API TESTING USING POSTMAN

MITHILESH SINGH

PREREQUISITE CHECKS

Postman

- Postman is a standalone software testing API (Application Programming Interface) platform to build, test, design, modify, and document APIs.

  Note:
  Application Programming Interface (API) is software that acts as an intermediary for two apps to communicate with each other. We use APIs whenever we use an application like Twitter, Facebook, sending text messages, or checking the weather over the phone.

- Postman can be downloaded from this link: https://www.postman.com/downloads/

Http Request

- HTTP (Hypertext Transfer Protocol), is the underlying format that is used to structure request and responses for effective communication between a client and a server. The message that is sent by a client to a server is what is known as an HTTP request. When these requests are being sent, clients can use various methods.

- An HTTP request is an action to be performed on a resource identified by a given Request-URL. Request methods are case-sensitive, and should always be noted in upper case. There are various HTTP request methods, but each one is assigned a specific purpose.

1. GET method is used to retrieve whatever information is identified by the Request-URL.
2. POST requests are utilized to send data to a server to create or update a resource.
3. PUT is similar to POST as it is used to send data to the server to create or update a resource.
4. DELETE request method is used to delete resources indicated by a specific URL.

Note:
A PATCH request is similar to POST and PUT. However, its primary purpose is to apply partial modifications to the resource

GET vs POST Request Methods

GET	POST
• Has restriction on data type as the only allowed data type is ASCII characters	• There is no restriction on data type, and binary data is also allowed
• Can be bookmarked.	• Cannot be bookmarked.
• maximum URL length is 2048 characters.	• There are no restriction on data length.
• Can be cached	• Can’t be cached

Basic Understanding:

- **URI**(Uniform Resource Identifiers).

  e.g. https://reqres.in/api/users?page=2
- **URL**(Uniform Resource Locators).

  e.g. https://reqres.in ---> Domain
- /api/users ---> **Path parameters**(It will get the data from the server based on path which we provided).
- ?page=2 ---> **Query Parameters**(It will filter the data).

Note: We will be using regres dummy API for learning API testing through Postman.

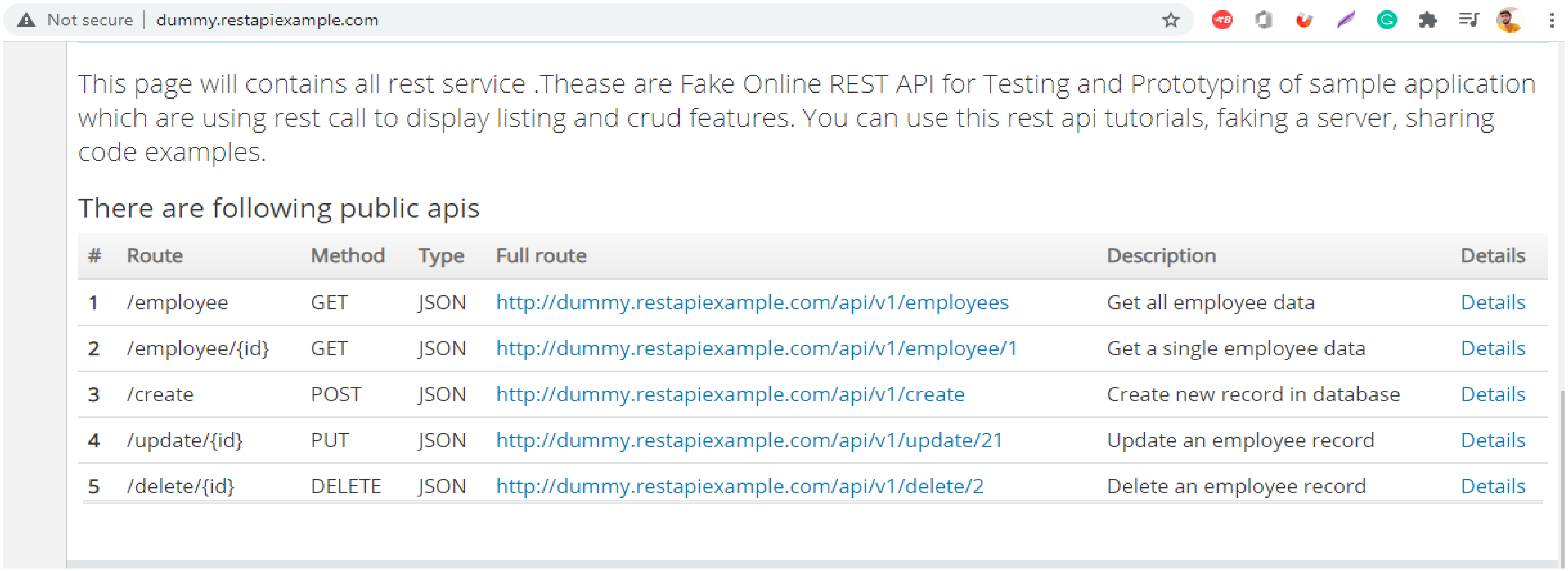
Dummy API's:

There are so many online sites which provide dummy API's for testing.

1. http://dummy.restapiexample.com/
2. https://reqres.in/api/users?page=2

Note:

we can easily find out most important http methods – GET, POST, PUT, DELETE in above links.



Not secure | dummy.restapiexample.com

This page will contains all rest service .Thease are Fake Online REST API for Testing and Prototyping of sample application which are using rest call to display listing and crud features. You can use this rest api tutorials, faking a server, sharing code examples.

There are following public apis

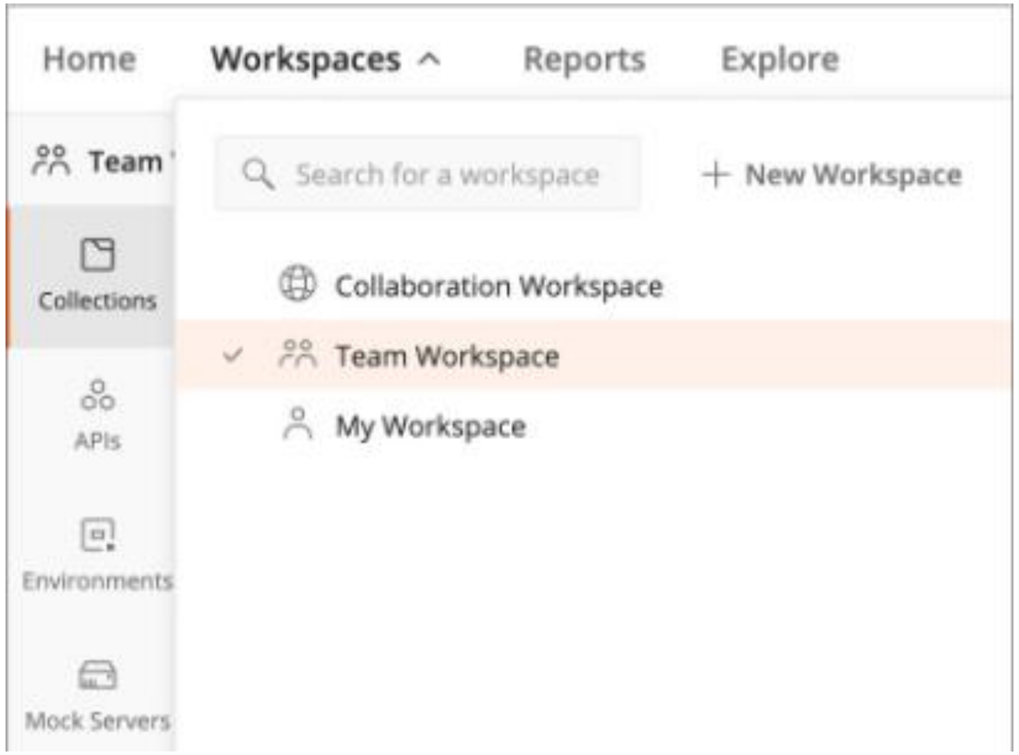
#	Route	Method	Type	Full route	Description	Details
1	/employee	GET	JSON	http://dummy.restapiexample.com/api/v1/employees	Get all employee data	Details
2	/employee/{id}	GET	JSON	http://dummy.restapiexample.com/api/v1/employee/1	Get a single employee data	Details
3	/create	POST	JSON	http://dummy.restapiexample.com/api/v1/create	Create new record in database	Details
4	/update/{id}	PUT	JSON	http://dummy.restapiexample.com/api/v1/update/21	Update an employee record	Details
5	/delete/{id}	DELETE	JSON	http://dummy.restapiexample.com/api/v1/delete/2	Delete an employee record	Details

Workspace in Postman

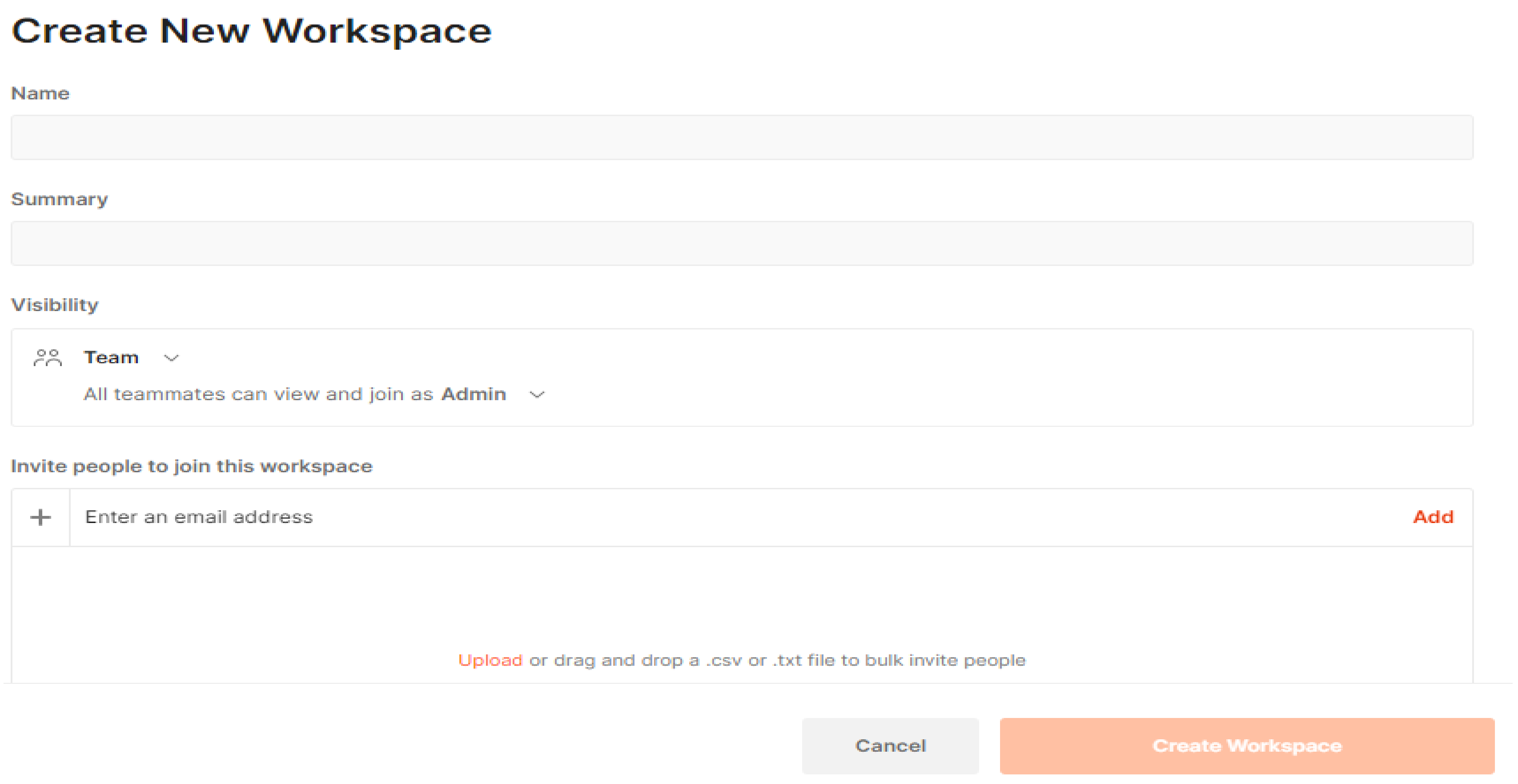
- Workspaces allow you to organize your Postman work and collaborate with teammates.
- Personal workspaces are visible only to you—with a Postman_account you can create unlimited workspaces. With team workspaces, you can share and manage access to project components with collaborators.

Creating a new workspace: Way 1

- 1. To create a new workspace, select the workspace dropdown menu at the left of Postman, and click **New Workspace**.



2.Use the visibility dropdown to choose a **Team**, **Private** or **Personal and** To create a personal workspace, enter the workspace name and summary. Select the visibility dropdown menu and choose **Personal**.



Create New Workspace

Name

Summary

Visibility

Team

All teammates can view and join as **Admin**

Invite people to join this workspace

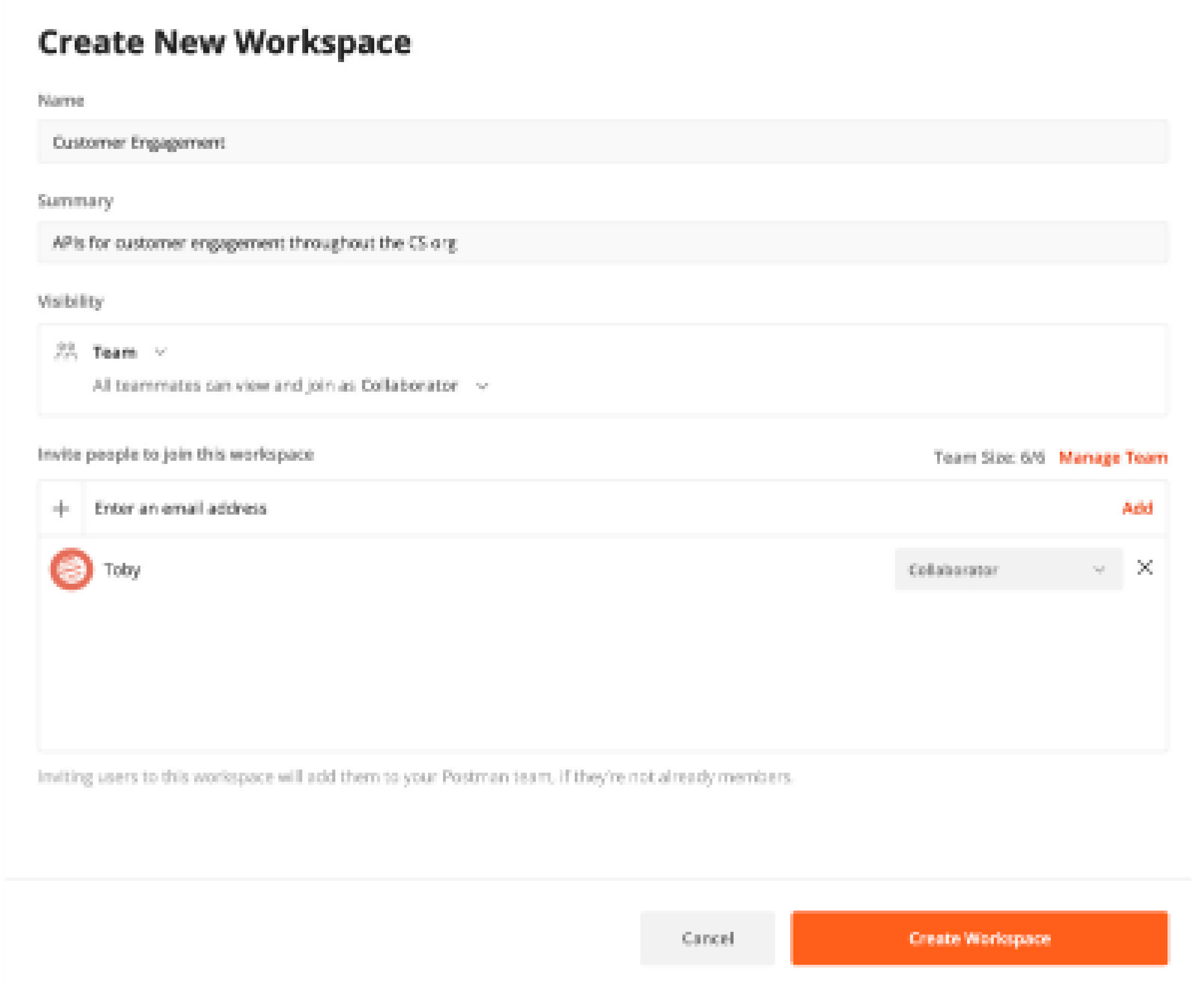
Enter an email address

Add

Upload or drag and drop a .csv or .txt file to bulk invite people

Cancel

Create Workspace



Create New Workspace

Name

Customer Engagement

Summary

APIs for customer engagement throughout the CS org

Visibility

Team

All teammates can view and join as Collaborator

Invite people to join this workspace

Team Size: 6/6 Manage Team

Enter an email address Add

Toby Collaborator

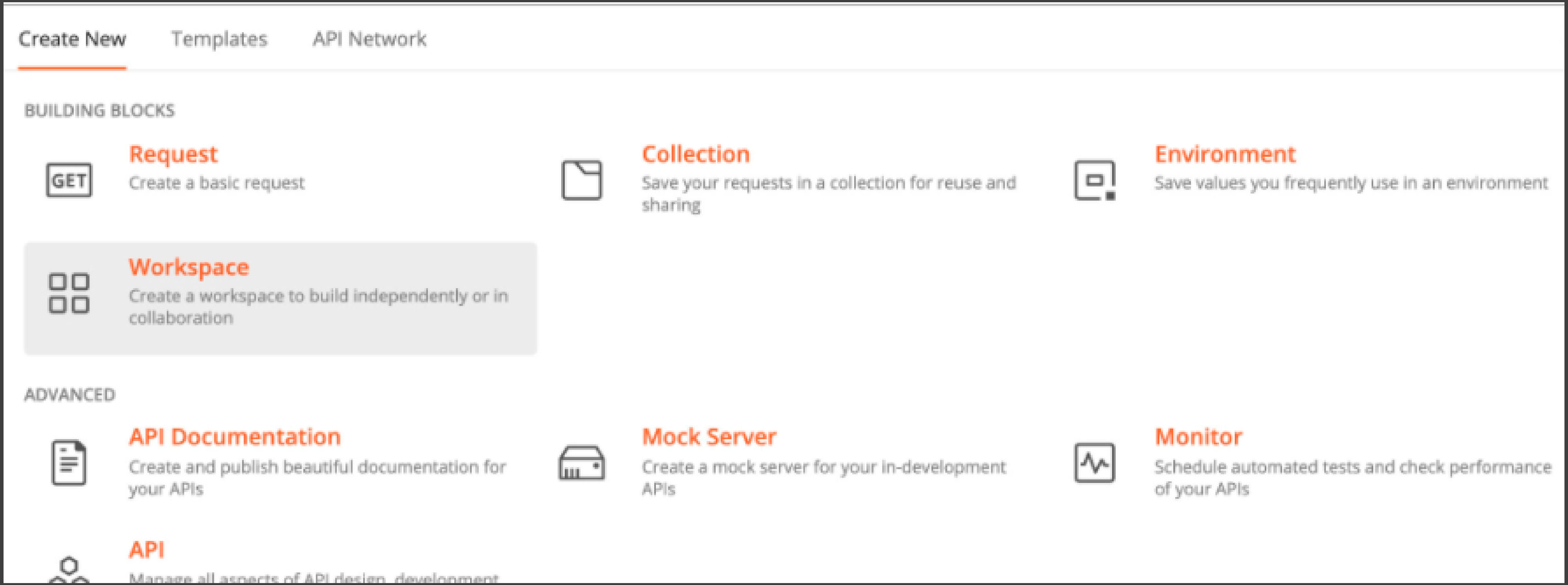
Inviting users to this workspace will add them to your Postman team, if they're not already members.

Cancel Create Workspace

2. To create a team workspace, enter the workspace name and summary. Select the visibility dropdown menu and choose **Team**. Add collaborators by entering their email addresses, then define their workspace roles

Note:

You can add elements to the workspace and invite new members using the **Invite** button at the top at any time.


Create New Templates API Network

BUILDING BLOCKS

Request
Create a basic request

Collection
Save your requests in a collection for reuse and sharing

Environment
Save values you frequently use in an environment

Workspace
Create a workspace to build independently or in collaboration

ADVANCED

API Documentation
Create and publish beautiful documentation for your APIs

Mock Server
Create a mock server for your in-development APIs

Monitor
Schedule automated tests and check performance of your APIs

API
Manage all aspects of API design, development

Creating a new workspace: Way 2

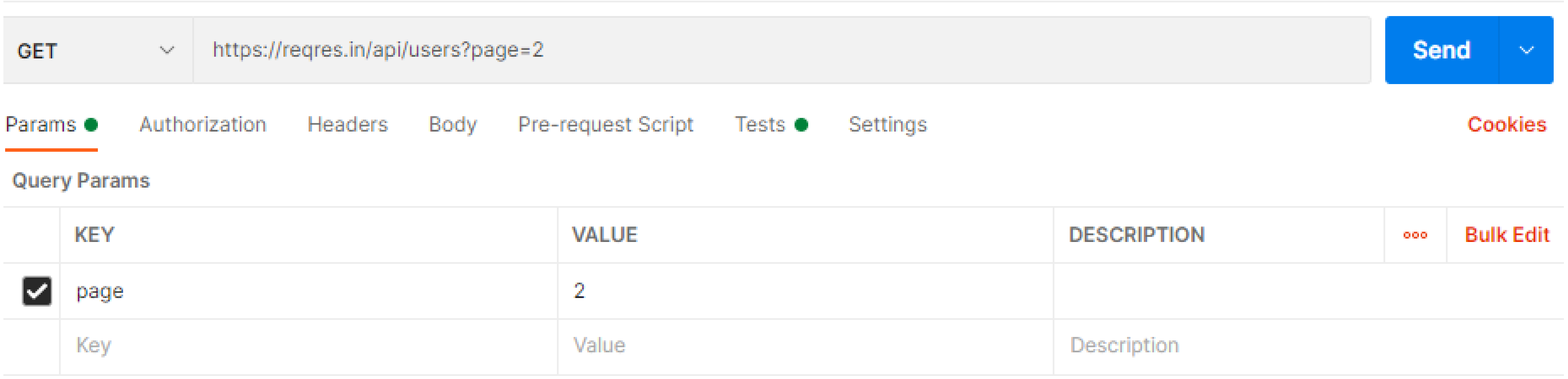
You can also create a new workspace in the Workspaces dashboard.

Click the **New** button above the navigation bar, select **Workspace** and follow the same steps.

Testing GET Request

- To test this we will have to use GET request.
- For sample requests, visit https://reqres.in/
- Make a collection in Postman — To make a collection in Postman, click on **New->Collection->CollectionDemo(Any Collection Name you wish)->Create**
- Make a Request — To make a request, click on **New->Request->GetUser(Any request name you wish)->Select the Collection** you wish to save request in(Present in bottom of dialog box)->Save to Collection Demo
- By now, we have created our first request, now we need to pass different parameters in the request to get the expected response.
- In the "Enter Request URL" text box type : https://reqres.in/api/users?page=2
- Click on "Send" Button

REQUEST

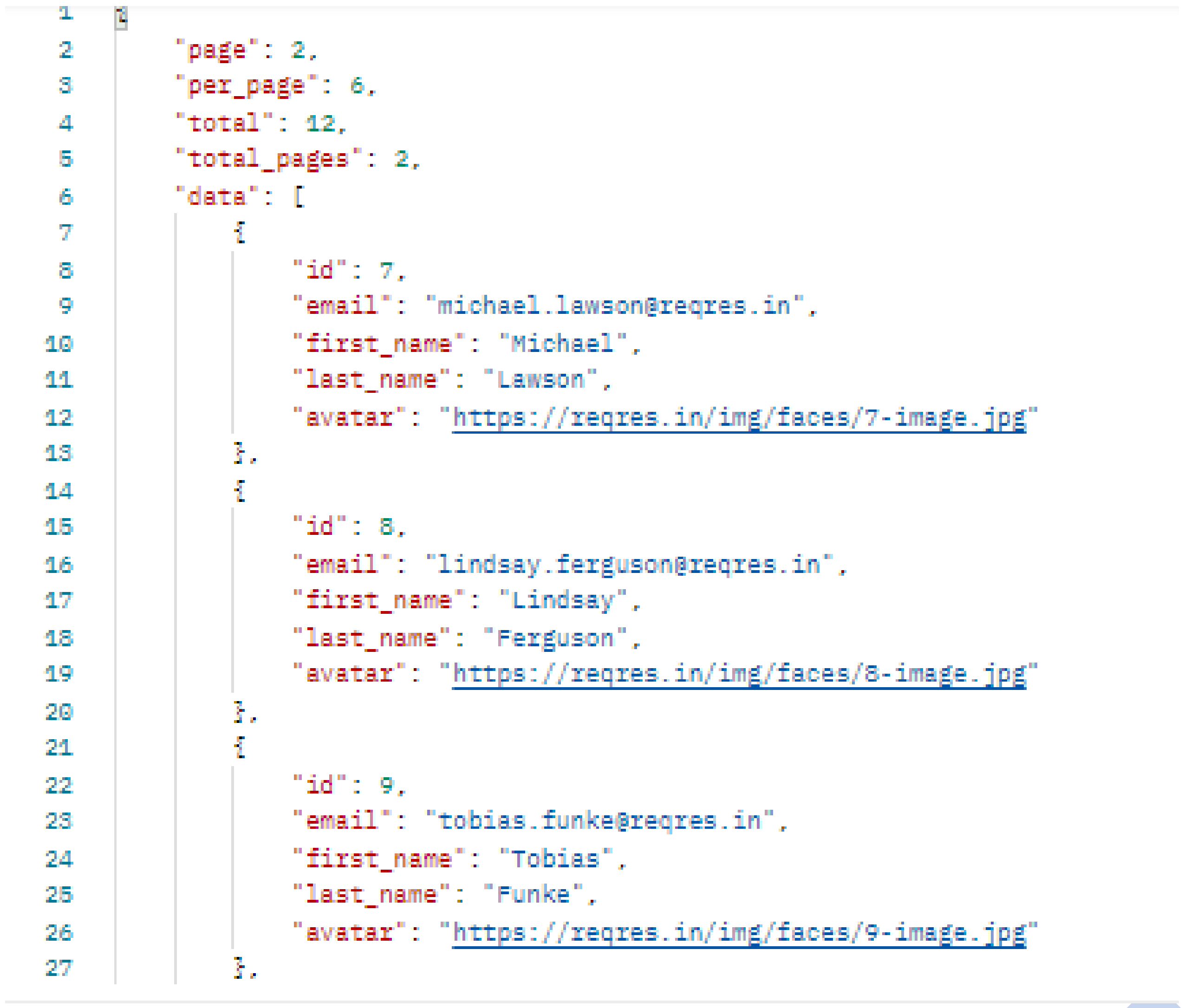


GET https://reqres.in/api/users?page=2

Query Params

	KEY	VALUE	DESCRIPTION
✓	page	2	
	Key	Value	Description

RESPONSE



```
{
    "page": 2,
    "per_page": 6,
    "total": 12,
    "total_pages": 2,
    "data": [
        {
            "id": 7,
            "email": "michael.lawson@reqres.in",
            "first_name": "Michael",
            "last_name": "Lawson",
            "avatar": "https://reqres.in/img/faces/7-image.jpg"
        },
        {
            "id": 8,
            "email": "lindsay.ferguson@reqres.in",
            "first_name": "Lindsay",
            "last_name": "Ferguson",
            "avatar": "https://reqres.in/img/faces/8-image.jpg"
        },
        {
            "id": 9,
            "email": "tobias.funke@reqres.in",
            "first_name": "Tobias",
            "last_name": "Funke",
            "avatar": "https://reqres.in/img/faces/9-image.jpg"
        },
```

How to validate Get Request

- We can write snippet of code to validate the GET request in the postman tool.
- Steps:
  Click on Tests(available below to URL textbox in postman) and add the code.
-
  1. tests["Validating Status Code"] = responseCode.code == 200;
  2. tests["Validating response body"] = responseBody.has("data");
  3. var response = JSON.parse(responseBody);
  4. tests["page no"] = response.page ==2;
- **Note:** line no.2 says "data" will be validated in response body, where ever it is present.
  Line no.4 represent the specific row where page no. Needs to be presented for validation.

Testing POST Requests

Now, suppose we need to create a user into a application that means we are sending data or feeding data to an application. For these type of requests we use POST request. In POST request we send data/parameter in the body of the request, and in response to that, API returns some data to us which validates the user has been created. The response can either be a success message or the id of the new user created and time when the user was created.

Post Request In Postman

1. To make a POST request, **click on New->Request->CreateUser(Any request name you wish)->Select the Collection you wish to save request in(Present in bottom of dialog box)->Save to Collection Demo**

2. From the Dropdown select POST

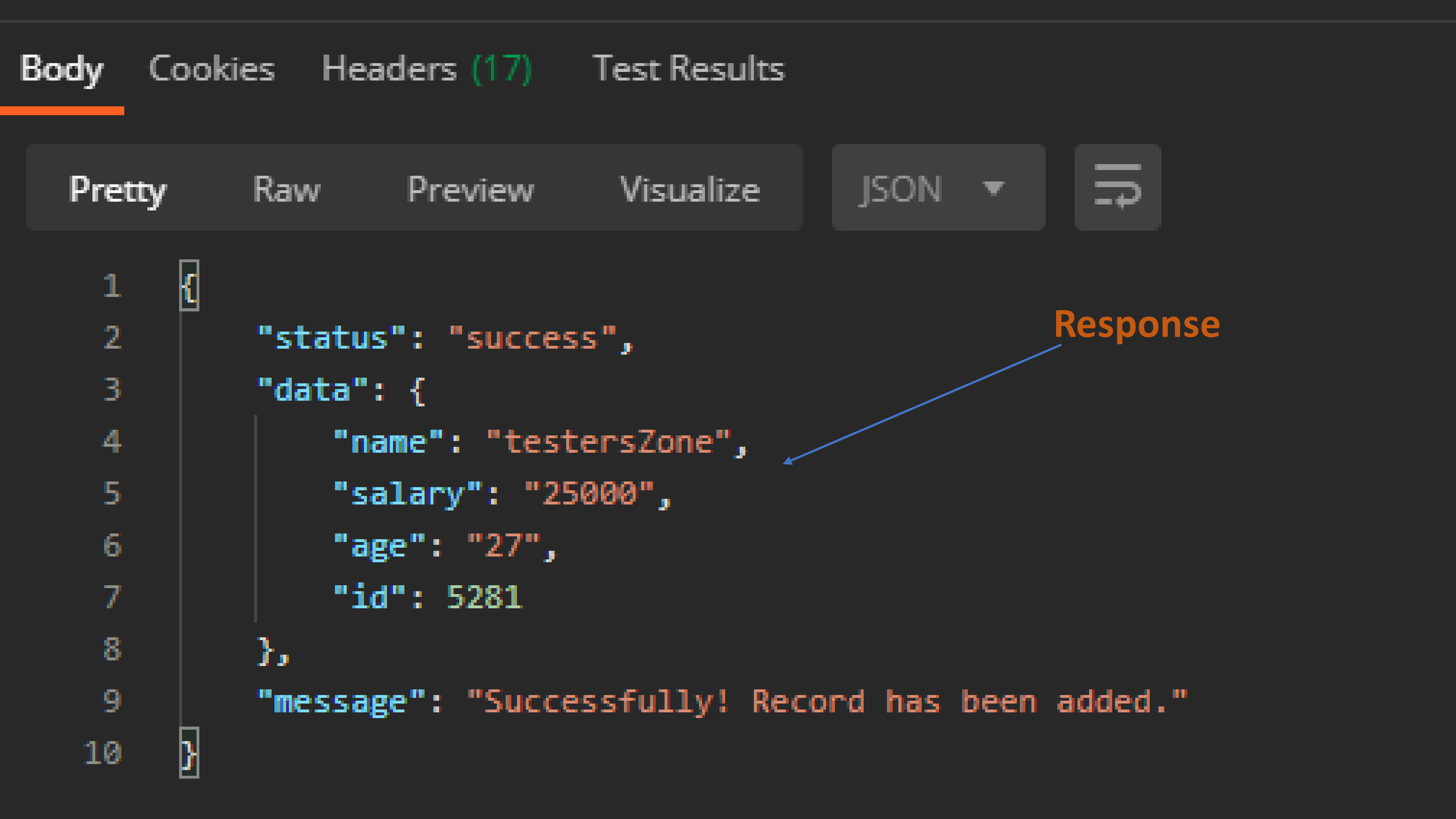
3. In the "Enter Request URL" text box, type : **http://dummy.restapiexample.com/api/v1/create**

4. Click on Body Tab and select "Raw" radio button and json type,

In the text box, paste :

```
{
   "name":"testersZone",
   "salary":"25000",
   "age":"27"
}
```

5. Click on Send button.



```
{
    "status": "success",
    "data": {
        "name": "testersZone",
        "salary": "25000",
        "age": "27",
        "id": 5281
    },
    "message": "Successfully! Record has been added."
}
```

Key Points:

- Some time we need authentication like user id-password, access key etc while sending post request. We can practice those kind of post request using below details.

- All the status code can be viewed in detail way in given link: https://developer.amazon.com/docs/amazon-drive/ad-restful-api-response-codes.html

1. **Basic Auth:**

https://postman-echo.com/basic-auth

username: postman

Password : password

2. **API Key Auth:**

https://api.openweathermap.org/data/2.5/forecast/daily?q=Delhi&cnt=1

3. **Bearer Token / OAuth 2.0**

https://developer.github.com/v3/repos/

API Key/appid: fe9c5cddb7e01d747b4611c3fc9eaf2c

Testing PUT Request

PUT is used to send data to a server to create/update a resource.
The difference between POST and PUT is that PUT requests are idempotent.

That is, calling the same PUT request multiple times will always produce the same result. In contrast, calling a POST request repeatedly have side effects of creating the same resource multiple times.

Put Request In Postman

1. To make a PUT request, **click on New->Request->UpdateUser(Any request name you wish)->Select the Collection you wish to save request in(Present in bottom of dialog box)->Save to Collection Demo**

2. From the Dropdown select PUT

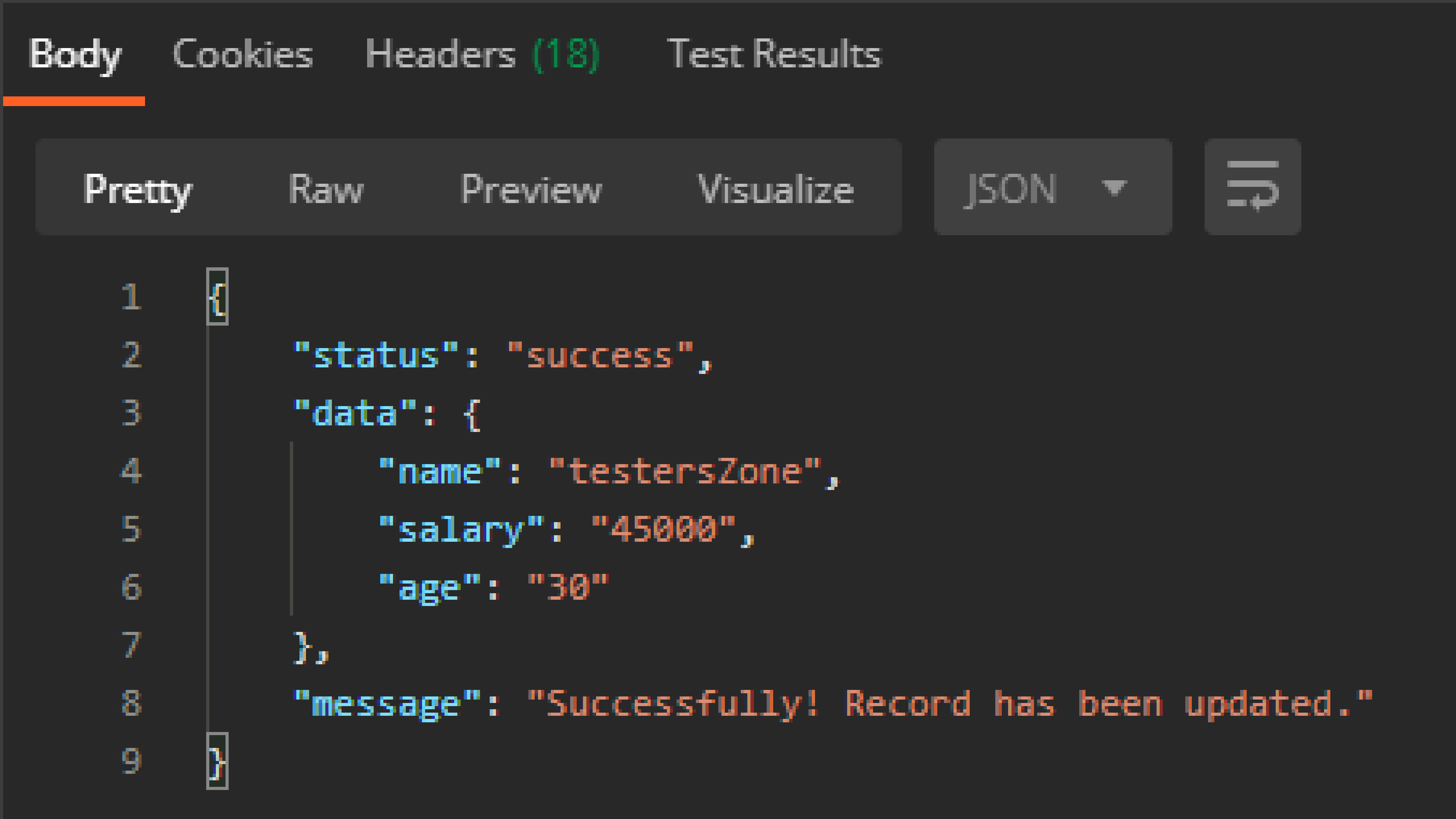
3. In the “Enter Request URL” text box, type : http://dummy.restapiexample.com/api/v1/update/5281
Note: 5281 is a id of user which we have created in post request.Now we want to update salary and age of that user.

4. Click on Body Tab and select “Raw” radio button and json type,

In the text box, paste :

```
{
   "name":"testersZone",
   "salary":"45000",
   "age":"30"
}
```

5. Click on Send button.



```
{
    "status": "success",
    "data": {
        "name": "testersZone",
        "salary": "45000",
        "age": "30"
    },
    "message": "Successfully! Record has been updated."
}
```

Testing Delete Request

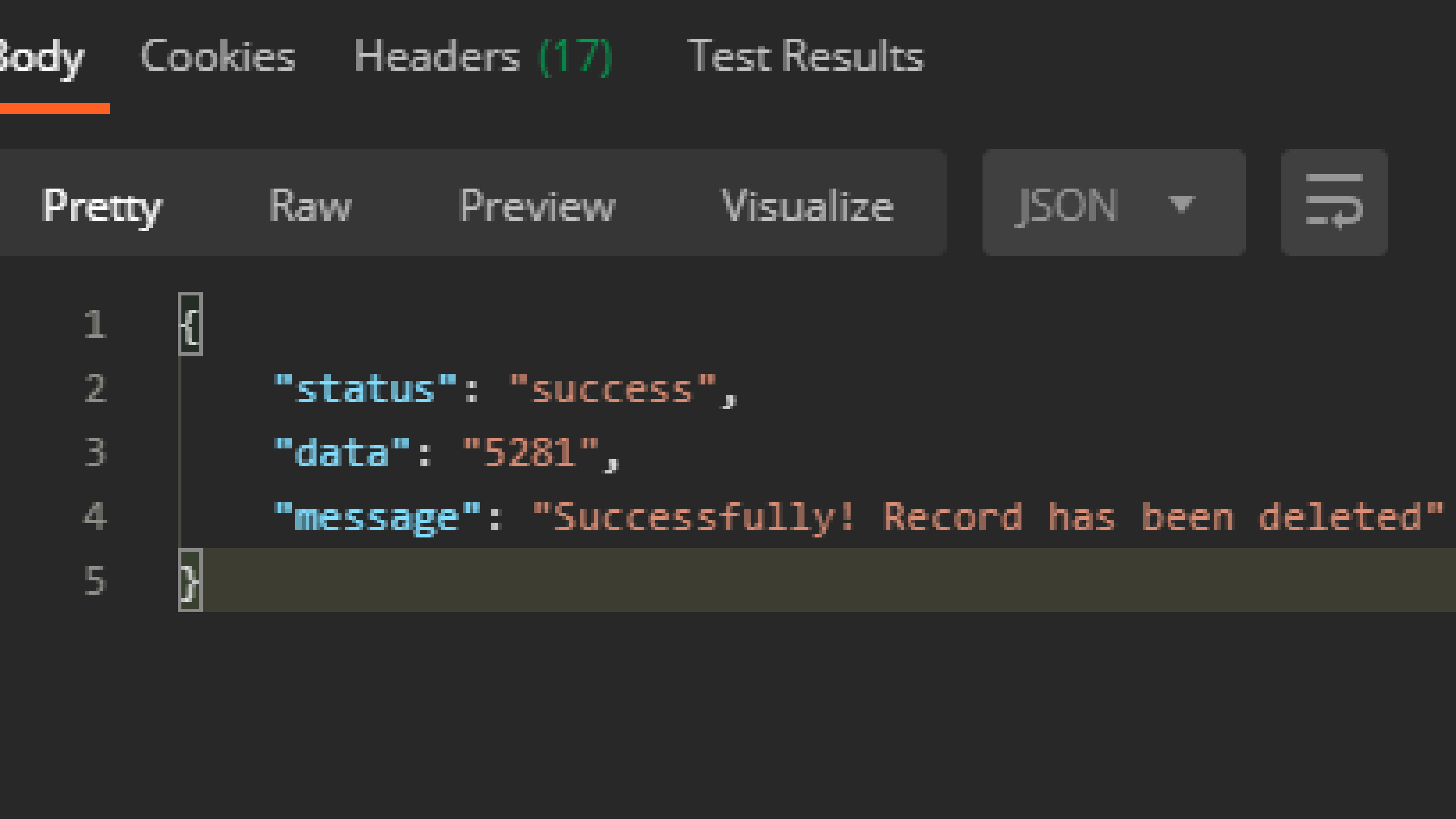
"The DELETE method deletes the specified resource."

1. To make a DELETE request, **click on New->Request->DeleteUser(Any request name you wish)->Select the Collection you wish to save request in(Present in bottom of dialog box)->Save to Collection Demo**

2. From the Dropdown select DELETE

3. In the "Enter Request URL" text box, type : http://dummy.restapiexample.com/api/v1/delete/5281
Note: **5281 is a id of user which we have created in post request.Now we want to delete that user.**

5. Click on Send button.



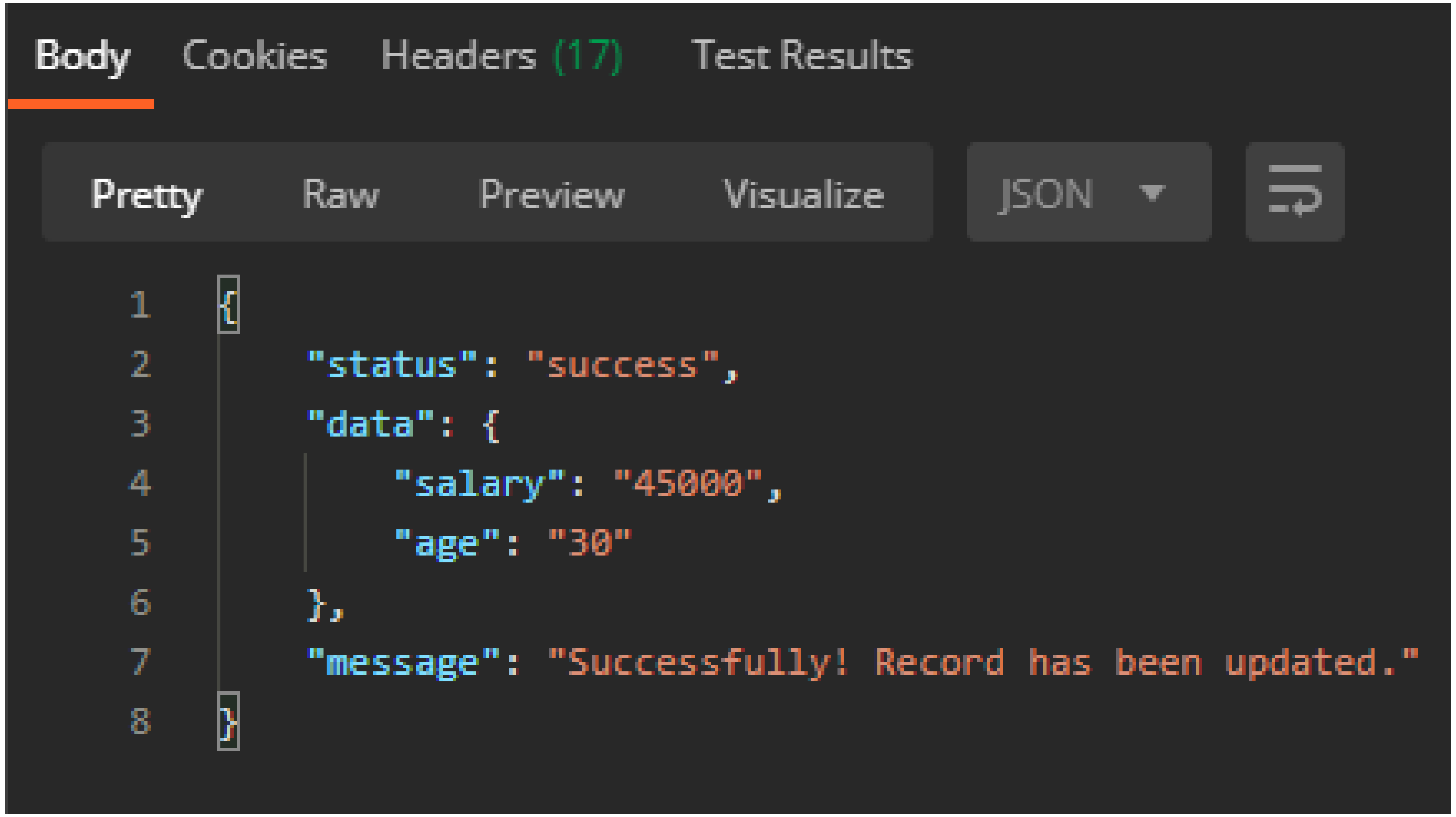
```
{
    "status": "success",
    "data": "5281",
    "message": "Successfully! Record has been deleted"
}
```

A **PATCH** request also modifies an existing resource but it only contains the data that needs to be changed.

Note

Request Body: In patch we pass only those key value pair which needs to be updated.

```
{
  "salary":"45000",
  "age":"30"
}
```



```
{
    "status": "success",
    "data": {
        "salary": "45000",
        "age": "30"
    },
    "message": "Successfully! Record has been updated."
}
```

Verify Response Header's values under postman

- **Here is the few lines of code which we can use:**

To verify the header content we have to use below java script code

```
pm.test("check Content-Type header", function()
{
  pm.response.to.be.header("Content-Type","application/json");
})
```

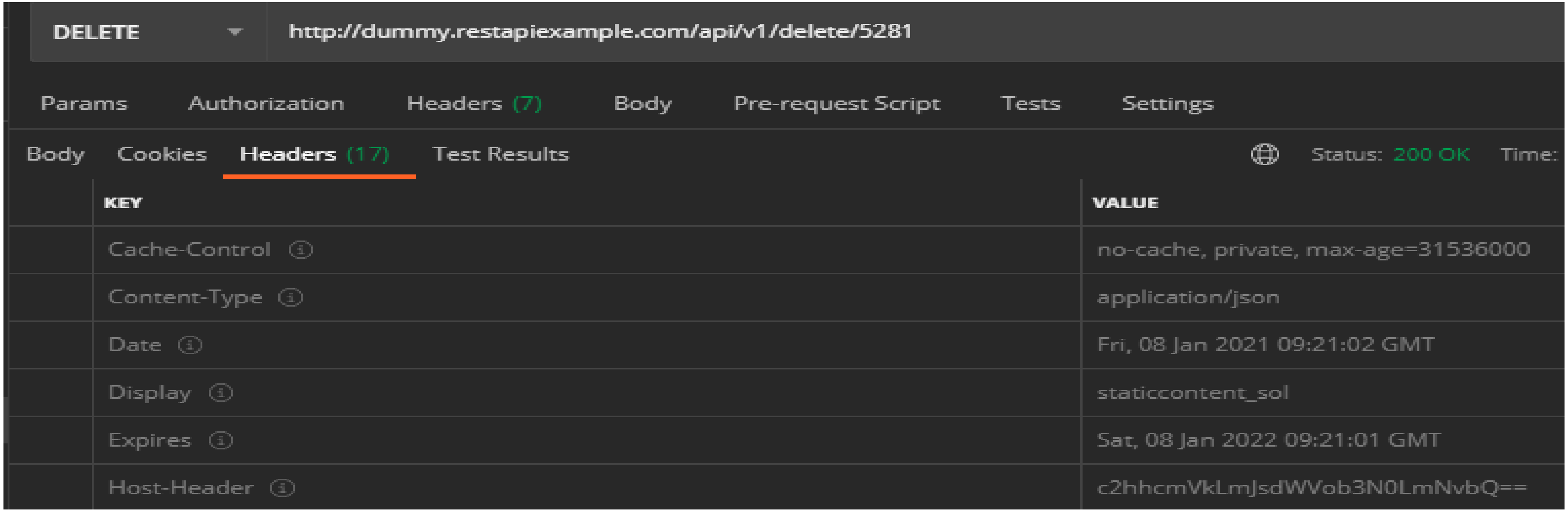
Above code will verify the content-Type's value. Observe in attachment.

Note:

Where I should write this code?

Ans: Under Tests option in the postman.

Observe in the attachment



DELETE http://dummy.restapiexample.com/api/v1/delete/5281

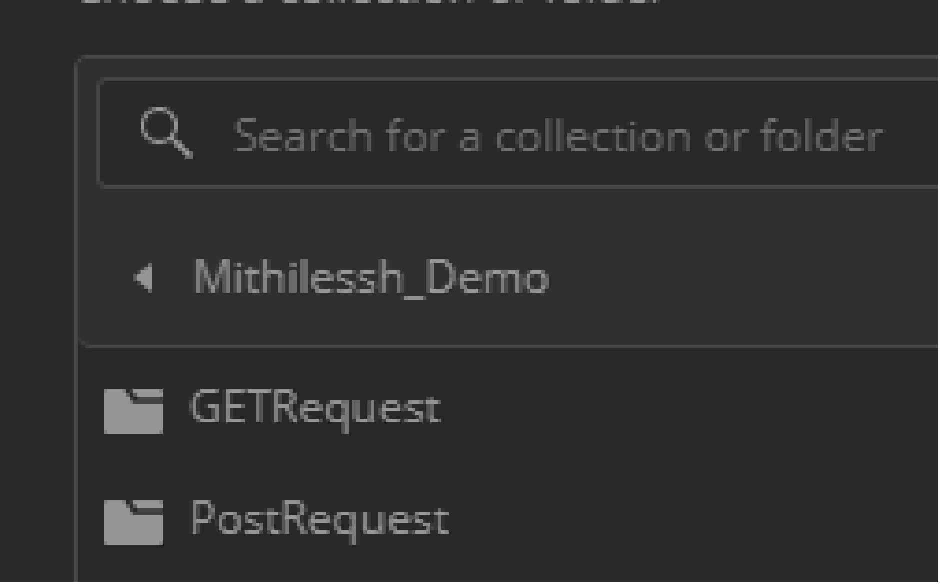
KEY	VALUE
Cache-Control	no-cache, private, max-age=31536000
Content-Type	application/json
Date	Fri, 08 Jan 2021 09:21:02 GMT
Display	staticcontent_sol
Expires	Sat, 08 Jan 2022 09:21:01 GMT
Host-Header	c2hhcmVkLmJsdWVob3N0LmNvbQ==

Data Driven Testing In Postman

- **Data-driven testing** is when we have one **test** that we run multiple times with different **data** variables.
- We can put those data variables in the csv or notepad file and use that file in postman
  Steps:
  Pre-requisite: create a csv file as per post request body(also know as payload) and save it in system.

****online json to csv or csv to json conversion link:**
https://csvjson.com/csv2json
https://csvjson.com/json2csv

- **Click on Runner option available at top left corner next to import option--> select the collection folder--> select the request under the collection folder--> put iteration value(total no of times you want to repeat this request)--> Select File(created csv file) in Data option--> Click on Run**.
  Request will take the data variables from the csv file and access the request as per iteration value.
  Observe the attachment in next slide.

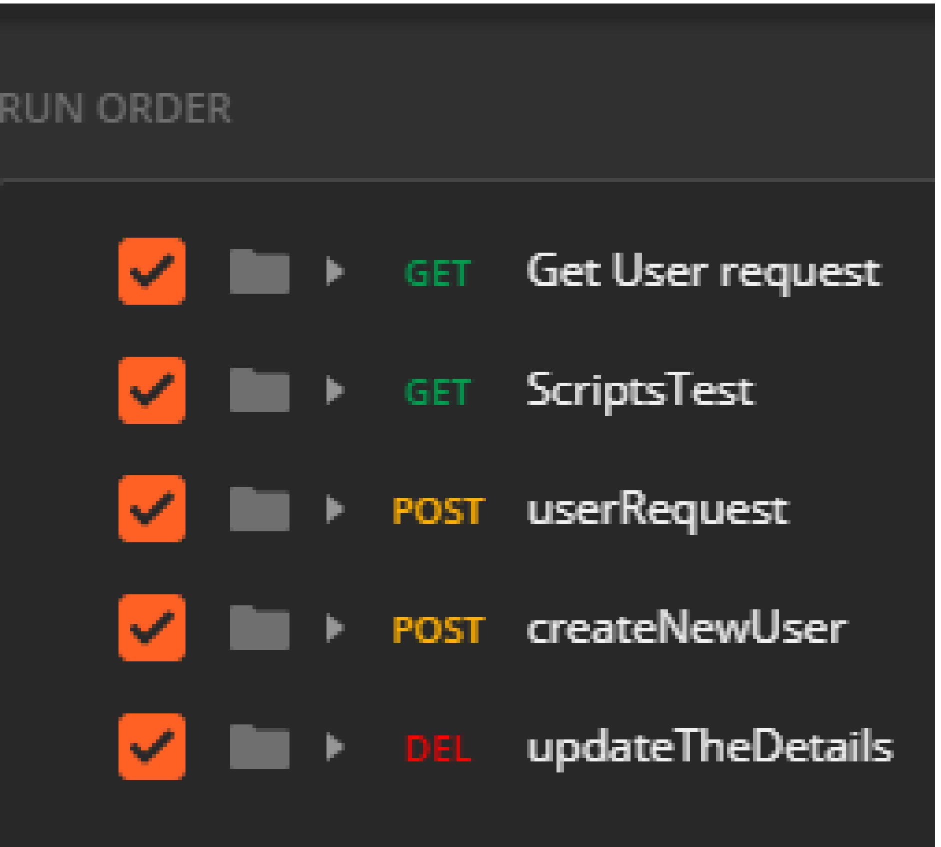


Search for a collection or folder

Mithilessh_Demo

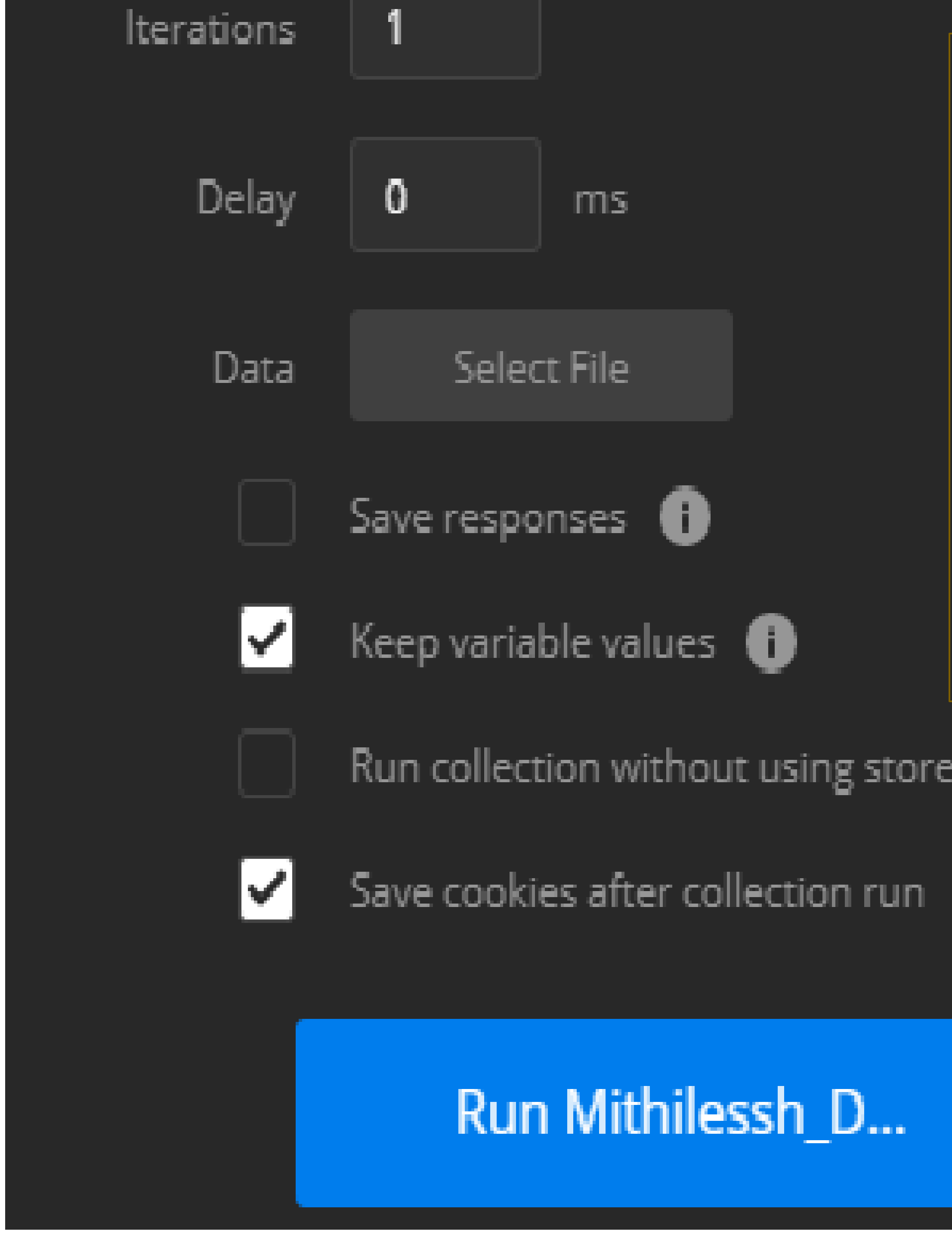
GETRequest

PostRequest



RUN ORDER

- ✓ GET Get User request
- ✓ GET ScriptsTest
- ✓ POST userRequest
- ✓ POST createNewUser
- ✓ DEL updateTheDetails



Iterations	1
Delay	0 ms
Data	Select File

- [ ] Save responses
- [✓] Keep variable values
- [ ] Run collection without using stored cookies
- [✓] Save cookies after collection run

Run Mithilessh_D...

Please remember: since we are taking data variables from the csv file we have to use key in request payload as mentioned below:

```
{
   "name":"{{name}}",
   "salary":"{{salary}}",
   "age":"{{age}}"

}
```

Run the collection in postman through Runner option

Steps:

1. go to the runner option in postman.
2. Select the collection
3. provide the iteration and delay time(ms) and Run.

Note:
1. All the available request in the collection will execute one by one as per order of creation.

2. We can also change the order of execution as per our wish.

Steps:

- 1. select the collection
- 2. select the request--> click on tests option and use the below command.

```
postman.setNextRequest(<name of API request>);
```

This will help to execute the mentioned api before the current one.

Execute postman collection from commands line

- **Pre-requisite:**

1. **Install node.js**

- follow this link to install the node.js https://nodejs.org/en/download/
- Verify the installation using command **node -v**
- NPM comes as a bundle under node.js so npm can be used to install and node module
- syntax **npm install <module name>**

2. **Install newman**

- **npm install -g newman**
- To generate the html report

  npm install newman-reporter-html

3. Export collection and then run from command prompt.

Note: How to export collection??

right click on collection-> Export-> mention the location and save.

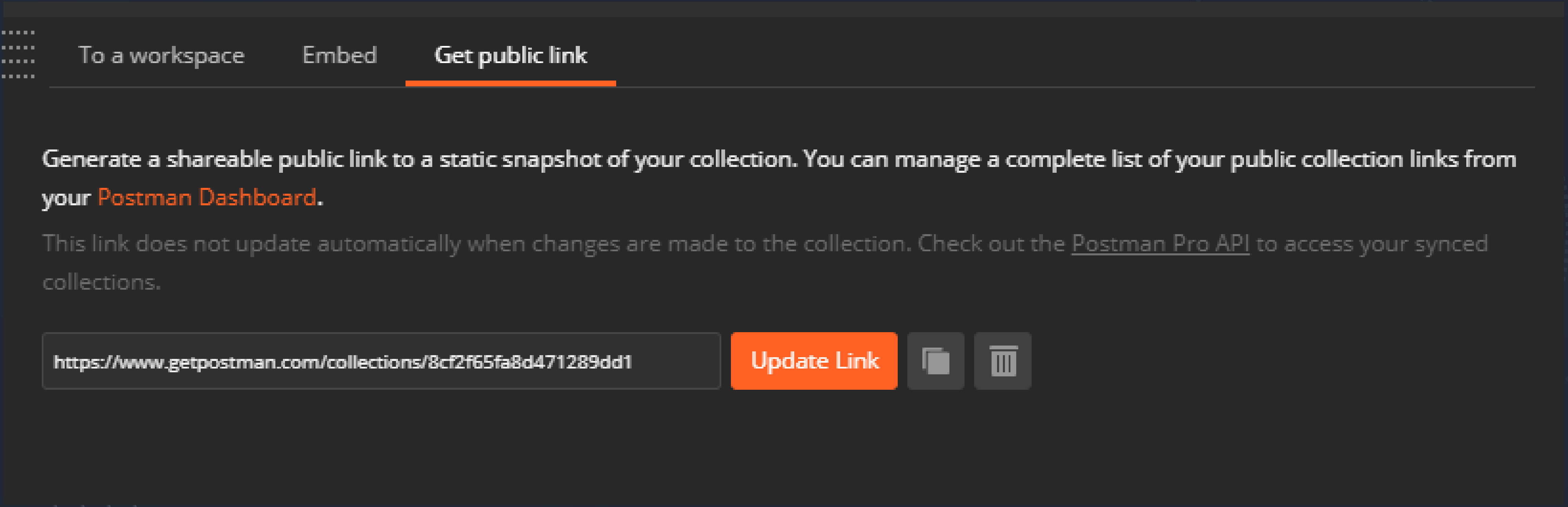
4. Navigate to that location in local system and open the cmd and use this command newman run <exported collection file name>.

***If we want to generate the html report of the execution use this command and check the report under the newman folder in your system.
newman run <exported collection file> -r html.

Execute collection remotely- without exporting it

For that we need to get the url of the collectiom

- 1. Click on three dot's available right side of the collection --> select the collection
- 2. Click on shared the link--> get public link
- use this public link to execute the collection.
- **command**: newman run <Url > -r html [it will generate report also].



Generate a shareable public link to a static snapshot of your collection. You can manage a complete list of your public collection links from your Postman Dashboard.

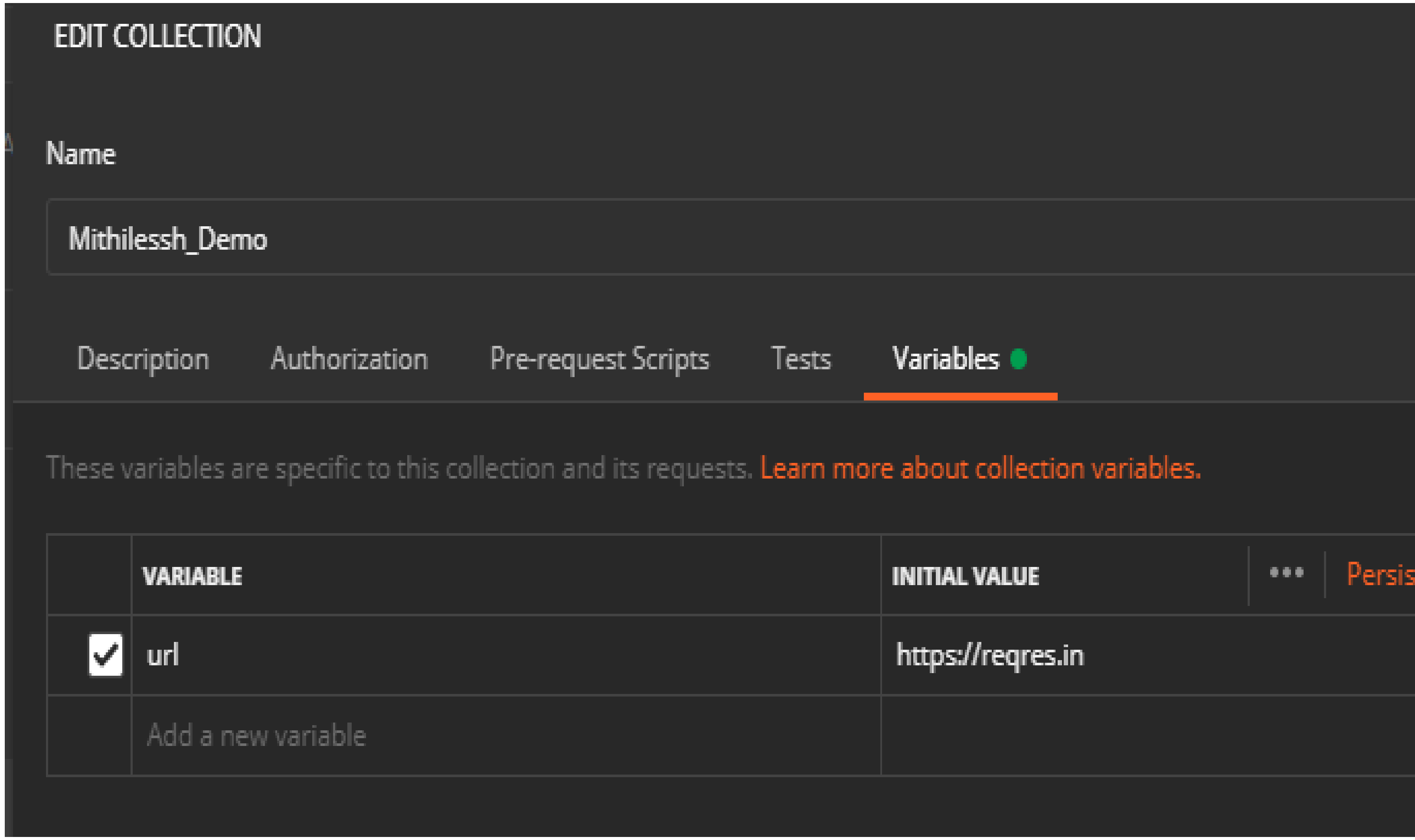
This link does not update automatically when changes are made to the collection. Check out the Postman Pro API to access your synced collections.

https://www.getpostman.com/collections/8cf2f65fa8d471289dd1

Variables concept in postman:

1. Collection variables/Global variables.
2. Environment variables.

- **Collection variables:**
- Way1:
- **Steps:**
- Click on three dot's available right on the collection--> click on edit
- --> click on variable option--> set the variable in key value pair.
- Note: Use same variable name in the request with syntax {{<variable name>}}.
- Way2:
- **Steps:(Global variable)**
- Click on three dot's available right on the collection--> click on edit
- --> Click on pre request scripts-->add the below line of code for environment variable
- **postman.setGlobalVariable("variable name","value")**


EDIT COLLECTION

Name

Mithilessh_Demo

These variables are specific to this collection and its requests. Learn more about collection variables.

	VARIABLE	INITIAL VALUE	••• Persis
✓	url	https://reqres.in	
	Add a new variable		

2. Environment Variable:

we can create a environment variable and use it in any collection.

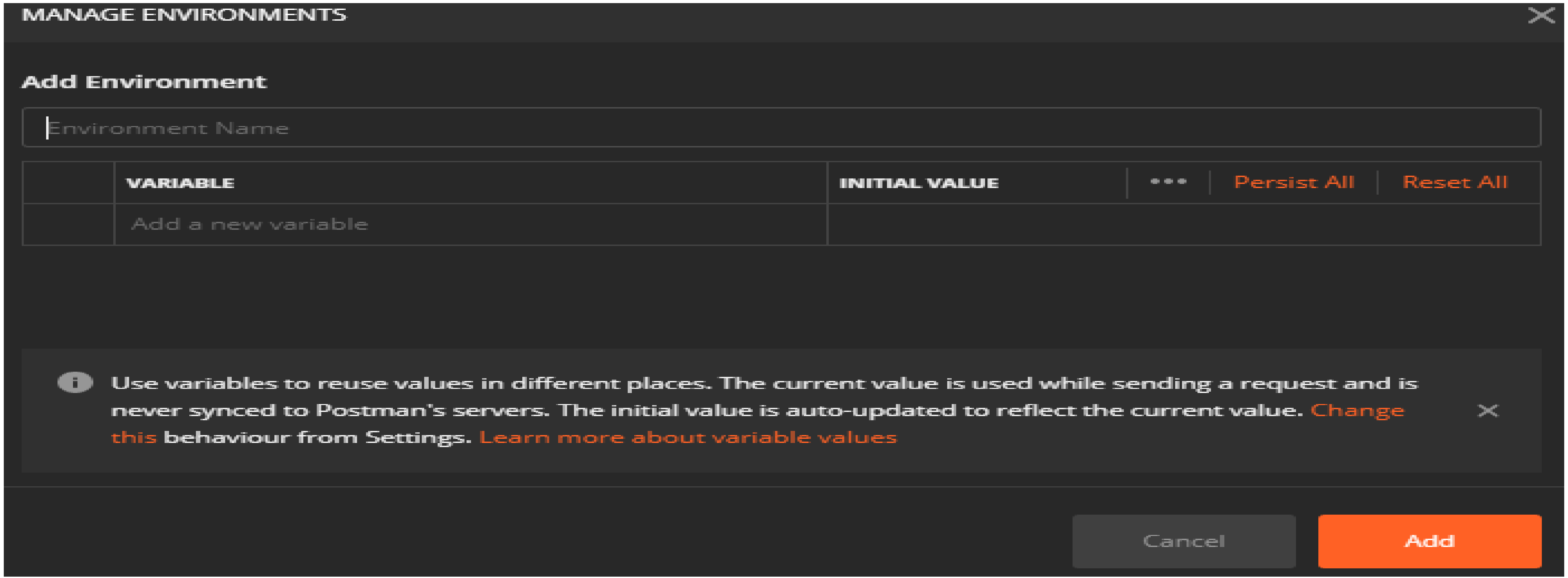
Steps to create Environment Variable:

1. Click on manage Environments icon(top right next to eye icon)-- click on add option.

Provide the Environment name and create a variable under this environment.

later we can use this environment variable under any collection in the postman.

Note: while using the collection using runner we can select environment variable from the dropdown



	VARIABLE	INITIAL VALUE
	Add a new variable	

Use variables to reuse values in different places. The current value is used while sending a request and is never synced to Postman's servers. The initial value is auto-updated to reflect the current value. Change this behaviour from Settings. Learn more about variable values

Postman Tool

Advantages

- 1) In postman it is easy to create test suits. We can create various test suits which contains multiple integration tests
- 2) There are multiple test cases which are dependent on each other i.e some times one test case require test data from other test cases so in that case postman store data for them
- 3) Some time we need to require different environments on which we run test cases so postman store information for those test cases
- 4) We can also integrate postman with different tools like Jenkins
- 5) One of the most important advantage of postman is we can easily move test case from one system to another system also one environment to other
- There are also various disadvantages of Postman testing tool:

Disadvantages

- 1) Postman has a disadvantage when it comes to monitoring of test cases. There are some other tools like Assertible which makes this task easy.

Testers Zone

Mithilesh Singh