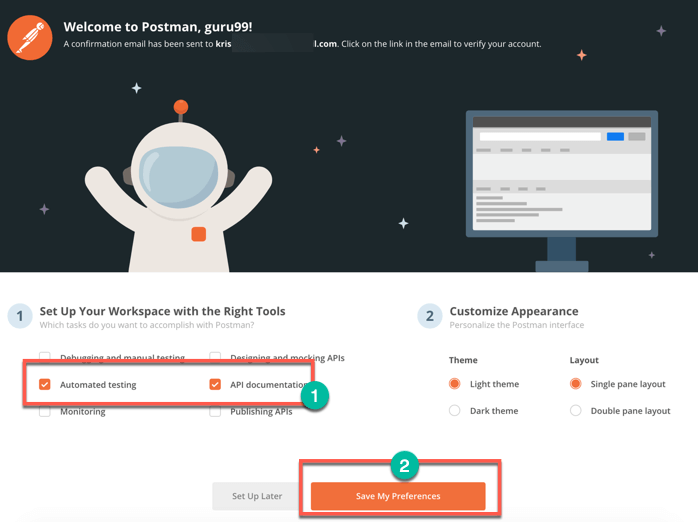
* Postman is already installed in your practice lab. Refer QA to QE lab guide -- Phase 3 for more information.

**Step 2.1.2: Setting up Postman**

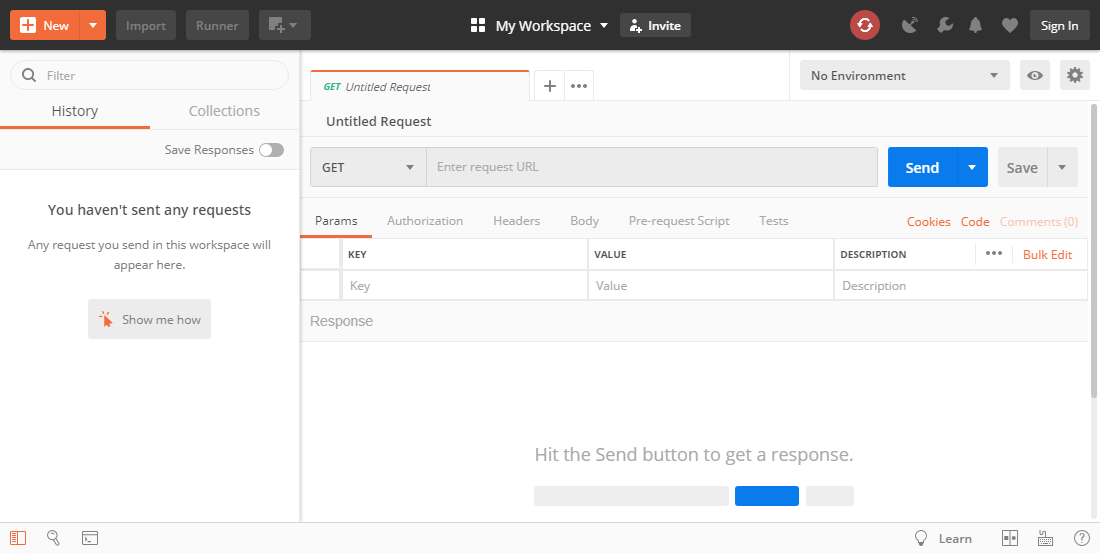
* Open Postman and click on the sign up button to create a Postman Account.



* NOTE: There are two ways to sign up for a Postman account. One is to create your own Postman account, and the other is to use a Google account. Though Postman allows users to use the tool without logging in, signing up ensures that your collection is saved and can be accessed for later use.
* Select the required workspace tools.
* Click on Save My Preferences.



* You will see this Startup Screen below:



2. Demonstrate how to create the first API request.

This section will guide you to:

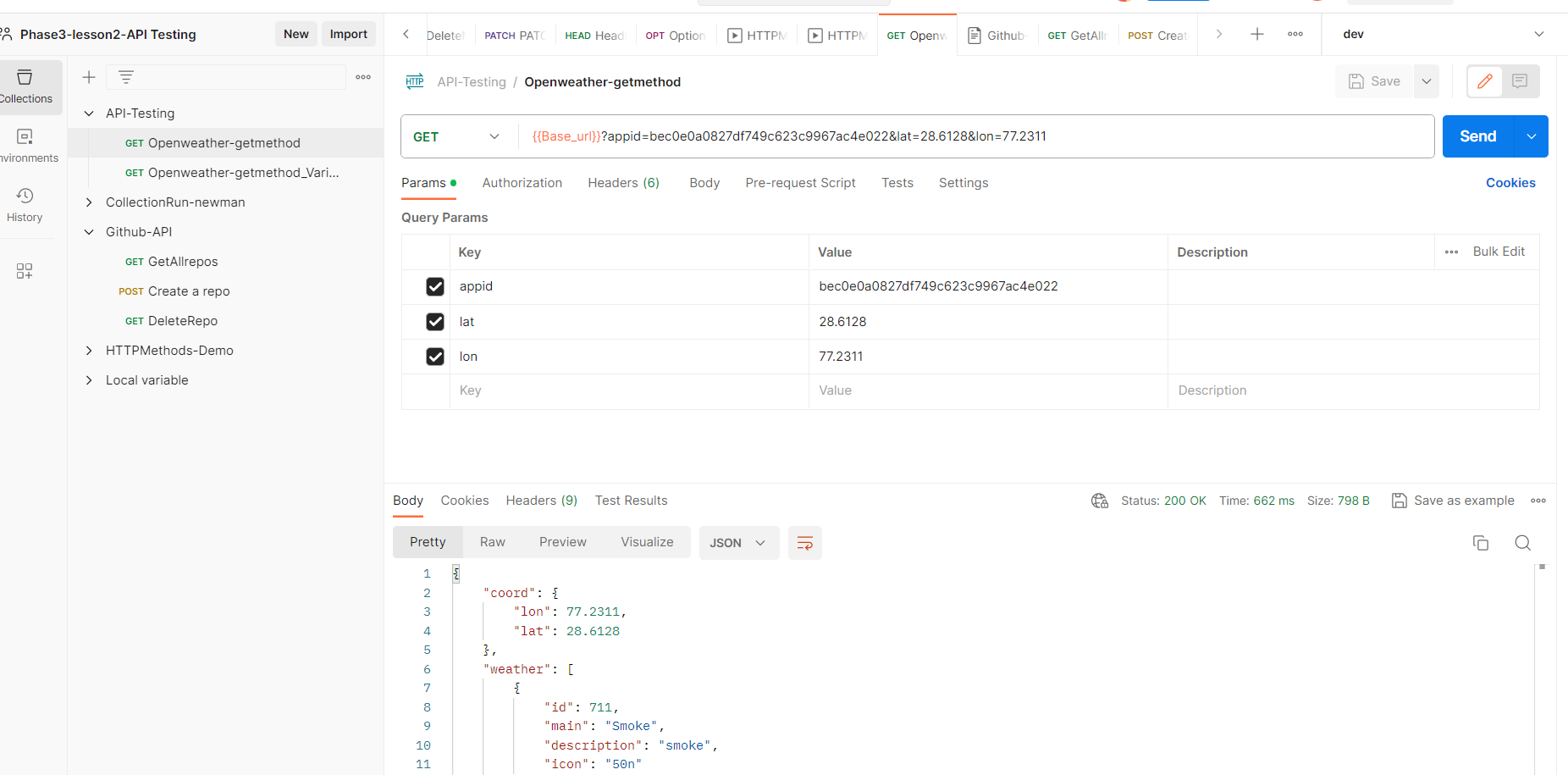
* Create first API request on Postman
* Get response of a request

This guide has three sections, namely:

* + 1. Setting up API request on Postman
    2. Getting response of API request on Postman
    3. Pushing the code to GitHub repositories

**Step 2.2.1:** Setting upAPI request on Postman

The screenshot below shows the Postman Workspace



3.Demonstrate how to use Postman with SOAP.

This section will guide you to:

* Use Postman with SOAP

**Development Environment:**

* Postman Tool
* Endpoint URL

This guide has three subsections, namely:

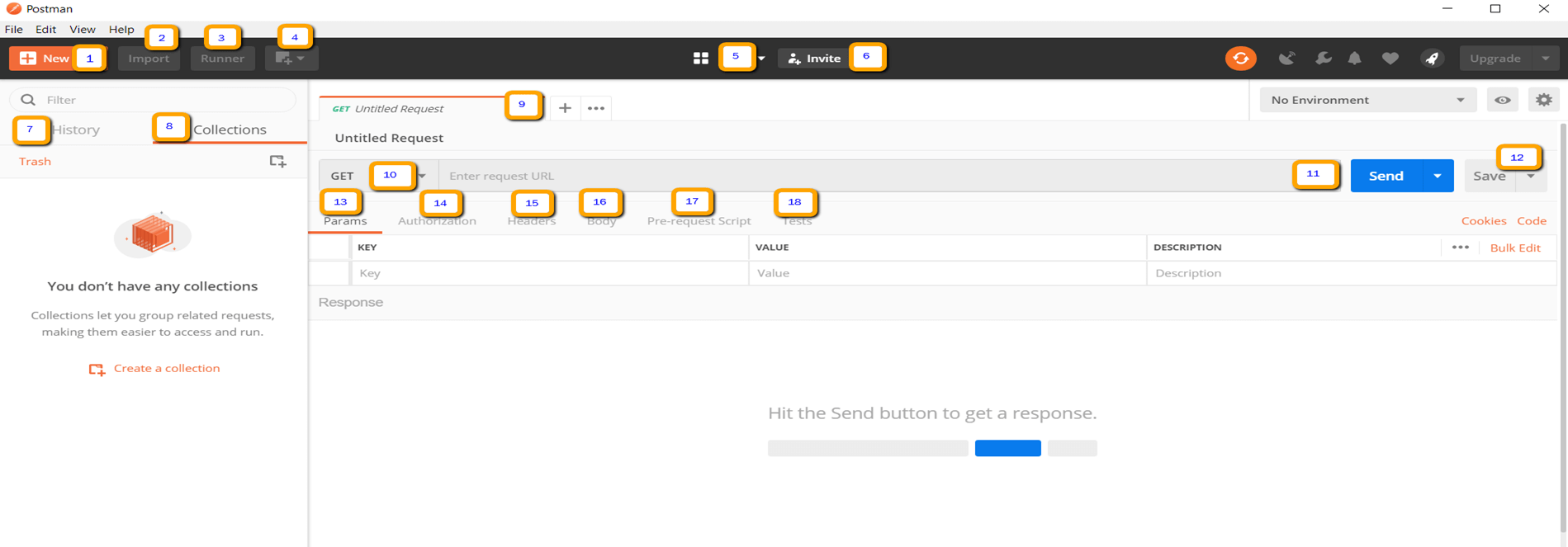
2.3.1 Adding SOAP request in postman

2.3.2 Running and validating the response

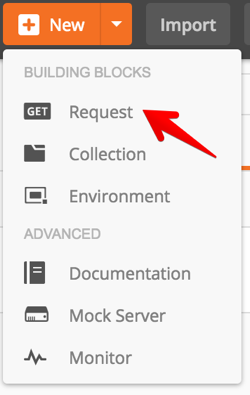
2.3.3 Pushing the code to your GitHub repositories

**Step 2.3.1:** Adding SOAP request in Postman

* The screenshot below shows the Postman Workspace.



* Open Postman WorkSpace.
* On the top left corner, click on the **New option** button.



* In the drop-down options available, click on the Request option.
* Your Request is ready now.
* Take the given URL as the Soap request URL.

<http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?WSDL>

* In the request URL field, input the above link.
* Then set the method as Post.
* Set Body as raw and set text as XML.
* Provide data in the body.

<Envelope xmlns="http:**//**schemas.xmlsoap.org/soap/envelope/">

<Body>

<CapitalCity xmlns="http:**//**www.oorsprong.org/websamples.countryinfo">

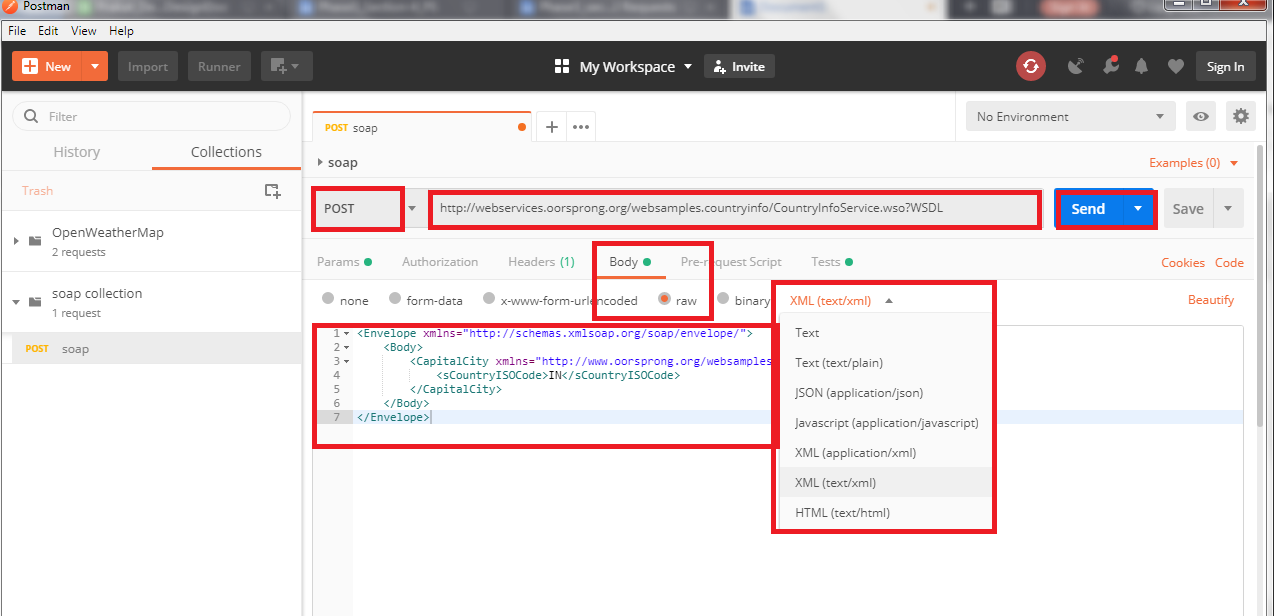
<sCountryISOCode>IN</sCountryISOCode>

</CapitalCity>

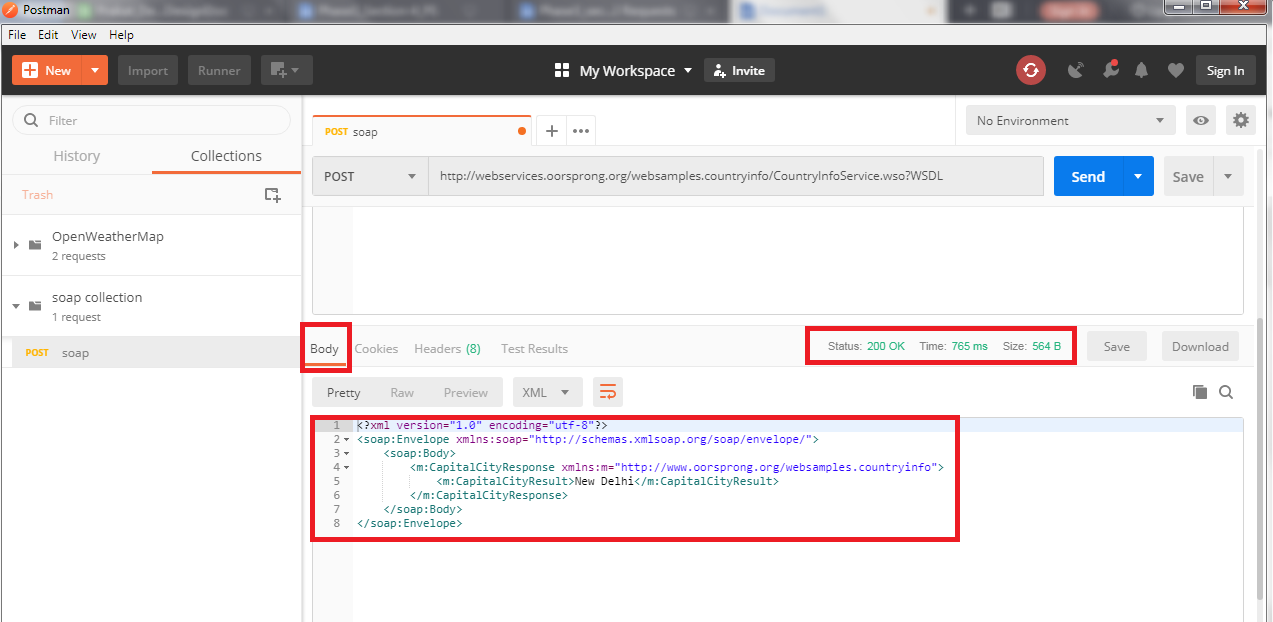
</Body>

</Envelope>

* **IN** keyword in the above code block is the Country code for India.
* Click on the Send.

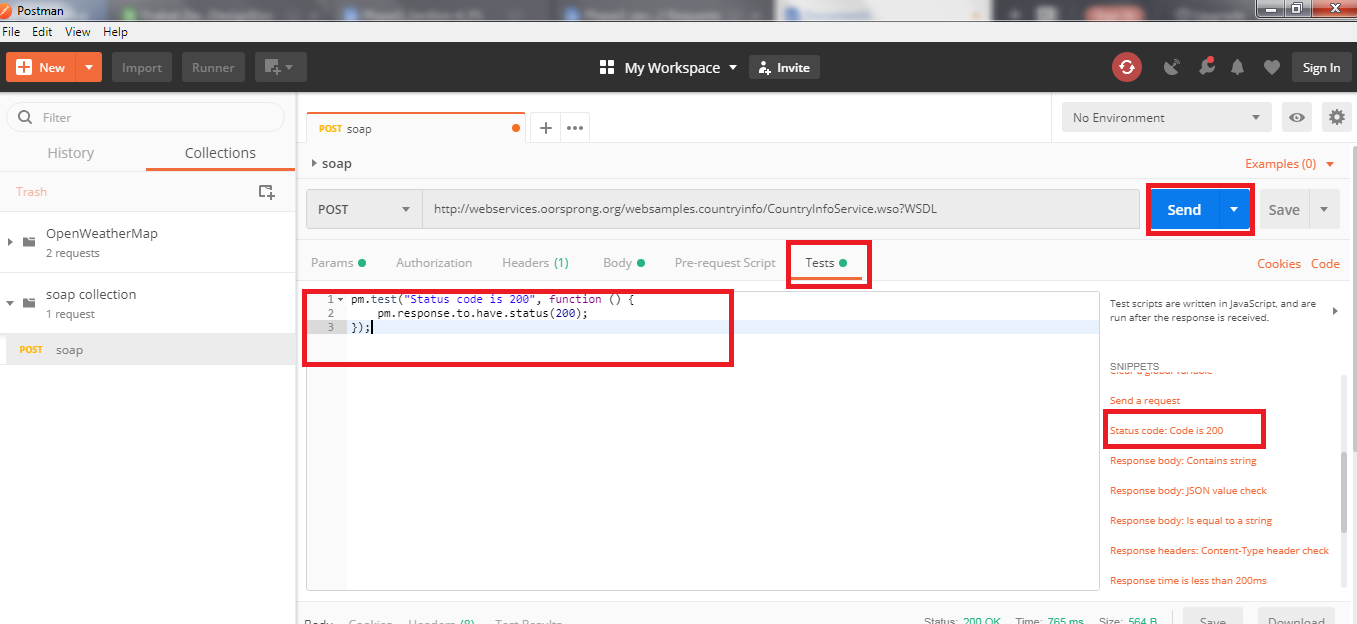


* Scroll down and click on the Body and verify the Response.

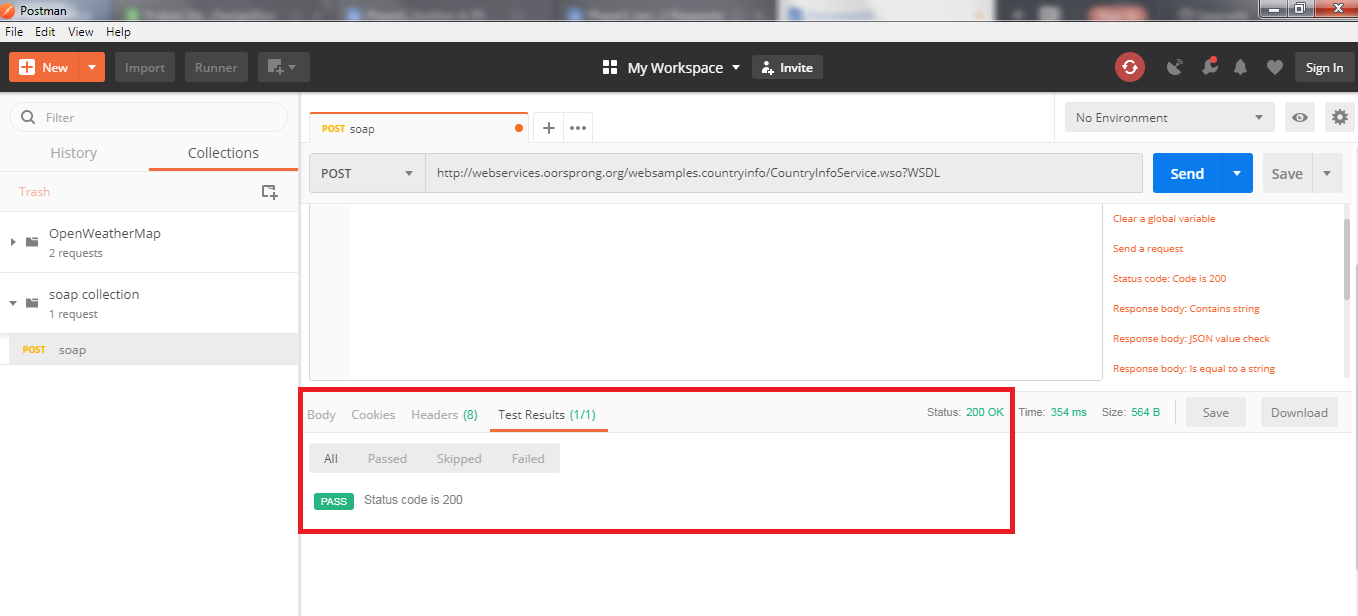


**Step 2.3.2:** Running and validating the response

* Click on Tests.
* Select Status code: Code is 200
* Click on Send again.

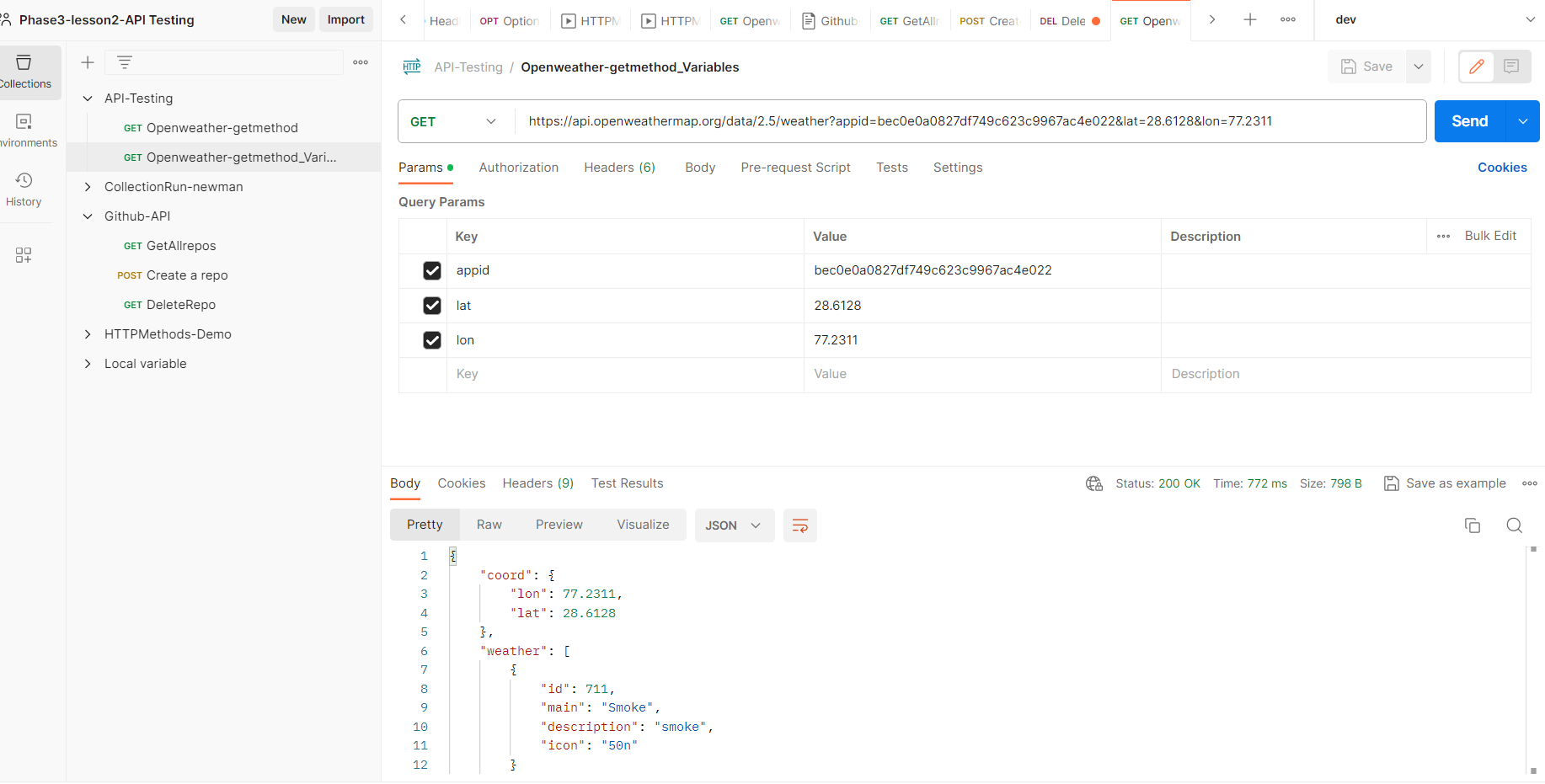


* Scroll down and click on TestResults and verify the response.



* You will see 200 OK Messages.

4.Explain how to work on GET requests, with an example.



5. Demonstrate how to work with POST requests

Working with POST Requests

This section will guide you to understand:

* How to POST a Request on Postman
* Response of POST Request

**Development Environment:**

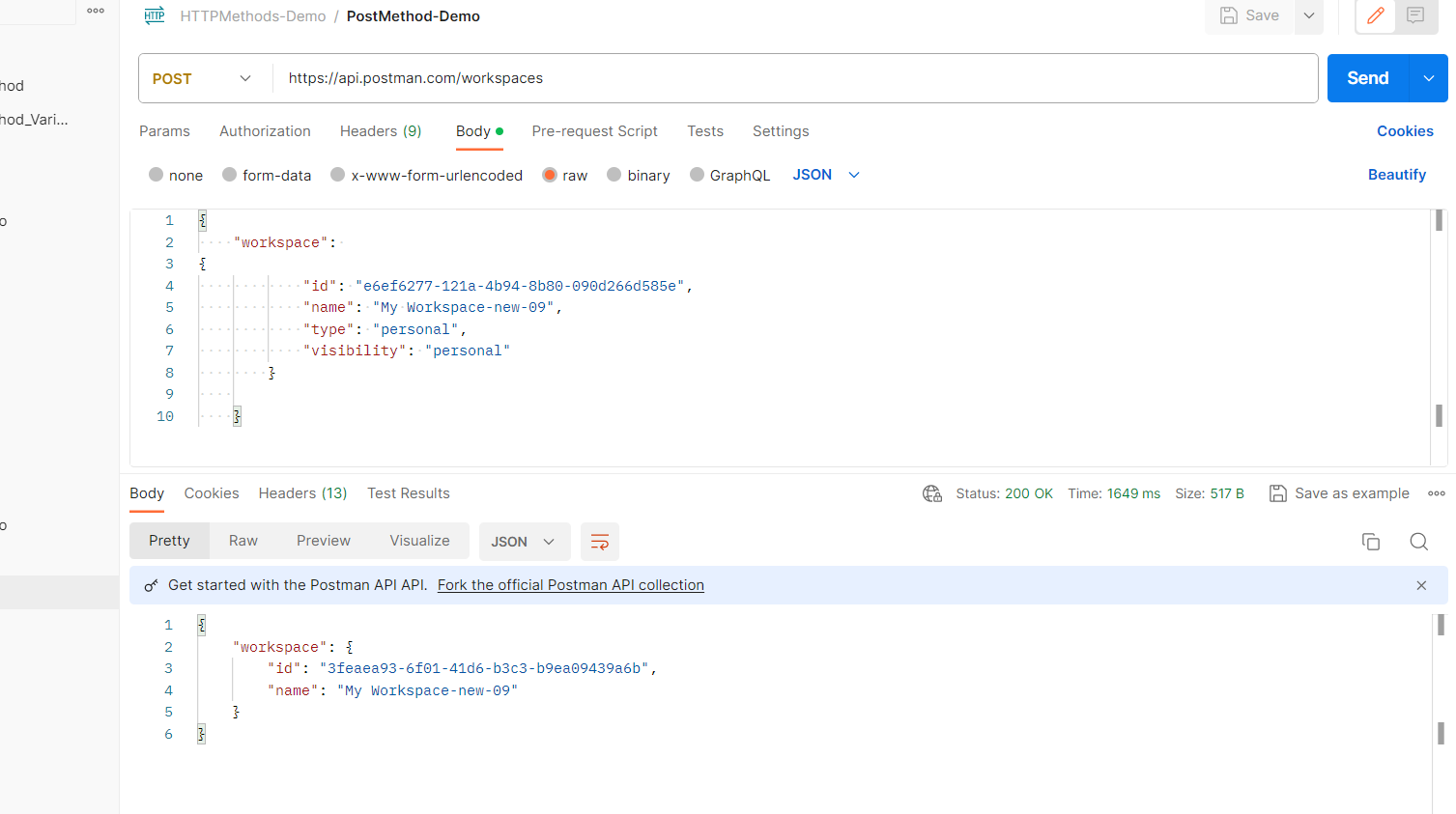
* Postman Tool
* URL

This guide has four subsections, namely:

2.5.1 Creating a POST Request

2.5.2 Writing the body of Request

2.5.3 Checking response



6. Demonstrate how to parameterize requests.

This section will guide you to understand:

* How to use variables with parameters
* Response of a parameterized request

Development Environment:

* Postman Tool
* Endpoint URL

This guide has four subsections, namely:

2.6.1 Checking response before creating Variable

2.6.2 Creating Variable

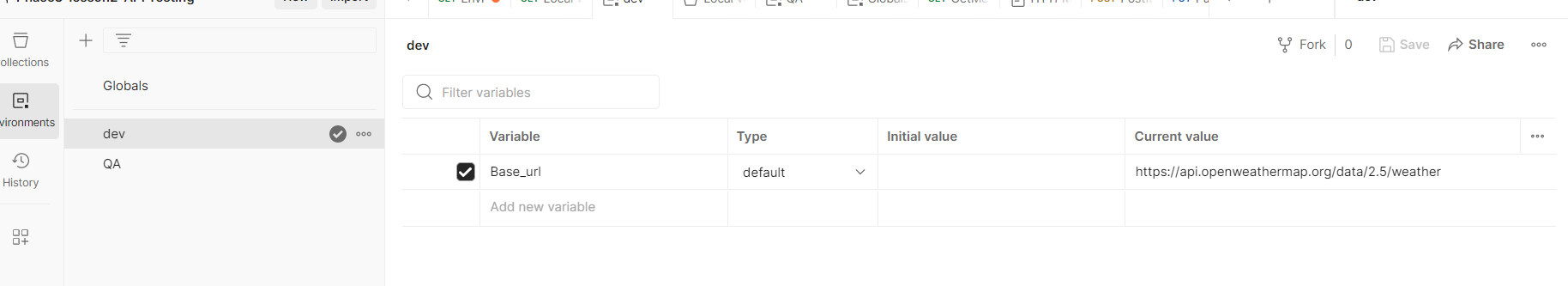
2.6.3 Getting the Response

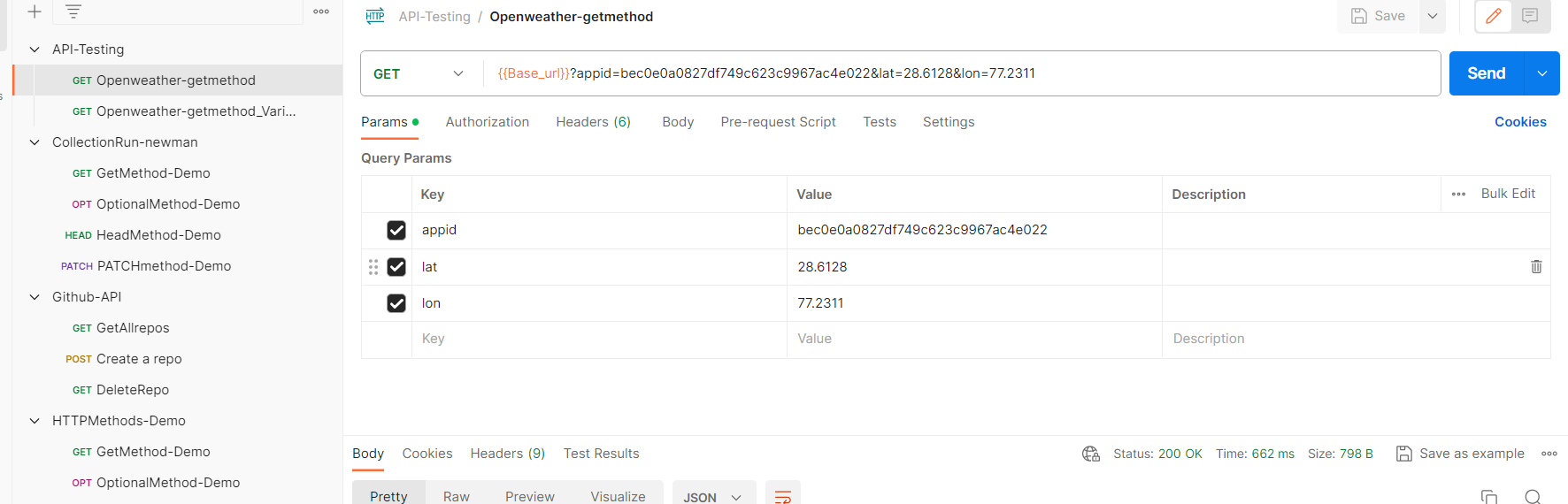
2.6.4 Pushing the code to GitHub repositories

**Steps 2.6.1:** Checking response before creating Variable

* Set your HTTP request to GET
* Input this link: https://reqres.in/api/users?page=2. Replace the first part of the link with a parameter, such as {{url}}. Request url should now be {{url}}/users.
* Click **Send**

**Note:** There should be no response since we have not set the source of our parameter.





7. Demonstrate how to create a Collection.

This section will guide you to understand:

* How to create a Collection
* How to use Collection
* Response Run

**Development Environment:**

* Postman Tool
* Collection
* Endpoint URL

This guide has six subsections, namely:

2.7.1 Creating a Collection

2.7.2 Adding the desired name

2.7.3 Adding a new request in Collection

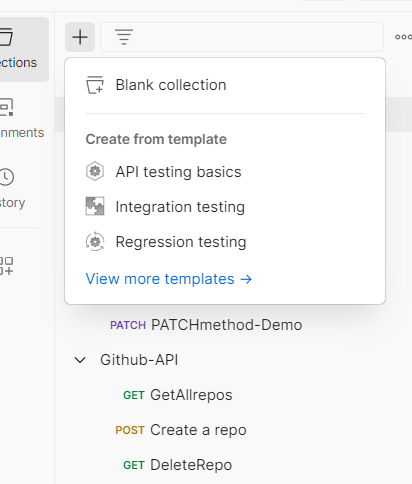
2.7.4 Running that Collection

2.7.5 Checking the output

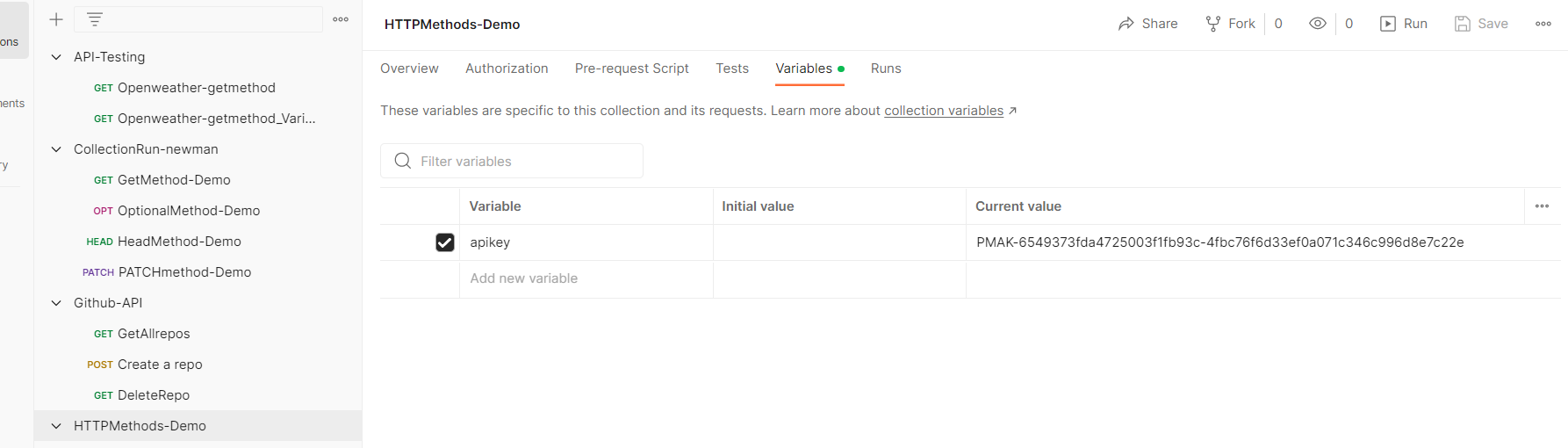
2.7.6 Pushing the code to GitHub repositories

**Steps 2.7.1:** Creating a Collection

* Click on the NEW button
* Then you get a pop up like below:



8.Demonstrate how to work with variables in Postman.



9. Create a request, and collections and variables using Postman.

This guide has three subsections, namely:

1. Problem Statement for creating requests, collections, and variables using Postman.
2. Solution for the Problem Statement.
3. Pushing the code to Github repositories.

**Step 2.1.1:** Problem statement for creating requests, collections, and variables using Postman.

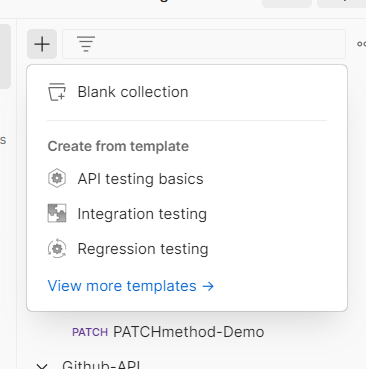
* Objective: Create a collection, add a request, create respective variables in Postman using any sample url, and check the response.
* Steps Involved:

1. Open Postman
2. Create a collection
3. Add requests by using sample url
4. Create variable
5. Validate the response

**Step 2.1.2:** Solution for the problem statement.

* Steps to create a collection

1. Open Postman
2. Go to **New** option
3. Create a collection

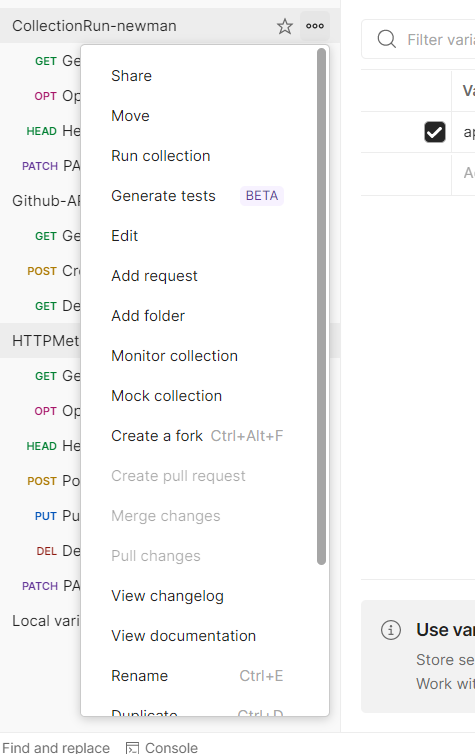


* Steps to set requests

1. Go to **New** option
2. Click on **Request**
3. Save it
4. Use the sample url:

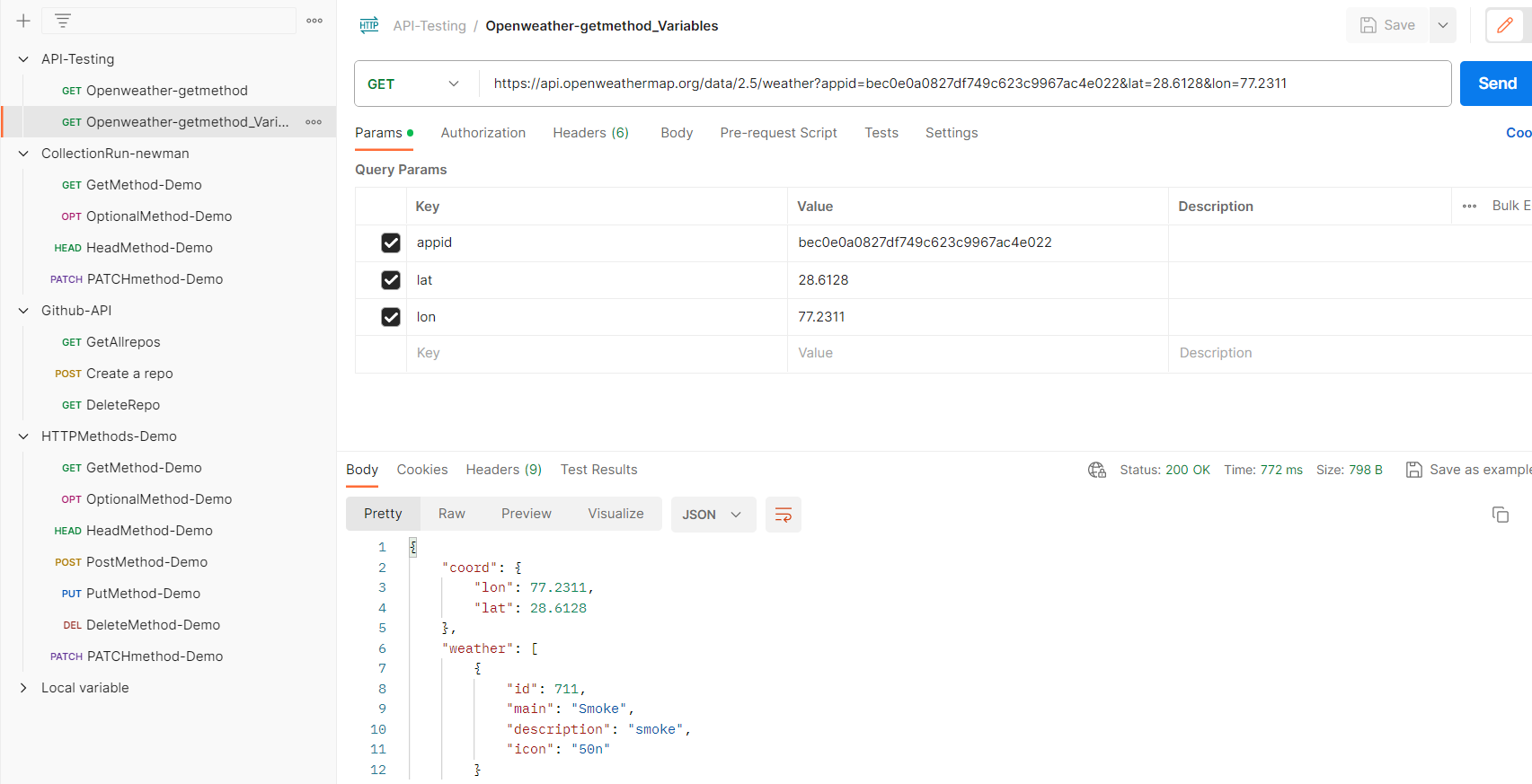
* Steps to set variables

1. Go to collection
2. Click on the Triple dot mark
3. Click **Edit**

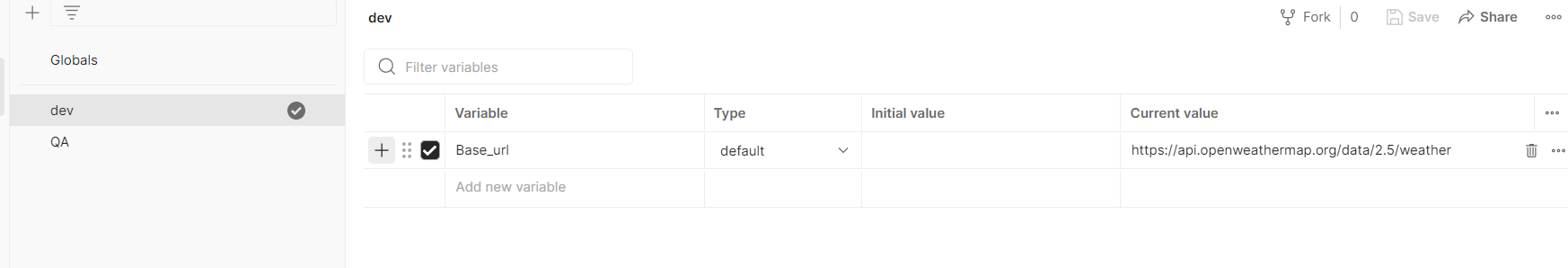


* Validate the response

1. Go to the request
2. Use the variable that was created earlier

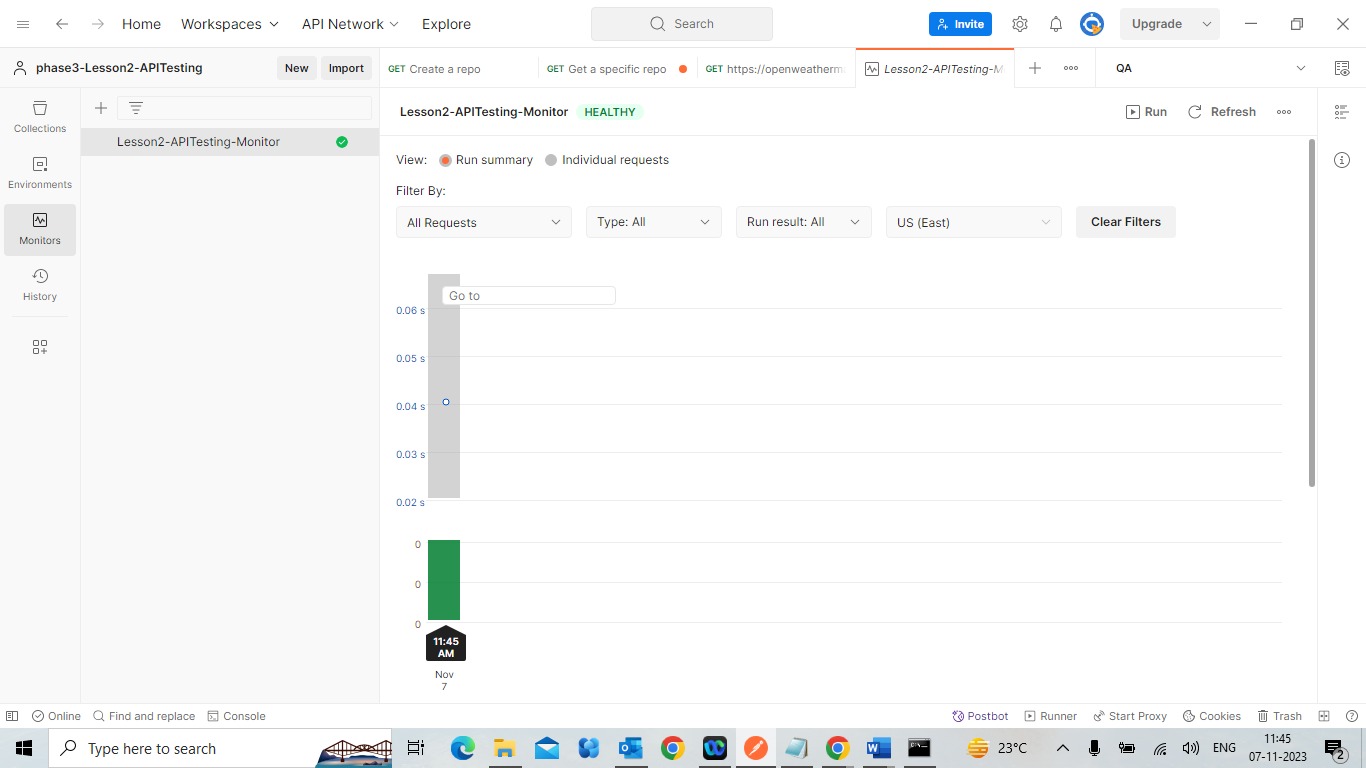


11.Demonstrate how environments are used in Postman.



15. Demonstrate how monitors are used.

Managing Collection

* Create a new Collection
* Scroll down and find the option Browse
* Click on Browse
* Click on Collections
* Click on Monitor Collect

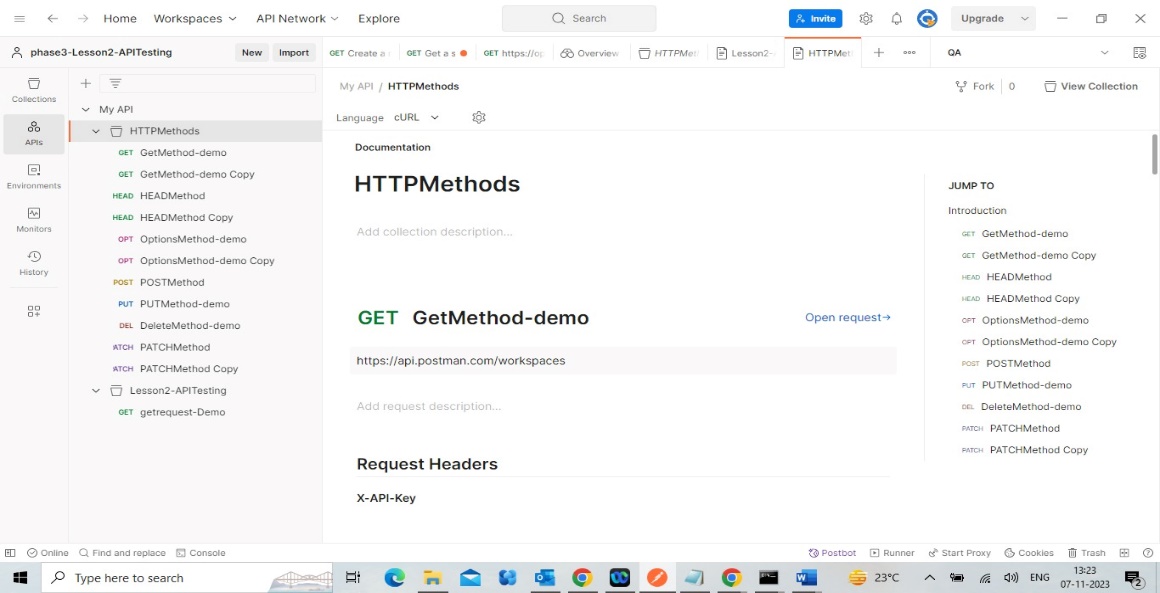
16.   
Demonstrate how to perform API documentation.

Creating a Collection

* Open Postman
* Go to New
* Click on Collection

Creating API documentation

* Go to New
* Click on API Documentation



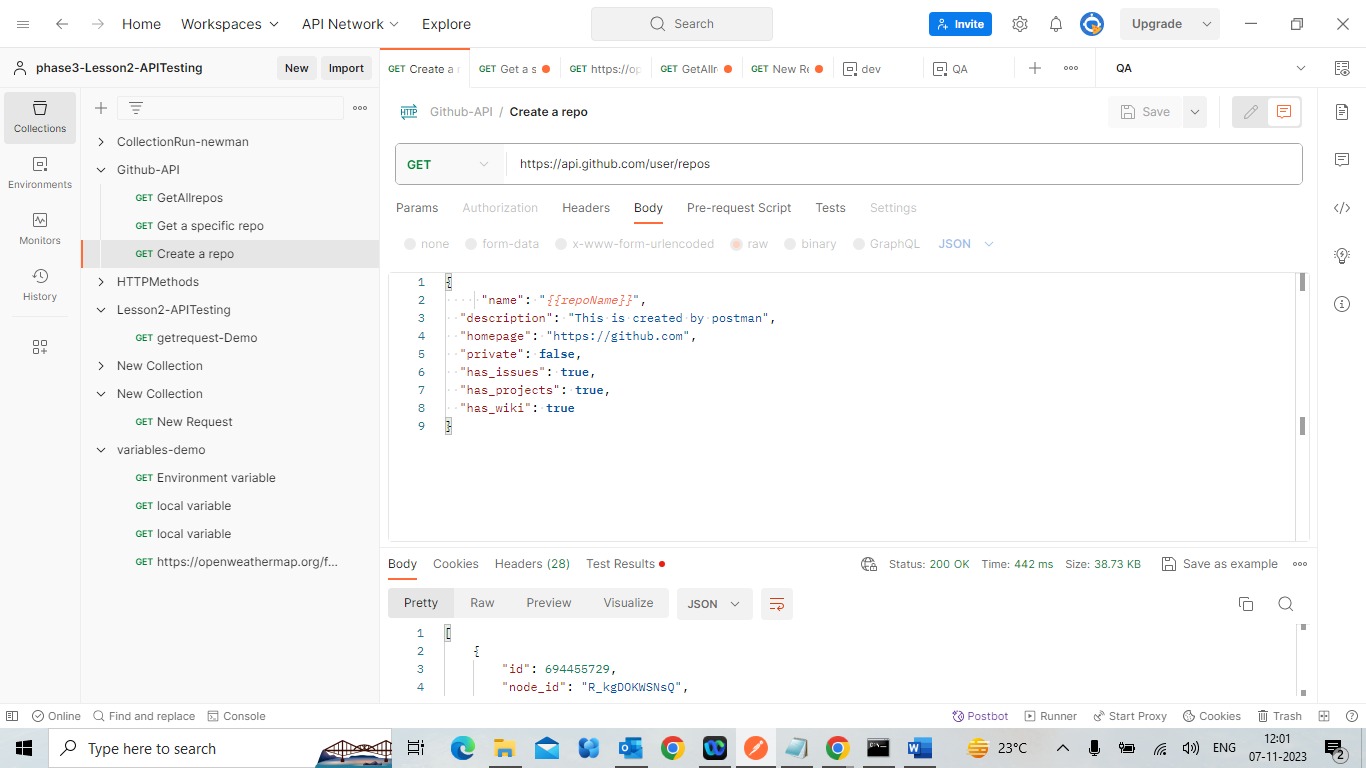
17. Demonstrate how to get data from CSV and JSON.

Creating a new request and setting the environment

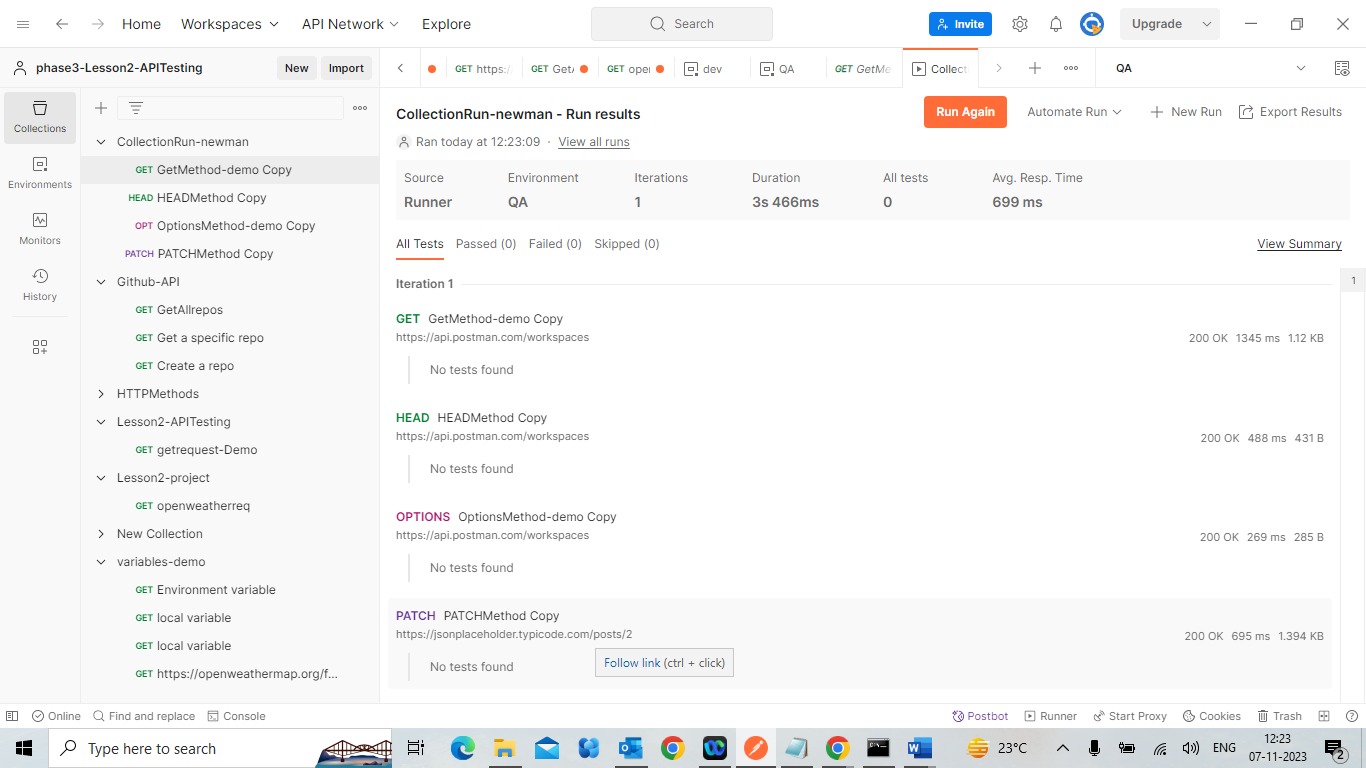
Getting the data from CSV file

Getting the data from JSON file

Pushing the code to your GitHub repositories



18. Demonstrate how to run a collection remotely with URL



20.Demonstrate API Chaining and REST in Postman

* + 1. Creating a Collection
    2. Creating two new requests
    3. Creating an environment
    4. Checking the API chaining response
    5. Pushing the code to GitHub repositories

**Step 2.20.1:** Creating a Collection

* Open Postman
* Click on **New**
* Select Collection

Creating two new requests

* Go to New
* Select Request
* Create GET Request

Create POST Request

