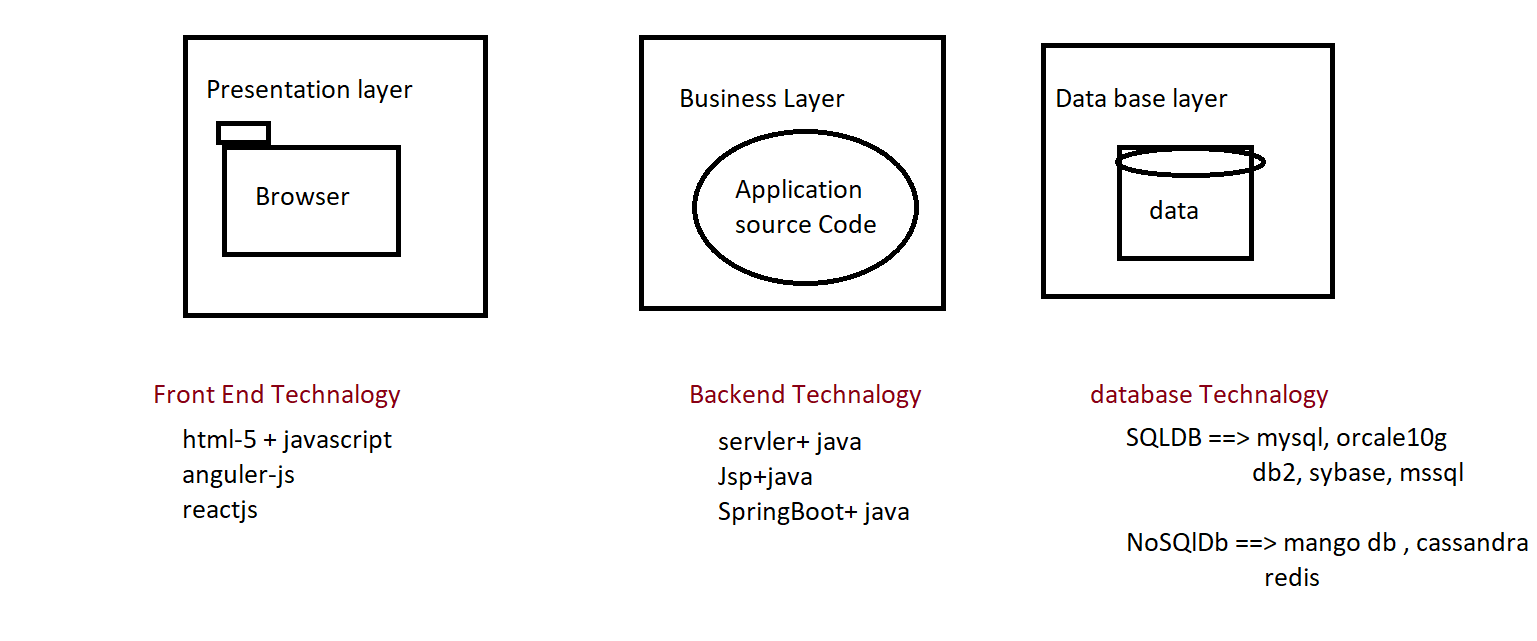
1. **What technology your application was developed?**



1. **Difference between Unit testing & API testing**

|  |  |
| --- | --- |
| Unit Testing | API testing(web service) |
| Done by development team | Done by testing team or SDET team |
| Should have knowledge on source code | Should have knowledge on expectations and agreements of API (Api-Functional Spec) |
| Test each and every source code of the application | Test the data flow between 2 application |
| It is a white box testing | It is a grey box testing |
| Tools: eclipse+testNG | Tools: SOAP UI, POSTMAN, RESTCLIENT, RESTASSURED |

1. **Difference between JSON & XML language**

|  |  |
| --- | --- |
| JSON | XML |
| JSON object has a type | XML data is type less |
| JSON types: string, number, array, Boolean, null, Object, arrayofObject | All XML data should be string |
| Data is readily accessible as JSON objects | XML data needs to be parsed. |
| JSON is supported by most browsers. | Cross-browser XML parsing can be tricky |
| Retrieving value is easy | Retrieving value is difficult |
| A fully automated way of DE serializing/serializing JavaScript. | Developers have to write JavaScript code to serialize/de-serialize from XML |
| Native support for object. | The object has to be express by conventions - mostly missed use of  attributes and elements. |
| It supports only UTF-8 encoding. | It supports various encoding. |
| Structure is simple | Structure is complex |
| It doesn't support comments. | It supports comments. |
| JSON files are easy to read as compared to XML. | XML documents are relatively more difficult to read and interpret. |

|  |  |
| --- | --- |
| SOAP Web Service | Rest Web Service |
| SOAP is a protocol | REST is Architectural style |
| Simple Object Access Protocol | Representational State Transfer |
| SOAP can't use REST Web services, it is a protocol | REST can use xml as well Web Services since it is concept |
| SOAP expose the services (Business Logic) via .WSDL file | REST expose the service (Business Logic) via URI |
| SOAP designed using too much standards | REST does not define too much standards |
| SOAP permits XML data format only | REST permits different data formats- |
|  | Plain text, html, JSON, etc |
| SOAP requires more bandwidth and resources | REST requires less bandwidth and resource than SOAP |
| SOAP defines its own security | Inherits security measures from underlined transport Protocol (ouath-1.0 outh-2.0 , Bearer token) |
| Less preferred than REST | REST more preferred than SOAP |

1. **Difference between SAOP & Rest WebService**
2. **Difference between Frontend & Backend language**

|  |  |
| --- | --- |
| Frontend testing | BackEnd testing |
| Front-end is always performed on GUI | Back-end testing is done on Business layer and should  have knowledge on Data base and business logic of software |
|  |  |
| Testing becomes slower coz it have to wait for browser rendering time | It is faster and saves time. |
| Performance testing Is not easy in front-end | Performance testing is easy and faster |
| Adoc, usability and compatibility testing is possible | Adoc, usability and compatibility testing is not possible |
| Manual & selenium automation will be done at front end(BBT) | whiteBox , Soap API & rest API tesing will be done in the backEnd(WBT. GBT) |
| End to end testing is possible | End to end testing is not possible |
| Tools: selenium, QTP, test complete etc | SOAP UI, POSTMAN, RESTCLIENT, RESTASSURED, testNG , JUNIT |

1. Which APi testing Tool you have used?

**PostMan & RestAssured** , RestClient , testOptimize , read API

1. Which type of Webservice testing you worked on

Rest WebServciec testing

1. Types of API testing ?
2. Soap WebServcie API testing
3. REST WebServcie API testing
4. What is API ?

Application programing Interface, testing interface between two application, or

Testing the application in business layer without browser is called API testing

EG : Rest webservcie is API testing

Soap WebServcie is API testing

1. What is the advantages of the webservice API testing?

a. API testing very Faster (because no to wait for Browser rendering time)

b. testing the functionality with out GUI(Browser)

c. testing the functionality early stages (We can start API testing in Sprint-1)

d. find defect in early stages

e. Time effective & fast to release

f. whenever API provider develop an API , every api should be tested the Functionality, performance security, reliability before exposing those API to consumer

6. what is the disadvantages API testing?

b. End to end workflow testing is difficult

c. testing should have knowledge on Programing & data base (Backend)

1. What is Unit testing ?

Testing the source of the application using another program is called unit API testing

1. Why JSON is very Popular?
   1. Java Script Object Notation
   2. It’s a Programing language to exchange information between 2 application
   3. The JSON format is syntactically identical to the code for creating JavaScript objects.
   4. Because of this similarity, a JavaScript program can easily convert JSON data into native JavaScript objects.
   5. JSON is a lightweight format for storing and transporting data
   6. JSON is often used when data is sent from a server to a web page
   7. JSON is "self-describing" and easy to understand
   8. All Browser & Mobile UI can easily consume json language
   9. NO SQL databases can directly store the data in the from of JSON (MangoDb, Casendra )
   10. Platform independent

1. What is Soap webservice testing
   * When two application exchange information via **soap** protocol, which is based on xml based , testing those request and response is called Soap webservice testing
   * SOAP Webserive exposed via .WSDL file
   * EG : SOAP API

<http://www.dneonline.com/calculator.asmx?WSDL>

<http://map.google/api/soap/service.wsdl>

1. What is Rest Webservcie testing ?

* When twoapplication exchange information via **http** protocol , which is based json/xml/text/html/js , testing those request and response is called Rest webservcie testing
* Rest API Webserice are exposed via URI
* EG : Rest API

[http://map.google/api/getlocation](http://map.google/api.getlocation)

<http://map.google/api/gettraffice?src=‘value’&> dest=value

<http://map.google/api/gettraffic>

1. How many years of Experience in API project?

2 years

1. Where did you used API in your Previous project & Purpose of the API?\*\*\*\*\*\*\*\*\*\*\*\*

Product , contacts , Quote , Case

1. How many API you have tested\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

50-60 RestFull API

1. Can you Write or list Out few API in your Previous Project

Product

Get <Http://autodesk-test-env.com:8084/products>

Post <Http://autodesk-test-env.com:8084/products/addproduct>

Put patch ,delete [Http://autodesk-test-env.com:8084/product/{productID}](Http://autodesk-test-env.com:8084/product/%7bproductID%7d)

Get <Http://autodesk-test-env.com:8084/products>?name =hplabtaop

1. Write on POST API request in your previous Project

EG:

Post <Http://autodesk-test-env.com:8084/products/addproduct>

{

"ProductNAme" : "name"

"SerailNum" : 123

"Manfauting date" : ""

"unitPrice" : 2000INR

"Qty" : 10

}

1. Write one API response in your previous Project

{

"name": "name",

"msg": "successfully added",

"id": "625",

"createdAt": "2020-11-02T08:05:40.255Z"

}

EG : get all Products

Get <Http://autodesk-test-env.com:8084/products>

{

"page": 2,

"per\_page": 6,

"total": 12,

"total\_pages": 2,

"data": [

{

"id": 7,

"product": "michael.lawson@reqres.in",

"first\_name": "Michael",

"last\_name": "Lawson",

"avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/follettkyle/128.jpg"

},

{

"id": 8,

"email": "lindsay.ferguson@reqres.in",

"first\_name": "Lindsay",

"last\_name": "Ferguson",

"avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/araa3185/128.jpg"

},

{

"id": 9,

"email": "tobias.funke@reqres.in",

"first\_name": "Tobias",

"last\_name": "Funke",

"avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/vivekprvr/128.jpg"

},

{

"id": 10,

"email": "byron.fields@reqres.in",

"first\_name": "Byron",

"last\_name": "Fields",

"avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/russoedu/128.jpg"

},

{

"id": 11,

"email": "george.edwards@reqres.in",

"first\_name": "George",

"last\_name": "Edwards",

"avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/mrmoiree/128.jpg"

},

{

"id": 12,

"email": "rachel.howell@reqres.in",

"first\_name": "Rachel",

"last\_name": "Howell",

"avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/hebertialmeida/128.jpg"

}

],

"ad": {

"company": "StatusCode Weekly",

"url": "<http://statuscode.org/>",

"text": "A weekly newsletter focusing on software development, infrastructure, the server, performance, and the stack end of things."

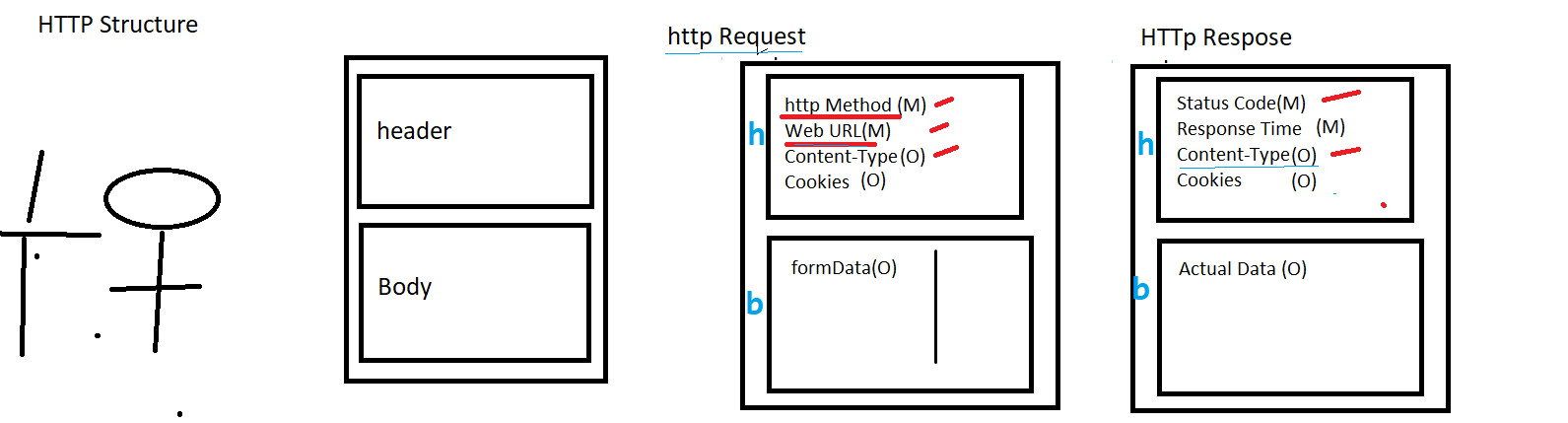
}

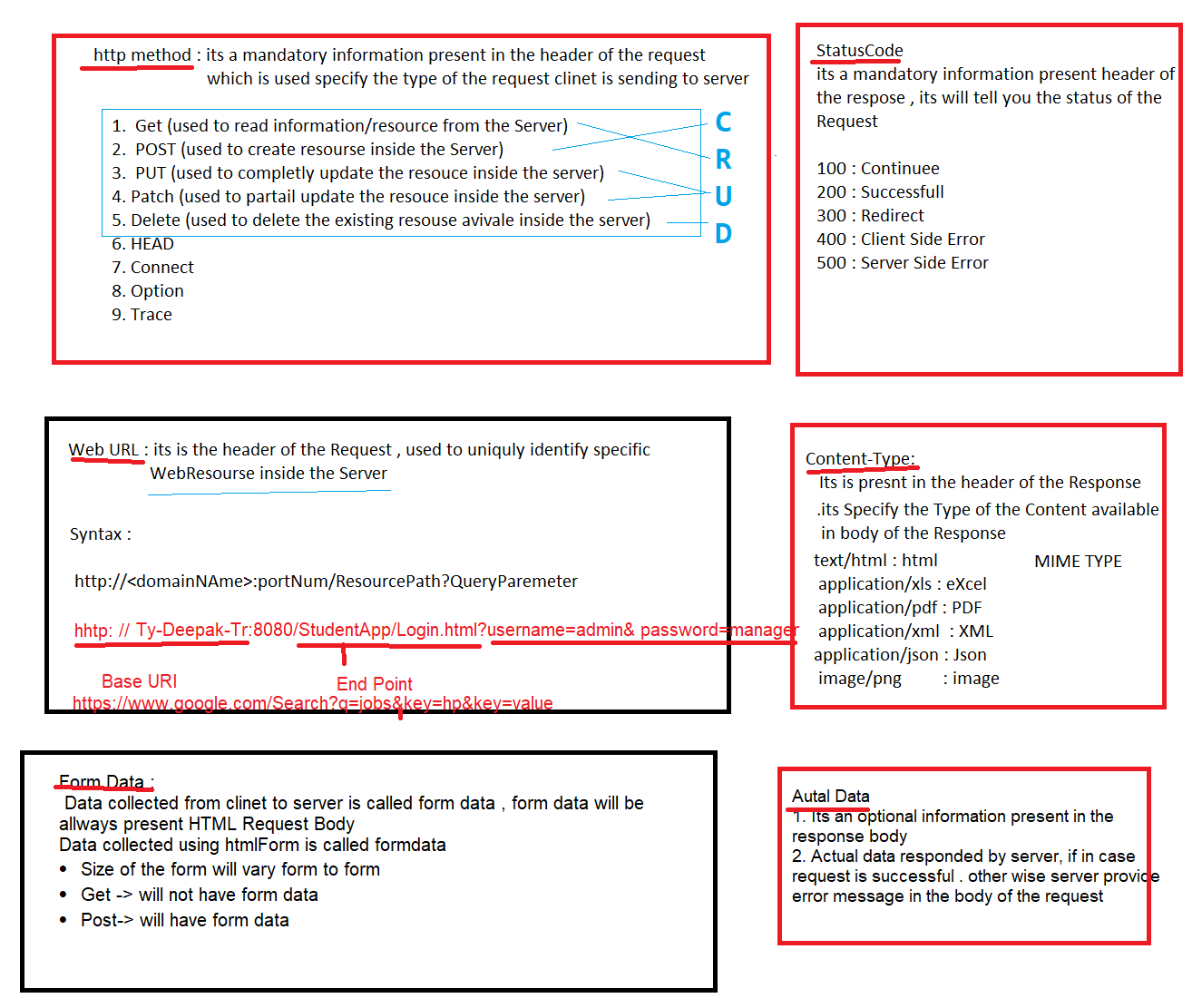
}

1. How many API test case you written

20 API 🡺 100 tc

1. Explain httpRequest / httpResponse structure





HttpREquest :

httpMtd : specify the type of the request

Web URL : uniquely identify the WebResourse inside the Web Application

ContentType : specify the type of content available in body the Request/Response

formData : data collected in html Form / or request body

HttpREsponse

Status Code : specify the status of the Request

ContentType

Actaul Data : data available in response body

1. Explain the http method which you used In your previous project

Get = > read resource from the sever

Post => create resource inside the sever

Put=> create resource/ complete update resource inside the sever

Patch => partial update resource inside the sever

Delete => Delete resource inside the sever

1. Explain the status code which you encounter in your previous project\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**100 Continue**

**--------------------------SucessFull----------------------**

[**200 (OK)**](https://restfulapi.net/http-status-200-ok/) **: able to read resource from the server**

[201 (Created)](https://restfulapi.net/http-status-201-created/) : able to create **resource inside the server**

[202 (Accepted)](https://restfulapi.net/http-status-202-accepted/) : able to accept the permission request inside the server

[204 (No Content)](https://restfulapi.net/http-status-204-no-content/) : got response from the server but no content

---------------------------------- **Redirected** ---------------------------------

300 Redirected : indicates that further action needs to be taken by the user agent in order to fulfill the request. http method is not supported

-------------------------------**Client Side Error-----------------------------------------------**

400 (Bad Request) : The request could not be understood by the server due to malformed syntax

401 (Unauthorized) : request is not authorized to access the resource inside the server

403 (Forbidden) : server understood the request but not authorized person to access

404 (Not Found) : not able to find the resource inside the server which you requested

405 Method Not Allowed

---------------------------------------**Server Side Error-------------------------------------------------**

500 Internal Server Error : problem from server side

502 Bad Gateway : proxy / firewall (network) issue in server side

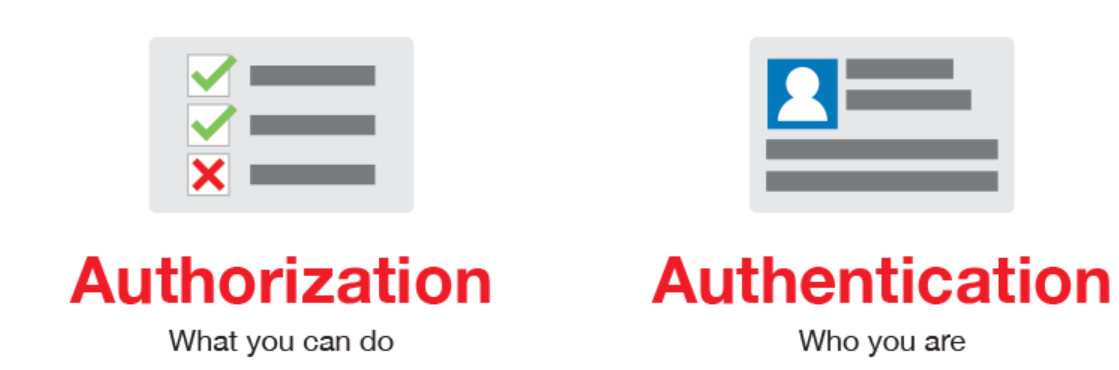
503 Service Unavailable : The server is currently unable to handle the request

504 Gateway Timeout : server not response even waiting for long time

1. **What is Crud operation**

**Create read update delete**

1. **What Authorization & Authentication**

****

Authentication to check whether you are authenticated client or not ?

Authorization to check what resource your accesses

1. **Challenges you faced in API Testing**
2. End to End Scenario testing is challenging because we have to do API Chaining

EG: Scenario 🡪Search Product + ADD to cart + Billing + Logistic

API : API -1 for Search Product

API-2 for ADD to cart

API-3 for Billing

API-4 for Logistic

1. API Document is not clear, API testing challenging
2. Negative testing is challenging because in api document will not have complete requirement
3. Validation of Complex response is challenging
4. Deriving api test scenario is challenging ,
5. **Prerequisite for APi testing\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

* Knowledge of client server architecture of the application
* Knowledge of HTTP request and HTTP response properties;
* Knowledge of JSON or XML language
* API documentation / API Functional spec (Swagger Document)from Development team
* Get the list of URI (BASEURL + ENDPOINT 🡺 URI)
* Knowledge of CRUD
* Write test case for every API that include CRUD operation
* Required tool for API TESTING

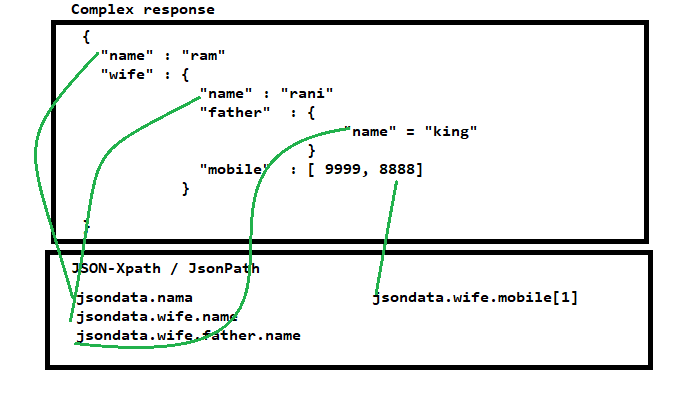
Postman 🡪 GUI based API testing tool

RestAssured 🡪 Headless Java libraries for REST API testing

* Collect API authentication if required (like Basic Auth , Bearer Auth , Oauth-1.0, Oauth-2.0

1. **How to verify the complex response in API testing?**

Will write jsonPath to navigate to specific data inside the response , then will go for “JSon Value checker ”Assertions in postMan

****

1. **How many ways we can post a request in POSTMAN**
2. Post a request body by JSONOBJECT
3. Post a request body by Key:value pair (hashMAp)
4. Post a request body by JSON File
5. **How to post Complex request in postman**

**We can post json File Itself in postman 🡺 body🡺binary🡺 select File**

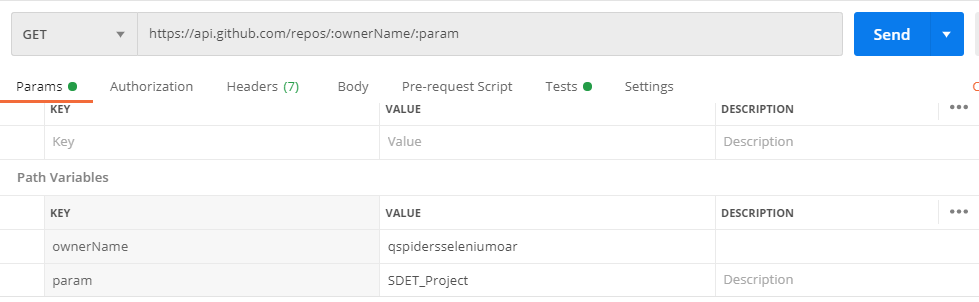
**Note: file extension should be file.json , if content Type in JSON**

1. **Explain the validation available in POSTMAN APi testing \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
2. ResponseBody : Verify status Code : used to verify the response status Code
3. ResponseBody : time is less than --- : used to verify the Response time
4. ResponseBody : json value Check : used to navigate to particular key
5. ResponseBody : contain string : used to check the particular String is available in entire response body
6. ResponseBody : equal to String
7. **sendRequest : we can send another request with in a request**

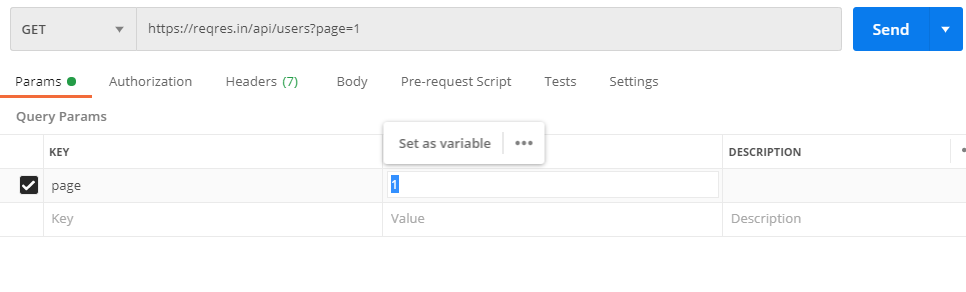
1. What is Parameter & types of Parameter?

To achieve data driven testing we go for Parameter

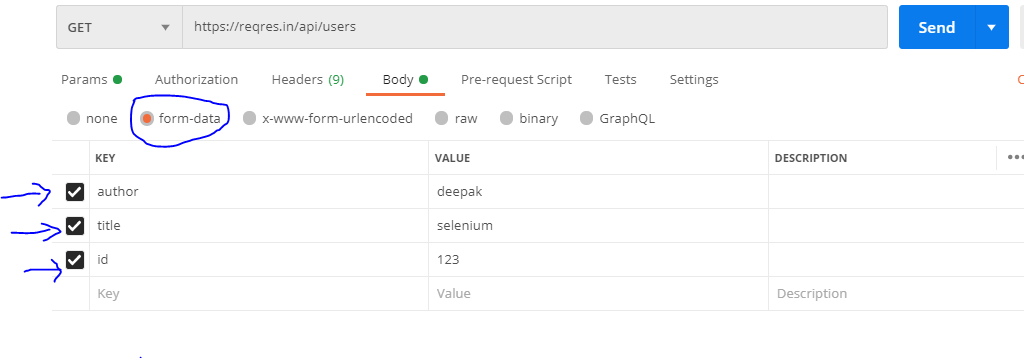
1. Path parameter: it’s the Part the ENDPoint, execute same request with different data we go for Path Parameter



1. Query parameter : it’s part of the Query String, execute same request with diff filter /search criteria



1. form Parameter : send complete request body in the form of Parameter



1. What Authentication & Autherization

Authentication: used to check whether you are valid user or not?

Authorization : used to check your permission / accesses to the resource

1. **Types of API Athentication & which Athentication you have used in your previous project**
2. Basic Auth (send a request using username/ password)
3. Bearer Token (send a request using tokenID, but token is fixed)
4. OAuth-1.0(older)
5. OAuth-2.0 (send a request using tokenID , but token ID is dynamic created via Outh-cleintAPP) or (Gmail app , allow grant permission to skillRaray app without sharing your username/password via Oauth-2 Protocol )
6. **What is Request chaining?**

**Capturing data from one response using jsonXpath() , & pass same data in to another request is called Request chaining .**

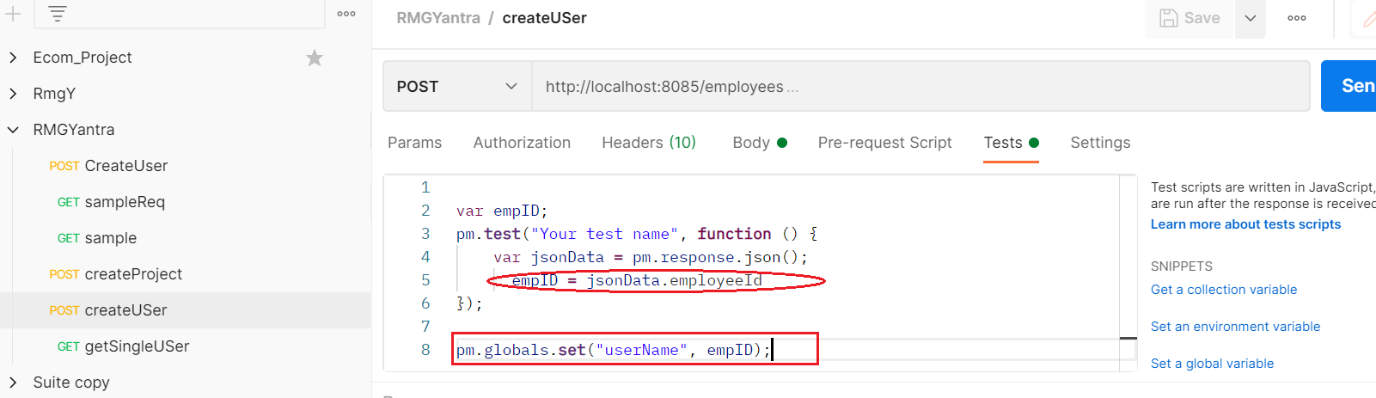
**In below example, capture “id” value from the first api request & pass same data in another URL**

EG :

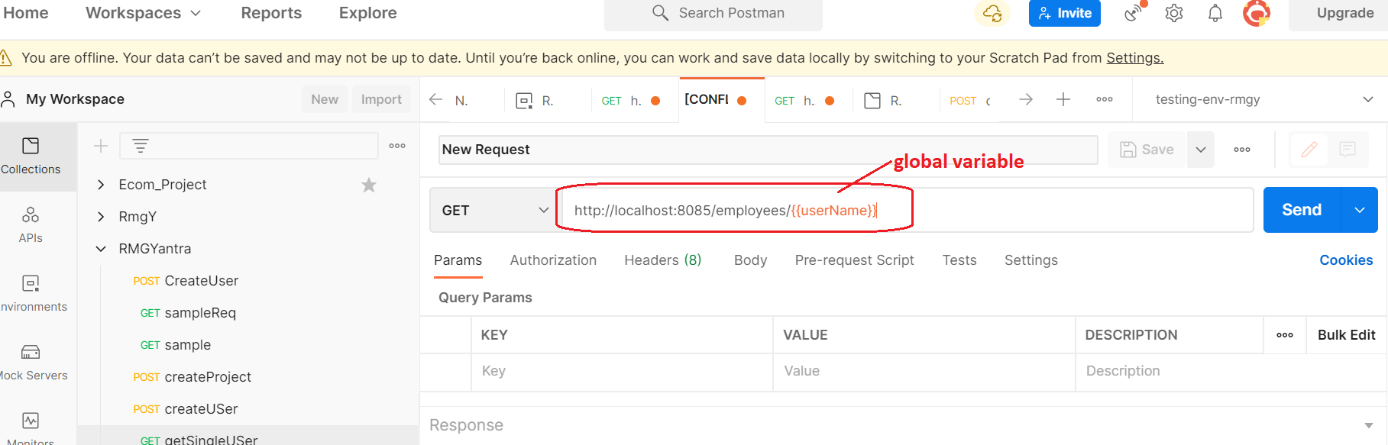
**In below example, capture “id” value from the first api request & pass same data in another URL within the same api execution**



EG : 2 🡺Send data from one request to Another request



Send “userName” data to another request



1. **What Is Collection in PostMan**

Collection of API requests is called collection or suite

1. **How to execute API collection in command line \*\*\*\*\*\*\*\*\*\*\*\*\***
2. Export Collection from POSTMAN
3. Download Collection in .json format
4. Go to Commandline
5. Install newMAN

cmd> npm install –g new man

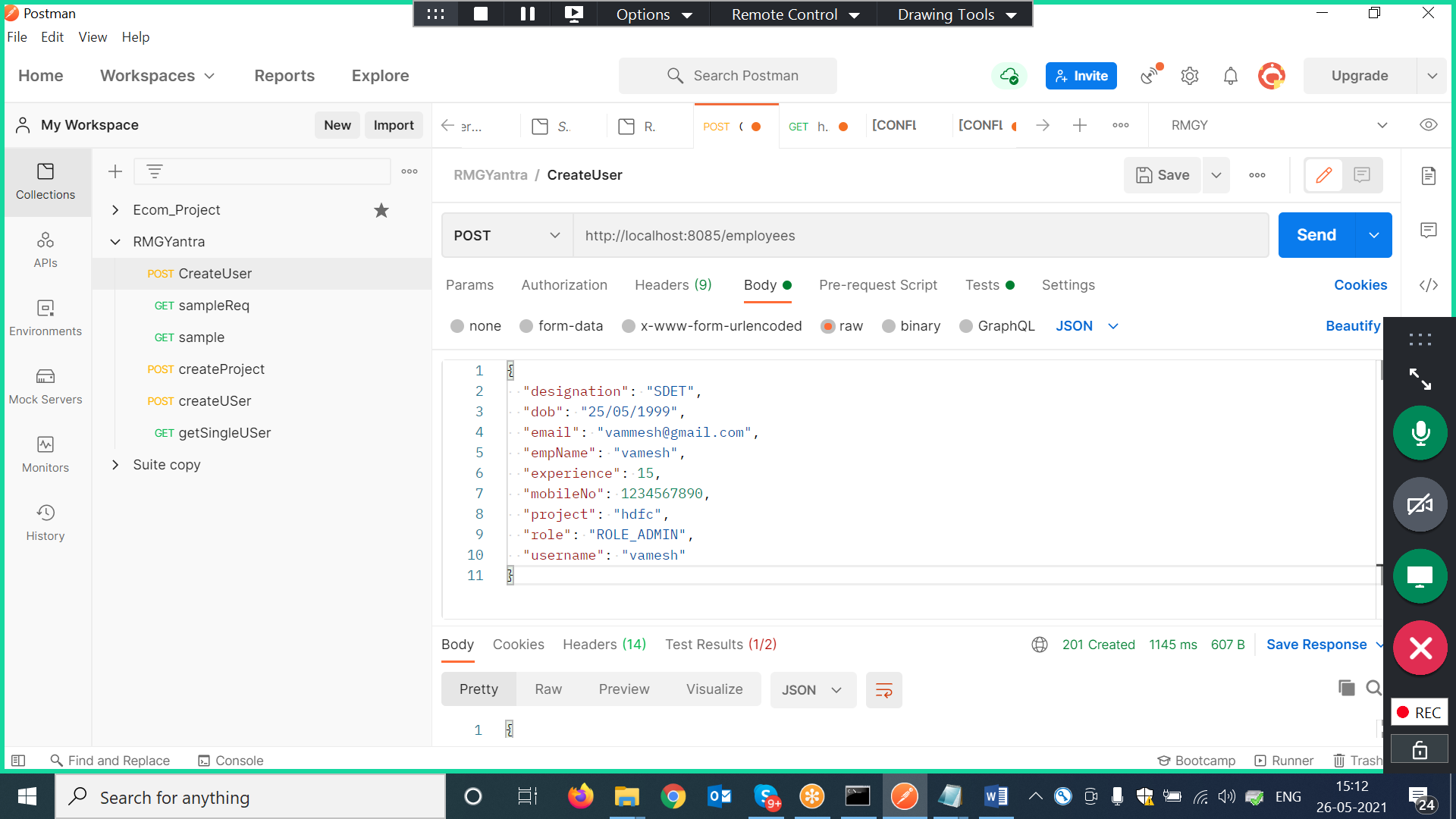
1. Execute collction in CMD

Cmd> newman run path

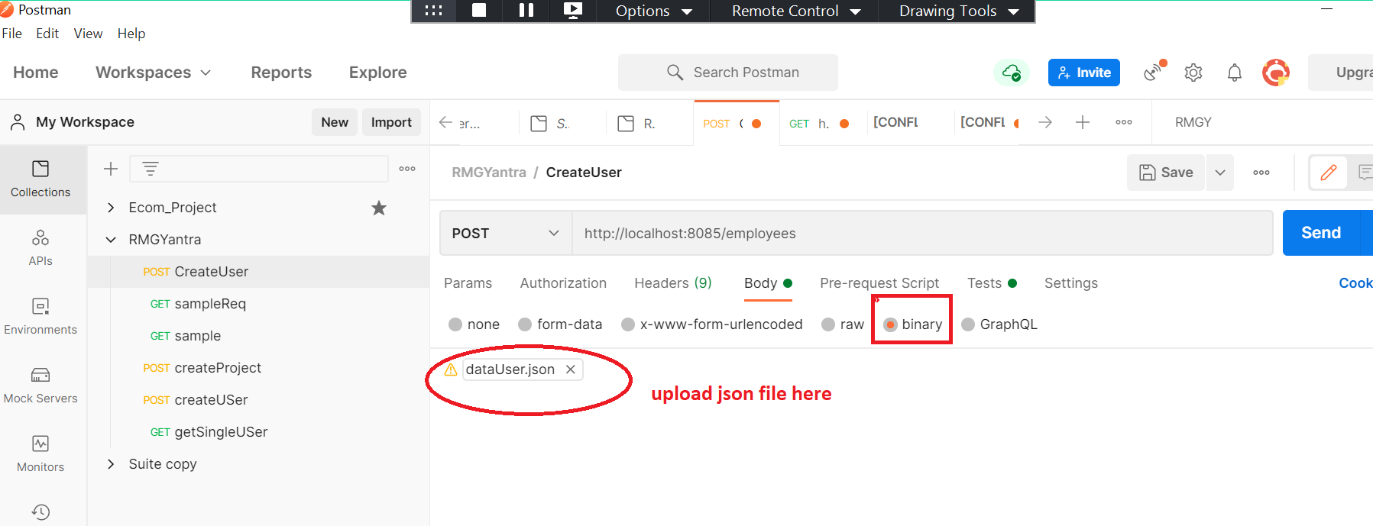
1. **How many ways we can post request in JSon in PostMan**

**In postman we can post request in 3 ways**

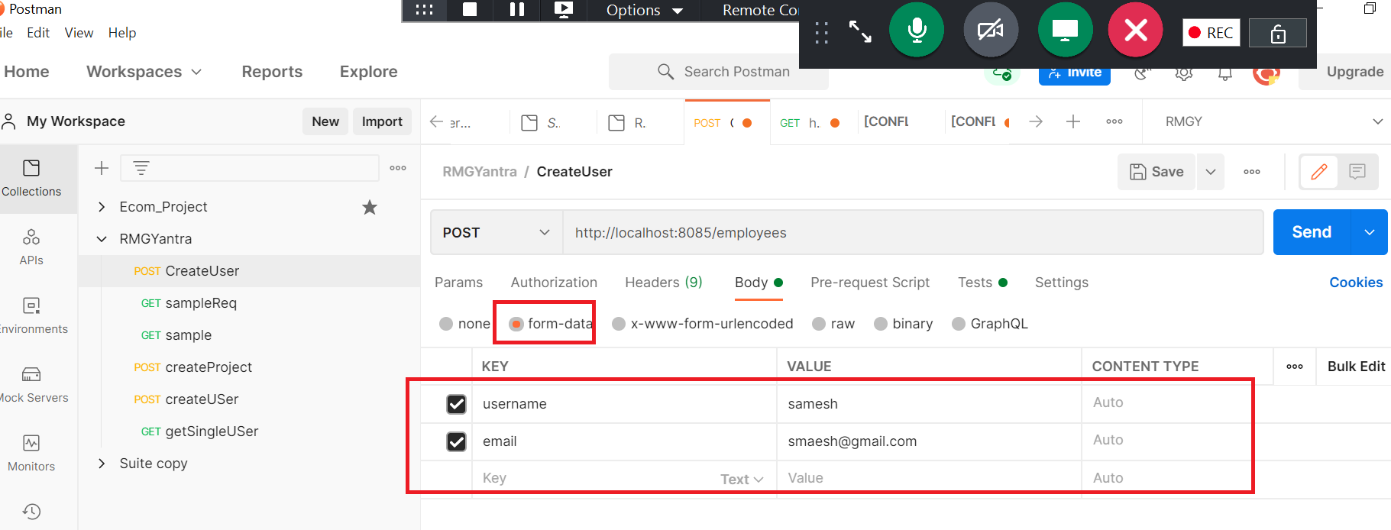
1. **Post a request using jsonObject**



1. **Post a request using Json File**

****

1. **Post a request in the form of hashmap**

****

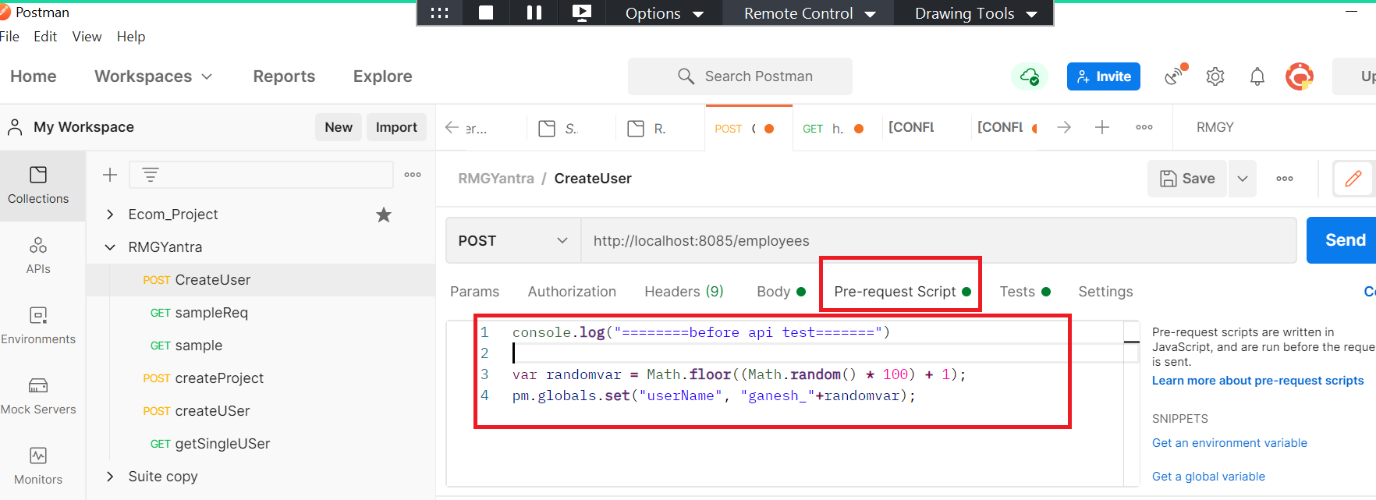
**36 How to execute same request with dynamic request body\*\*\*\*\*\*\*\*\***

We can post same request with dynamic data without any changes in the request body,

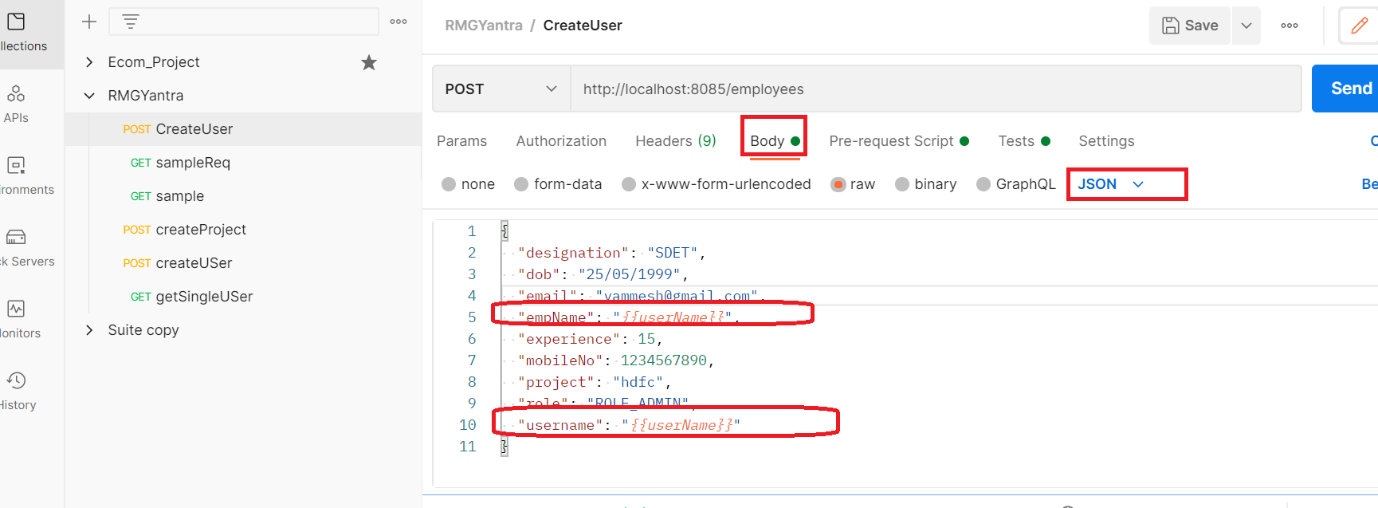
in order the achieve this fallow below steps

1. should write random math function in “pre-request-script” , then store the random data in global variable

Note : pre-request script will be executed before the sending the api request



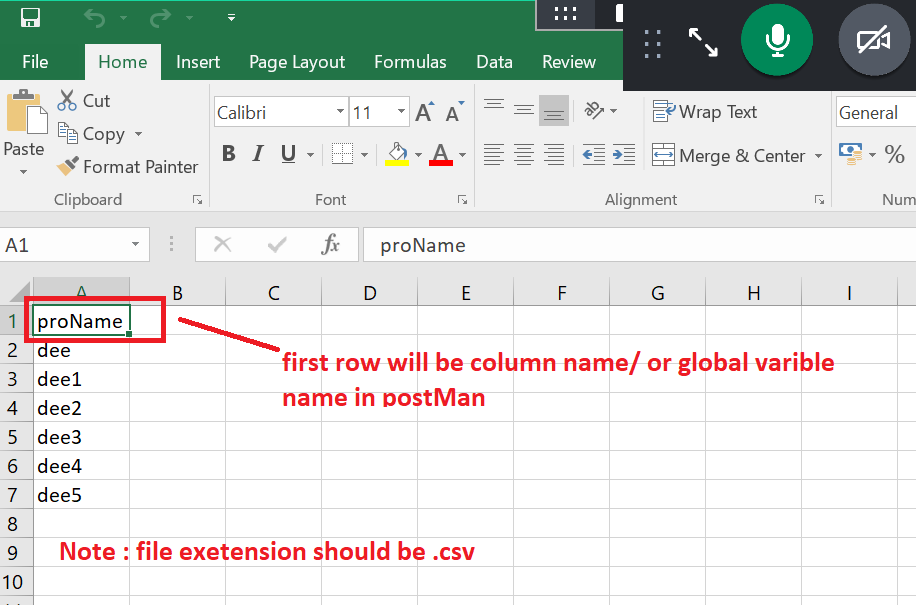
1. use the same random variable in request body



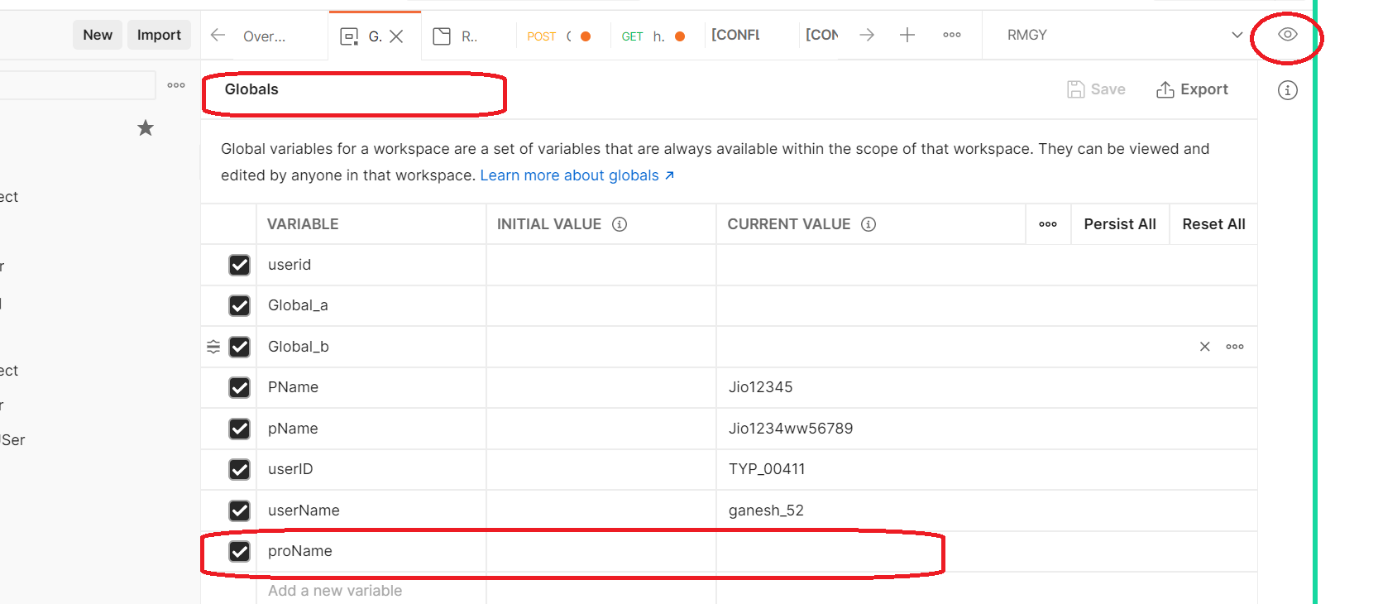
1. **How to execute same request multiple time with different data \*\*\*\*\*\*\*\*\***

**We can execute same api with multiple data using .csv file , to achieve fallow the below steps**

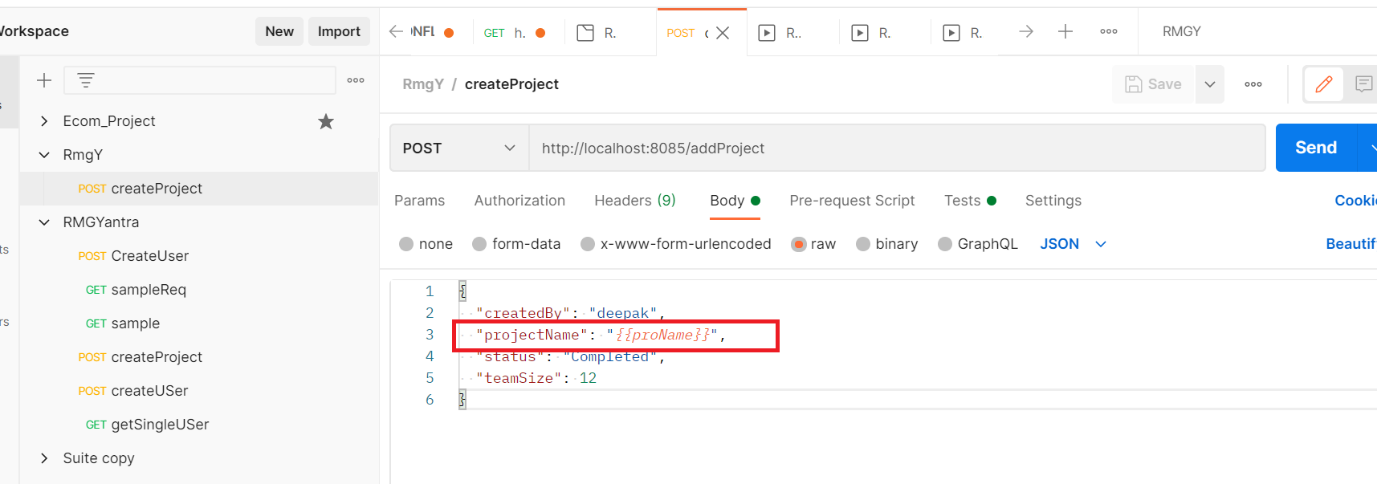
1. **Create .csv file with column name fallowed by data**

****

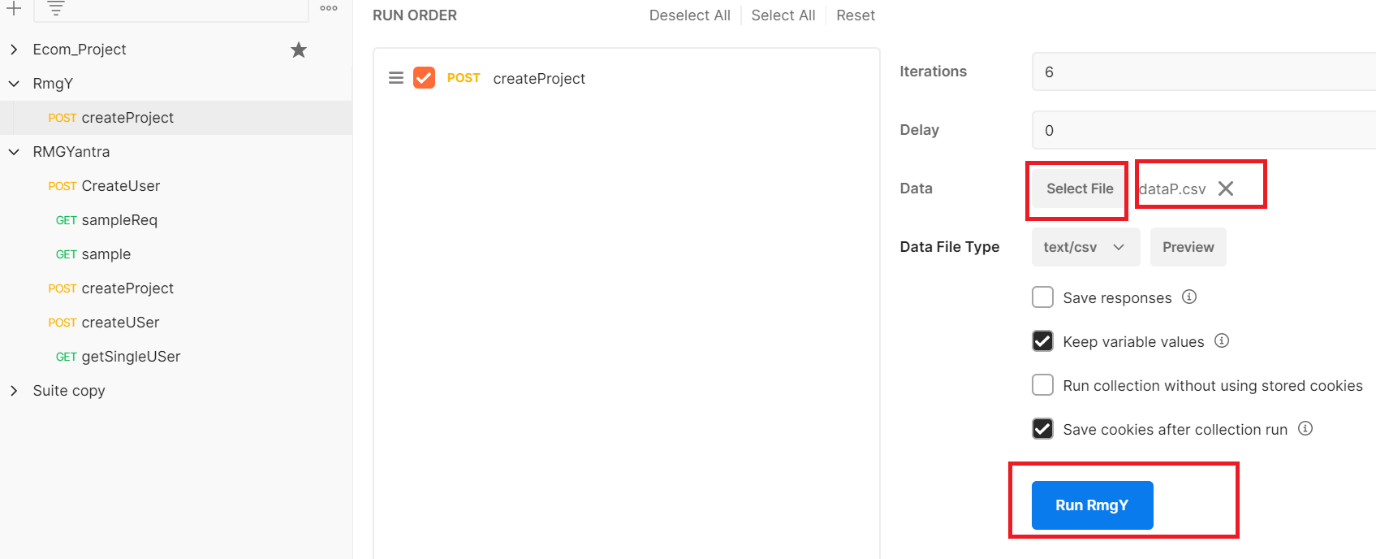
1. **Create on one global variable in postName , name should be same as column name in .csv file**

****

1. **Set the global variable in request body**

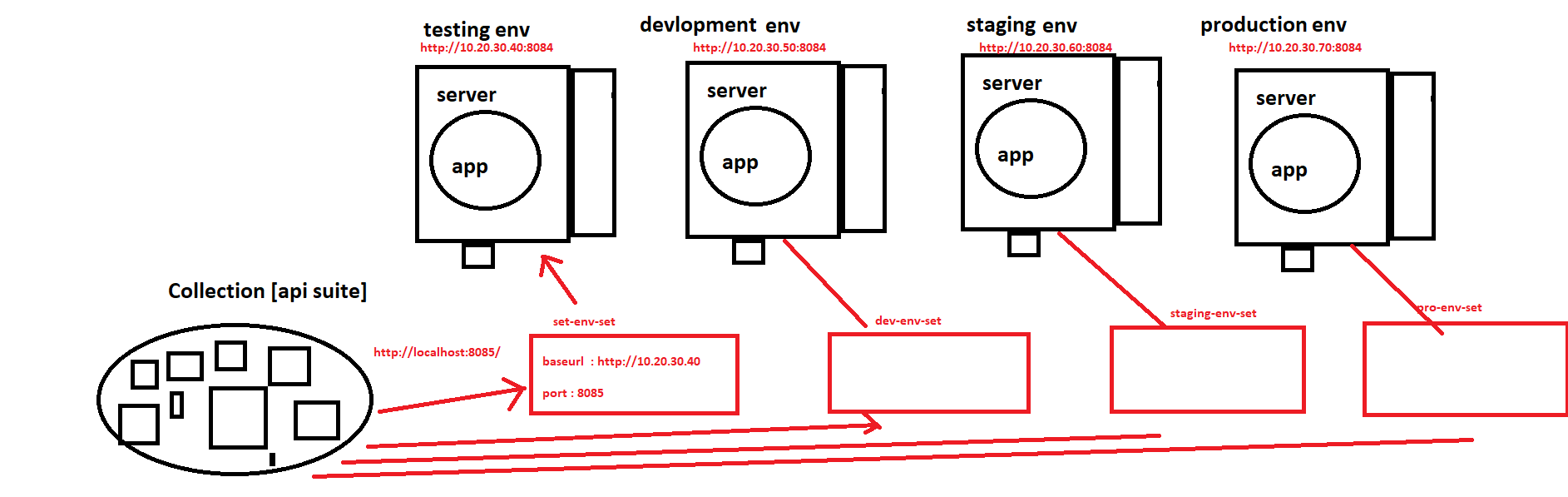
****

1. **Run the collection with .csv file**

****

**38. how to execute API collection with different Environment like testing env , development, staging , production**

In order to run same collection with different environment, take a help environment variable available in postman



EG : API URI :

Get🡺 {{baseuri}}/{{port}}/endpoint

**39. Explain the variables available in postman**

1. local variable: (scope: it’s a data within the api request) can be used with in a request

2. Global variable: (scope: it’s a data across the collection) can be used between the api request, outside the collection also

3. Environment variable: (scope with in the collection) it’s a configuration data to run the collection

39. What is the difference between Oauth1.0 Oauth2.0\*\*\*\*\*\*

|  |  |
| --- | --- |
| Oauth1.0 | Oauth2.0 |
| older version of protocol for authentication | latest version of protocol for authentication |
| Two level authentication required for every api | One level authentication required for every api |
| complex authentication approach | Simple authentication approach compare to Oath1.0 |
| To get bearer token , should pass consumerID ,  consumer secret & accessID , access secret | To get bearer token , should pass Client & Client Secret |