

REST API & test using POSTMAN

REST API

POSTMAN

get

request

communitycreator

REST is an acronym for **Representational State Transfer**. It is an architectural style for distributed hypermedia systems.

An API is an **application programming interface**. It is a set of rules that allow programs to talk to each other. The developer creates the API on the server and allows the client to talk to it.

REST determines how the API will look. It is a set of rules that developers follow when they create their API. One of these rules states that you should be able to get a piece of data (i.e., a resource) when you hit a specific URL.

The key abstraction of information in REST is a **resource**. Any information that can be named can be a resource:

- a document
- an image
- a temporal service
- a collection of other resources
- a non-virtual object (e.g., a person)

REST uses a resource identifier to identify the particular resource involved in an interaction between components.

Another important thing associated with REST is resource methods, which perform the desired transition. REST APIs enable us to develop any kind of web application that has all possible **CRUD (create, retrieve, update, delete)** operations.

HTTP methods

The suitable HTTP method for the action performed by API are:

- GET
- PUT
- POST
- DELETE

GET:

GET requests only retrieve resource representation/information – they do not modify it in any way. Since GET requests do not change the state of the resource, these are said to be safe methods.

PUT:

PUT APIs are primarily used to update existing resources (if the resource does not exist, then the API may or may not decide to create a new resource). If a new resource has been created by the PUT API, the origin server MUST inform the user agent via the HTTP response code

POST:

POST APIs create new subordinate resources (e.g., a file is subordinate to a directory containing it and a row is subordinate to a database table). Strictly talking in terms of REST, POST methods are used to create a new resource in the collection of resources.

DELETE:

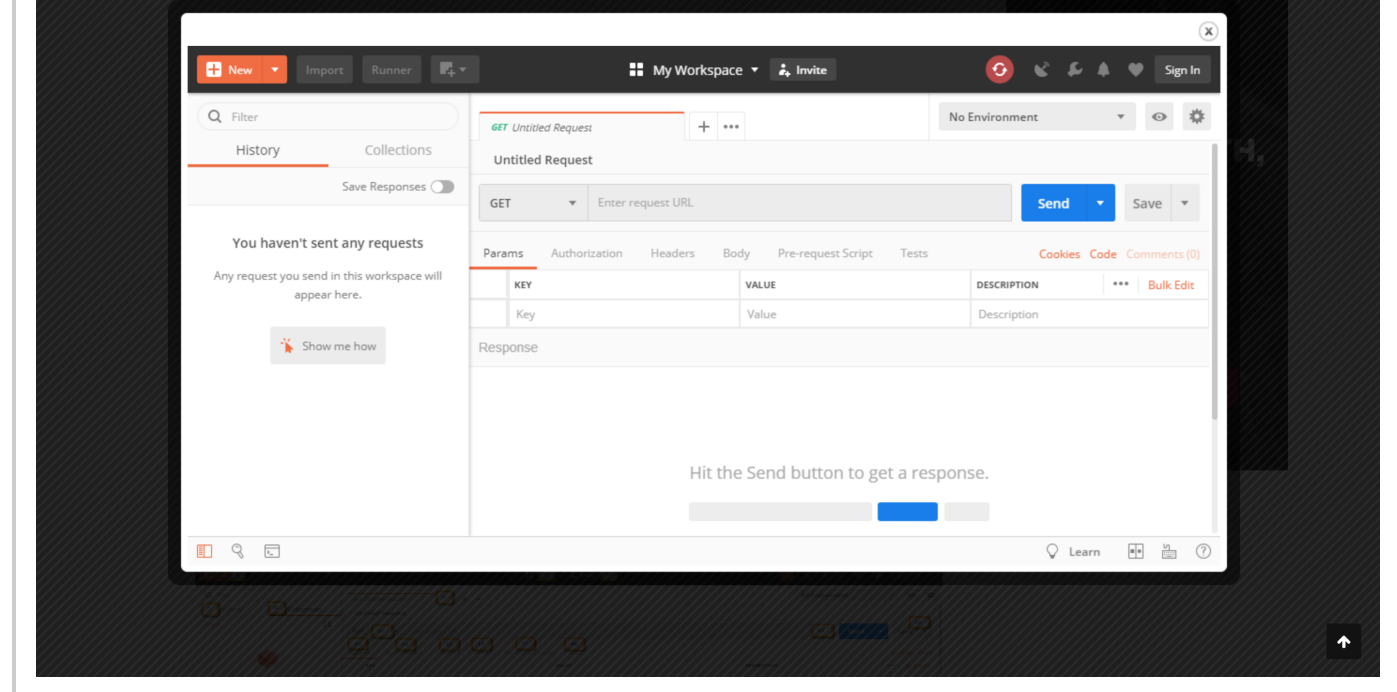
DELETE APIs are used to delete resources (identified by the Request-URI).

POSTMAN

Postman is a scalable API testing tool that quickly integrates into the CI/CD pipeline. It started in 2012 as Abhinav Asthana's side project to simplify API workflow in testing and development.

Before working with the testing of API using POSTMAN, install the application from this [link](#).

After installation, the POSTMAN application interface can be used for reading, writing, and deleting the contents in API using client HTTP.

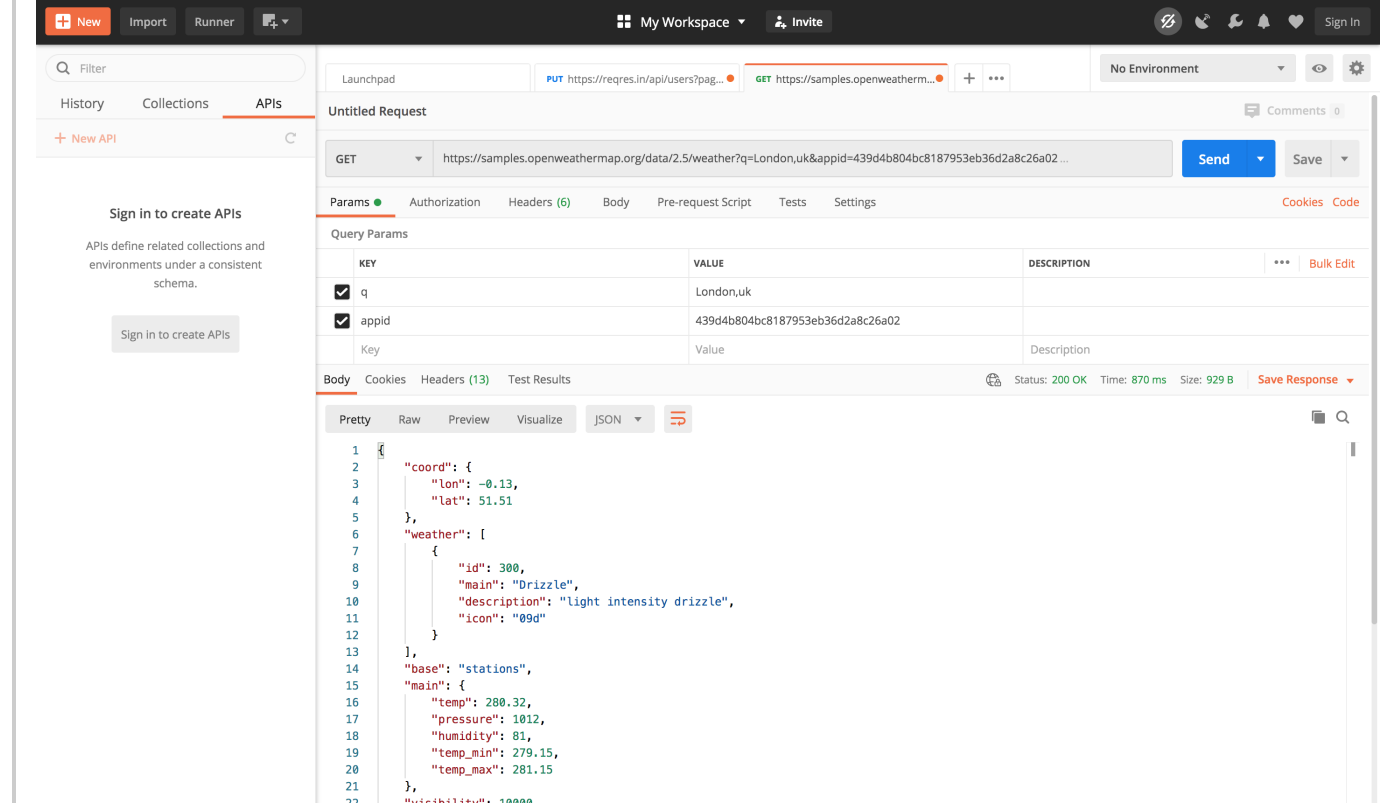


POSTMAN Interface

Let's do an example to illustrate the working of a **GET** request in POSTMAN. Consider a weather API data and access the data from the API source.

The API source link is : [OpenWeatherMap](#).

The data returned from the OpenWeatherMap is in JSON format.



GET Request in POSTMAN for an arbitrary weather API

Let us consider another example where we try to PUT data in some API.

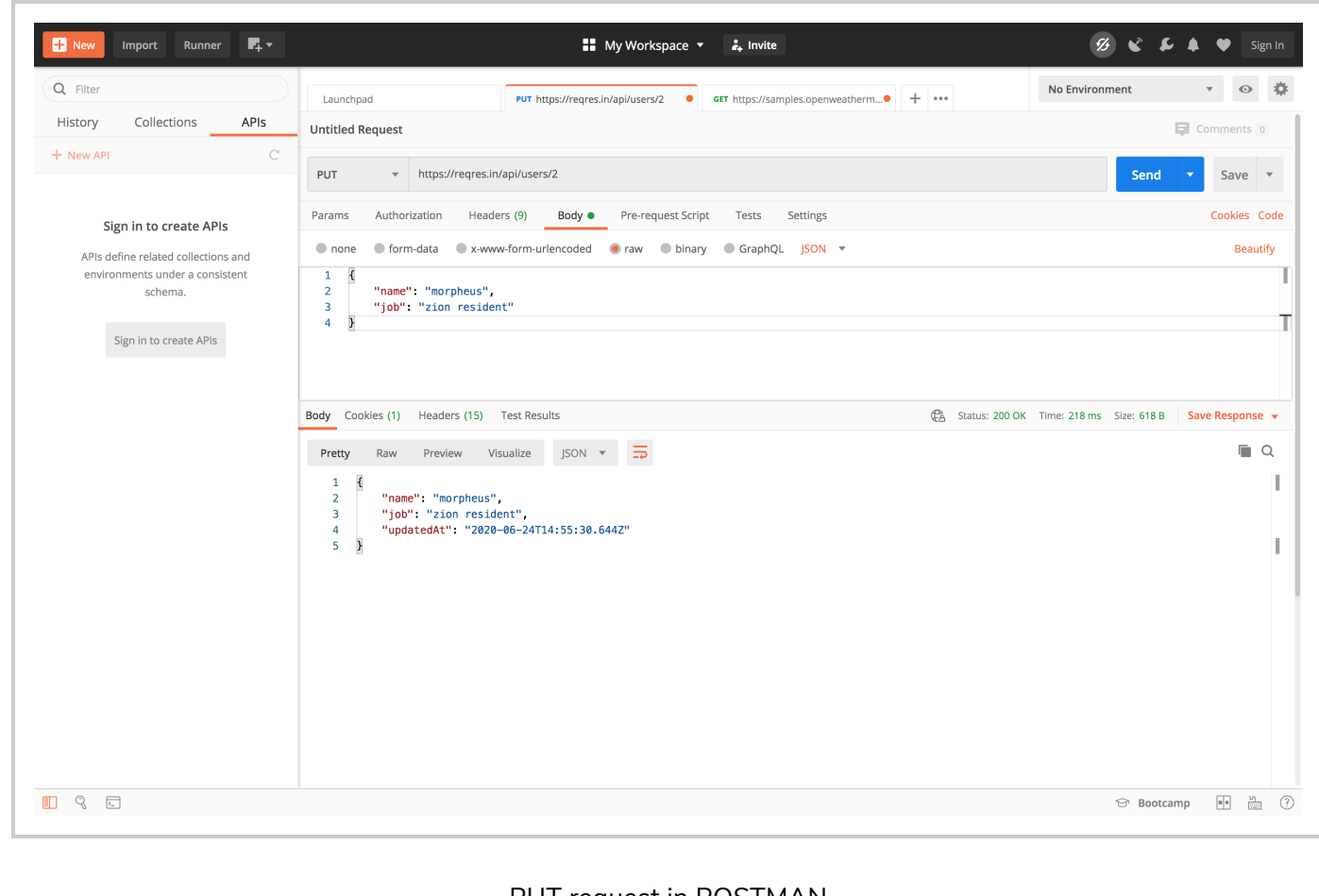
PUT requests are different from **Get** requests as there is data manipulation with the user adding data to the endpoint.

Let us consider an API [link](#) where we can PUT the data into the API source.

The JSON data to be PUT into the source API is:

```
{
  "name": "morpheus",
  "job": "zion resident"
}
```

After the execution of the command, the JSON data is successfully inserted into the source API.



PUT request in POSTMAN

We can similarly delete the data from the source API using the **DELETE** request in POSTMAN.

Contributor: Aravind Kumar

License: Creative Commons -Attribution -

ShareAlike 4.0 (CC-BY-SA 4.0)

