**What is web services?**

A service which is available over internet is called web service.

Consume user is Human than its not a webservice [Youtube, face book and all, consume user is Human, so Youtube, face book is not a web service eventhough its available over internet]

Consume user is machine or software than its is a webservice[API is web service, here 2 diff application interact through any software or application program code ]

**What is API?**

It is application programming interface, this acts as intermediary between 2 different applications

Eg ->Train booking in redbus...in real life, lot of places we are using API nowadays..

In Indian railway have separate irctc site to book the ticket & also thru redbus we can book train ticket..

railway given permission to redbus & provide their API details to redbus, ,thru API only redbus can access(red/write) their DB, so that ticket can be booked

Thru both site, user can book ticket & DB should update in sycn..**thru API only its achieved**

IRCTC Booking App

API

**Important Terminologies**

**Client** (anyone who makes the request – a human, a piece of software code). software code can also make a request.eg weather report we are receive daily on mobile, on behalf of us software code can also make a request & provide the details

**Server** ( a program which receives the request from the client, process it & retunes the response)

**URL**-> Uniform resource locator(eg www.facebock.com)

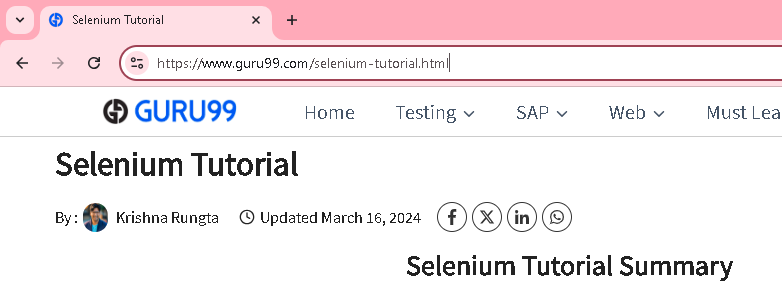
**HTTP request, HTTP Methods, Headers, content types**

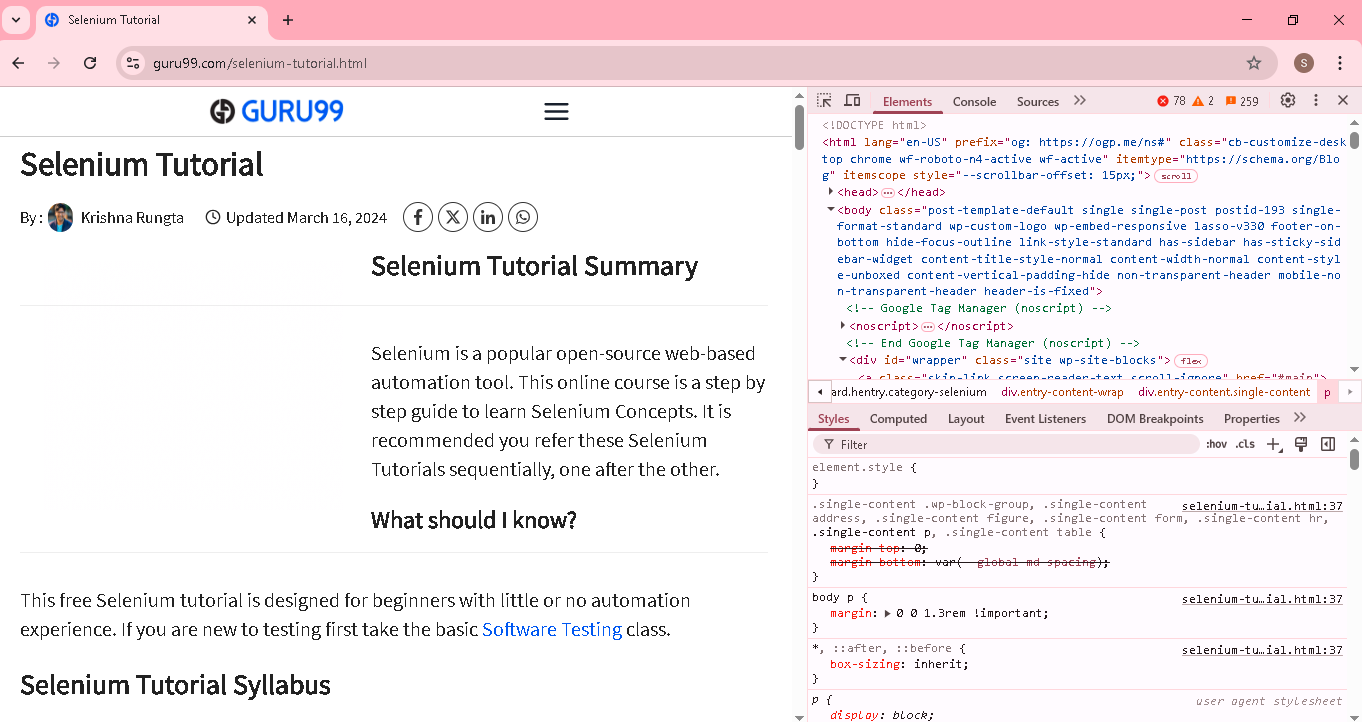
**HTTP**->Hypertext transfer protocols..Client will make the Request in HTTP in web browser,

if web browser, then request should be HTTP only

In web browser, am giving request(HTTP) & getting response in HTML page(Hyper text markup language).

HTML resources achived thru HTTP.





**HTTP Methods.->**

HTTP request done/achived thru HTTP methods.. **HTTP Methods** -> Get, post,put, delete etc

**Headers**:

Server first check the headers part in HTTP Request...where they request originated, what they asked, what response they needed all in Headers

**Content Type:**

Normally In web browser, am giving request(HTTP) & getting response in HTML page

In API, we are interacting with software/Program code, in that We need to receive response/ content type in another format(text, **JSON,XML**) , its achieved thru content type

**Response, status code**

Thru status code, API are interacting.[eg 404 – Error status Code]

**JSON/XML – as above**

**HTTP Methods**

In web page, creating new form/account, modify the details, deleting the account all achived thru http methods

HTTP request done/achived thru HTTP methods.. **HTTP Methods** -> Get, post, put,delete etc

**GET**

To retrieve resource representation/information only..and not to modify it in any way

Its do not change the state of the resources, these are said to be safe methods

GET APIs should be **idempotent[status/info will not change in server]**

Get request will return the resources if found on the server, with status code and the response body

If the resources is not found, then the status code will be returned

Eg of GET URl

<Http://www.google.com> [just we are retrieve resource & not modify the resources /information]

**Post** [eg creating new signin account in facebook with all information or in facebook/insta we are posting something]

Create a new resource [ Eg in facebook/insta we are posting something..that is creating new resources]

Post request are not safe methods

Post request are not idempotent (i.e) invoking 2 identical post request will result in 2 different resources containing the same information(except resources ids)

It will return a status code, if the resources is successfully created

Eg of post URL:( Post request wil have POST BODY.we will see that in details)

**Put**

To update/modify the existing resources [ eg if any error, then edit the facebook post]

If the resource does not exist (while updating the existing resources ] then API may decode to create a new resources .

**Delete**

Used to delete resources

If the resources are present, the resources will be deleted & the relevant status code will be sent back.

Delete method is **idempotent in official documentation**

**HTTP Response**

When a request is submitted by the client, the server has to send back the response along with the status code

Response can be anytime like XML, JSON, plain text etc

JSON is the widely used representation of the resources because of its lightweight & verbose(readable) nature

**Status code:**

1xx -> 100 series

2xx-> 200 series

3xx-> 300 series

4xx-> 400 series

5xx-> 500 series

**Most used & very important status code**

Under each status series(2xx) we have multiple status code(200,201 etc)..but we are reading only important code

**2xx ->indicates the success reason** ..under 200 series we have 201, 202 etc

200-Ok(for Get request), 201 ->Created(for post request)

**3xx ->indicates redirection**...page redirection

**4xx->Error on the client side**(user can handle this) eg..user giving incorrect url..www.gogle.com

400 bad request, 401 unauthorized, 403 Forbidden, 404 Page not found

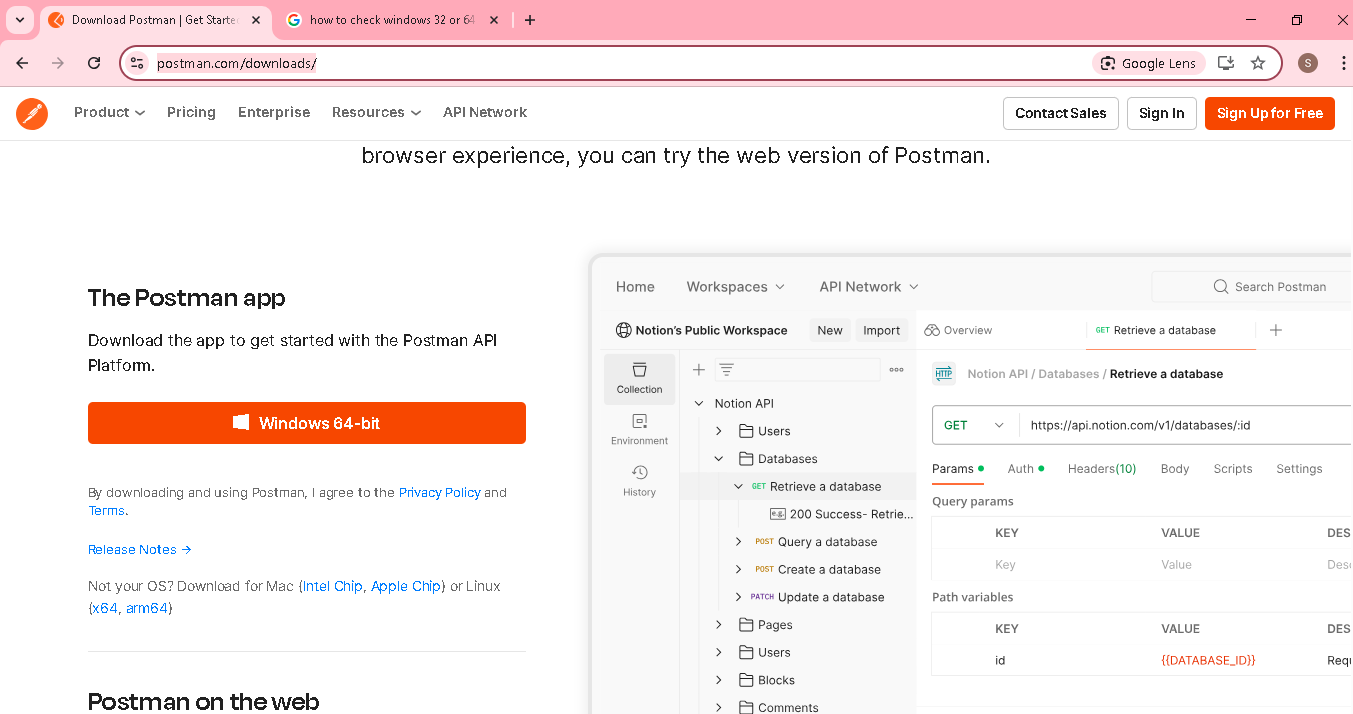
**5xx-> Error on the server side**(user can’t handle this, only the server maintenance people can handle)

500 Internal server error

Lets get into the actual Testing

**Tools needed:** Install postman for your OS

<https://www.postman.com/downloads/>

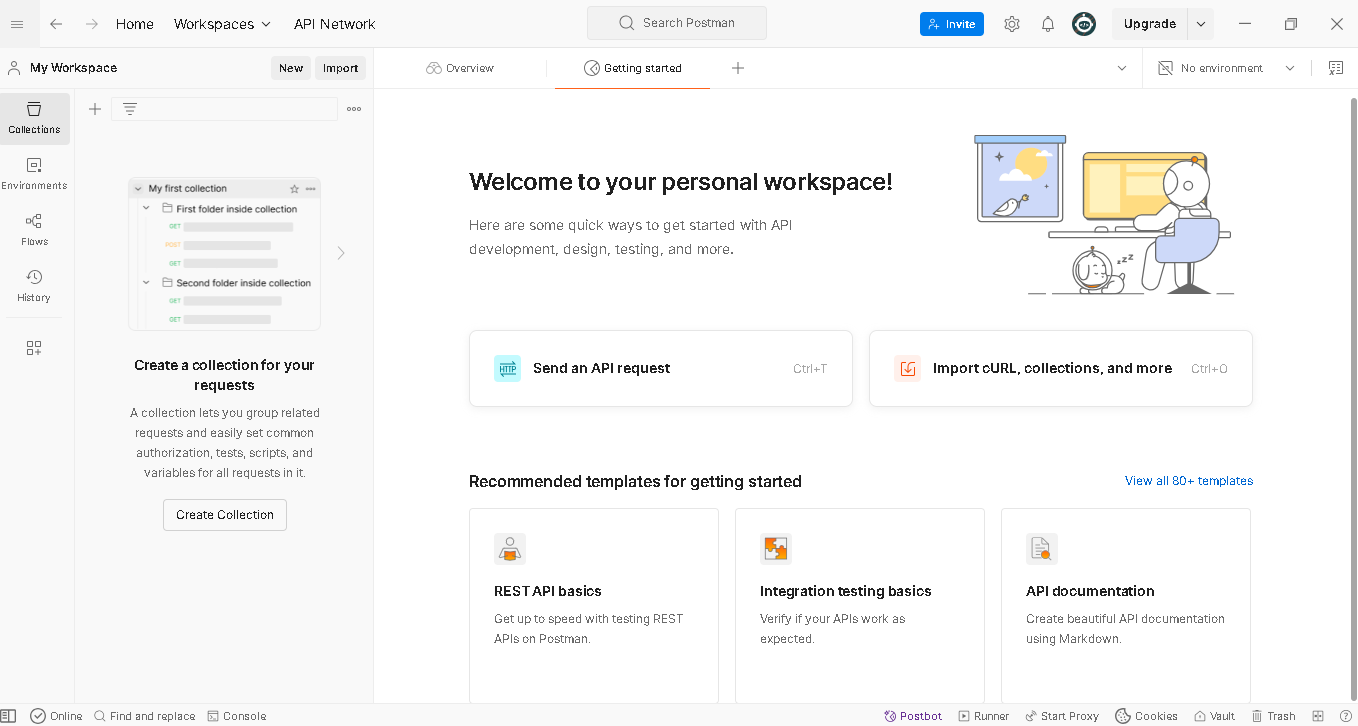


Create a login account & sign up

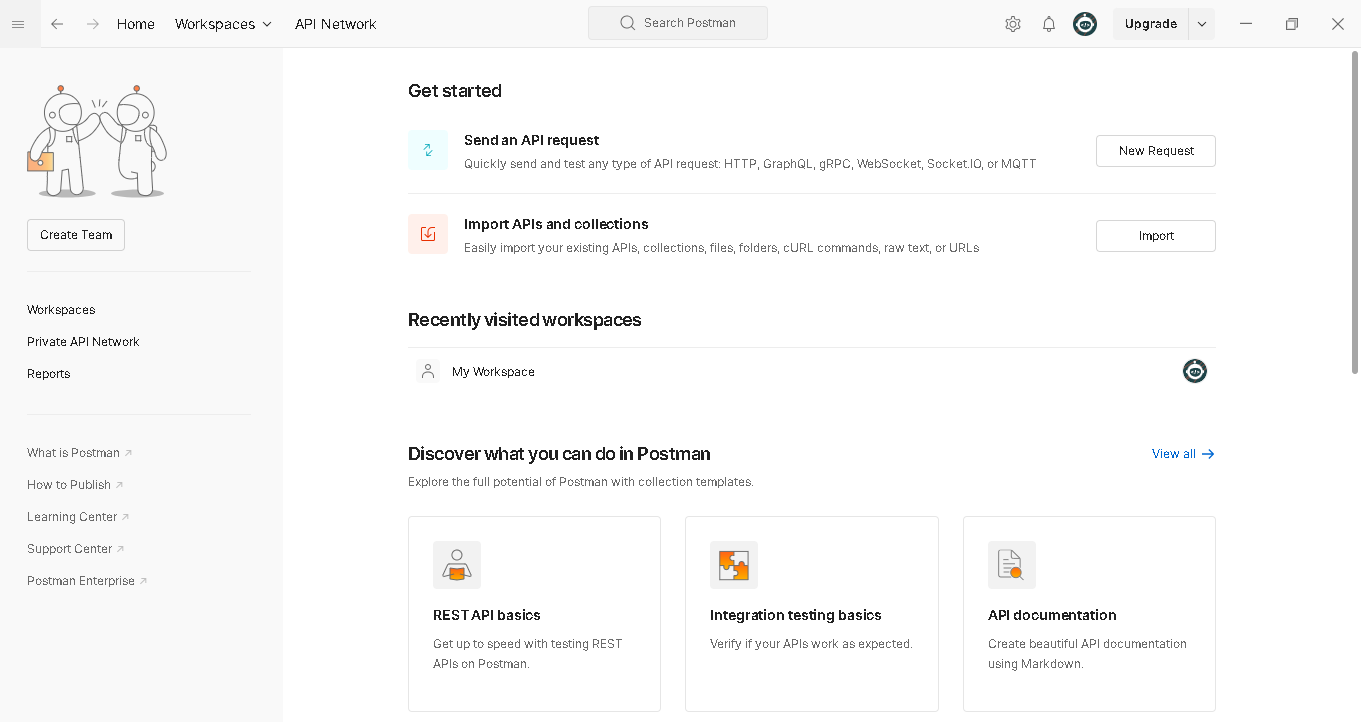
**Email:** [**Sukumarvijayan1990@gmail.com**](mailto:Sukumarvijayan1990@gmail.com)

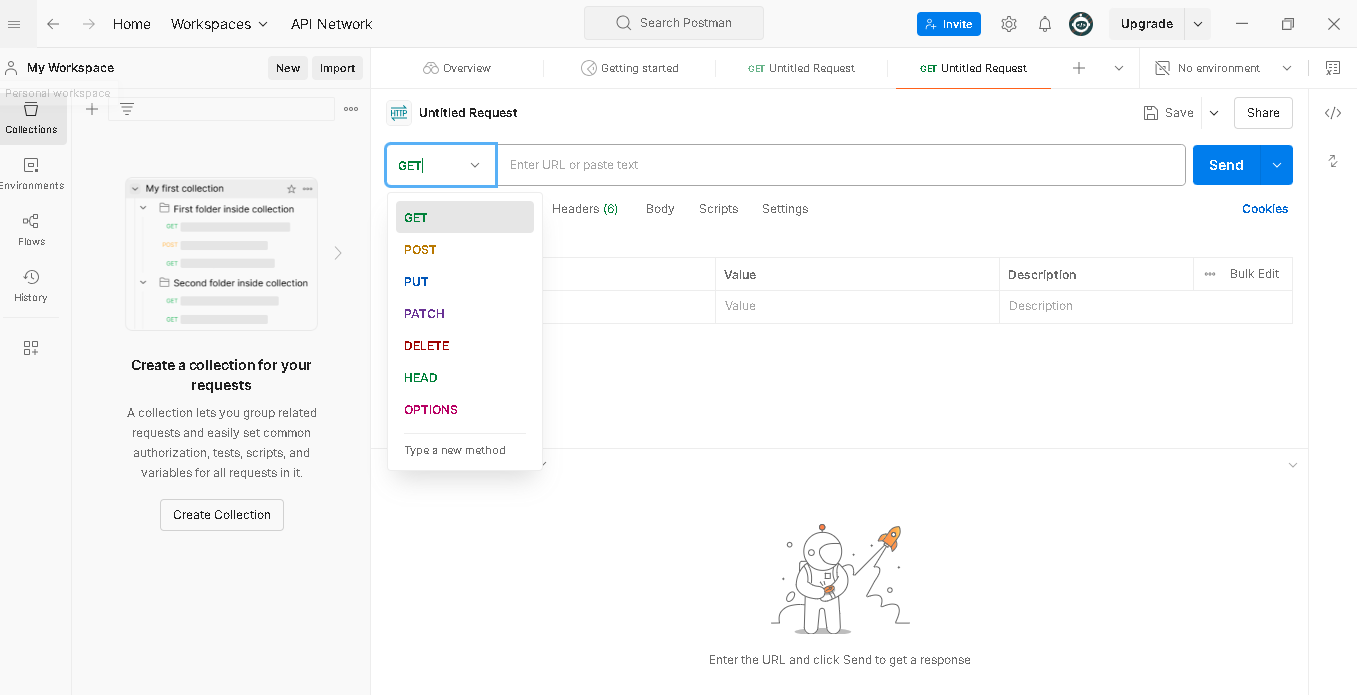
**PW: Aarinew1$**

**Username: Sukumarvijayan**



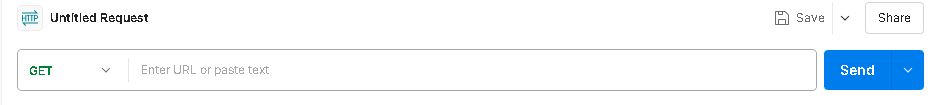
Navigate to Home->New Request



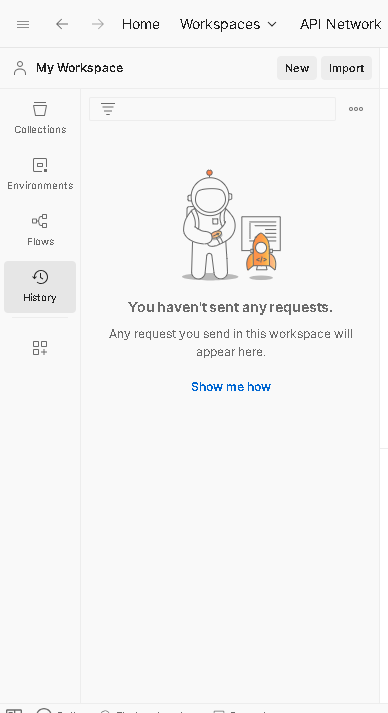


**Important Section in postman:**

**Request Bar-> have (GET,post,put, delete) & url...eg Perform GET request in XX url**



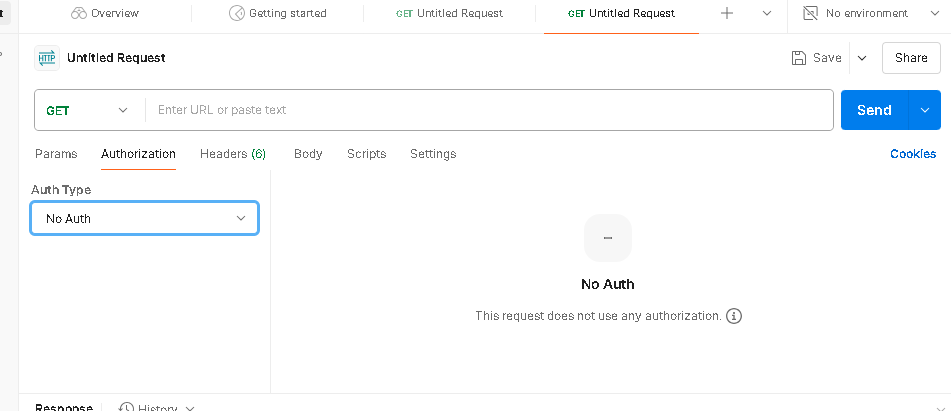
**History** ->it will show the history which url tested previously

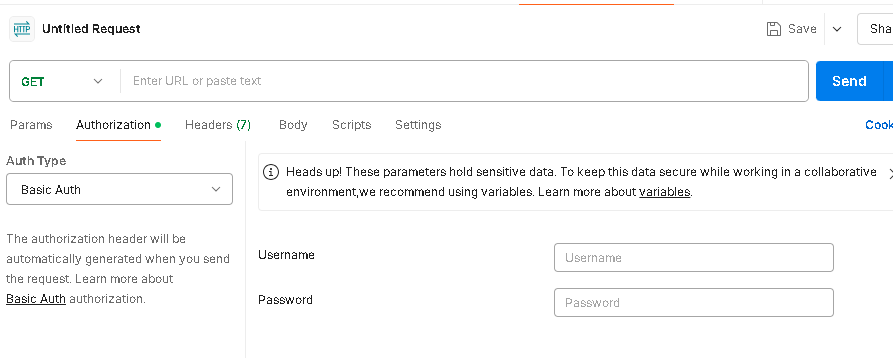


**Authorization**

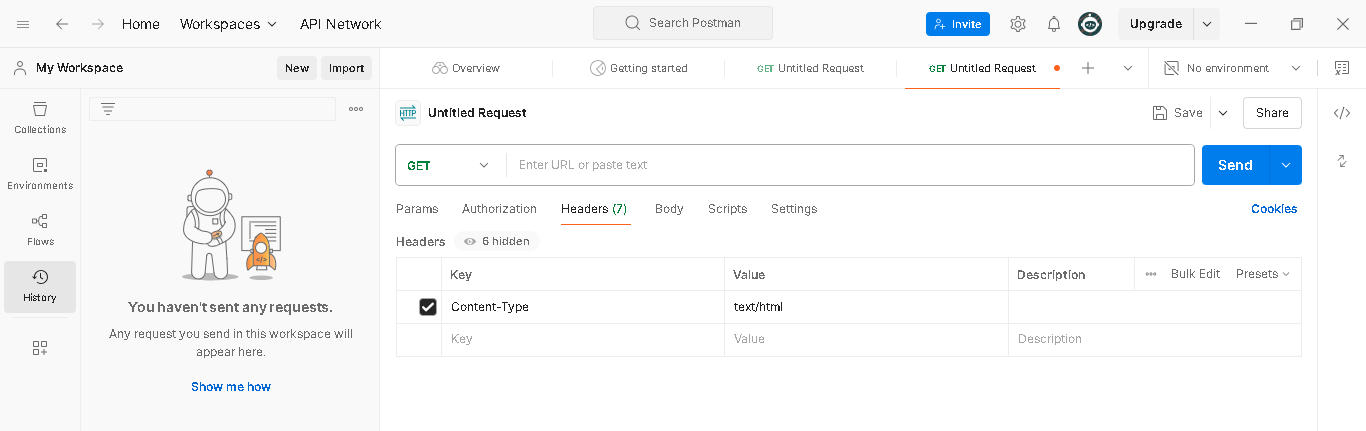
For accessing google url, we don’t need any authorization, but while accessing gmail login url..need to provide username & password

Kindly check AuthType dropdown, it will show multiple auth type

****

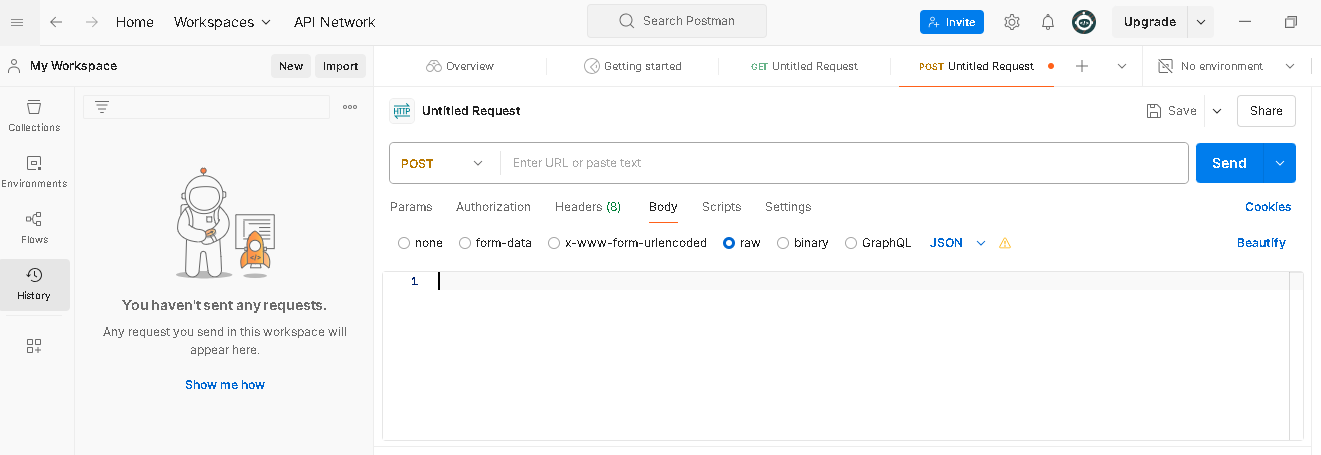
****

**Headers -> we will give content type(XML, JSCON, Text, HTML)**

****

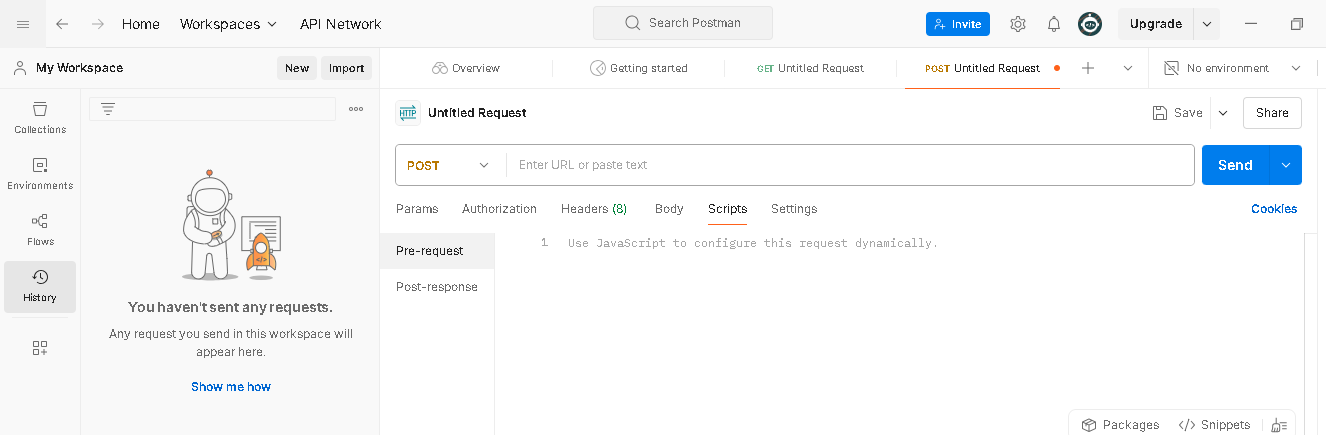
**Body**

For post request have body details..we have mention what need to create in body section during post request



**Scripts**:

Here we maintain pre-request details before URL executed & Post Response details after URL Tested



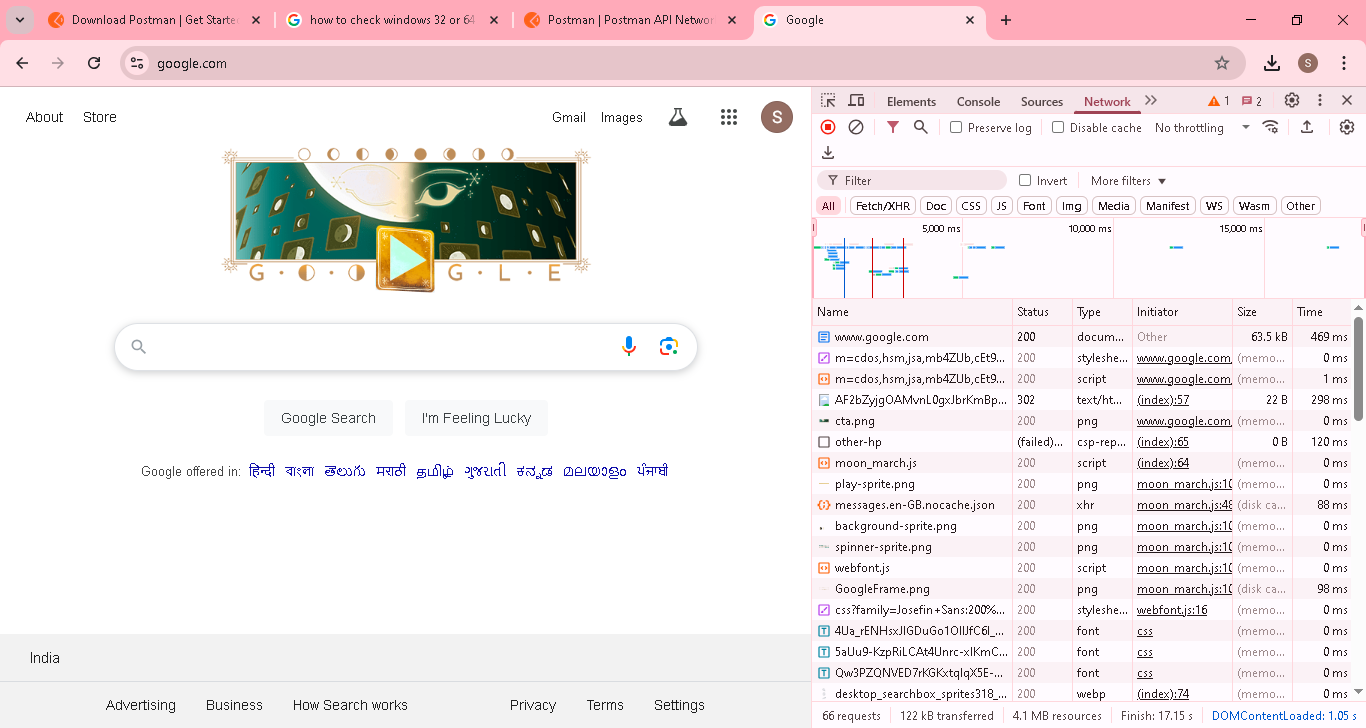
**Tests**

**Response console** ( Headers, body, cookies, Test Result)

**Let’s write our first API test with postman**

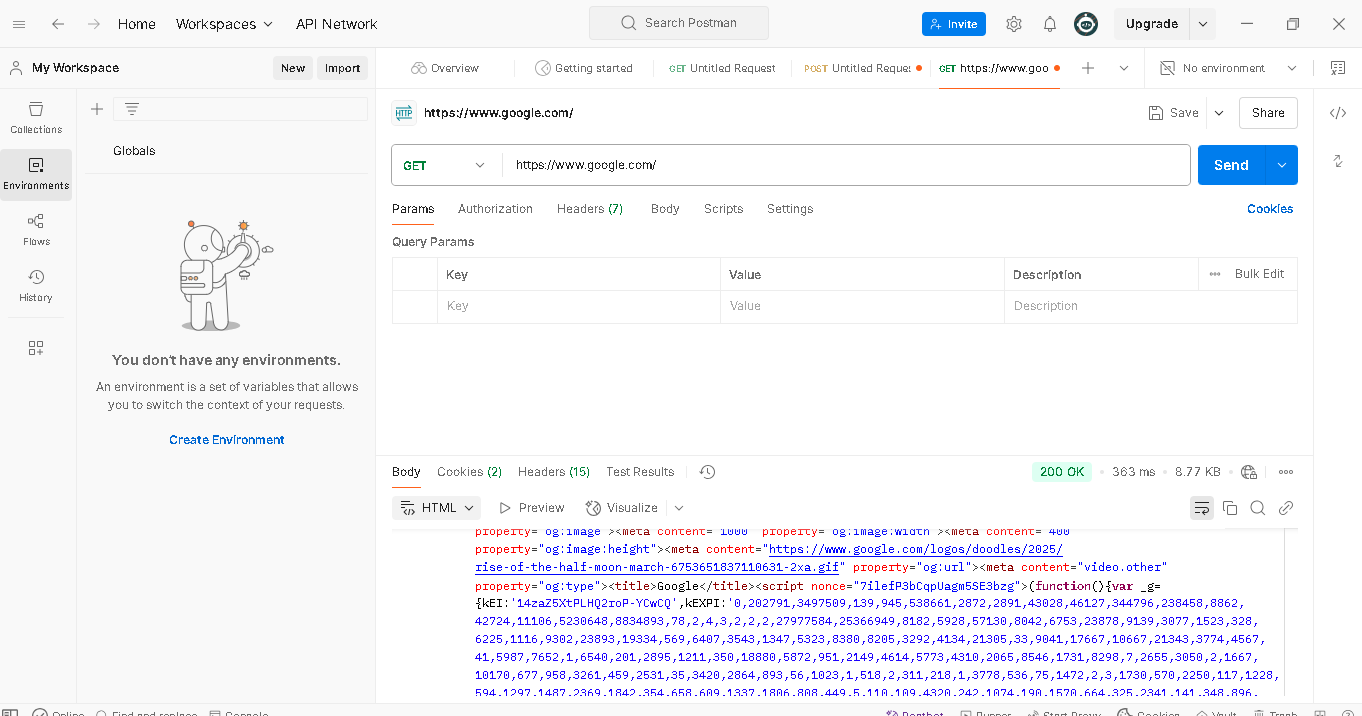
Go to google.co.in & check the page is loaded or not & check the status code(manually)

Open html(F12) & go to network..then refresh the url..u can see the status code as below



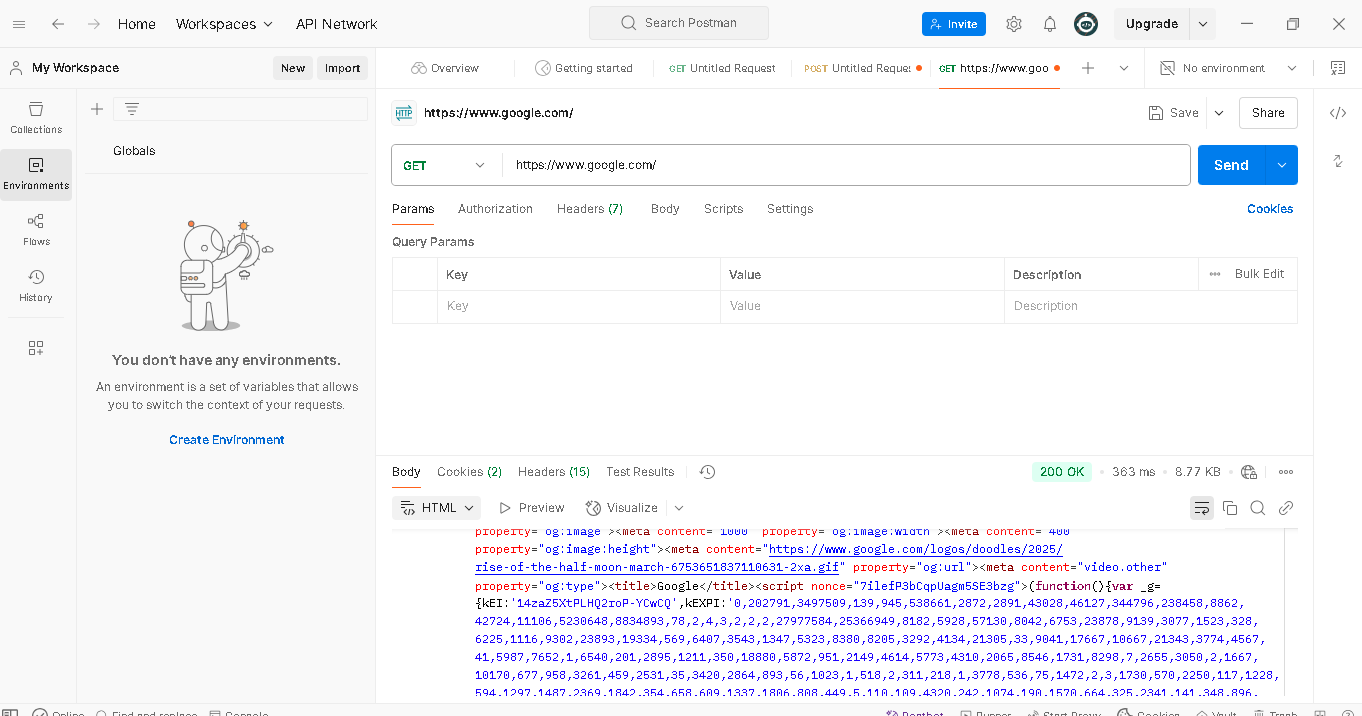
Lets do with postman

Give the url & click Send

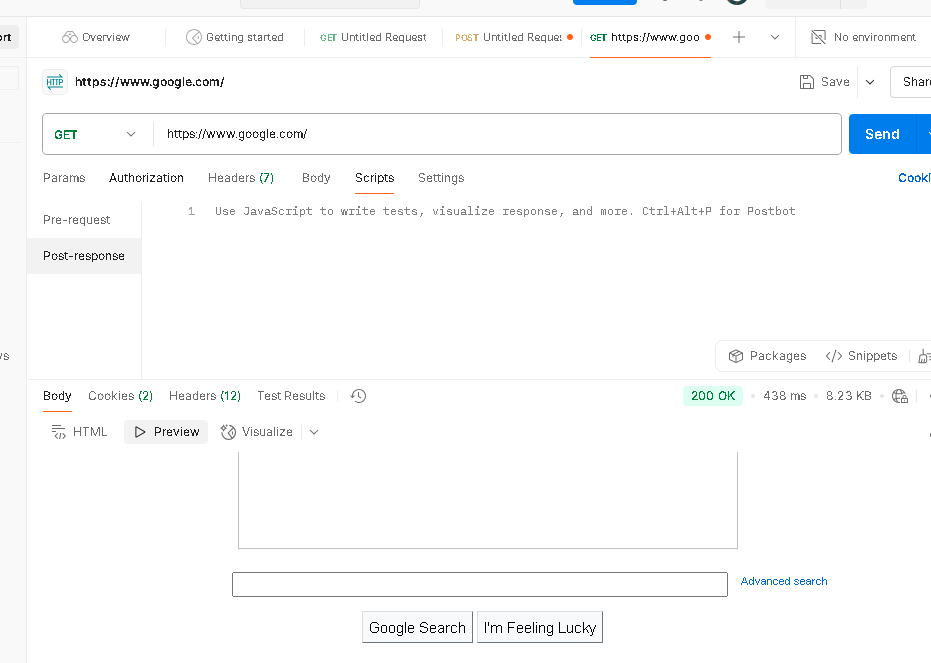


**Response console** contains Headers, body, cookies, Test Result/

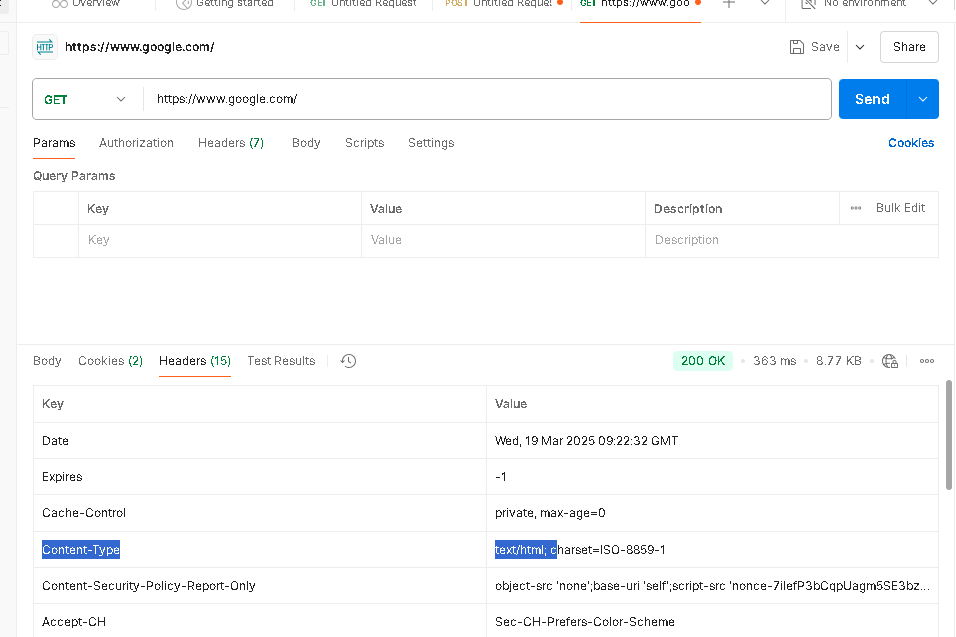
Body contains html..Note: Normally google page will come as html web page only after load..its design on that way



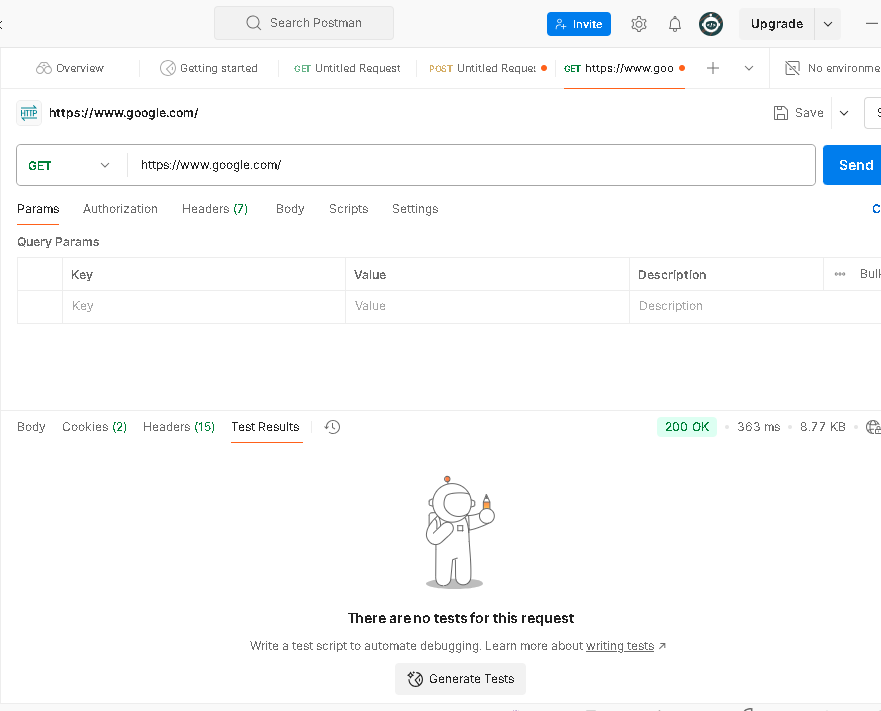
Under preview..its show page



Headers contain content type & other details

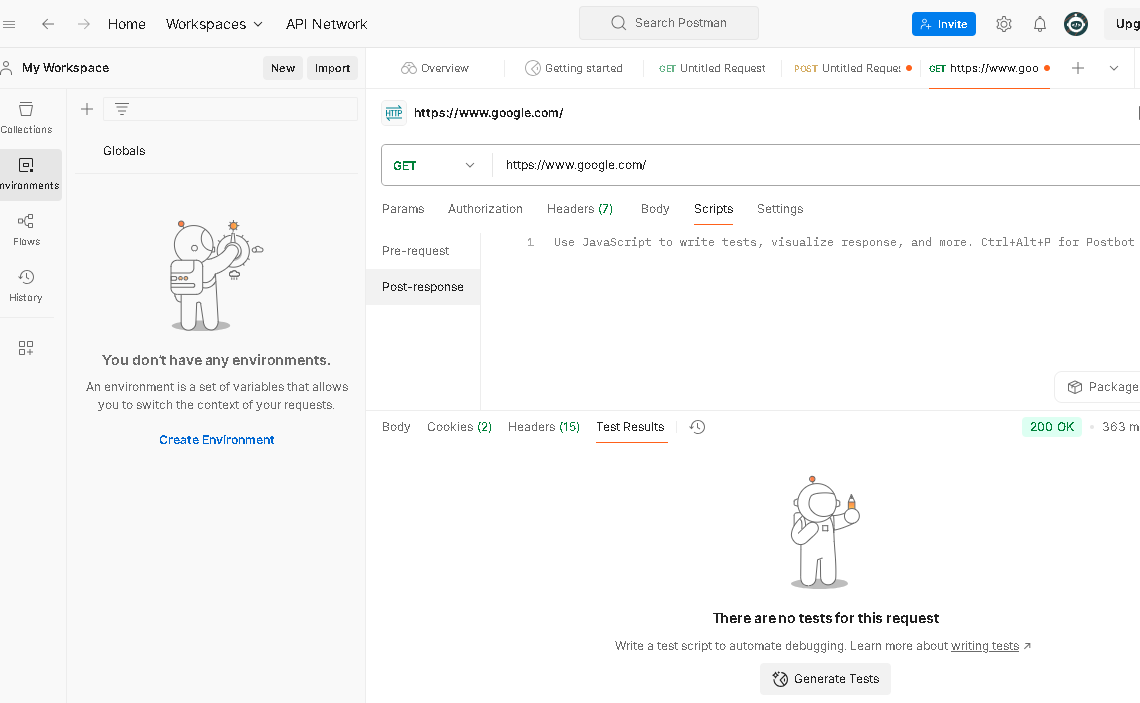


**Test Results:**



**Write a test case in postman for verifying the status code**

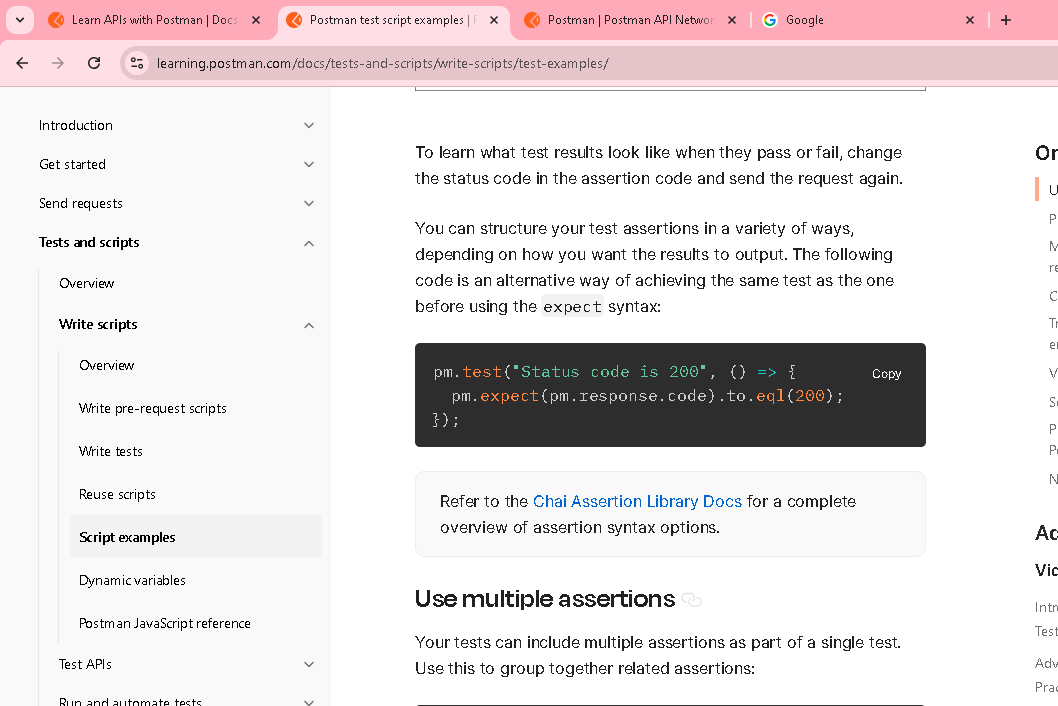
Scripts->post response



In postman, need to write JavaScript in order to test [

<https://learning.postman.com/docs/tests-and-scripts/write-scripts/test-examples/>

in postman site , itself they have provided test case how to write



Copy the code & paste in postman & execute

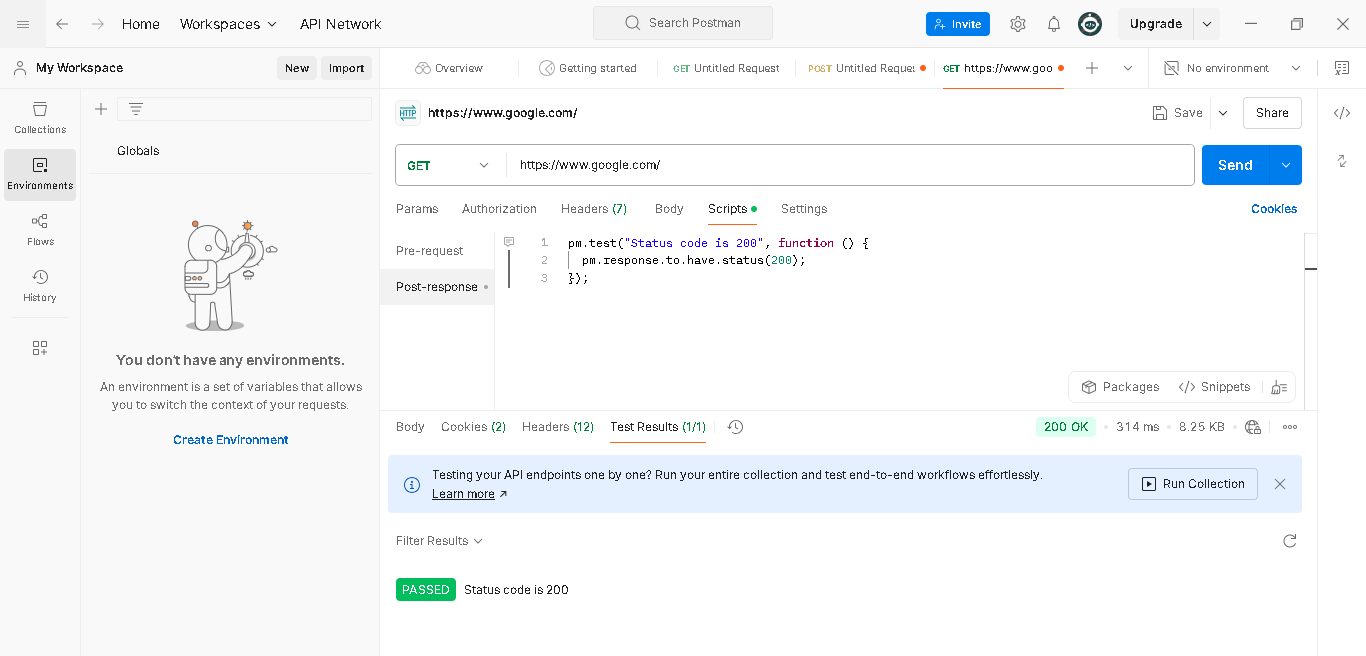
pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

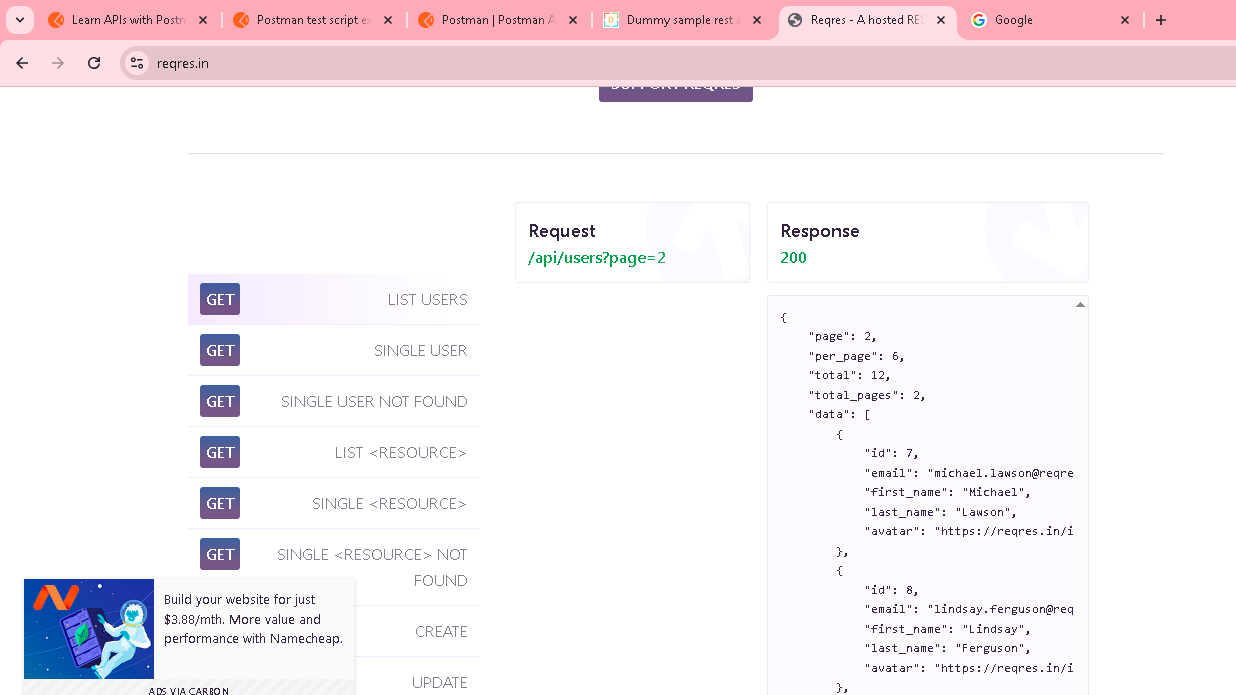
});

Pm->postman object

Status code is 200->test case name



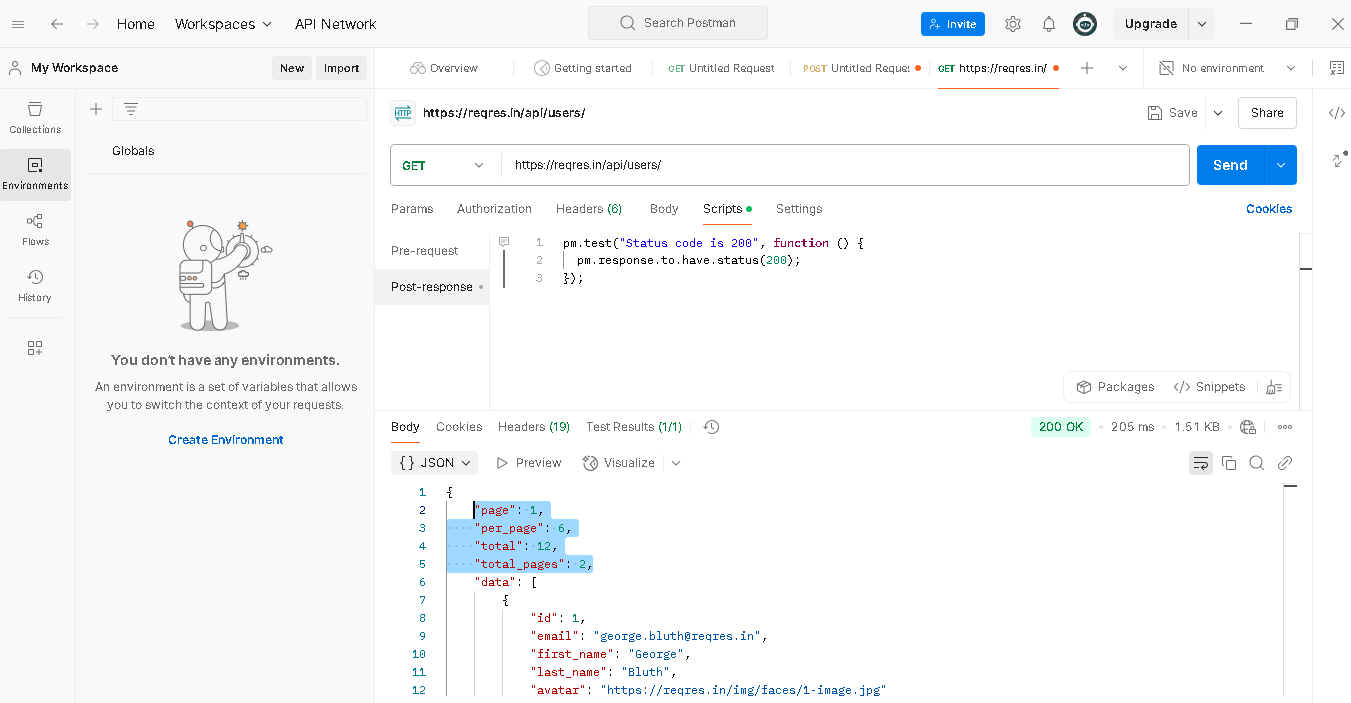
<https://reqres.in/>



<https://reqres.in/api/users/>

to test above url with test case for the field[page, per\_page, total]

**JSON file**



pm.test("Check Page Count", **function** () {

    //parse the response JSON and test three properties

**var** jsonData = pm.response.json(); ->it will move all json value to jsonData

    pm.expect(jsonData.page).to.eql(1); -> jsonData.page to get the page number & check equal to 1

});

pm.test("Check per\_page Count", **function** () {

    //parse the response JSON and test three properties

**var** jsonData = pm.response.json();

    pm.expect(jsonData.per\_page).to.eql(6);

});

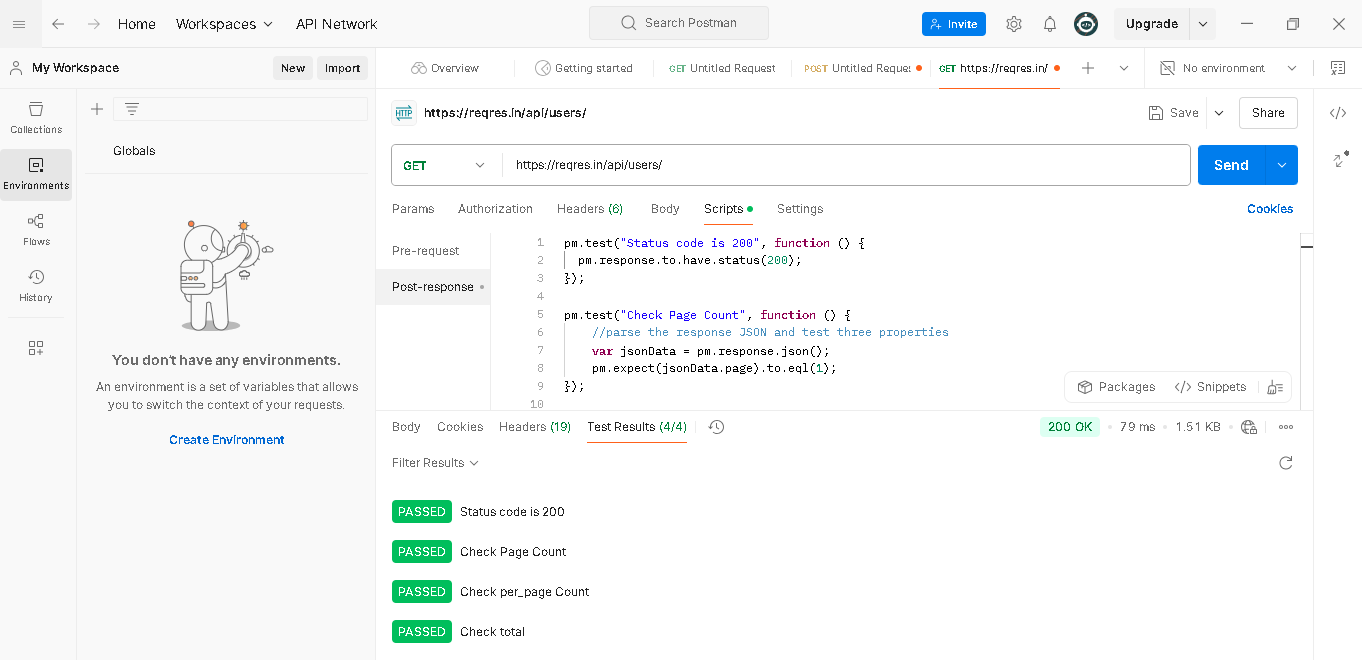
pm.test("Check total", **function** () {

    //parse the response JSON and test three properties

**var** jsonData = pm.response.json();

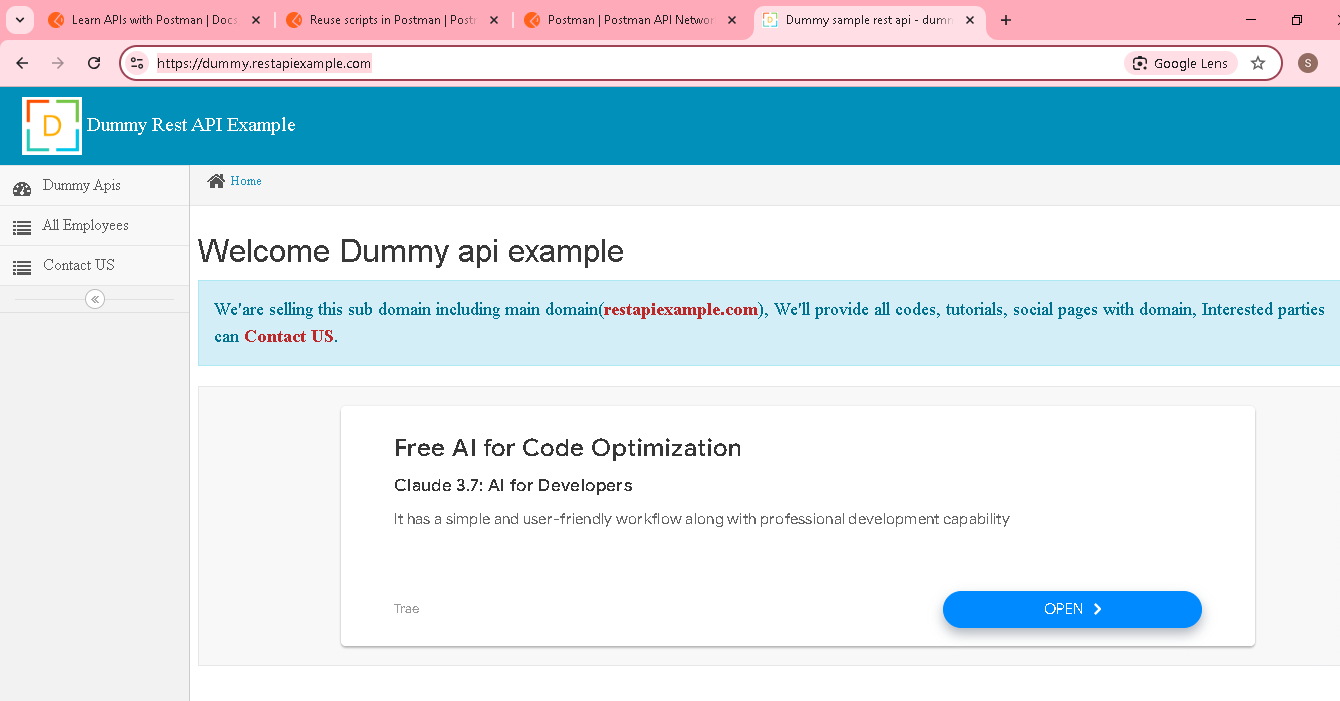
    pm.expect(jsonData.total).to.eql(12);

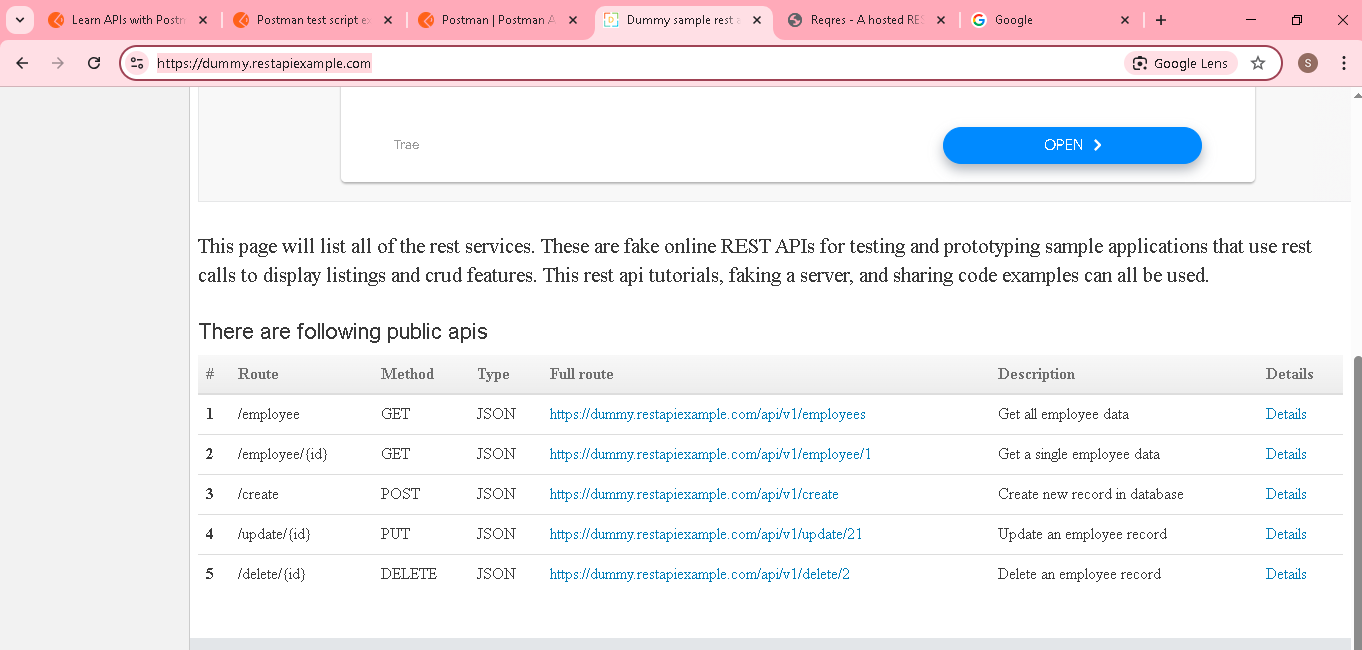
});



**Let’s write more test case for the API with get,post,put & delete request**

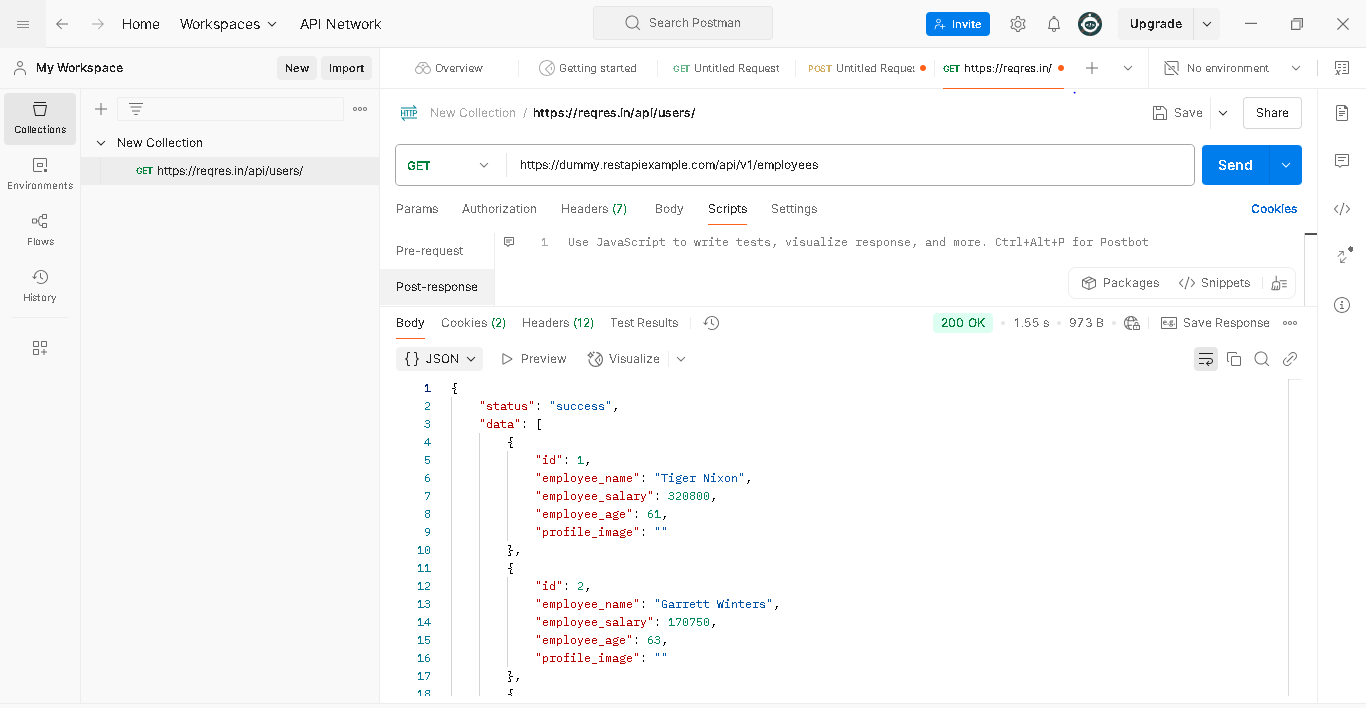
[**https://dummy.restapiexample.com/**](https://dummy.restapiexample.com/)

****

****

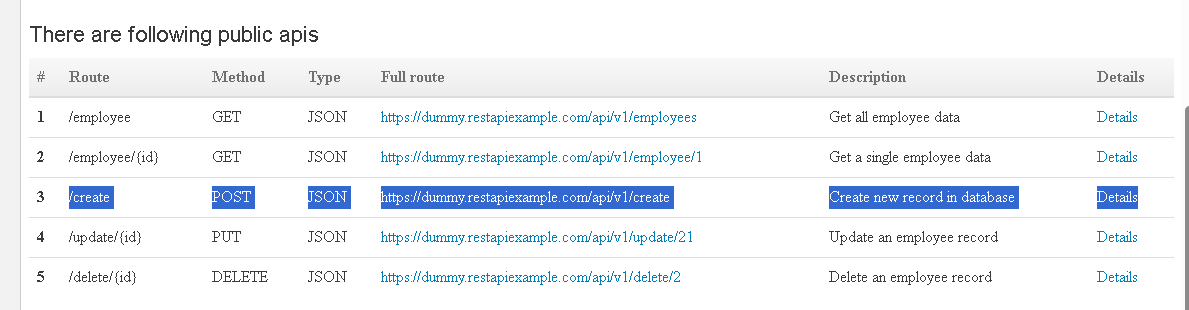
**First get all employee details**

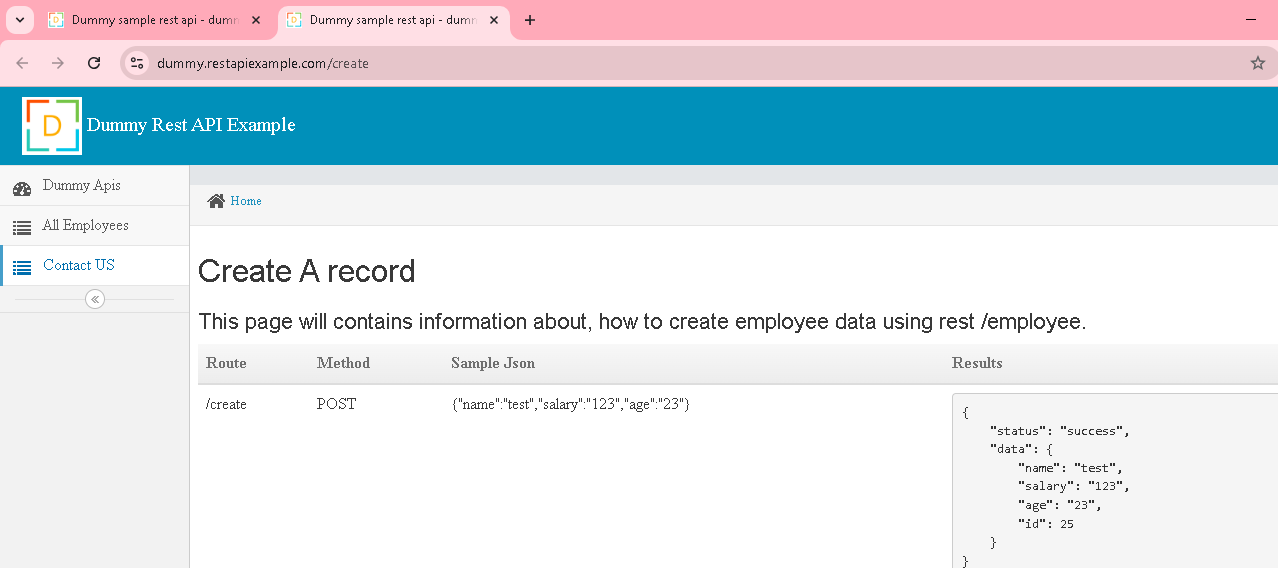
<https://dummy.restapiexample.com/api/v1/employees>

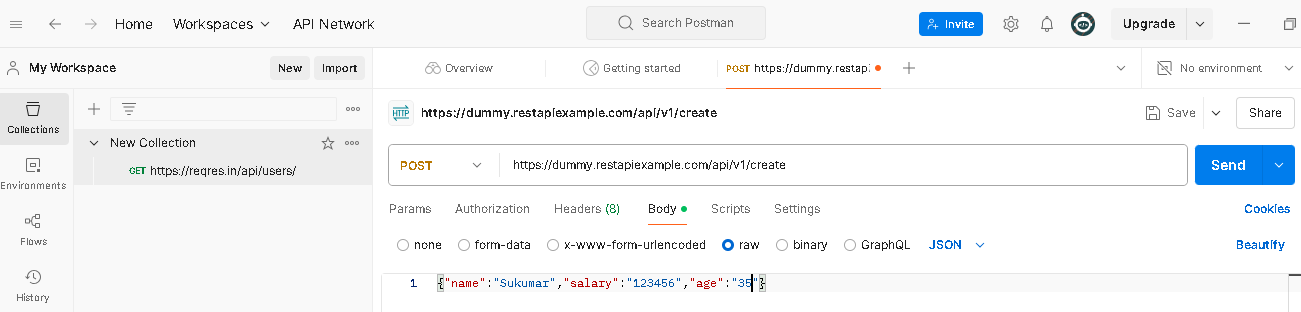
****

Create new record in database using post request

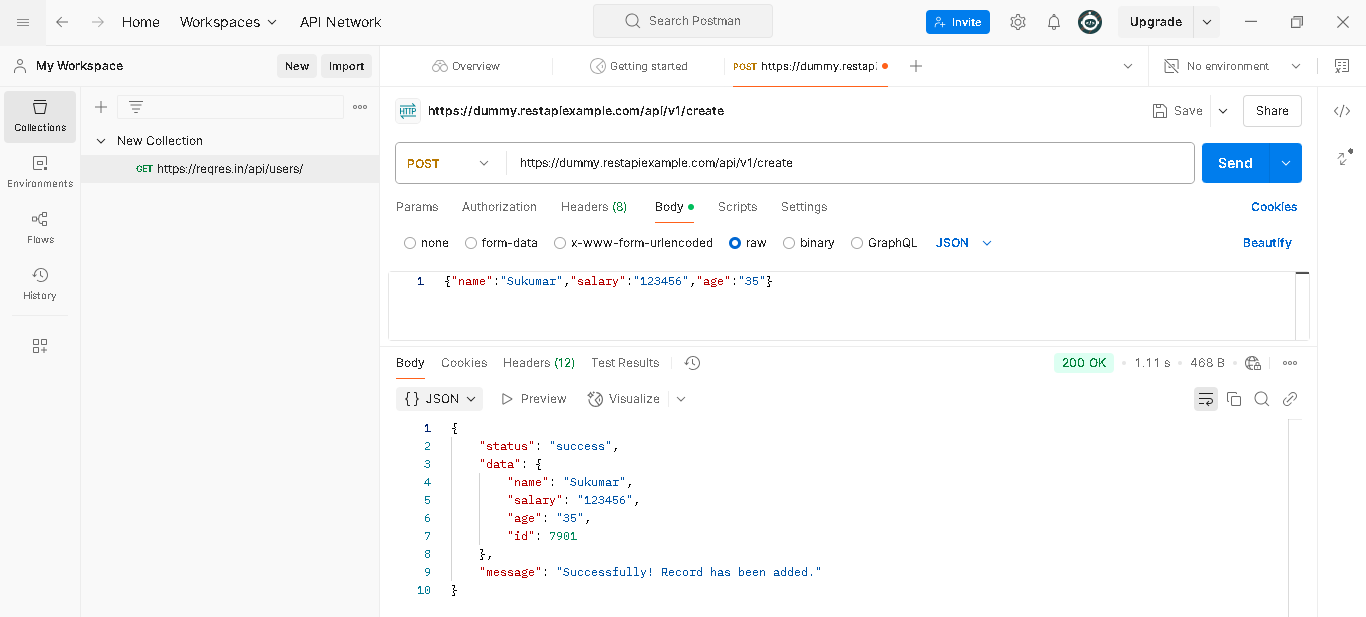
<https://dummy.restapiexample.com/api/v1/create>

****

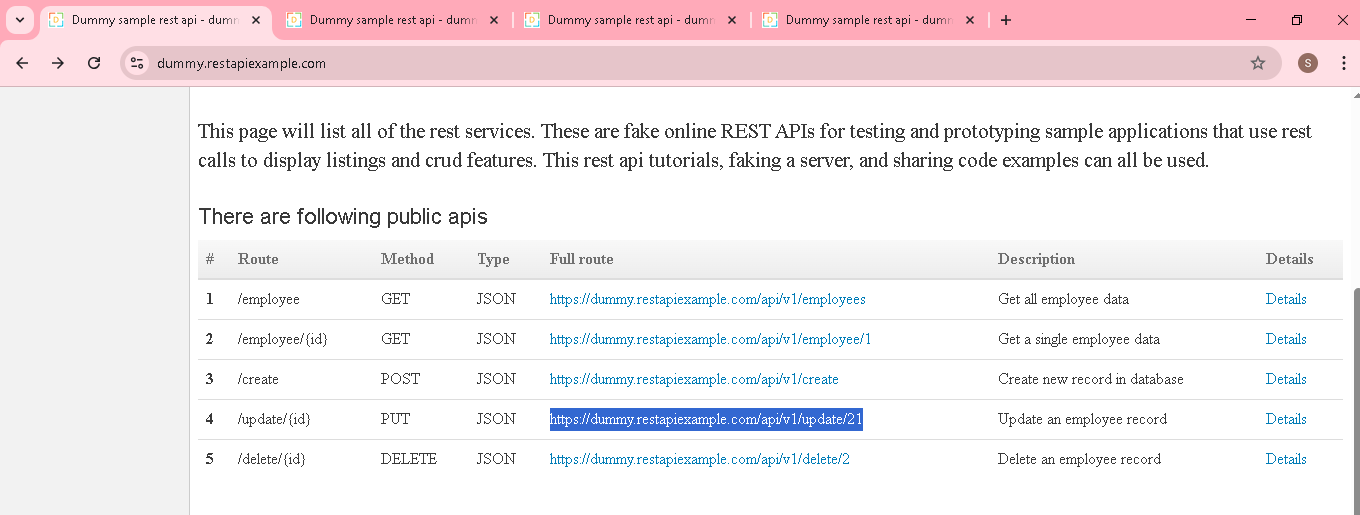
****

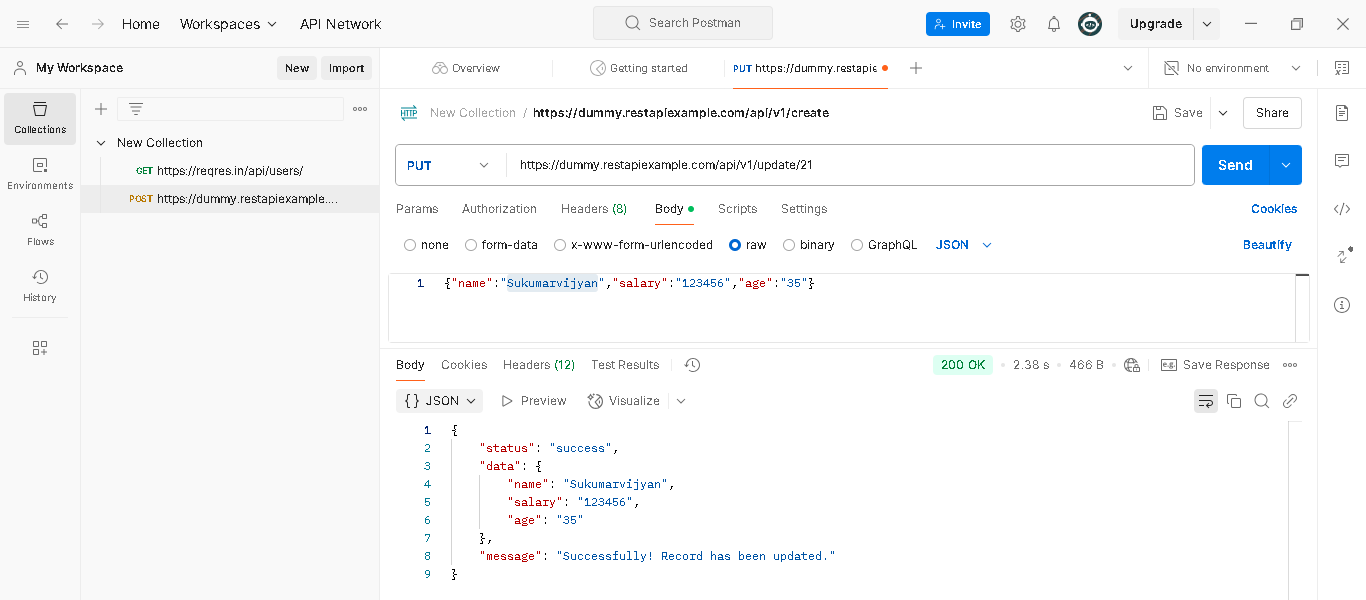
****

**New record created, ID automatically generated**

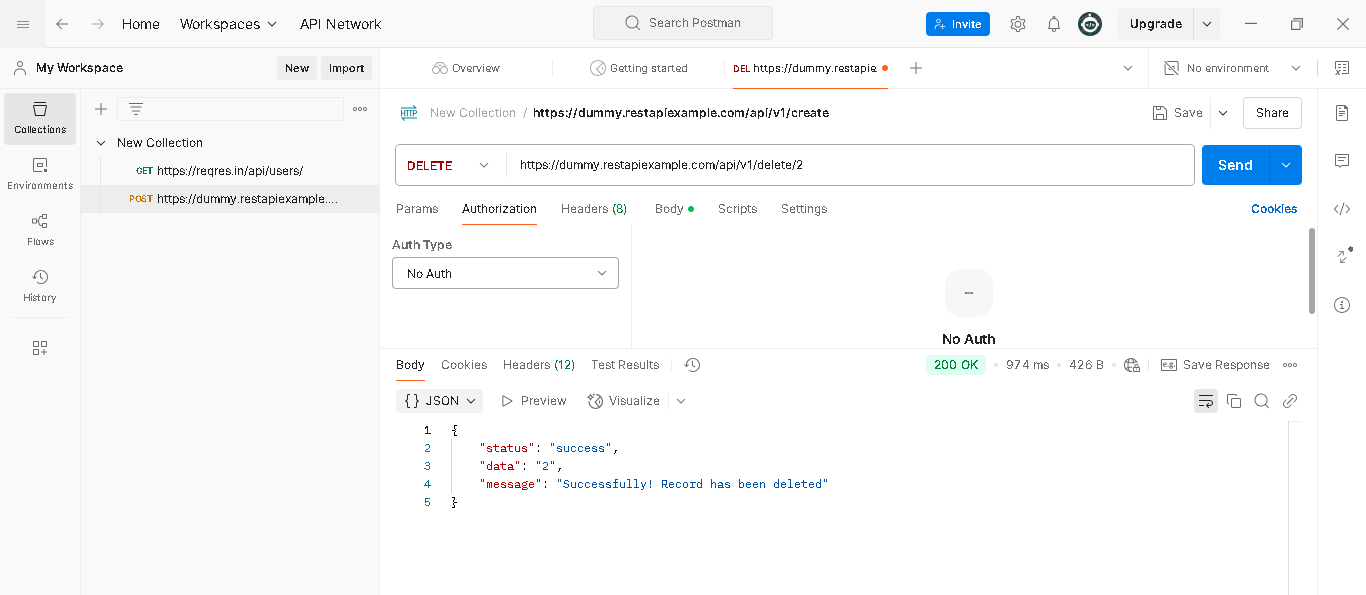
****

Update an employee record





Delete an employee record

****

**Automating API Tests**

There is a great misconception when it comes to automation. Selenium is not the solution for everything

API’s cant be automated using selenium..selenium is only for web application automation

**Then how to automate API? ->** Using Java & other third party jar as below

Standard HttpURLConnection(java.net) – default available inside JAVA itself

Apache HttpClientlibrary - third party

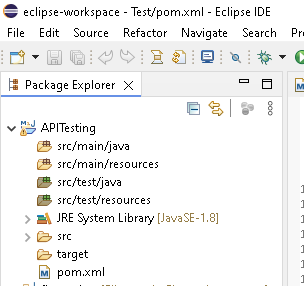
Unirest - third party

Google- Http-Client - third party library

Rest Assured - third party library

**By using Standard HttpURLConnection(java.net) to test API (get)**

**Create a maven project [ easy to maintain all above lib]**

****

**Create package & class**

**package** apiTest;

**import** java.io.BufferedReader;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.InputStreamReader;

**import** java.net.HttpURLConnection;

**import** java.net.URL;

**import** java.net.URLConnection;

**public** **class** APITestingGET {

**public** **void** getmethod() **throws** IOException {

//create object for URL(import java.net.URL)

URL url = **new** ~~URL~~("https://reqres.in/api/users/");

//openConnection retunn type URLconnection & URLconnection parent is HttpURLConnection

//need to use parent HttpURLConnection

//(HttpURLConnection) - Type cast , openConnection retunn type is URLconnection but we are using HttpURLConnection, so use type cast

HttpURLConnection connection = (HttpURLConnection) url.openConnection();

//below line to connect with that URL

connection.setRequestMethod("GET");

connection.connect();

//To get the responsecode

**int** statuscode = connection.getResponseCode();

System.***out***.println("Status code is" + statuscode);

//To get the response message

String statusmsg = connection.getResponseMessage();

System.***out***.println("Status Message is" + statusmsg);

//to read all the values in that url

//getInputStream read all values & stored in InputStream, need to use InputStreamReader & BufferedReader

InputStream inputstrem = connection.getInputStream();

InputStreamReader inputStreamReader = **new** InputStreamReader(inputstrem);

BufferedReader bufferedReader = **new** BufferedReader(inputStreamReader);

//String str = bufferedReader.readLine();

//System.out.println(str);

String line;

StringBuffer buffer = **new** StringBuffer();

**while** ((line = bufferedReader.readLine()) != **null**) {

buffer.append(line);

}

System.***out***.println(buffer);

}

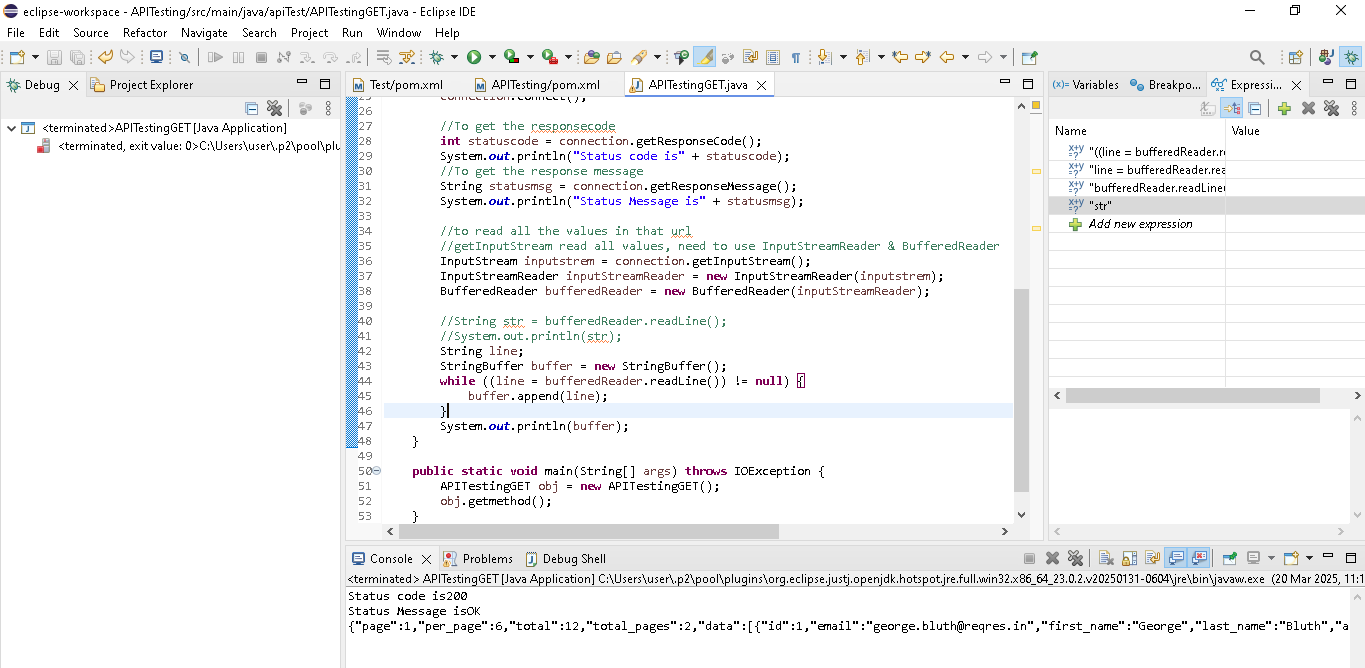
**public** **static** **void** main(String[] args) **throws** IOException {

APITestingGET obj = **new** APITestingGET();

obj.getmethod();

}

}

****

**Post request thru Java**

<https://dummy.restapiexample.com/api/v1/create>

**public** **void** putmethod() **throws** IOException {

//create object for URL(import java.net.URL)

URL url = **new** ~~URL~~("https://dummy.restapiexample.com/api/v1/create");

//openConnection retunn type URLconnection & URLconnection parent is HttpURLConnection

//need to use parent HttpURLConnection

//(HttpURLConnection) - Type cast , openConnection retunn type is URLconnection but we are using HttpURLConnection, so use type cast

HttpURLConnection connection = (HttpURLConnection) url.openConnection();

//below line to connect with that URL for post operation

connection.setRequestMethod("POST");

//--------------below line of code for post operation

//above url, reponse is jcon..so need to give content type as below

connection.setRequestProperty("Content-Type", "application/json");

//For post opeartion, setDoOutput need to set as true..

connection.setDoOutput(**true**);

//need to give the body details in order to create the details

String jsonbody = "{\"name\":\"Jothi\",\"salary\":\"123456\",\"age\":\"43\"}";

//server will accept only bytes, so convert string to bytes by using getBytes.

**byte**[] inputjscon =jsonbody.getBytes();

//need OutputStream in order to write

OutputStream outputStream = connection.getOutputStream();

outputStream.write(inputjscon);

//----------------------------------------

//To get the responsecode

**int** statuscode = connection.getResponseCode();

System.***out***.println("Status code is" + statuscode);

//To get the response message

String statusmsg = connection.getResponseMessage();

System.***out***.println("Status Message is" + statusmsg);

//to read all the values in that url

//getInputStream read all values, need to use InputStreamReader & BufferedReader

InputStream inputstrem = connection.getInputStream();

InputStreamReader inputStreamReader = **new** InputStreamReader(inputstrem);

BufferedReader bufferedReader = **new** BufferedReader(inputStreamReader);

//String str = bufferedReader.readLine();

//System.out.println(str);

String line;

StringBuffer buffer = **new** StringBuffer();

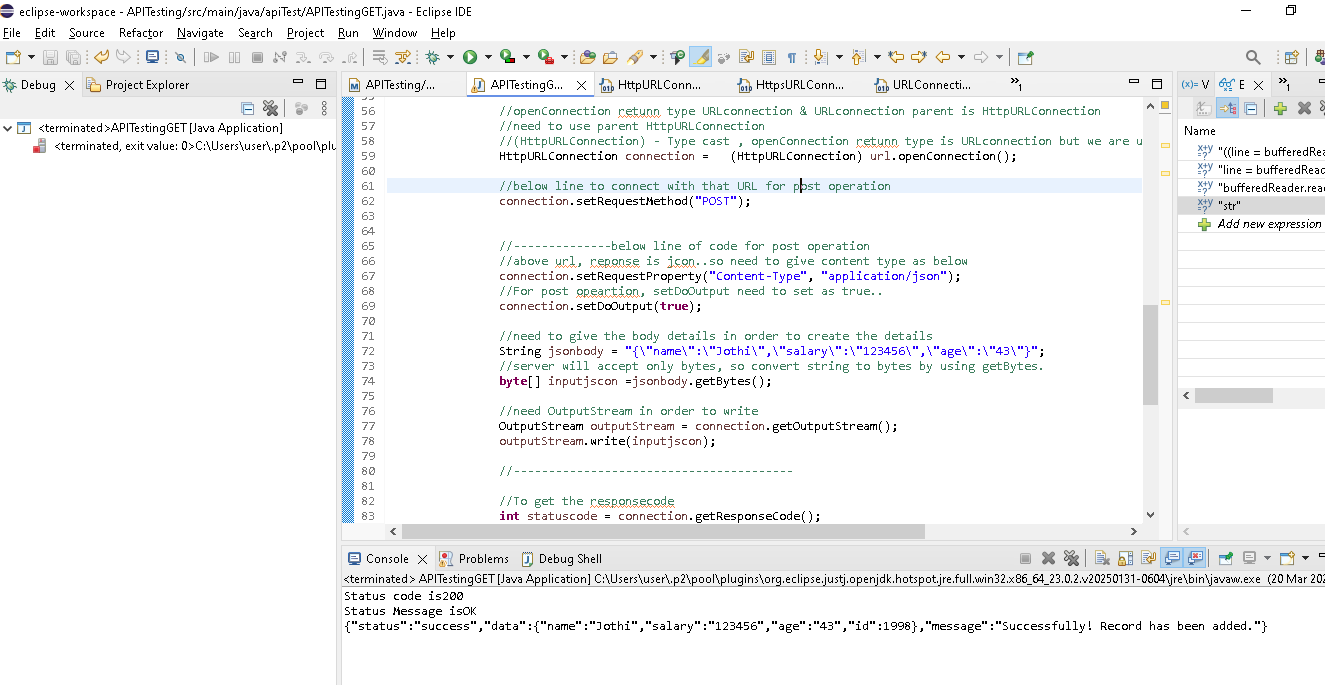
**while** ((line = bufferedReader.readLine()) != **null**) {

buffer.append(line);

}

System.***out***.println(buffer);

}

****

**Update & delete in Java**

**Update**

<https://dummy.restapiexample.com/api/v1/update/(id)>

update the ages alone..same code as Post..change URL & PUT type

connection.setRequestMethod("PUT");

**public** **void** updatemethod() **throws** IOException {

//create object for URL(import java.net.URL)

URL url = **new** ~~URL~~("https://dummy.restapiexample.com/api/v1/update/1998");

//openConnection retunn type URLconnection & URLconnection parent is HttpURLConnection

//need to use parent HttpURLConnection

//(HttpURLConnection) - Type cast , openConnection retunn type is URLconnection but we are using HttpURLConnection, so use type cast

HttpURLConnection connection = (HttpURLConnection) url.openConnection();

//below line to connect with that URL for put operation

connection.setRequestMethod("PUT");

//--------------below line of code for post operation

//above url, reponse is jcon..so need to give content type as below

connection.setRequestProperty("Content-Type", "application/json");

//For post opeartion, setDoOutput need to set as true..

connection.setDoOutput(**true**);

//need to give the body details in order to create the details

String jsonbody = "{\"name\":\"Jothi\",\"salary\":\"123456\",\"age\":\"100\"}";

//server will accept only bytes, so convert string to bytes by using getBytes.

**byte**[] inputjscon =jsonbody.getBytes();

//need OutputStream in order to write

OutputStream outputStream = connection.getOutputStream();

outputStream.write(inputjscon);

//----------------------------------------

//To get the responsecode

**int** statuscode = connection.getResponseCode();

System.***out***.println("Status code is" + statuscode);

//To get the response message

String statusmsg = connection.getResponseMessage();

System.***out***.println("Status Message is" + statusmsg);

//to read all the values in that url

//getInputStream read all values, need to use InputStreamReader & BufferedReader

InputStream inputstrem = connection.getInputStream();

InputStreamReader inputStreamReader = **new** InputStreamReader(inputstrem);

BufferedReader bufferedReader = **new** BufferedReader(inputStreamReader);

//String str = bufferedReader.readLine();

//System.out.println(str);

String line;

StringBuffer buffer = **new** StringBuffer();

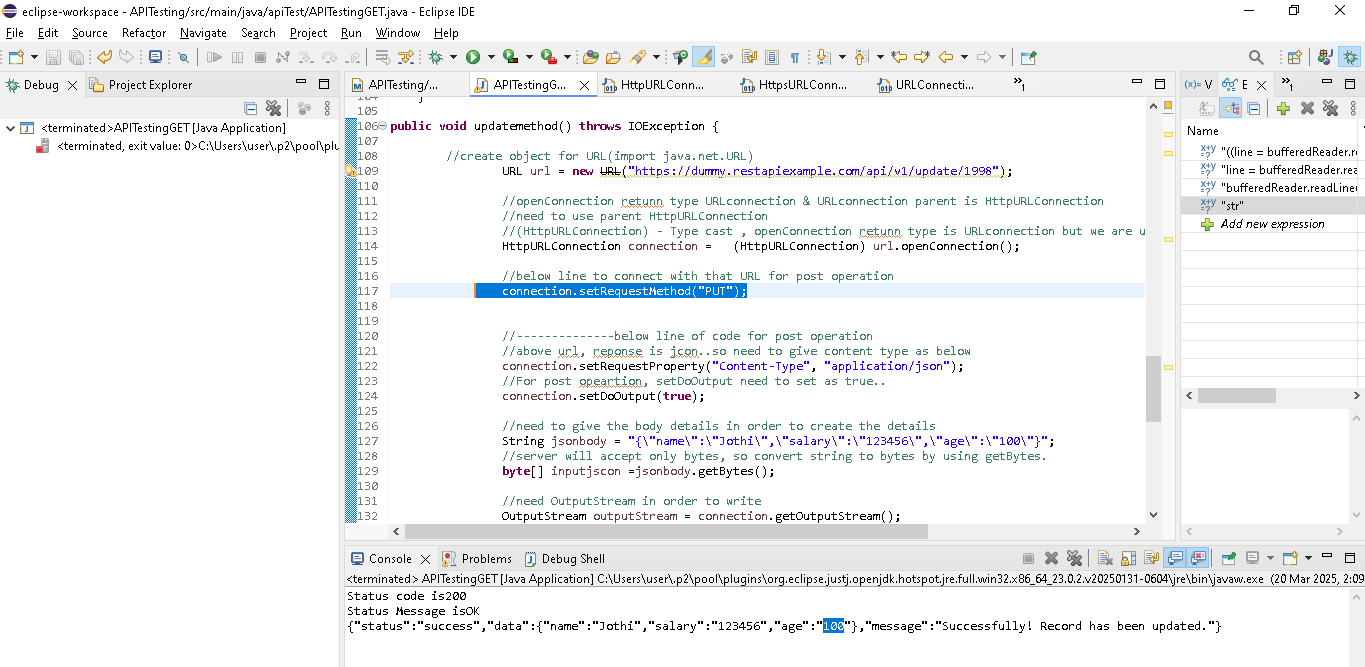
**while** ((line = bufferedReader.readLine()) != **null**) {

buffer.append(line);

}

System.***out***.println(buffer);

}



**Delete**

[https://dummy.restapiexample.com/api/v1/delete/2**(id)**](https://dummy.restapiexample.com/api/v1/delete/2(id))

DELETE the CREATED RECORD..same code as Post(byte, outputstream line can be removed)..change URL & delete type

connection.setRequestMethod("DELETE");

**public** **void** deletmethod() **throws** IOException {

//create object for URL(import java.net.URL)

URL url = **new** ~~URL~~("https://dummy.restapiexample.com/api/v1/delete/1998");

//openConnection retunn type URLconnection & URLconnection parent is HttpURLConnection

//need to use parent HttpURLConnection

//(HttpURLConnection) - Type cast , openConnection retunn type is URLconnection but we are using HttpURLConnection, so use type cast

HttpURLConnection connection = (HttpURLConnection) url.openConnection();

//below line to connect with that URL for post operation

connection.setRequestMethod("DELETE");

//--------------below line of code for post operation

//above url, reponse is jcon..so need to give content type as below

connection.setRequestProperty("Content-Type", "application/json");

//For post opeartion, setDoOutput need to set as true..

connection.setDoOutput(**true**);

//----------------------------------------

//To get the responsecode

**int** statuscode = connection.getResponseCode();

System.***out***.println("Status code is" + statuscode);

//To get the response message

String statusmsg = connection.getResponseMessage();

System.***out***.println("Status Message is" + statusmsg);

//to read all the values in that url

//getInputStream read all values, need to use InputStreamReader & BufferedReader

InputStream inputstrem = connection.getInputStream();

InputStreamReader inputStreamReader = **new** InputStreamReader(inputstrem);

BufferedReader bufferedReader = **new** BufferedReader(inputStreamReader);

//String str = bufferedReader.readLine();

//System.out.println(str);

String line;

StringBuffer buffer = **new** StringBuffer();

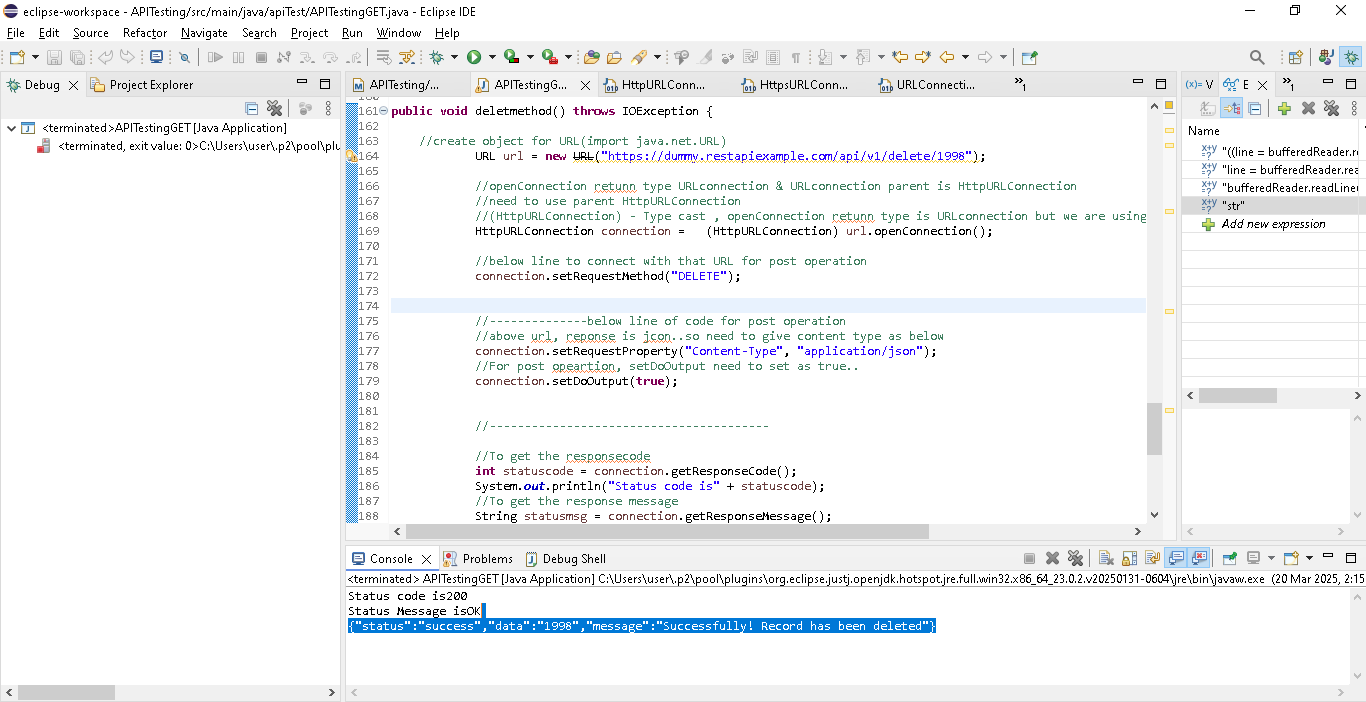
**while** ((line = bufferedReader.readLine()) != **null**) {

buffer.append(line);

}

System.***out***.println(buffer);

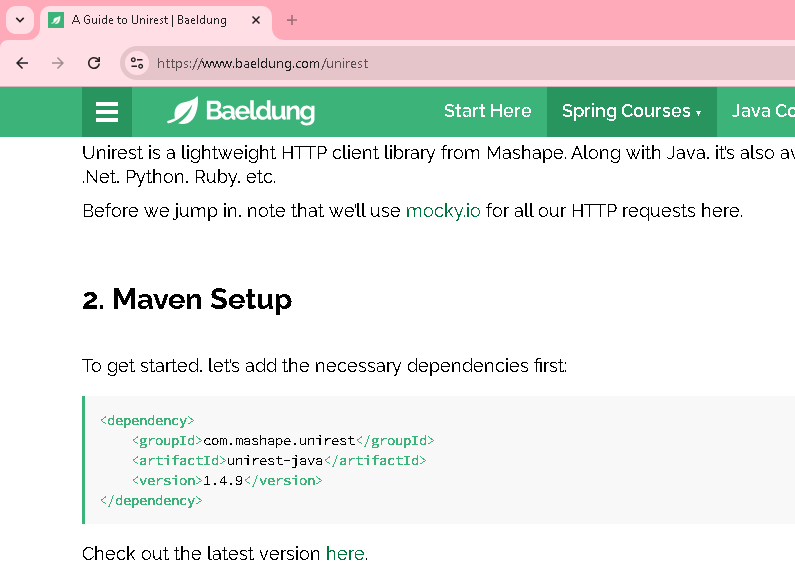
}

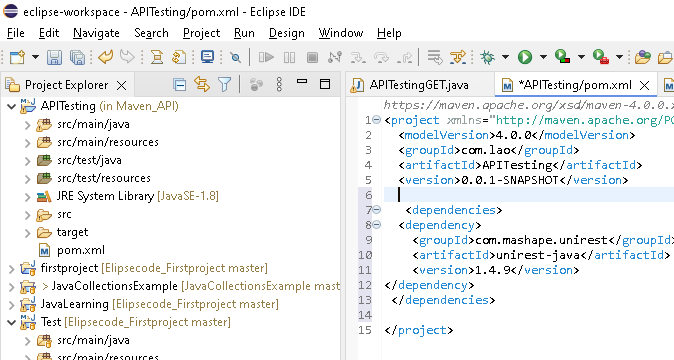
****

**Unirest lib for java API Automation ->Its very light weight & easy to use compare to above default lib**

Add the lib in porn.xml of maven project

[**https://www.baeldung.com/unirest**](https://www.baeldung.com/unirest)

****

****

Check the folder structure after save & refresh

Maven dependencies’ folder created & contains all jar files related

**Get Method using Unirest – less code compare to above java Http default lib**

**public** **void** Unirestget() **throws** UnirestException {

//below one line perform get with url & retrieve the response(Json, so use .asJscon) & stored in Jsondata variable

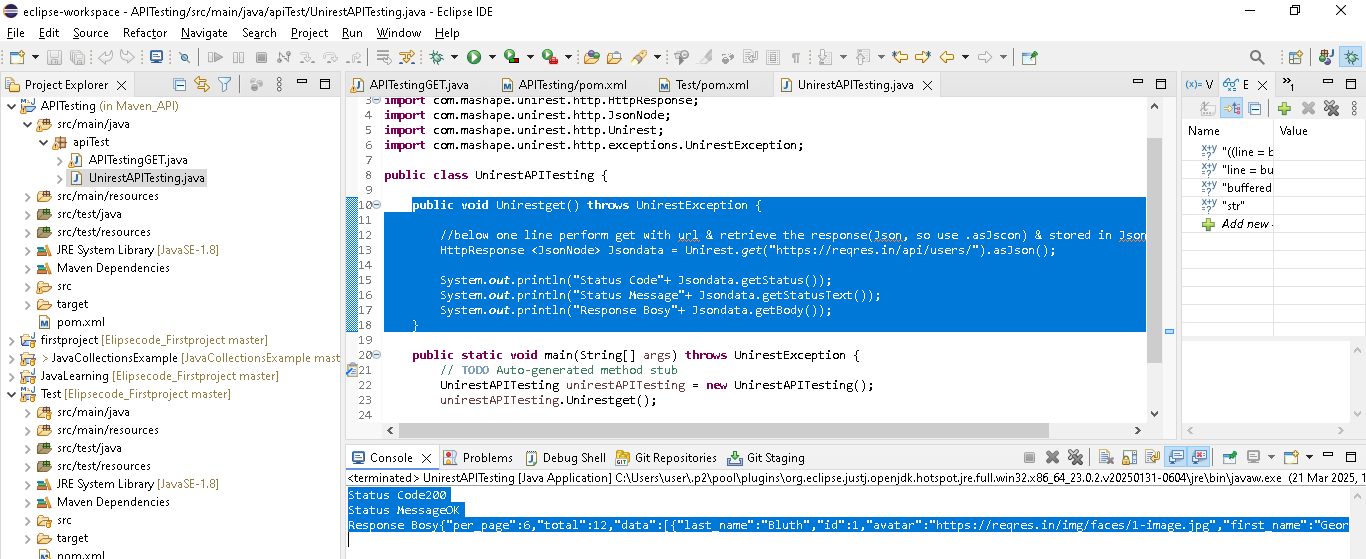
HttpResponse <JsonNode> Jsondata = Unirest.*get*("https://reqres.in/api/users/").asJson();

System.***out***.println("Status Code"+ Jsondata.getStatus());

System.***out***.println("Status Message"+ Jsondata.getStatusText());

System.***out***.println("Response Bosy"+ Jsondata.getBody());

}

****

**Post Method**

**public** **void** Unirestpost() **throws** UnirestException {

//below one line perform post with url & create with body details(body() & Json response, so use .asJscon) & stored in Jsondata variable

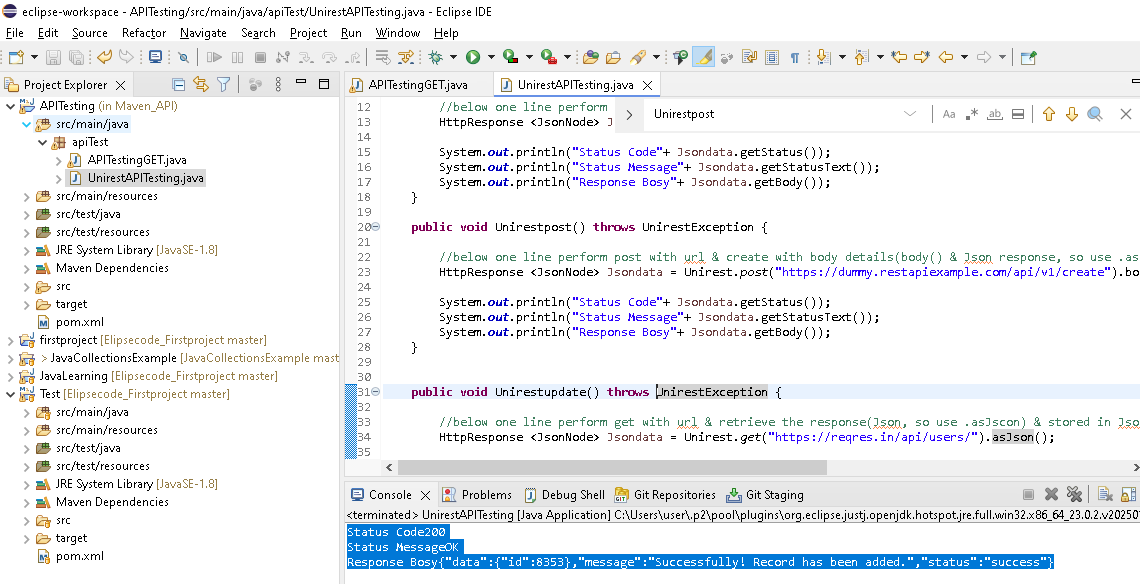
HttpResponse <JsonNode> Jsondata = Unirest.*post*("https://dummy.restapiexample.com/api/v1/create").body("{\"name\":\"JothiSuku\",\"salary\":\"123456\",\"age\":\"43\"}").asJson();

System.***out***.println("Status Code"+ Jsondata.getStatus());

System.***out***.println("Status Message"+ Jsondata.getStatusText());

System.***out***.println("Response Bosy"+ Jsondata.getBody());

}

****

**Update method**

**public** **void** Unirestupdate() **throws** UnirestException {

//below one line perform get with url & update the response(Json, so use .asJscon) & stored in Jsondata variable

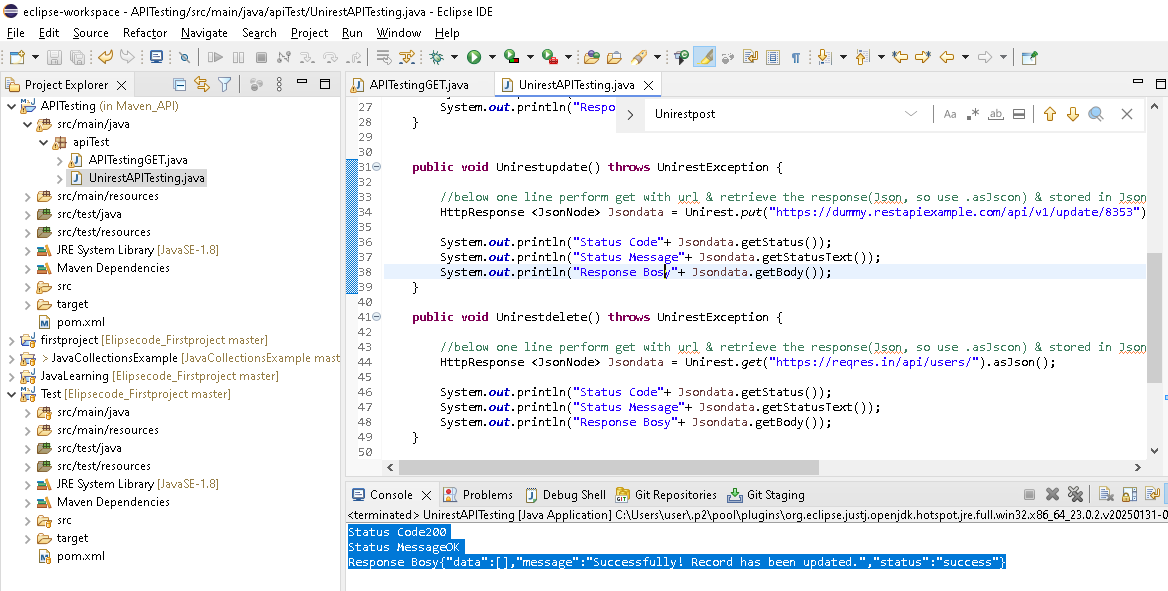
HttpResponse <JsonNode> Jsondata = Unirest.*put*("https://dummy.restapiexample.com/api/v1/update/8353").body("{\"name\":\"NaveenSuku\",\"salary\":\"50000\",\"age\":\"43\"}").asJson();

System.***out***.println("Status Code"+ Jsondata.getStatus());

System.***out***.println("Status Message"+ Jsondata.getStatusText());

System.***out***.println("Response Bosy"+ Jsondata.getBody());

}

****

**Delete method**

**public** **void** Unirestdelete() **throws** UnirestException {

//below one line perform post with url & delete (Json response, so use .asJscon) & stored in Jsondata variable

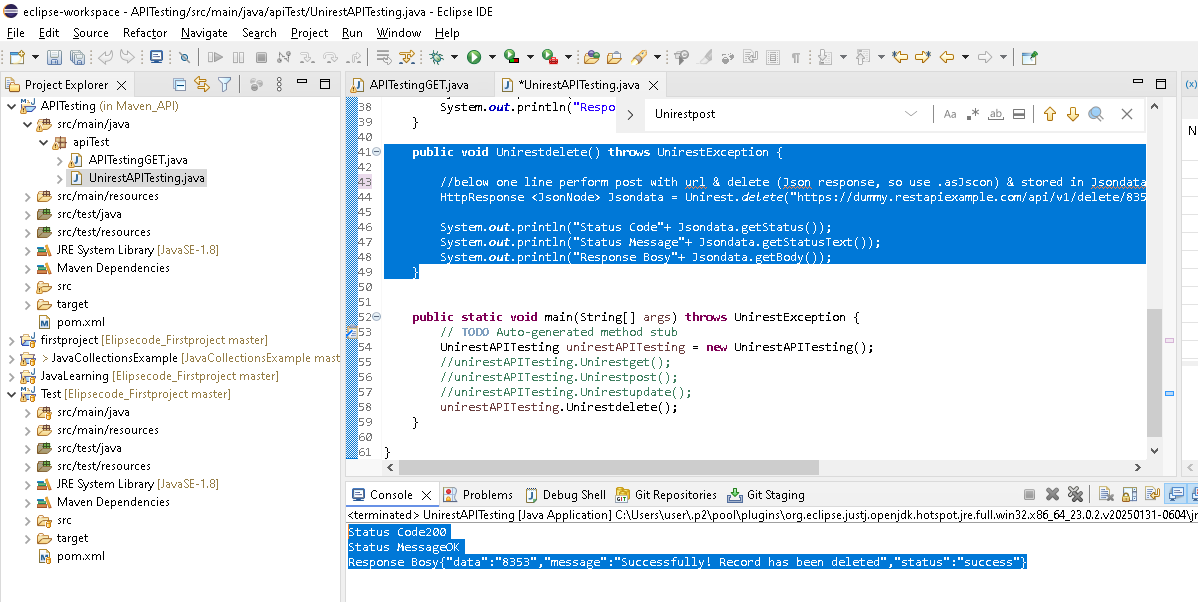
HttpResponse <JsonNode> Jsondata = Unirest.*delete*("https://dummy.restapiexample.com/api/v1/delete/8353").asJson();

System.***out***.println("Status Code"+ Jsondata.getStatus());

System.***out***.println("Status Message"+ Jsondata.getStatusText());

System.***out***.println("Response Bosy"+ Jsondata.getBody());

}

****

**Why Rest is a stateless architecture?**

**What is stateful application, why stateful application is it a failure approach & how stateless overcomes it.**

**Stateful ->means user login ID/credential stored in server state.]**

**Problem:** During peak time, company have multiple server(amazon big million day), no of user will be more, User searching the one product & so load balancer first stored the user detail in server1..after some time user searching diff product..during the time server1 peack,so load balancer changing the request to server2..but in server 2 login details not present..this a problem for stateful application **[ Refer below diagram]**

To overcome these , we have **stateless application**..here user details not stored instead Token will be generaeted from server 1when user send first request to server...next time if load balancing change request to server2 along with that Token for user identification/reference..

**Difference between Rest & soap(simple object access protocol)**

Soap -> its a set of very strict rules, all application need to follow the rules...old approach, out dated & data transfer in XML only

Rest->here no rules & simple process[as above all points from page1 to last]..modern approach now

Problem

**User ->**

**Token**

Token

**User ->**