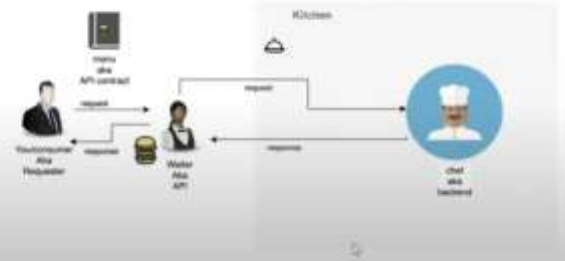
API Testing

Introduction

API - Restaurant analogy



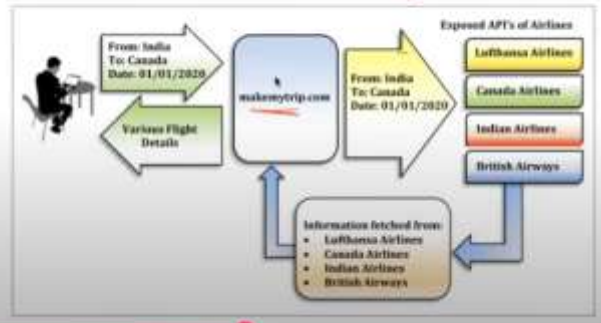
15 / 1:28:05 Bilingual

What is Client & Server?

A client is a computer hardware device or software that accesses a service made available by a server. The server is often (but not always) located on a separate physical computer.

A server is a physical computer dedicated to run services to serve the needs of other computers. Depending on the service that is running, it could be a file server, database server, home media server, print server, or web server.

Makemytrip example:



Client/Server Architecture

Types Of API

There are two types of API's.

1. Simple Object Access Protocol (SOAP)
2. REST (Representational State Transfer).

Both are the web services.

API Vs Webservice

- Web Service is an API wrapped in HTTP.
- All Web Services are API but APIs are not Web Services.
- A Web Service needs a network while an API doesn't need a network for its operation.

25 / 1:28:05 Bilingual

What is an API?

Application Program Interface (API): is the way of communication between two applications where applications may differ in their platforms or in terms of technology.

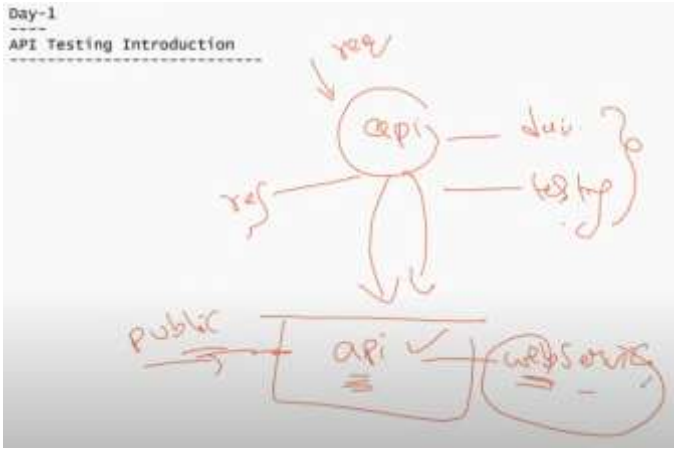
** API is a way of communication between two layers.

Day-1

API Testing Introduction

Day-1

API Testing Introduction



Feature & Resource

'Feature' is the term used in manual testing to test some functionality and similarly 'Resource' is the term used in API Automation testing referring some functionality.

Payload

payload means **body in the HTTP request and response message**

- Request Payload
- Response Payload

Rest API HTTP Methods

(HTTP REQUEST)
(get, post, put, patch, delete)

(VALIDATIONS PERFORMED)
(Response Status code, Data etc.)

REST API Methods

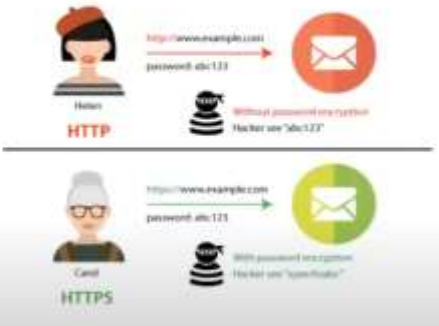
Terminologies

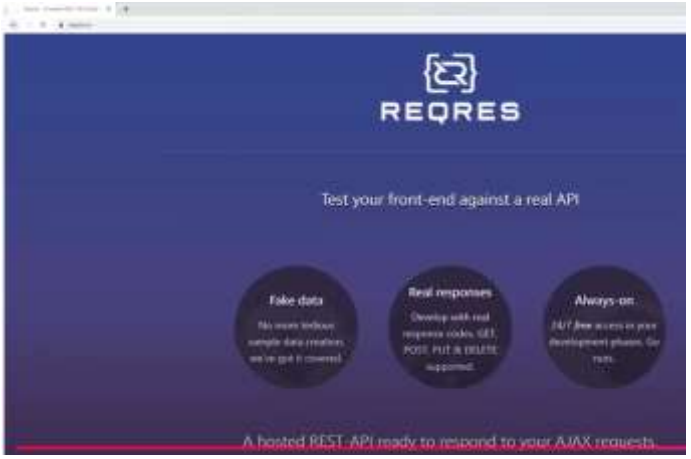
URI – Uniform Resource Identifier

URL – Uniform Resource Locator

URN – Uniform Resource Name

http Vs https



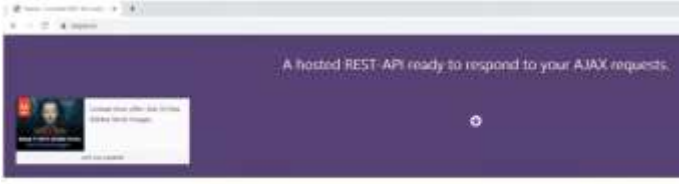


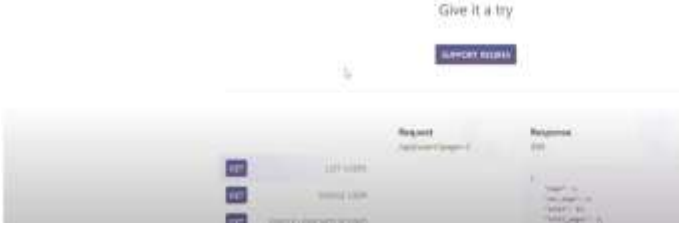
Terminologies

URI – Uniform Resource Identifier

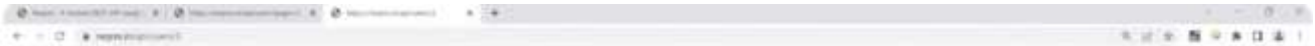
URL – Uniform Resource Locator

URN – Uniform Resource Name





```
Day-1
----
API Testing Introduction
-------------------------


Rest API methods / http request
-----------
get
post
put
delete
--


--

http://www.google.com/

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



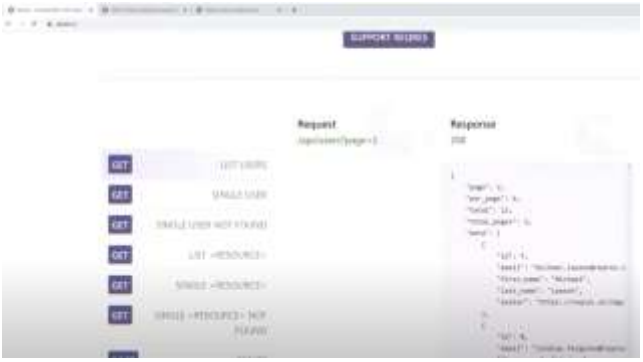

```
{"data":{"id":4,"email":"emma.wong@reqres.in","first_name":"Emma","last_name":"Wong","avatar":"https://reqres.in/img/faces/3-image.jpg"},"support":{"url":"https://reqres.in/#support-
heading","text":"To keep ReqRes free, contributions towards server costs are appreciated!"}}
```

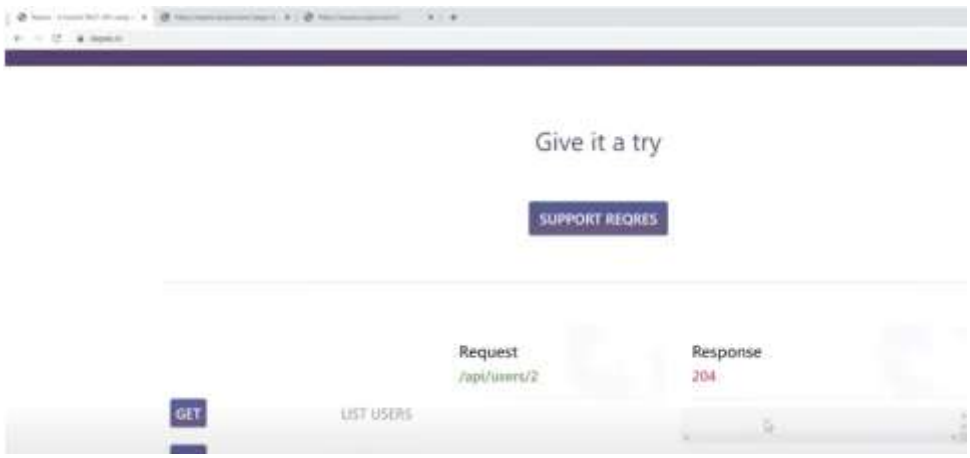


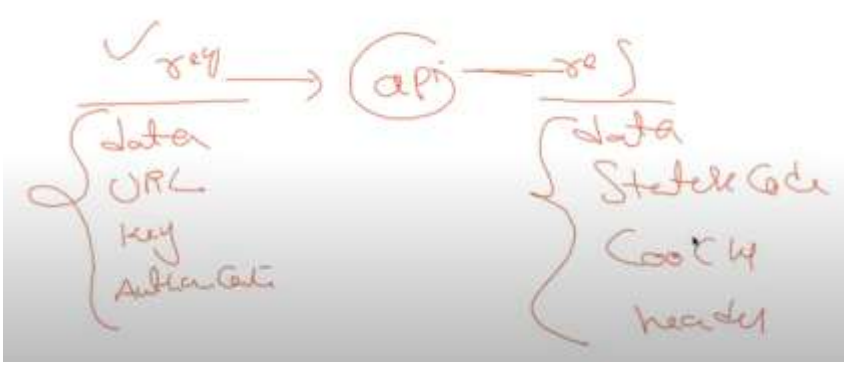
```
{"page":2,"per_page":6,"total":12,"total_pages":2,"data":
[{"id":7,"email":"michael.lawson@reqres.in","first_name":"Michael","last_name":"Lawson","avatar":"https://reqres.in/img/faces/7-image.jpg"},
{"id":8,"email":"lindsay.ferguson@reqres.in","first_name":"Lindsay","last_name":"Ferguson","avatar":"https://reqres.in/img/faces/8-image.jpg"},
{"id":9,"email":"tobias.funke@reqres.in","first_name":"Tobias","last_name":"Funke","avatar":"https://reqres.in/img/faces/9-image.jpg"},
{"id":10,"email":"byron.fields@reqres.in","first_name":"Byron","last_name":"Fields","avatar":"https://reqres.in/img/faces/10-image.jpg"},
{"id":11,"email":"george.edwards@reqres.in","first_name":"George","last_name":"Edwards","avatar":"https://reqres.in/img/faces/11-image.jpg"},
{"id":12,"email":"rachel.howell@reqres.in","first_name":"Rachel","last_name":"Howell","avatar":"https://reqres.in/img/faces/12-image.jpg"}],"support":
{"url":"https://reqres.in/#support-heading","text":"To keep ReqRes free, contributions towards server costs are appreciated!"}}
```



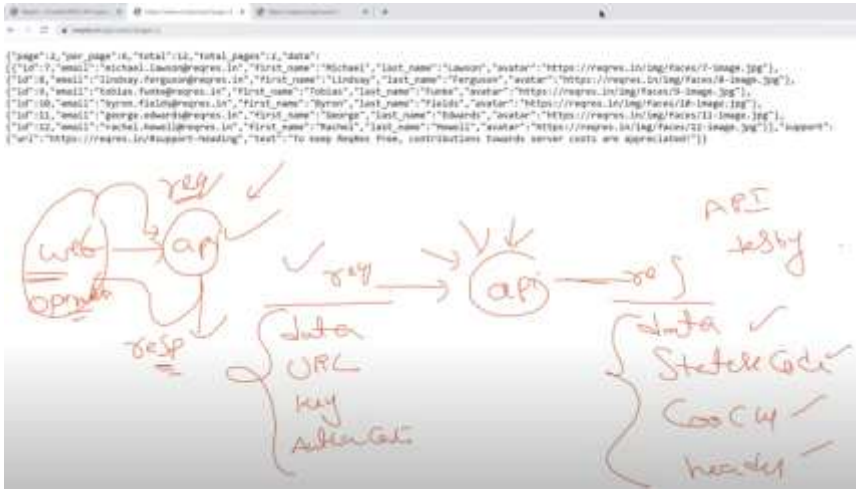
```
{"page":2,"per_page":6,"total":12,"total_pages":2,"data":
[{"id":7,"email":"michael.lawson@reqres.in","first_name":"Michael","last_name":"Lawson","avatar":"https://reqres.in/img/faces/7-image.jpg"},
{"id":8,"email":"lindsay.ferguson@reqres.in","first_name":"Lindsay","last_name":"Ferguson","avatar":"https://reqres.in/img/faces/8-image.jpg"},
{"id":9,"email":"tobias.funke@reqres.in","first_name":"Tobias","last_name":"Funke","avatar":"https://reqres.in/img/faces/9-image.jpg"},
{"id":10,"email":"byron.fields@reqres.in","first_name":"Byron","last_name":"Fields","avatar":"https://reqres.in/img/faces/10-image.jpg"},
{"id":11,"email":"george.edwards@reqres.in","first_name":"George","last_name":"Edwards","avatar":"https://reqres.in/img/faces/11-image.jpg"},
{"id":12,"email":"rachel.howell@reqres.in","first_name":"Rachel","last_name":"Howell","avatar":"https://reqres.in/img/faces/12-image.jpg"}],"support":
{"url":"https://reqres.in/#support-heading","text":"To keep ReqRes free, contributions towards server costs are appreciated!"}}
```

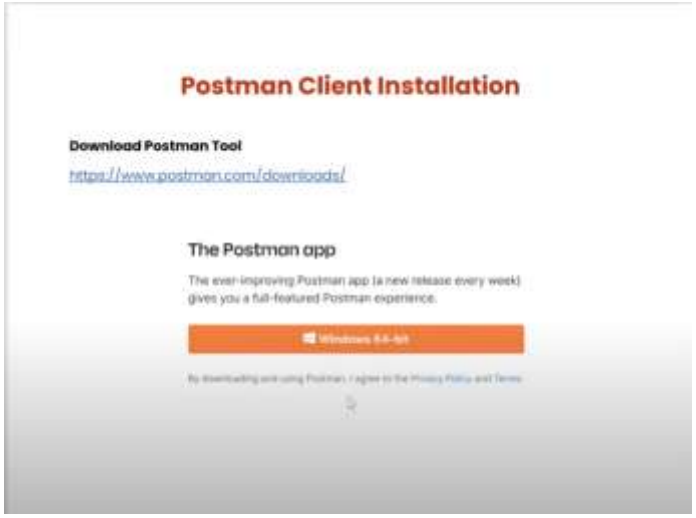



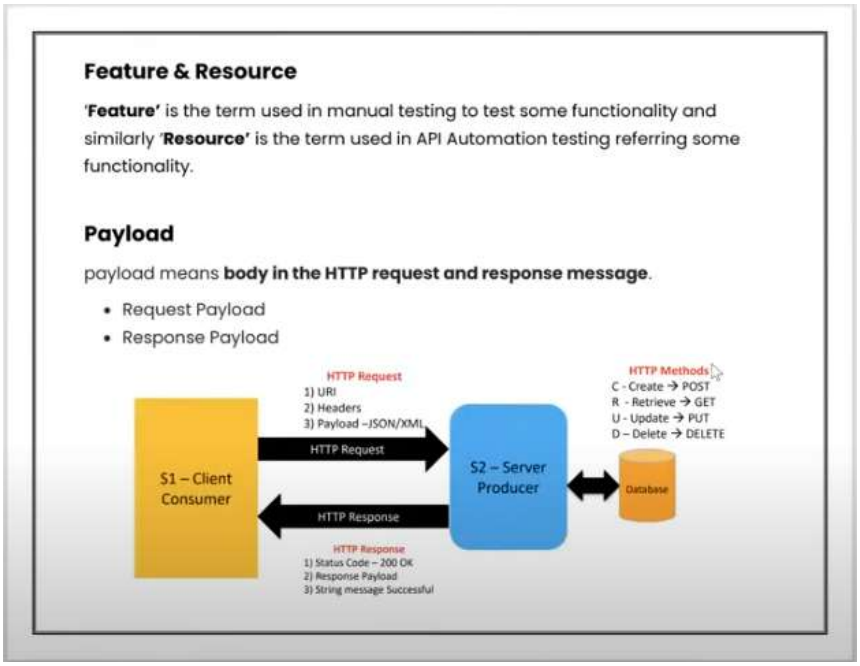


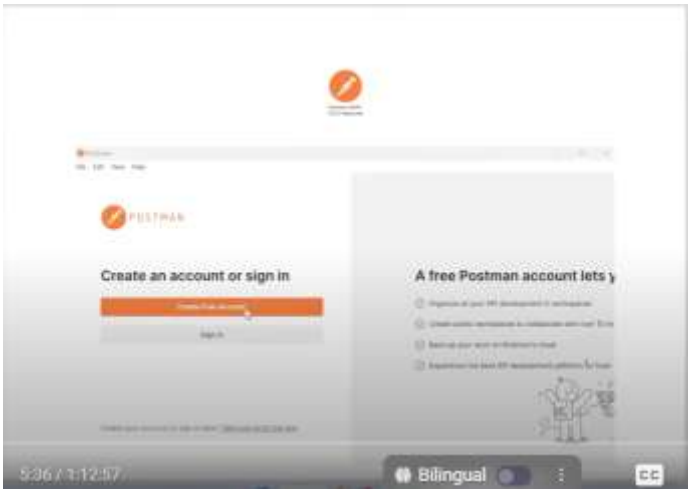

req
api
res
data
URL
key
Authentication
data
Status Code
Cookie
header

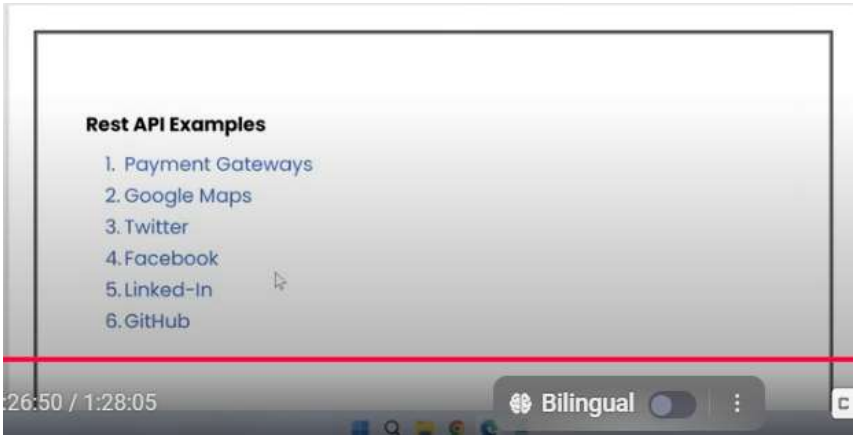

Download Postman
The Postman app


{"page":1,"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"},
{"id":10,"email":"byron.fields@reqres.in","first_name":"Byron","last_name":"Fields","avatar":"https://reqres.in/img/faces/10-image.jpg"},
{"id":11,"email":"george.edwards@reqres.in","first_name":"George","last_name":"Edwards","avatar":"https://reqres.in/img/faces/11-image.jpg"},
{"id":12,"email":"rachel.howell@reqres.in","first_name":"Rachel","last_name":"Howell","avatar":"https://reqres.in/img/faces/12-image.jpg"}],"support":
{"url":"https://reqres.in/#support-heading","text":"To keep ReqRes free, contributions towards server costs are appreciated!"}}
web
app
API
req
res
data
URL
key
Authentication
data
Status Code
Cookie
header

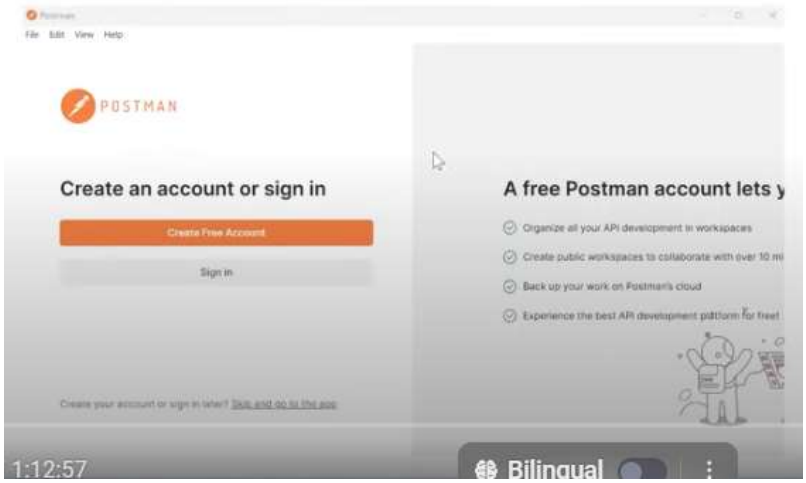

Postman Client Installation
Download Postman Tool
https://www.postman.com/downloads/
The Postman app
The ever-improving Postman app (a new release every week)
gives you a full-featured Postman experience.

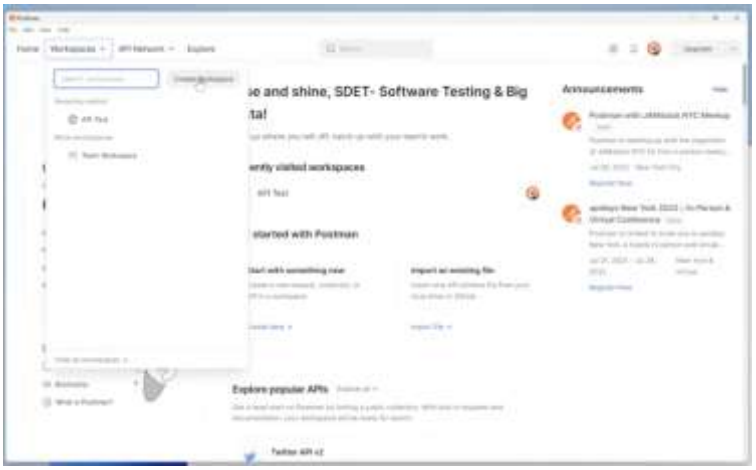

Feature & Resource
'Feature' is the term used in manual testing to test some functionality and similarly 'Resource' is the term used in API Automation testing referring some functionality.
Payload
payload means body in the HTTP request and response message.
Request Payload
Response Payload
HTTP Request
1) URI
2) Headers
3) Payload –JSON/XML
HTTP Request
S1 – Client Consumer
S2 – Server Producer
Database
HTTP Response
HTTP Response
1) Status Code – 200 OK
2) Response Payload
3) String message Successful
HTTP Methods
C - Create → POST
R - Retrieve → GET
U - Update → PUT
D – Delete → DELETE

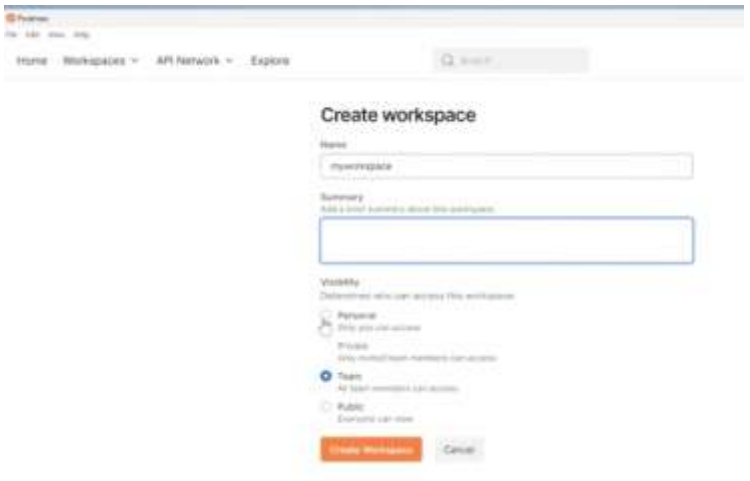

Create an account or sign in
A free Postman account lets
5:36 / 1:12:57
Bilingual

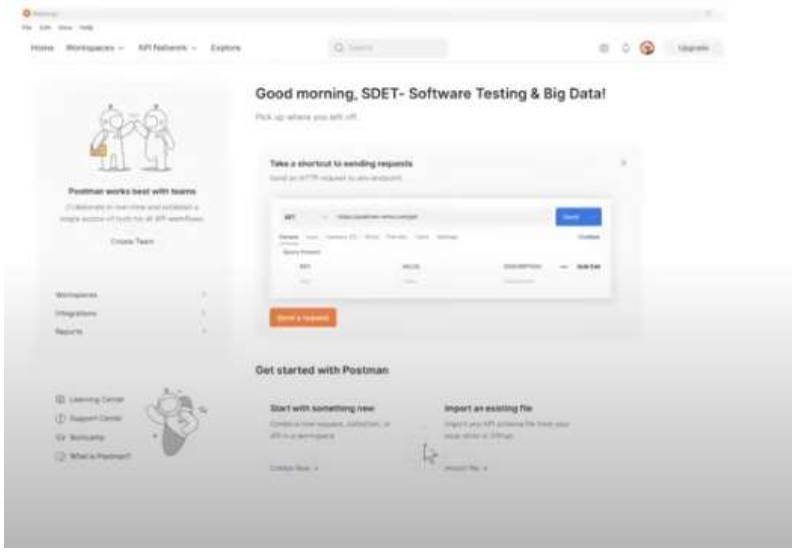

Rest API Examples
1. Payment Gateways
2. Google Maps
3. Twitter
4. Facebook
5. Linked-In
6. GitHub
:26:50 / 1:28:05
Bilingual

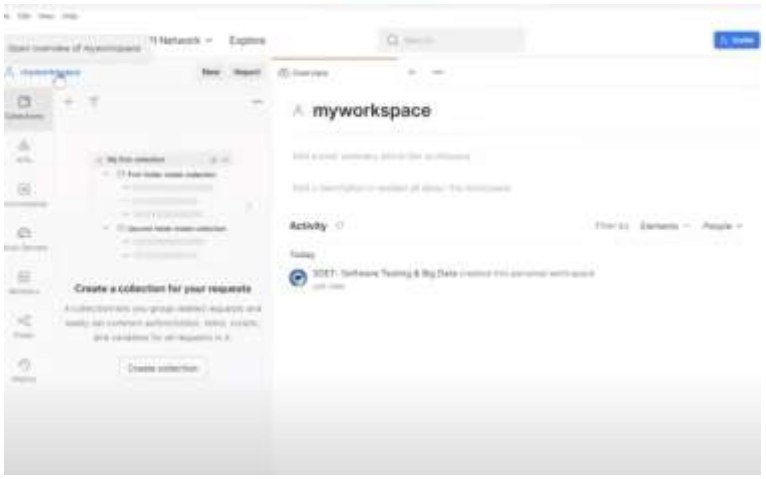

Bilingual

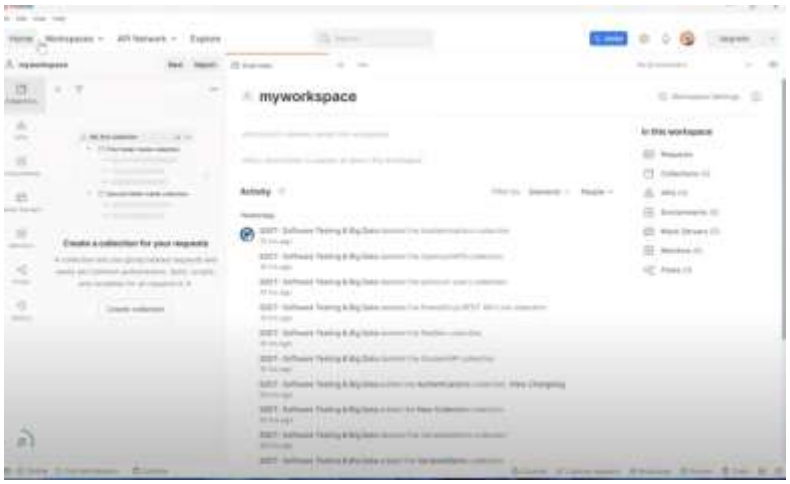

POSTMAN
Create an account or sign in
Create Free Account
Sign in
A free Postman account lets y
Organize all your API development in workspaces
Create public workspaces to collaborate with over 10 mi
Back up your work on Postman's cloud
Experience the best API development platform for free!
1:12:57
Bilingual

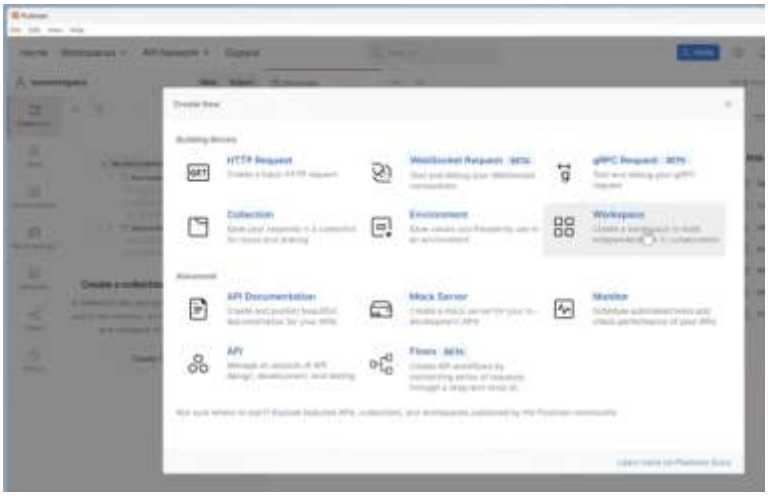

e and shine, SDET- Software Testing & Big
ta!
Announcements
Explore popular APIs

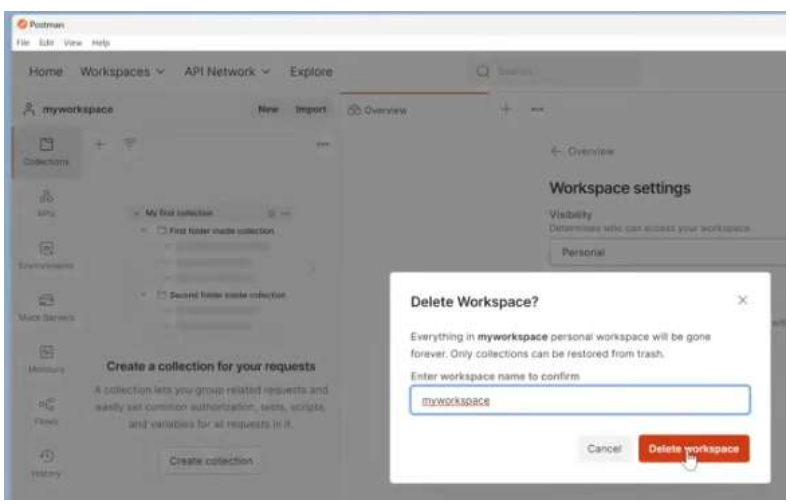

Home
Workspaces
API Network
Explore
Create workspace
myworkspace
Summary
Visibility
Personal
Team
Public
Cancel

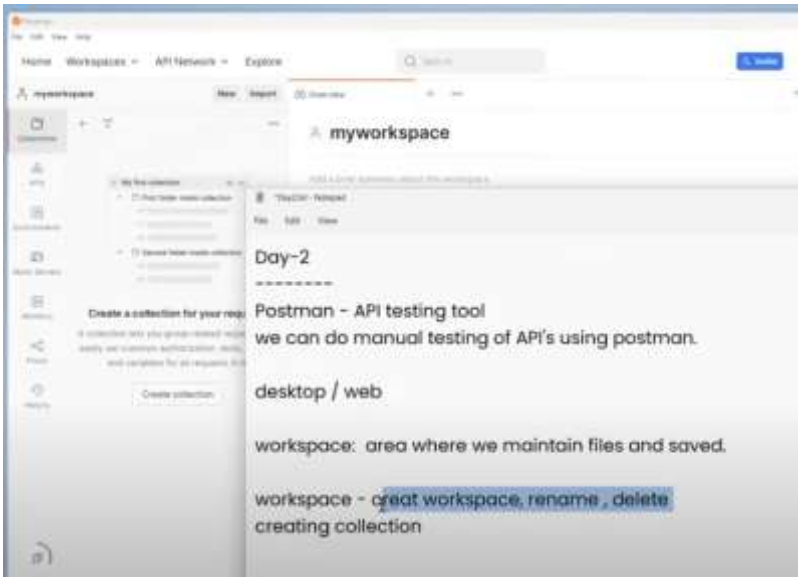

Good morning, SDET- Software Testing & Big Data!
Take a shortcut to sending requests
Postman works best with teams
Get started with Postman
Start with something new
Import an existing file

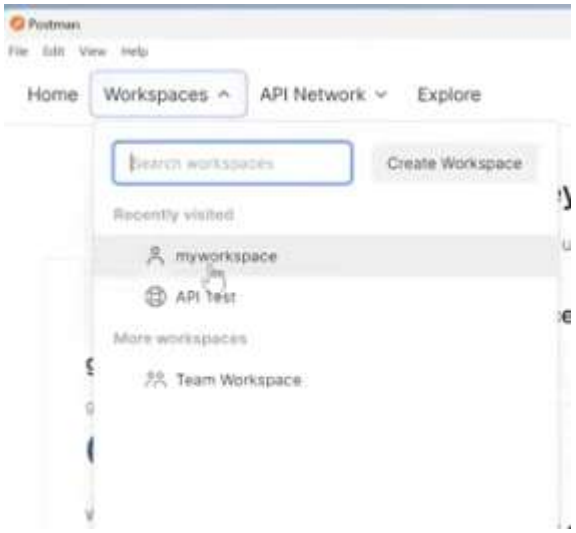

myworkspace
Activity
Create a collection for your requests
Create collection

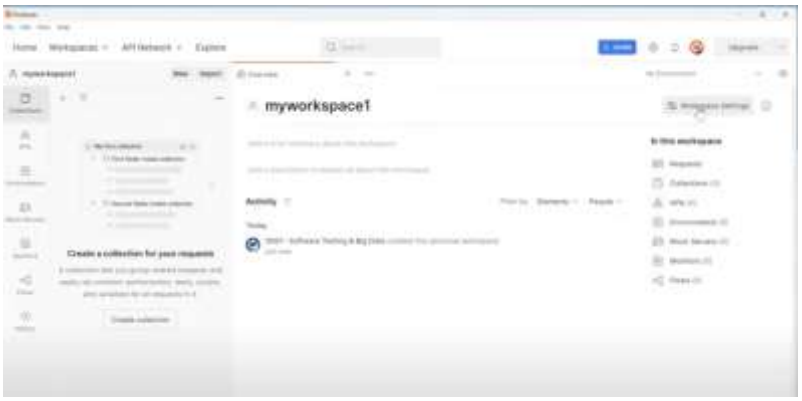

myworkspace
Activity
Create a collection for your requests
Create collection

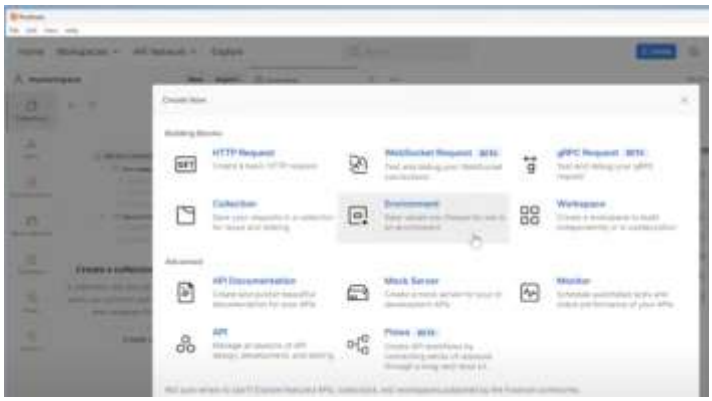

HTTP Request
WebSocket Request
gRPC Request
Collection
Environment
Workspace
API Documentation
Mock Server
Monitor
API
Flows

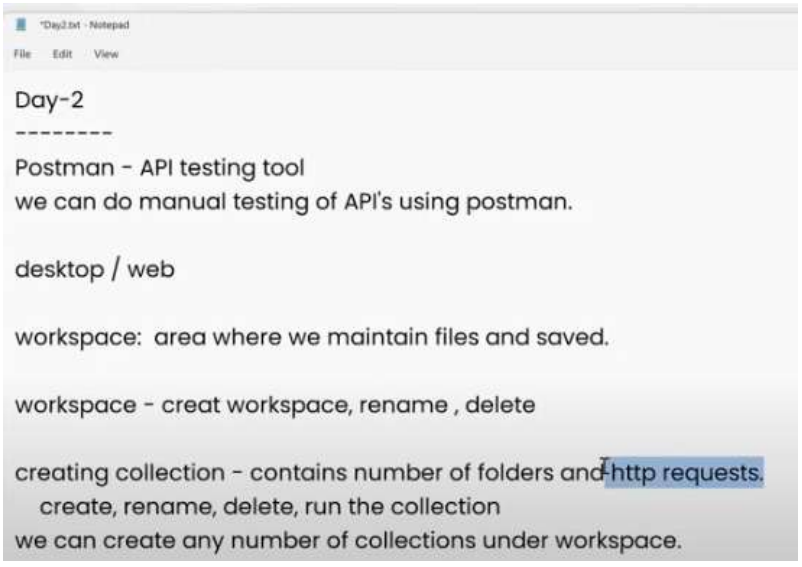

Home
Workspaces
API Network
Explore
myworkspace
New
Import
Overview
Workspace settings
Visibility
Personal
Create a collection for your requests
A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.
Create collection
Delete Workspace?
Everything in myworkspace personal workspace will be gone forever. Only collections can be restored from trash.
Enter workspace name to confirm
myworkspace
Cancel
Delete workspace


myworkspace
Day-2
Postman - API testing tool
we can do manual testing of API's using postman.
desktop / web
workspace: area where we maintain files and saved.
workspace - creat workspace, rename , delete
creating collection

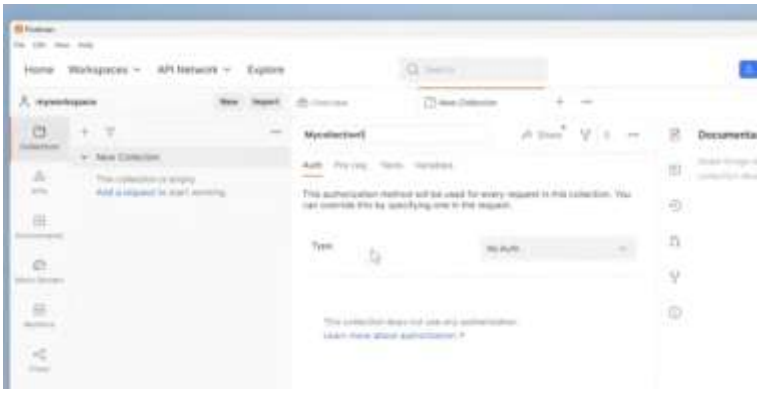

Postman
File Edit View Help
Home
Workspaces
API Network
Explore
Search workspaces
Create Workspace
Recently visited
myworkspace
API Test
More workspaces
Team Workspace

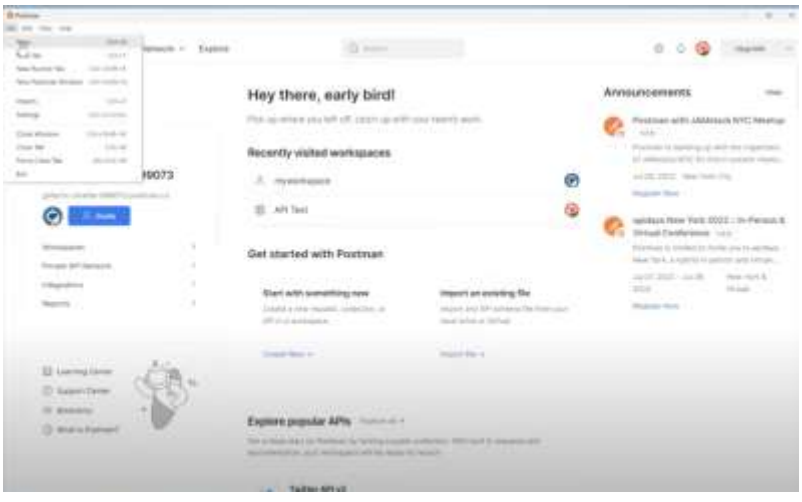

myworkspace1

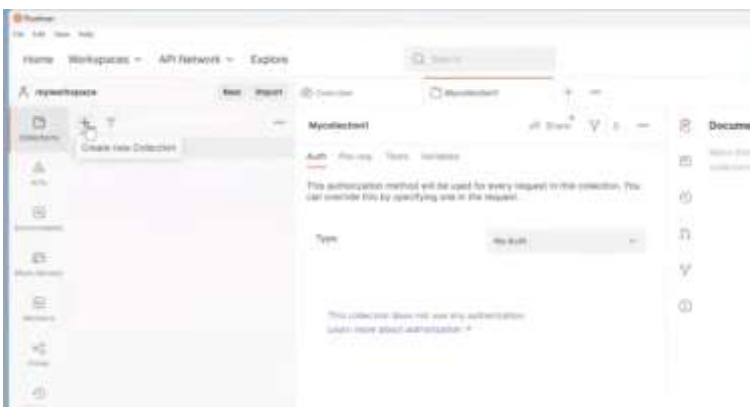


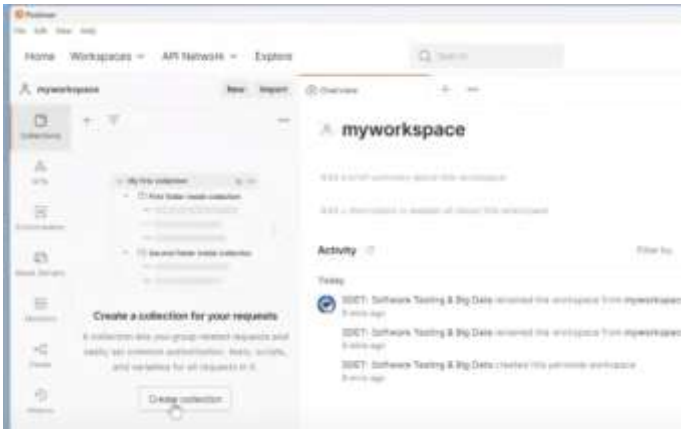

*Day2.txt - Notepad
File Edit View
Day-2
Postman - API testing tool
we can do manual testing of API's using postman.
desktop / web
workspace: area where we maintain files and saved.
workspace - creat workspace, rename , delete
creating collection - contains number of folders and http requests.
create, rename, delete, run the collection
we can create any number of collections under workspace.

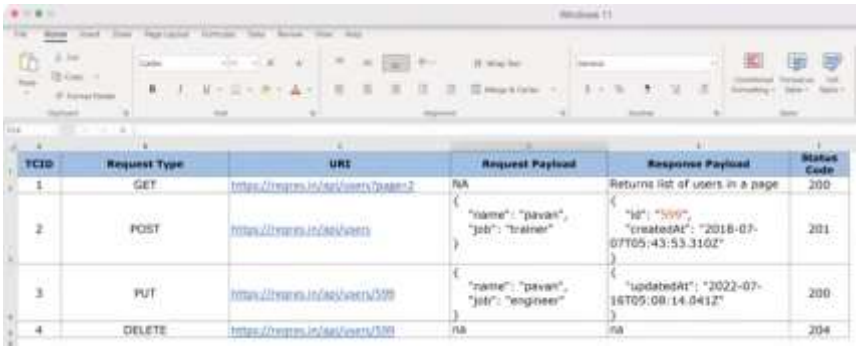

myworkspace


Home
Workspaces
API Network
Explore

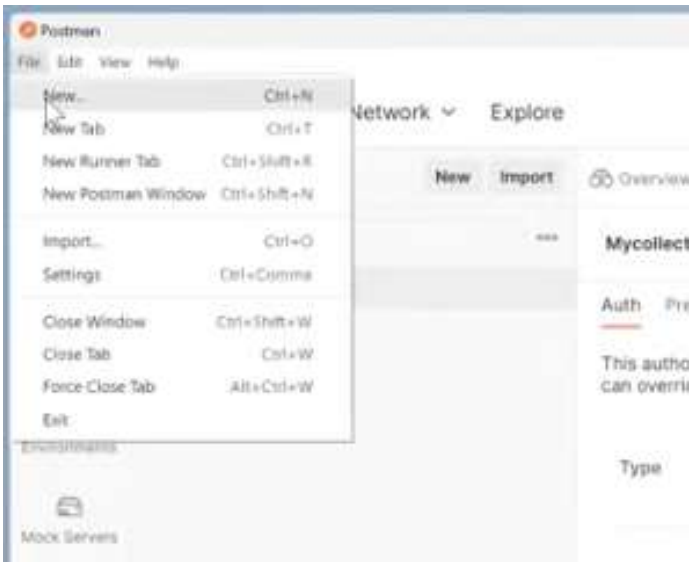

Hey there, early bird!
Announcements
Recently visited workspaces
myworkspace
API Test
Get started with Postman
Explore popular APIs

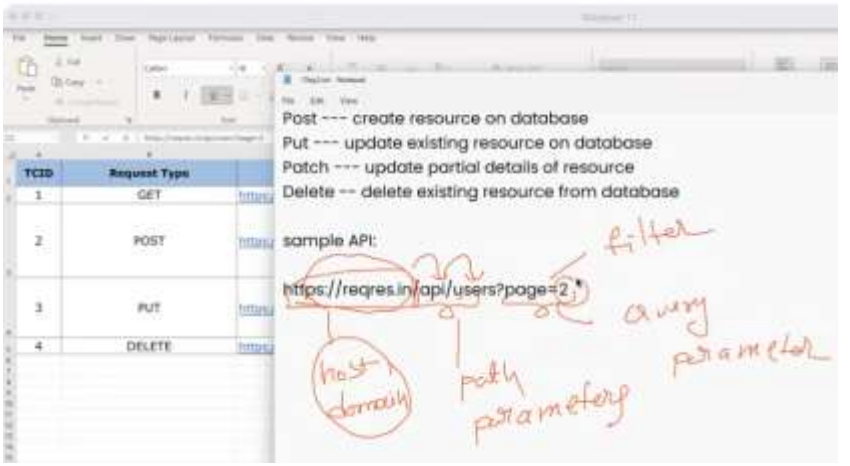

Home
Workspaces
API Network
Explore

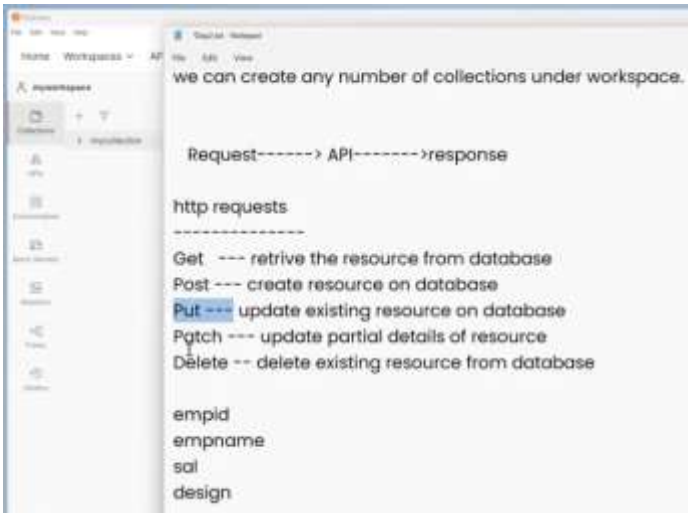


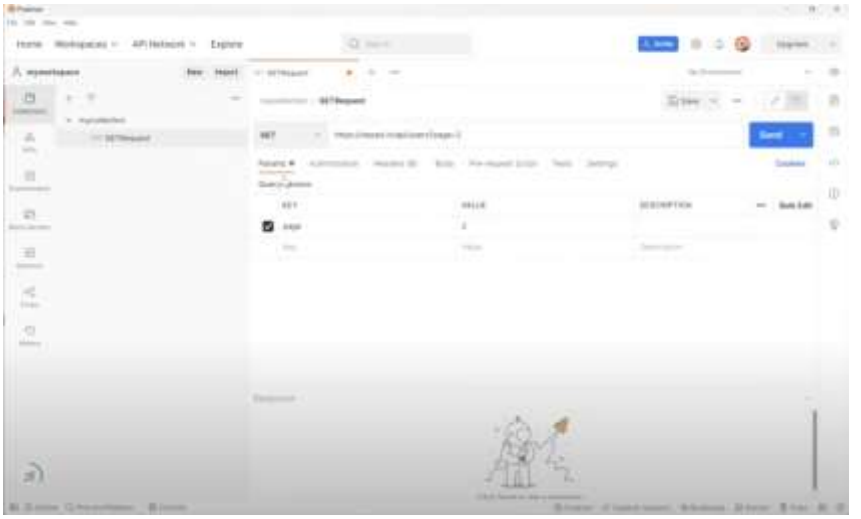


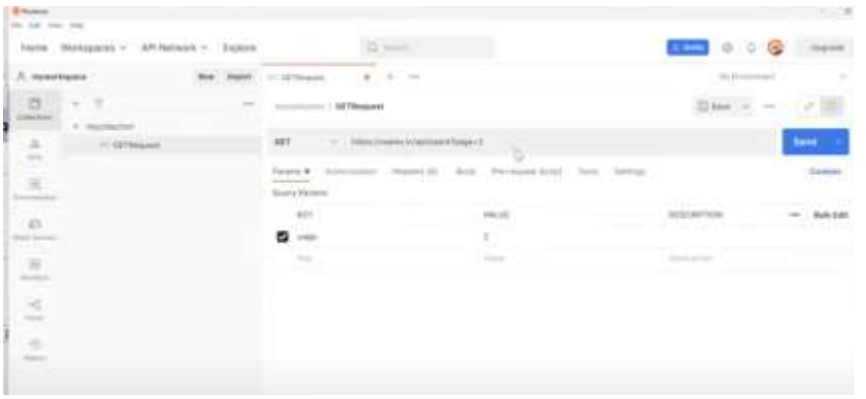
TCID	Request Type	URI	Request Payload	Response Payload	Status Code
1	GET	https://reqres.in/api/users?page=2	NA	Returns list of users in a page	200
2	POST	https://reqres.in/api/users	{ "name": "pavan", "job": "trainer" }	{ "id": "599", "createdAt": "2018-07-07T05:43:53.310Z" }	201
3	PUT	https://reqres.in/api/users/599	{ "name": "pavan", "job": "engineer" }	{ "updatedAt": "2022-07-16T05:08:14.041Z" }	200
4	DELETE	https://reqres.in/api/users/599	na	na	204

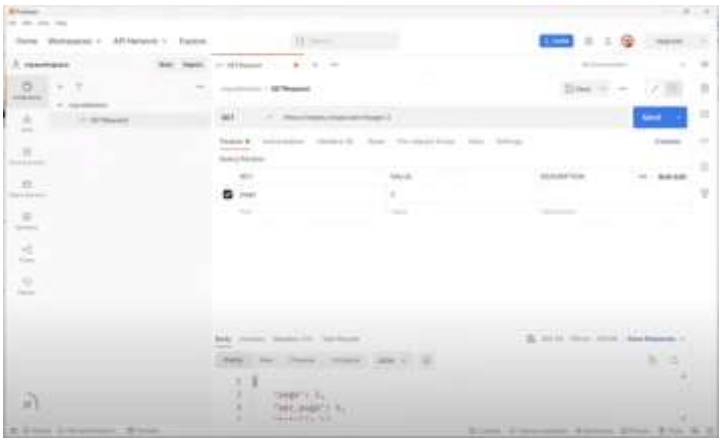


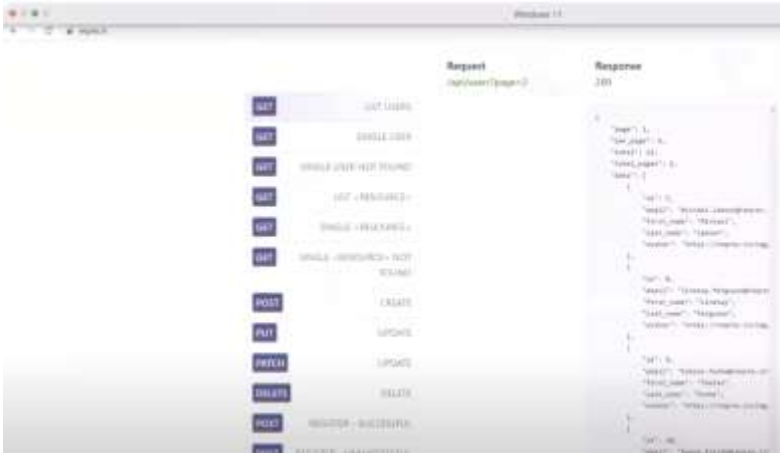




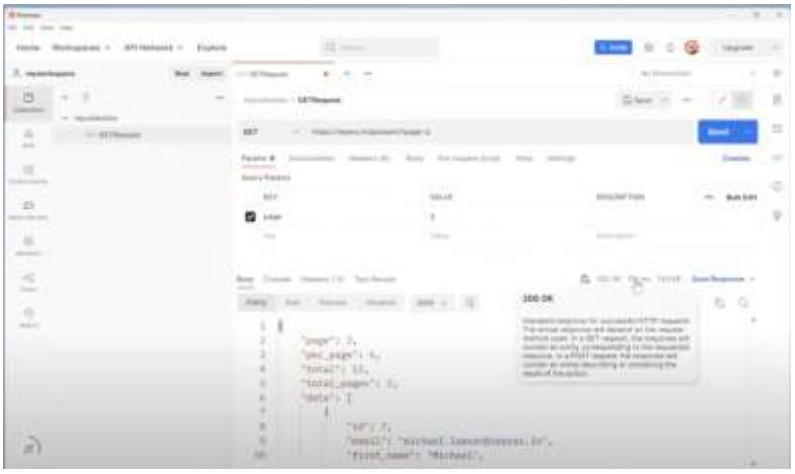


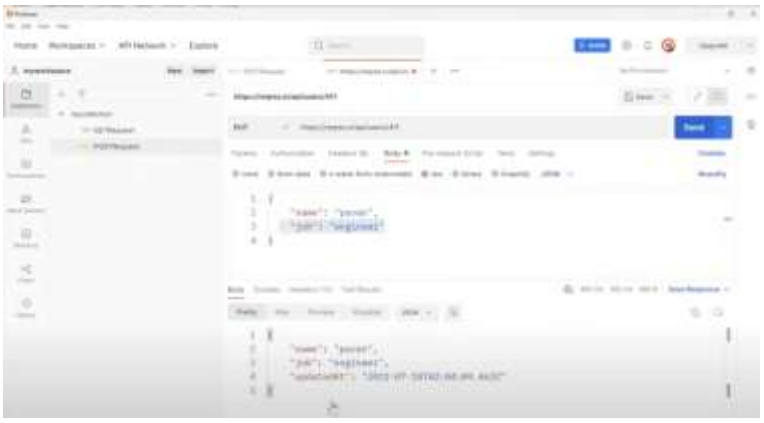


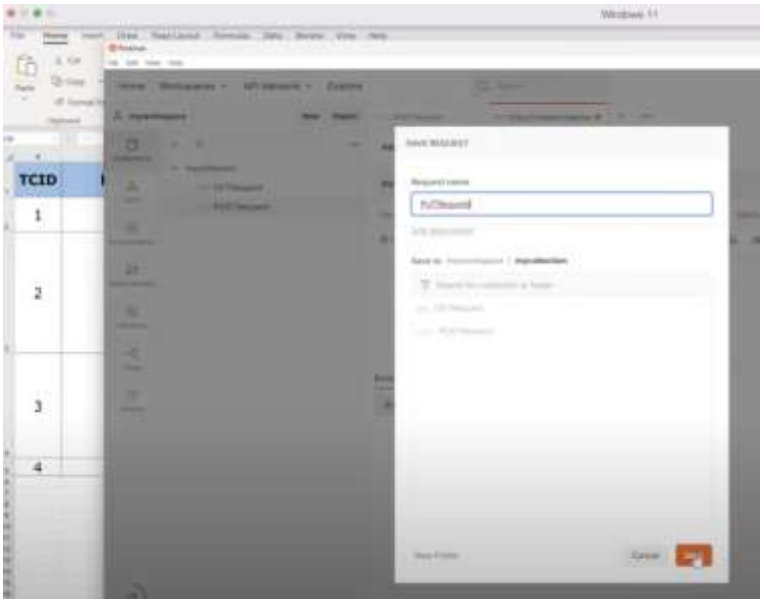


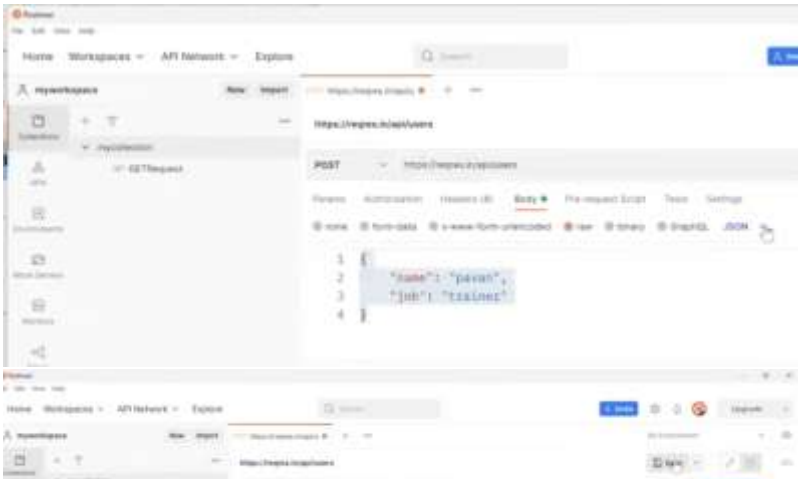


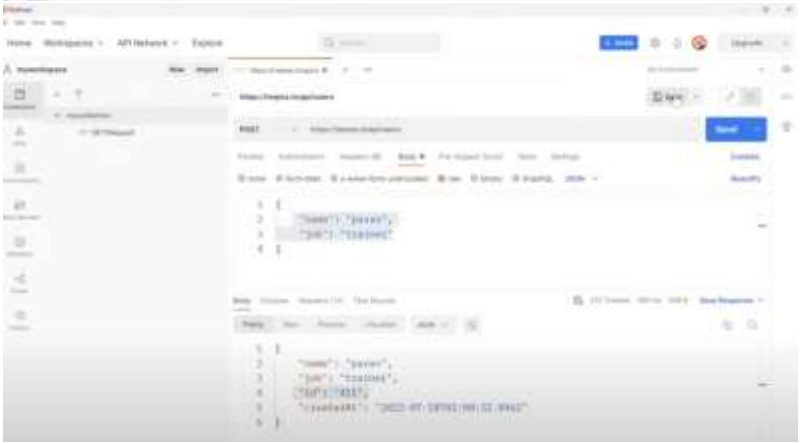
TCID	Request Type	URI	Request Payload	Response Payload	Status Code
1	GET		NA	Returns list of users in a page	200
2	POST		{ "name": "pavan", "job": "trainer" }	{ "name": "pavan", "job": "trainer", "id": "411", "createdAt": "2022-07-18T02:00:32.846Z" }	201
3	PUT	https://reqres.in/api/users/599	{ "name": "pavan", "job": "engineer" }	{ "updatedAt": "2022-07-16T05:08:14.041Z" }	200
4	DELETE	https://reqres.in/api/users/599	na	na	204

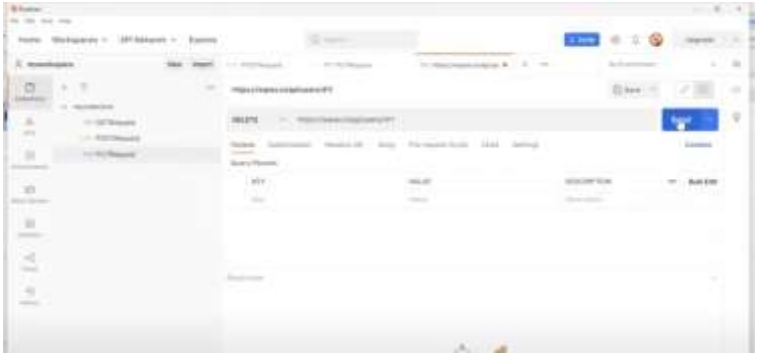


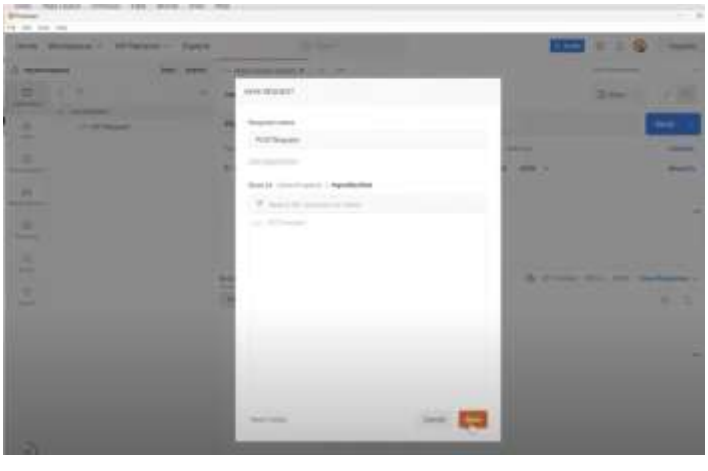


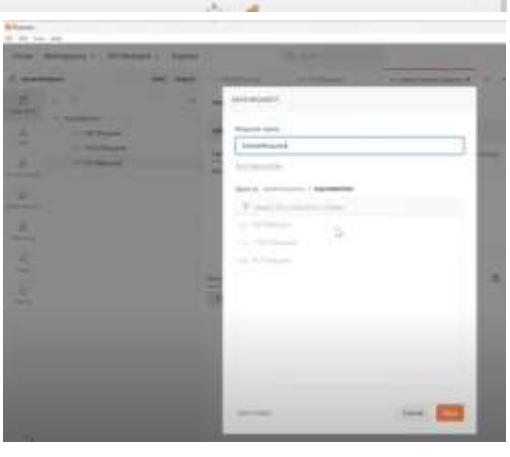


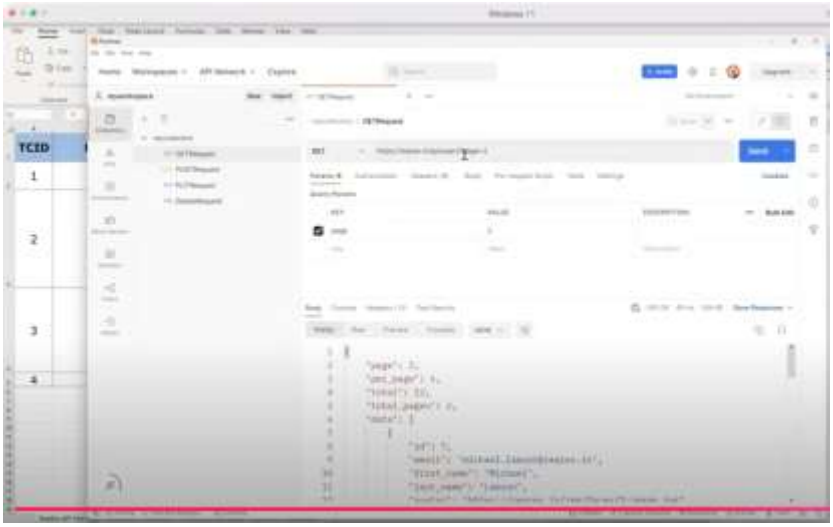


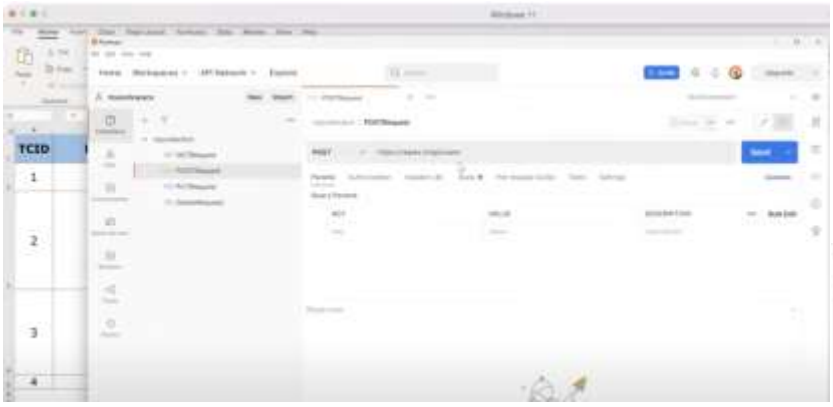


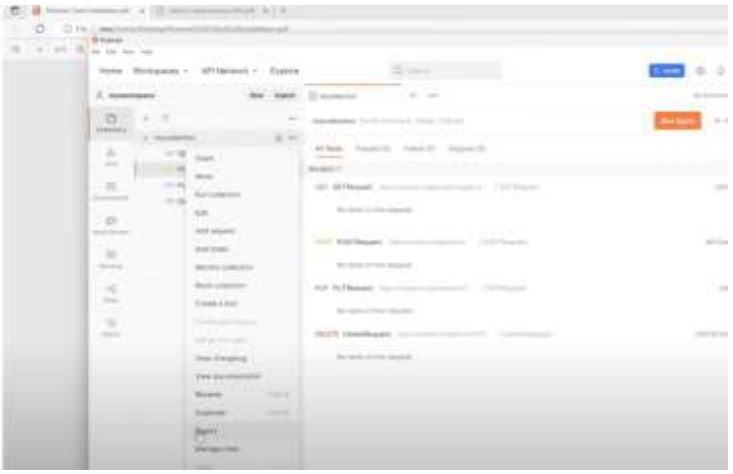












Delete -- delete existing resource from database

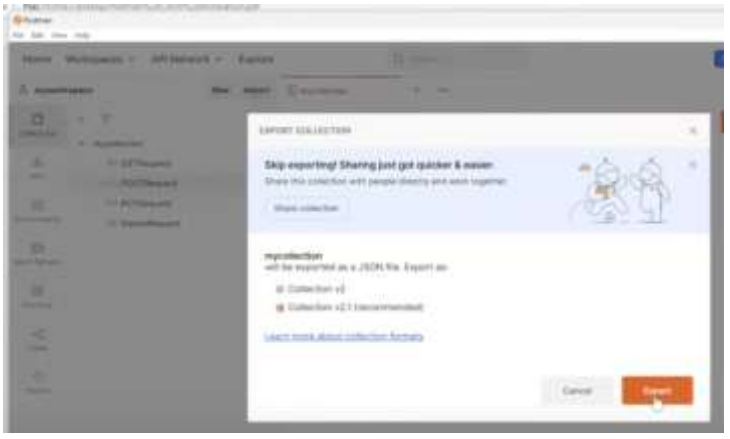
sample API:

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

validations

---

status code
time
size data
response body (json/xml)
cookies
headers



validations

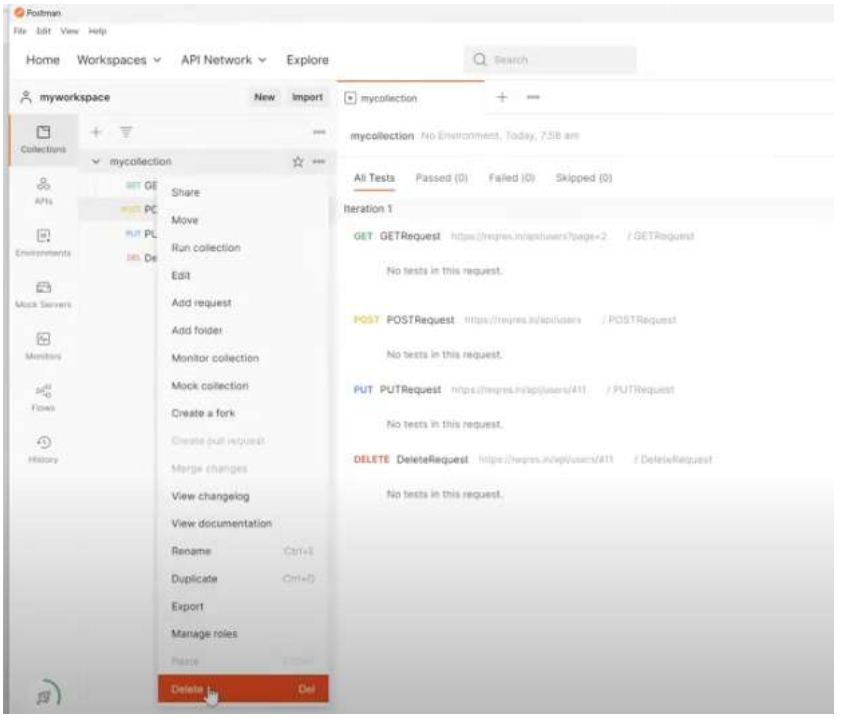
---

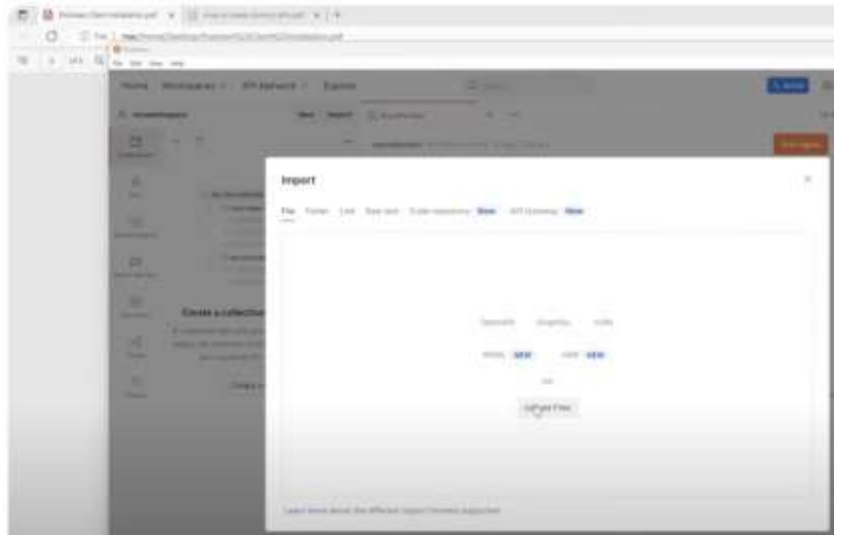
status code
time
size data
response body (json/xml)
cookies
headers

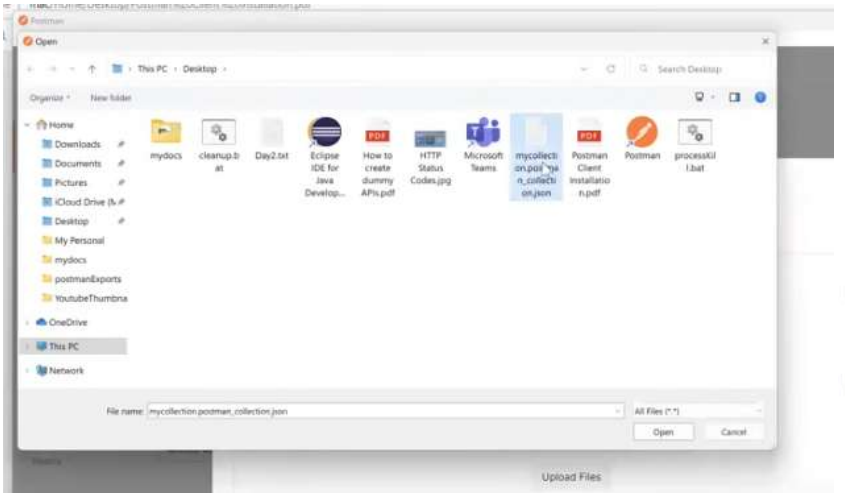
http status codes

3 levels

200
400
500







Day-3

Istep1) NodeJS

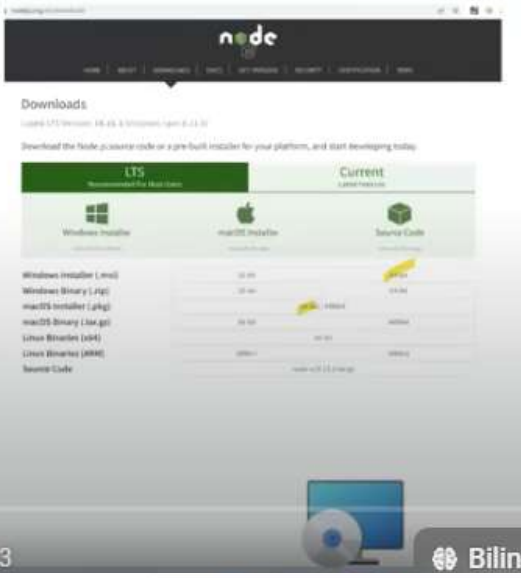
npm - node package manager

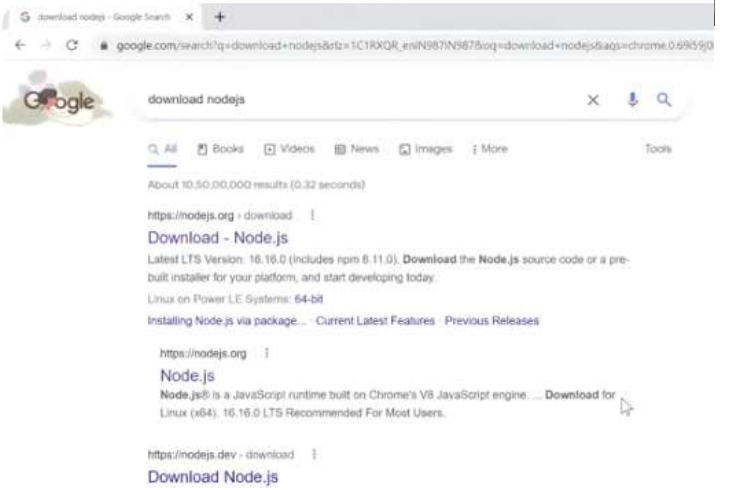
step2) json-server

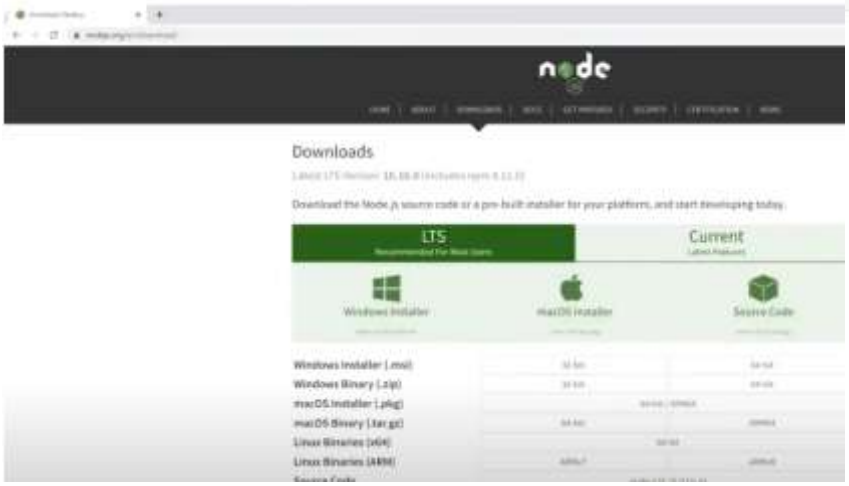
How to create dummy API's

1) Install NodeJS

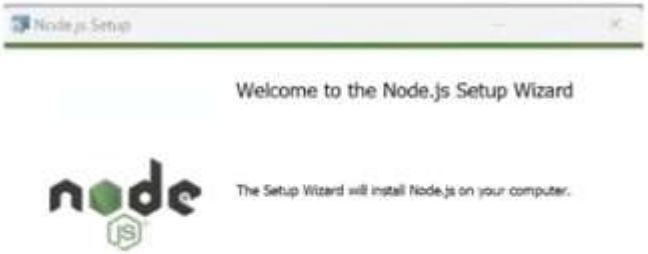
Download link: https://nodejs.org/en/download/



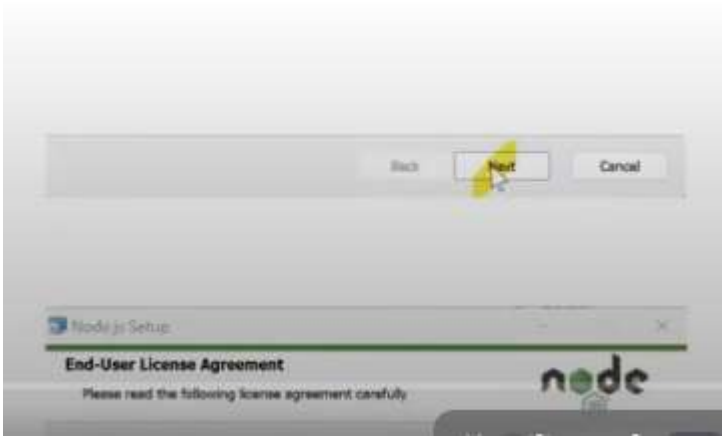



Downloads
LTS
Current
Windows Installer
macOS Installer
Source Code


Automatically install the necessary tools. Note that this will also install Chocolatey. The script will pop-up in a new window after the installation completes.
Alternatively, follow the instructions at https://github.com/nodejs/node-gyp#on-windows to install the dependencies yourself.
Back
Next
Cancel


Node.js Setup
Welcome to the Node.js Setup Wizard
The Setup Wizard will install Node.js on your computer.


Node.js Setup
Ready to install Node.js
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Install
Cancel


Back
Next
Cancel
Node.js Setup
End-User License Agreement
Please read the following license agreement carefully


Node.js Setup
Installing Node.js
Please wait while the Setup Wizard installs Node.js.
Status: Copying new files
Back
Next
Cancel

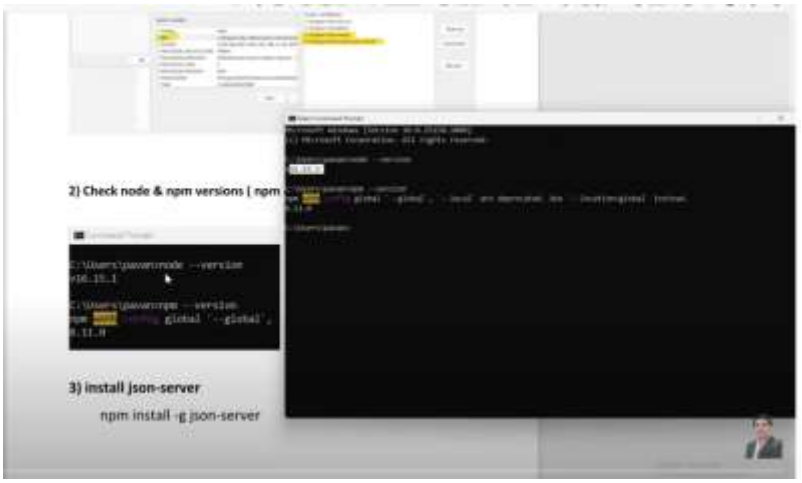

The Setup Wizard will install Node.js on your computer.

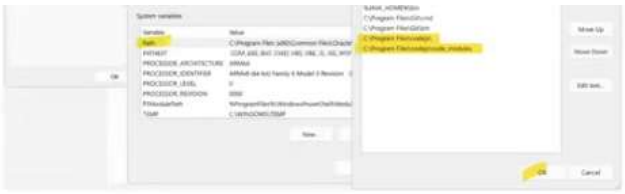

Back
Next
Cancel

Node.js Setup
End-User License Agreement
Please read the following license agreement carefully
Node.js is licensed for use as follows:
Copyright Node.js contributors. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the
I accept the terms in the License Agreement
Print
Back
Next
Cancel

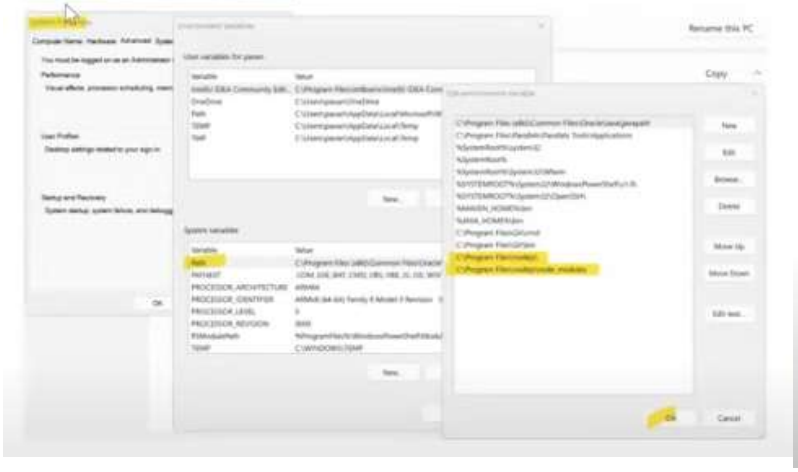

Node.js Setup


Completed the Node.js Setup Wizard
Click the Finish button to exit the Setup Wizard.
Node.js has been successfully installed.





Set Environment variable:

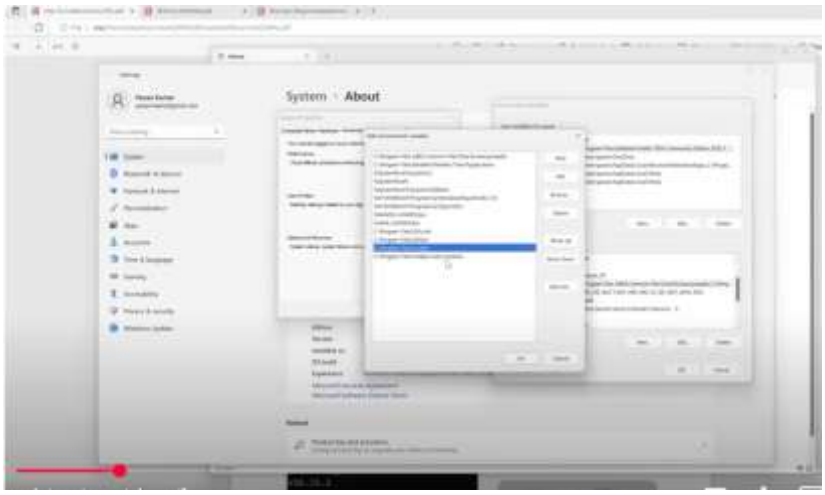


2) Check node & npm versions (npm comes along with node.js)

```
Command Prompt

C:\Users\pavan>node --version
v16.15.1

C:\Users\pavan>npm --version
npm WARN config global `--global`,
8.11.0
```

3) install json-server

npm install -g json-server



Day-3

---

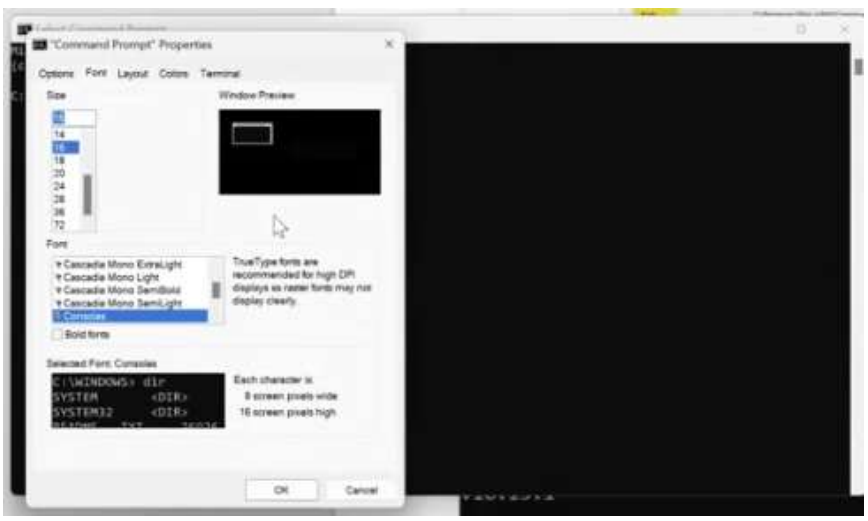
step1) NodeJS

npm - node package manager

node --version

npm --version

step2) json-server



Day-3

---

step1) NodeJS

npm - node package manager

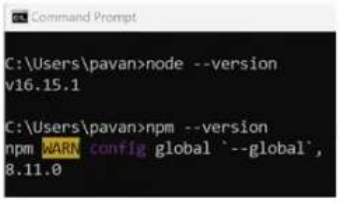
node --version

npm --version

step2) json-server

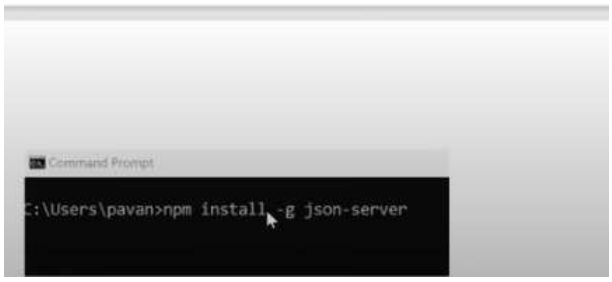
runthe below command in the cmd/terminal

npm install -g json-server



3) install json-server

npm install -g json-server



Day-3

Creating our own API's

---

step1) NodeJS

npm - node package manager

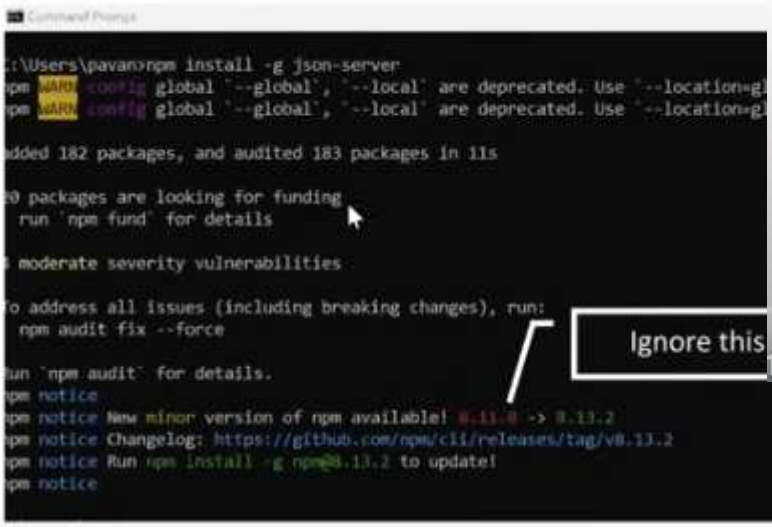
node --version

npm --version

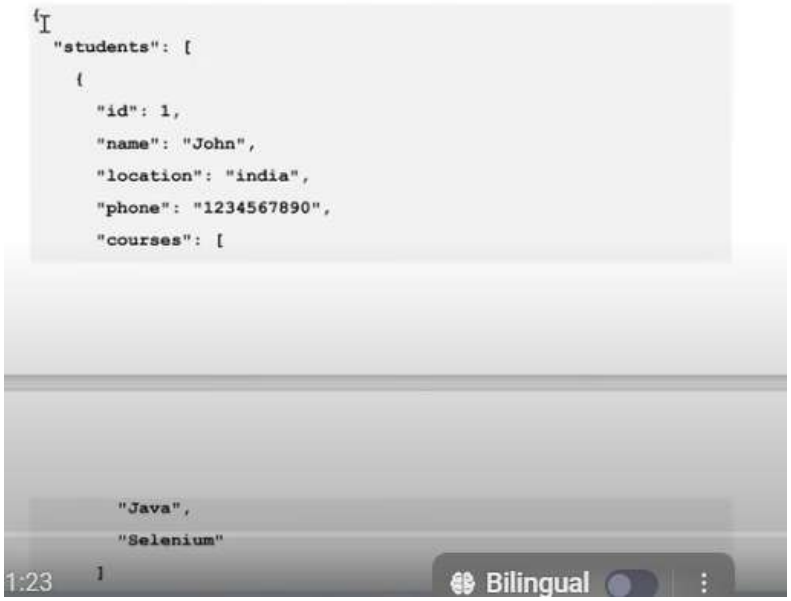
step2) json-server

runthe below command in the cmd/terminal

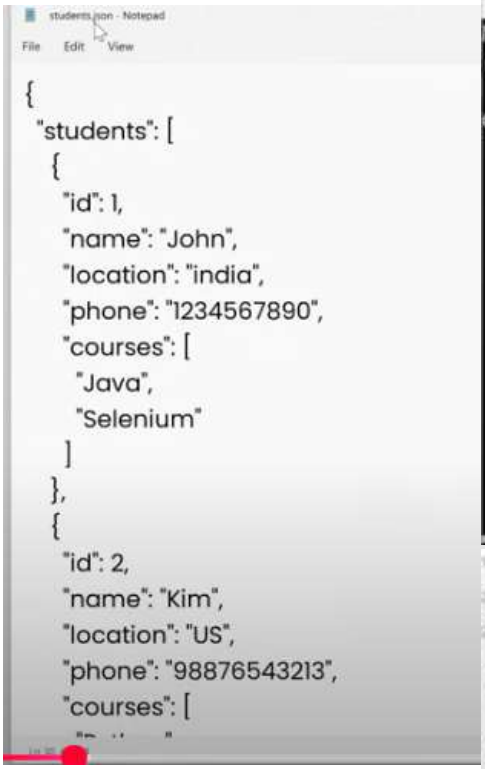
npm install -g json-server

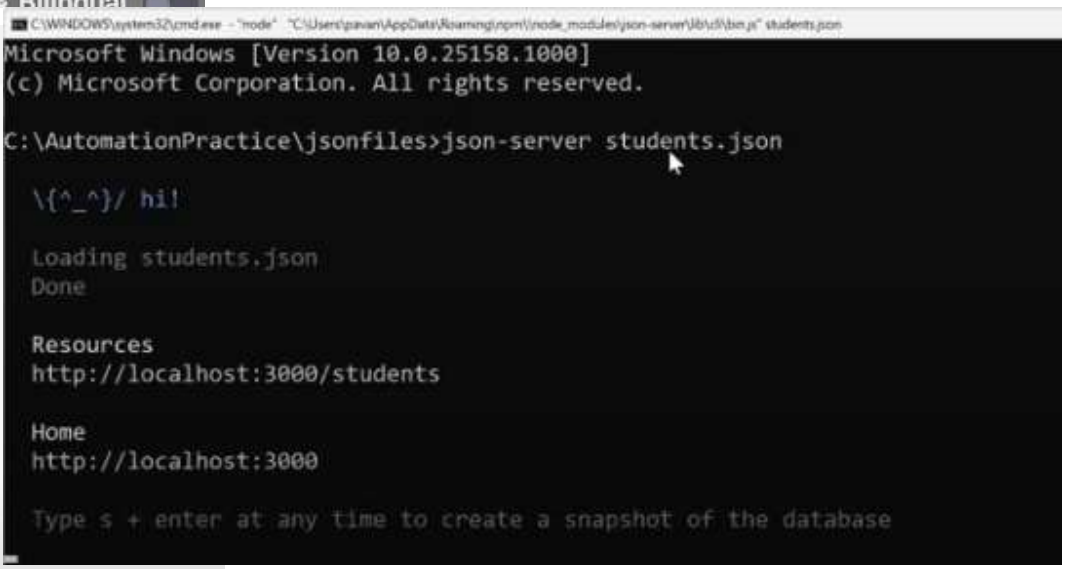


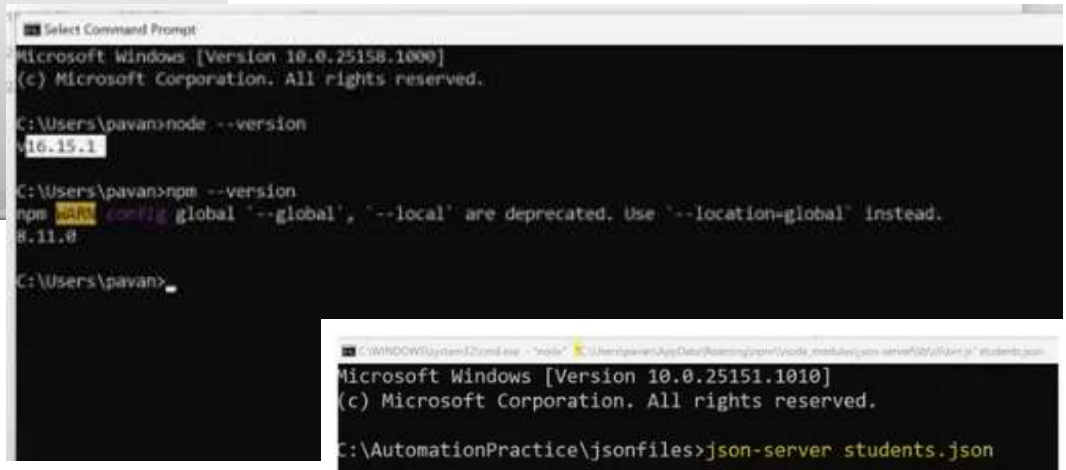
4) create students.json file with the following data

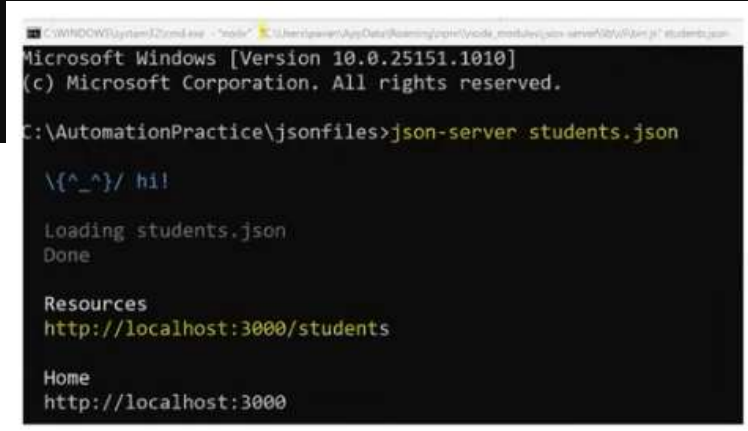


"students": [
{
"id": 1,
"name": "John",
"location": "india",
"phone": "1234567890",
"courses": [

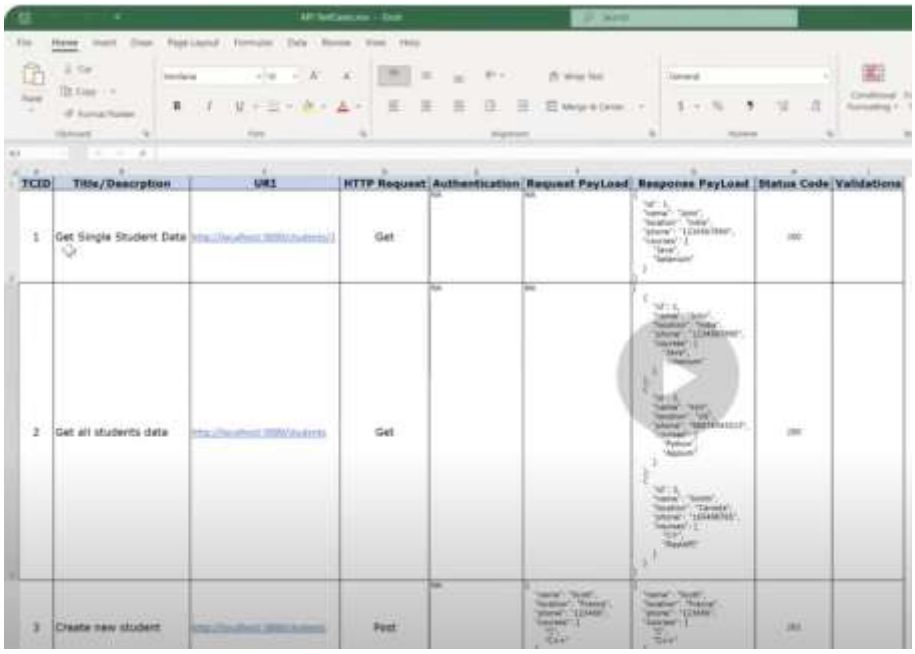




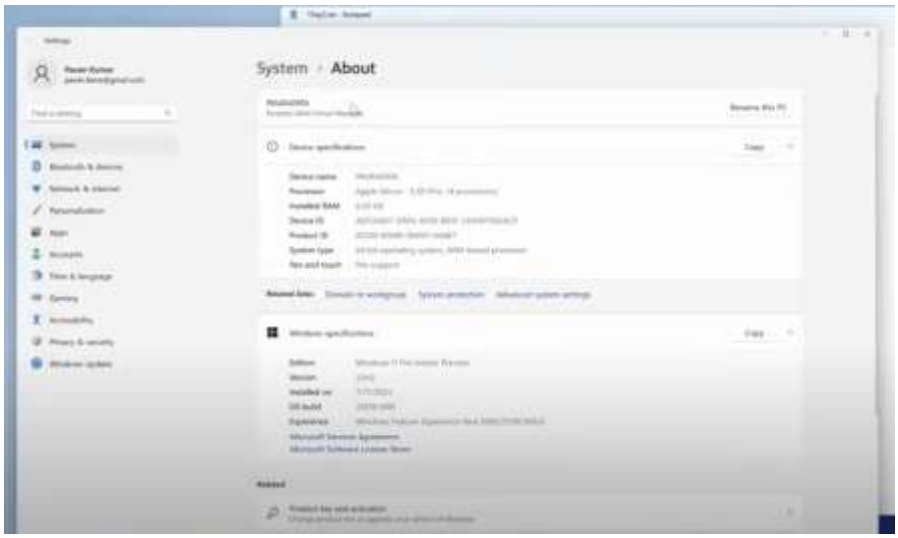


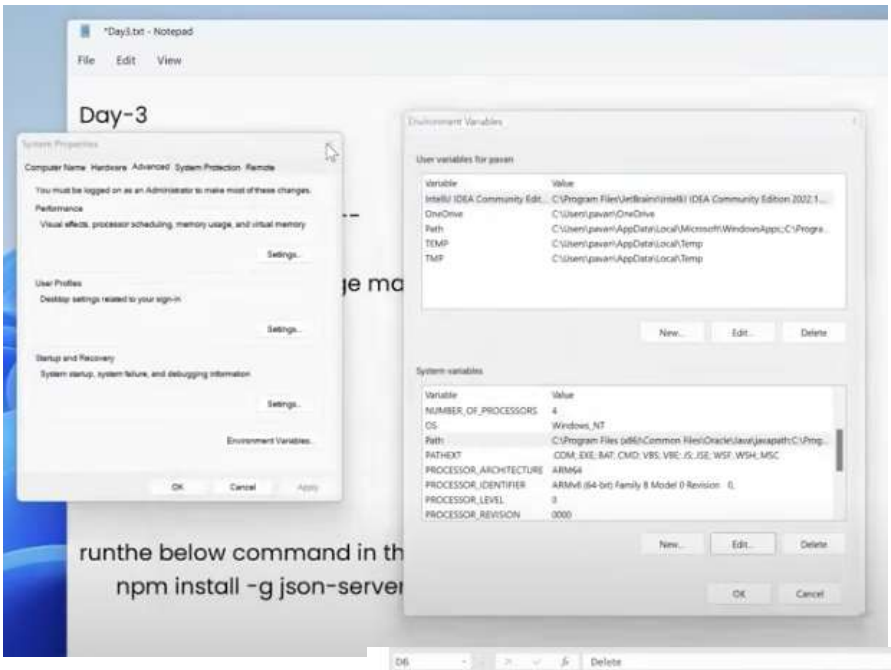


http://localhost:3000/students

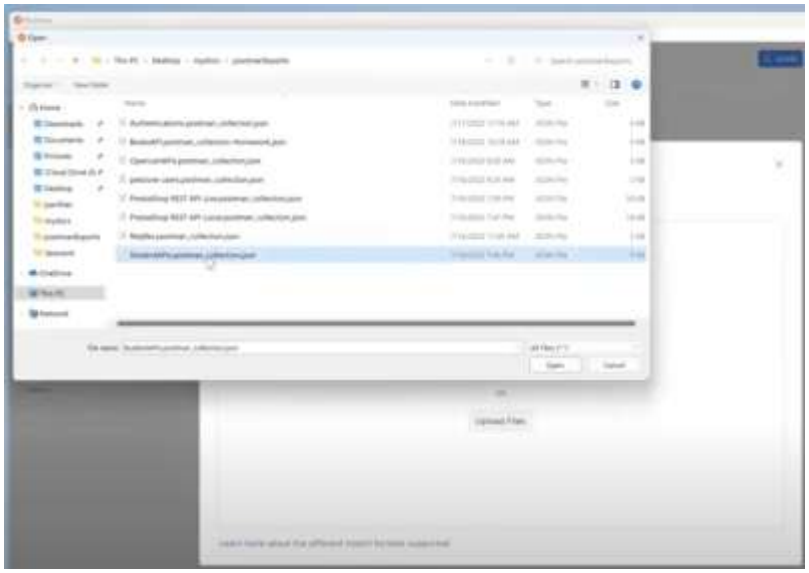


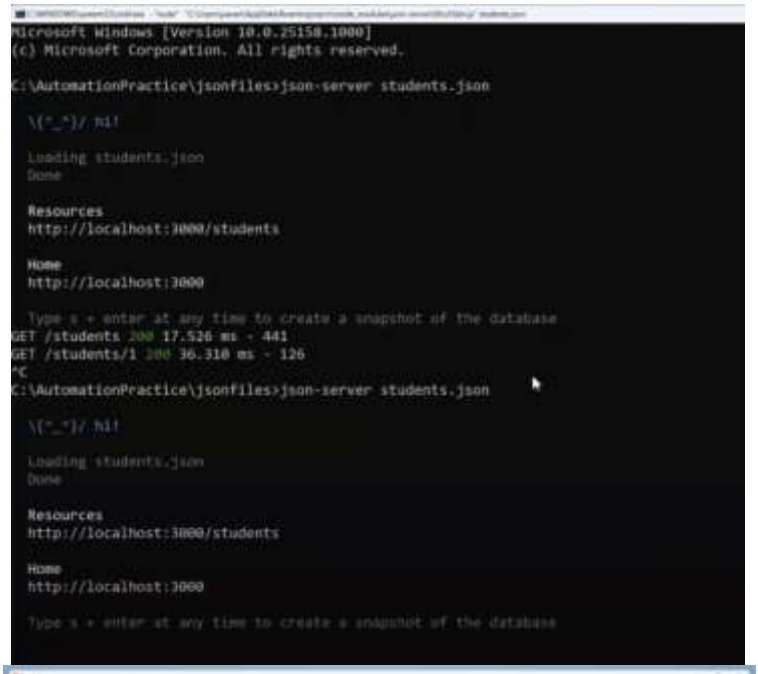
TCID	Title/Description	URI	HTTP Request	Authentication	Request PayLoad	Response PayLoad	Status Code	Validations
1	Get Single Student Data	http://localhost:3000/students/1	Get				200	
2	Get all students data	http://localhost:3000/students	Get				200	
3	Create new student	http://localhost:3000/students	Post				201	

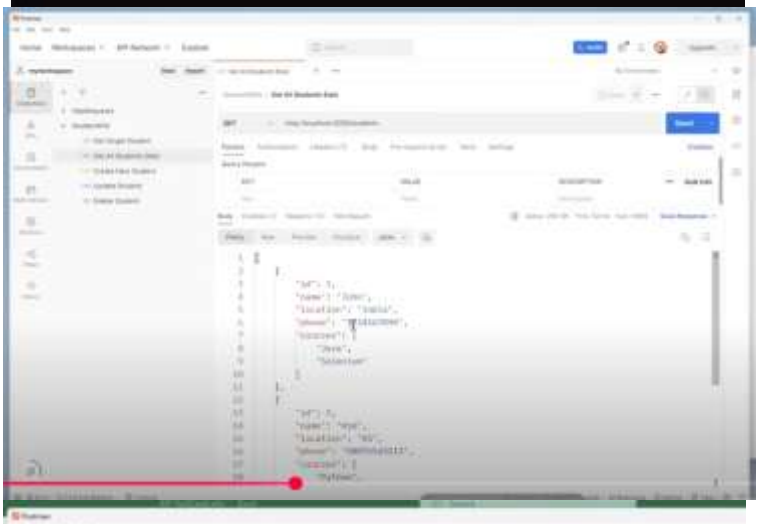


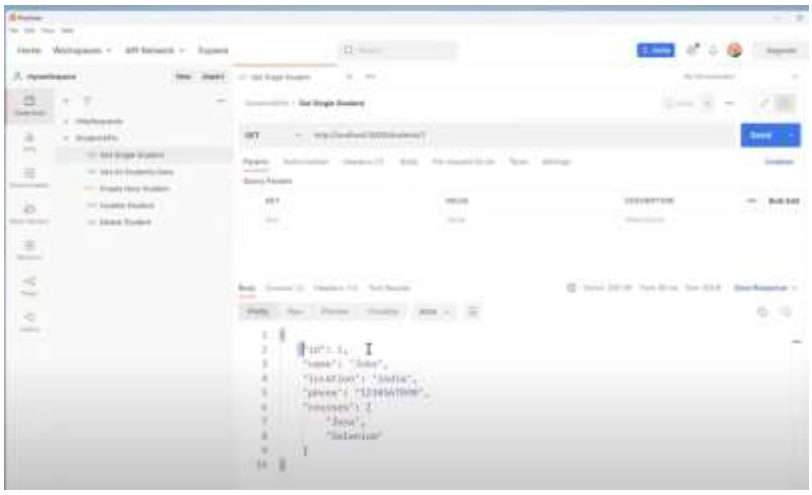


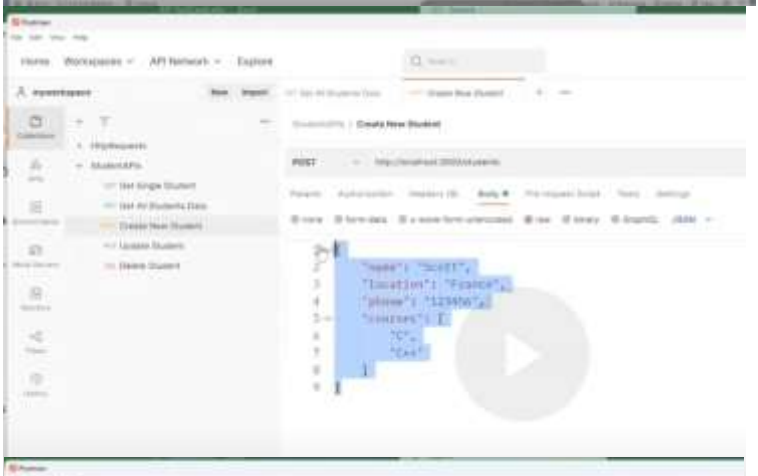
3	Create new student	http://localhost:3000/students	Post	NA	{"name": "Scott", "location": "France", "phone": "123456", "courses": ["C#", "C++"]}	{"name": "Scott", "location": "France", "phone": "123456", "courses": ["C#", "C++"], "id": 4}	201	
4	Update Student	http://localhost:3000/students/4	Put	NA	{"name": "Scott", "location": "Germany", "phone": "654321", "courses": ["C#", "C++"]}	{"name": "Scott", "location": "Germany", "phone": "654321", "courses": ["C#", "C++"], "id": 4}	200	
5	Delete Student	http://localhost:3000/students/4	Delete	NA	NA	{}	200	

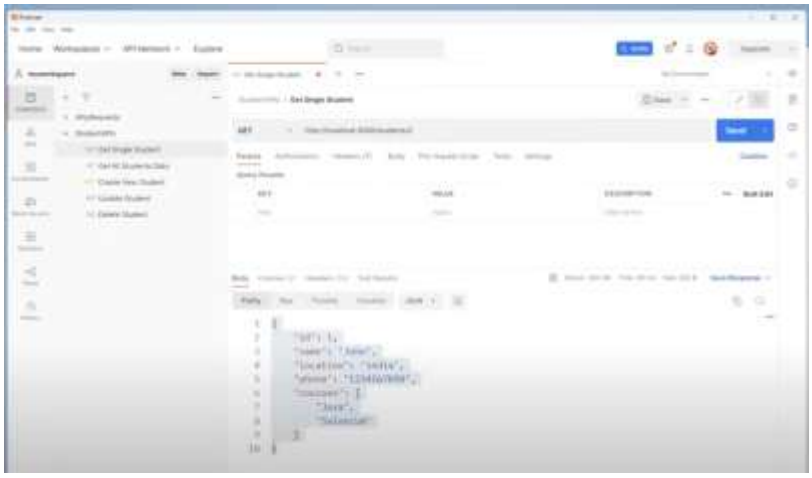


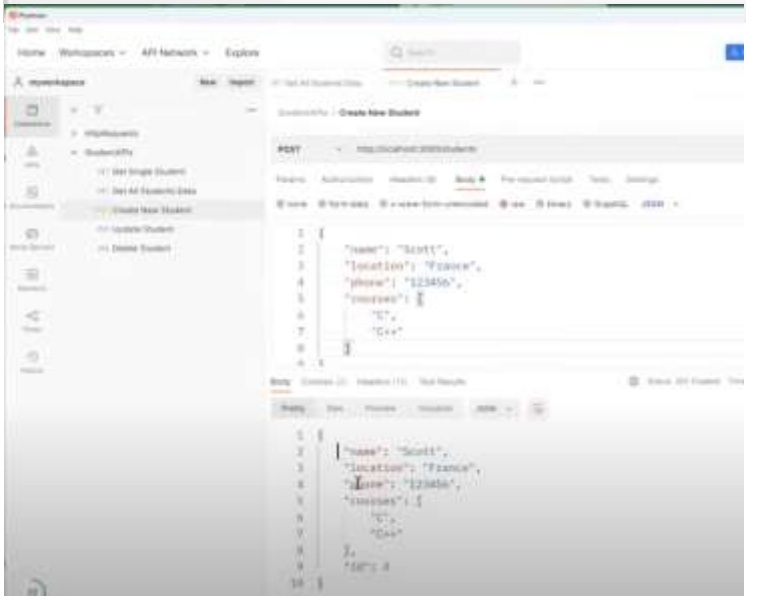

Microsoft Windows [Version 10.0.25158.1000]
(c) Microsoft Corporation. All rights reserved.
C:\AutomationPractice\jsonfiles>json-server students.json
\{^_^}/ hi!
Loading students.json
Done
Resources
http://localhost:3000/students
Home
http://localhost:3000
Type s + enter at any time to create a snapshot of the database
GET /students 200 17.526 ms - 441
GET /students/1 200 36.310 ms - 126
^C
C:\AutomationPractice\jsonfiles>json-server students.json
\{^_^}/ hi!
Loading students.json
Done
Resources
http://localhost:3000/students
Home
http://localhost:3000
Type s + enter at any time to create a snapshot of the database

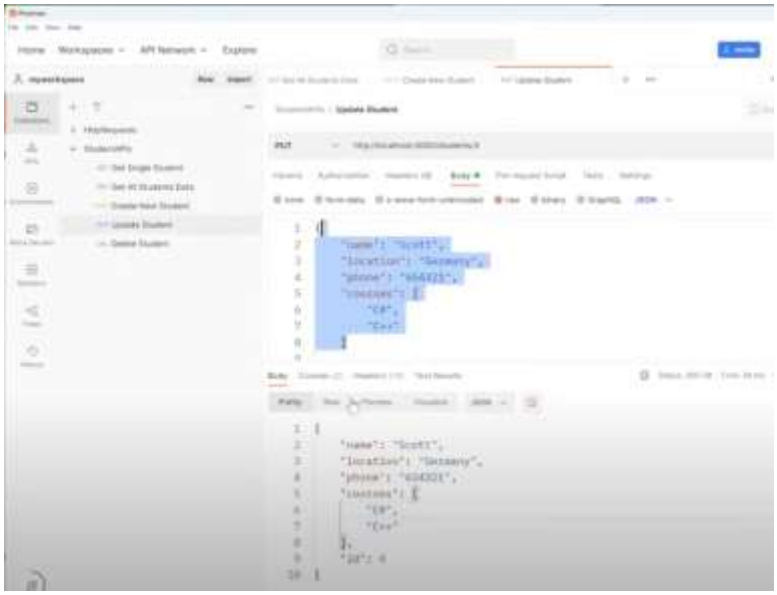


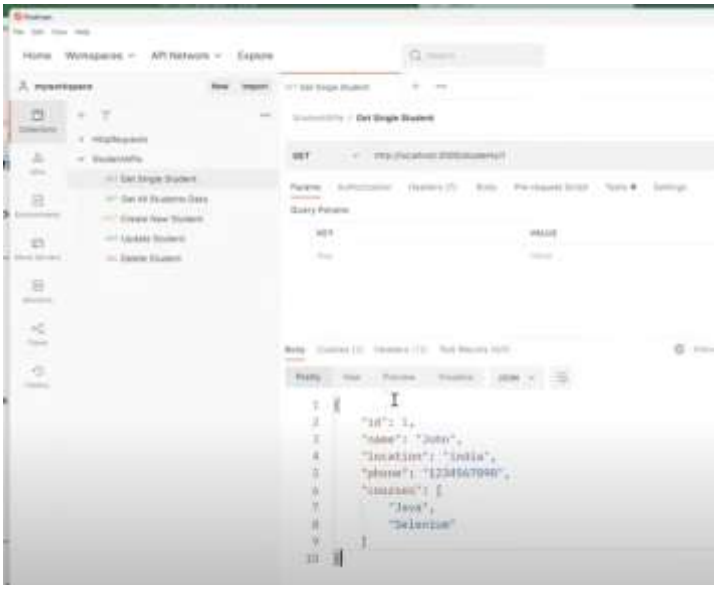


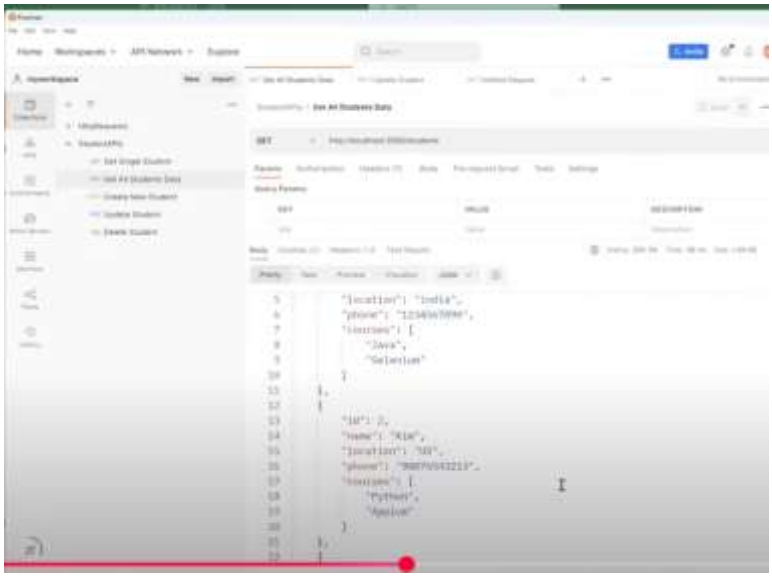


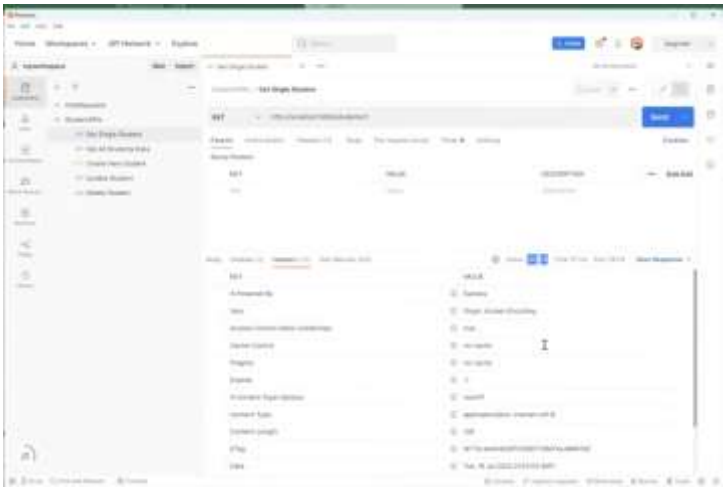


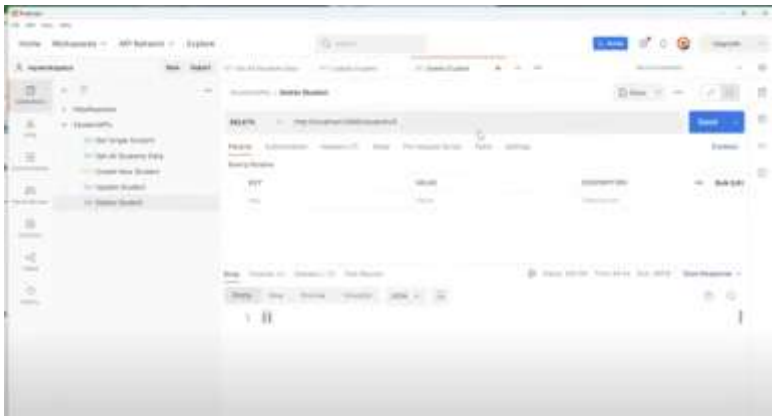


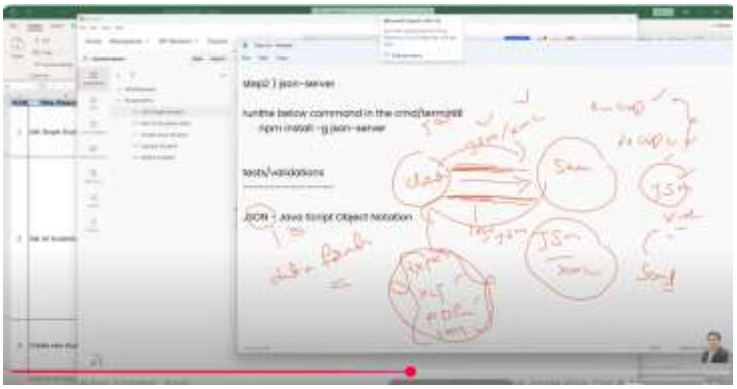

{
"name": "Scott",
"location": "Germany",
"phone": "654321",
"courses": [
"C#",
"C++"
]
}

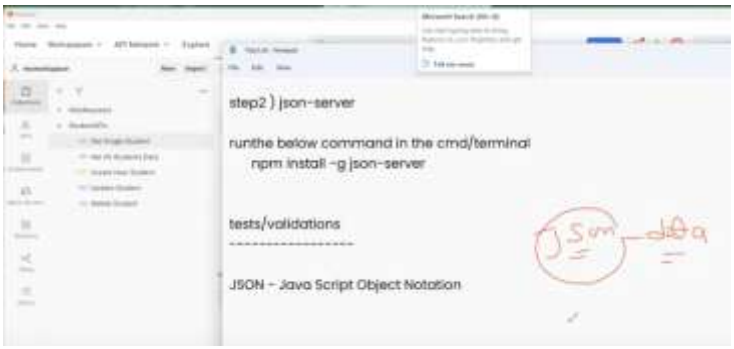

{
"id": 1,
"name": "John",
"location": "India",
"phone": "1234567890",
"courses": [
"Java",
"Selenium"
]
}

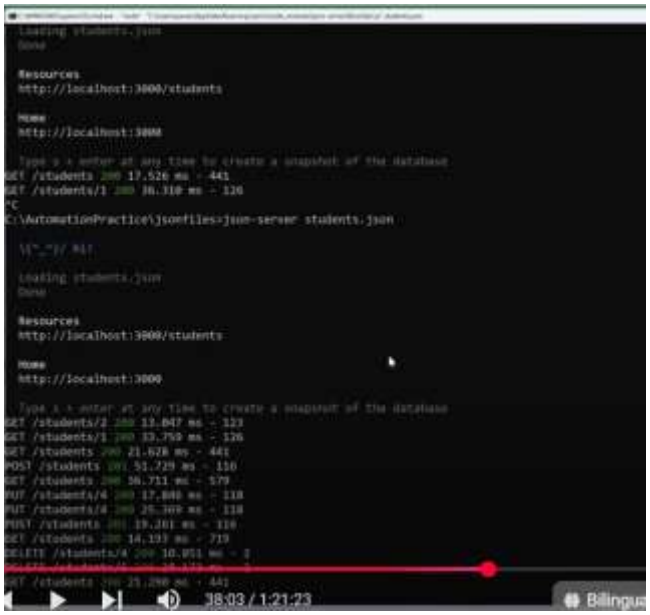

"location": "India",
"phone": "1234567890",
"courses": [
"Java",
"Selenium"
]
},
{
"id": 2,
"name": "Rim",
"location": "US",
"phone": "9087654321",
"courses": [
"Python",
"Appium"
]
},






step2) json-server
runthe below command in the cmd/terminal
npm install -g json-server
tests/validations
JSON - Java Script Object Notation


step2) json-server
runthe below command in the cmd/terminal
npm install -g json-server
tests/validations
JSON - Java Script Object Notation


Resources
http://localhost:3000/students
Home
http://localhost:3000
GET /students 200 17.526 ms - 441
GET /students/1 200 36.310 ms - 126
^C
C:\AutomationPractice\jsonfiles>json-server students.json
Loading students.json
Done
Resources
http://localhost:3000/students
Home
http://localhost:3000
GET /students/2 200 13.047 ms - 123
GET /students/1 200 33.759 ms - 126
GET /students 200 21.628 ms - 441
POST /students 201 51.729 ms - 116
GET /students 200 36.711 ms - 579
PUT /students/4 200 17.808 ms - 118
PUT /students/4 200 25.309 ms - 118
POST /students 201 19.201 ms - 116
GET /students 200 14.193 ms - 719
DELETE /students/4 200 10.951 ms - 2
38:03 / 1:21:23

tests/validations

JSON - Java Script Object Notation

key values pais

key: value

key is always included in " " quotation

```
{
"name": "John",
"age": 30
}
```

tests/validations

JSON - Java Script Object Notation

key values pais

key: value

key is always included in " " quotation

```
{
"name": "John",
"age": 30
"phone": [12345,678976 ]
}
```

Data Types

- **Array**
  - Values in JSON can be arrays.
  - Example:
    ```
    {
    "employees":[ "John", "Anna", "Peter" ]
    }
    ```
- **Boolean**
  - Values in JSON can be true/false.
  - Example:
    ```
    { "sale":true }
    ```
- **Null**
  - Values in JSON can be null.
    ```
    { "middlename":null }
    ```

```
[
"firstname": "John",
"secondname": null
"age": 30,
"phone": [12345,678976 ]
"status": true
]
```

JSON - Syntax

- Data should be in name/value pairs
- Data should be separated by commas
- Curly braces should hold objects
- Square brackets hold arrays

students data

student -- sid, sname , grad

```
{
"students":[
        {
            "sid" : 101,
            "sname" : "John",
            "grad": "A"
        },
        {
            "sid": 102,
            "sname": "Kim",
            "grad": "B"
        },
        {
            "sid": 103,
            "sname": "Scott",
            "grad": "C"
        },
    ]
}
```

JSON vs XML

What is JSON?

- JSON – Java Script Object Notation
- JSON is a syntax for storing and exchanging data
- Basically it was designed for human-readable data interchange.
- JSON is text, written with Java Script Object Notation.
- It has been extended from the JavaScript scripting language
- The filename extension is **.json**
- JSON internet Media type is **application/json**

Examples

JSON Example | XML Example

JSON Data Types

- Number

JSON Object

- JSON object holds key/value pair. Each key is represented as a string in JSON and value can be of any type.
- The keys and values are separated by colon. Each key/value pair is separated by comma.
- The curly brace { represents JSON object.
- Example:

```
{
    "employee": {
        "name":     "James",
        "salary":   56000,
        "married":  true
    }
}
```

```
{
"students": [100,200,300]
}
```

```
{
"students": ["A","B","C"]
}
```

JSON Data Types

- Number
- String
- Boolean
- Null
- Object
- Array

- JSON Object with Strings

```
{
"name":  "Scott",
"email":  "Scottjaiswal1987@gmail.com"
}
```

- JSON Object with Numbers

```
{
"integer": 34,
"fraction": .2145,
"exponent": 6.87789e+0
}
```

- JSON Object with Booleans

```
{
"first": true,
"second": false
}
```

- JSON Nested Object

```
{
"firstname": "Scott",
"lastname": "Jaiswal",
"age": 27,
"address": {
  "streetaddress": "Plot-9, Mohan Nagar",
  "city": "Hyderabad",
  "state": "TS",
  "postalCode": "500059"
}
}
```

Data Types

- **String**
  - Strings in JSON must be written in double quotes.
  - Example:
    ```
    { "name":"John" }
    ```
- **Numbers**
  - Numbers in JSON must be an integer or a floating point.
  - Example:
    ```
    { "age":30 }
    ```
- **Object**
  - Values in JSON can be objects.
  - Example:
    ```
    {
    "employee":{ "name":"John", "age":30, "city":"New York" }
    }
    ```

JSON Array

- JSON array represents ordered list of values.
- JSON array can store multiple values. It can store string, number, boolean or object in JSON array.
- In JSON array, values must be separated by comma.
- The [(square bracket) represents JSON array.

JSON Array of Strings

["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]

JSON Array of Numbers

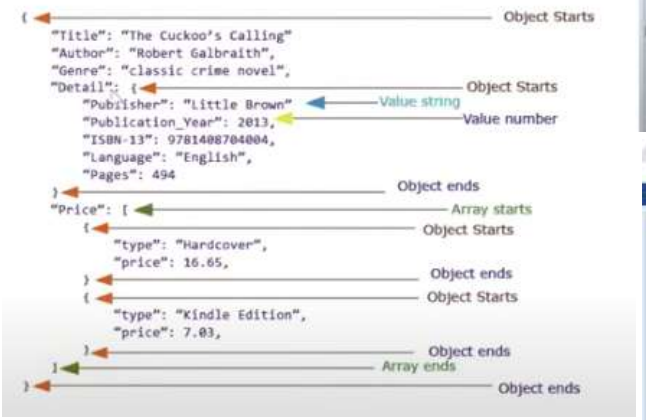
[12, 34, 56, 43, 95]

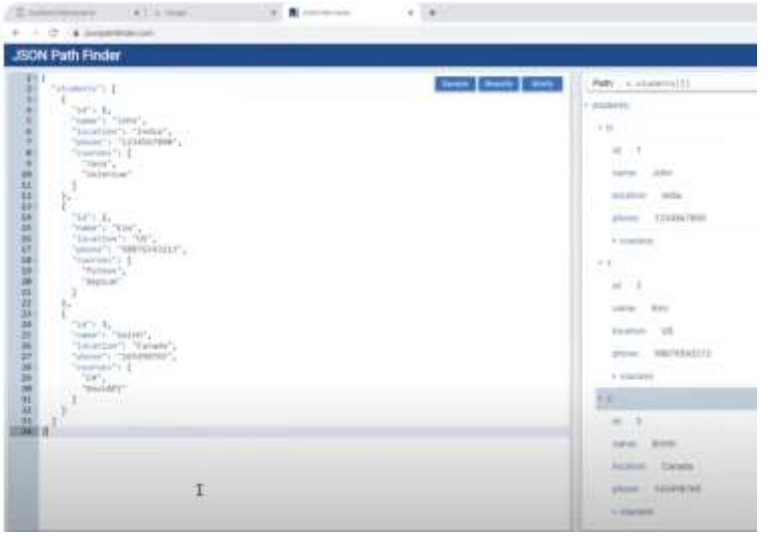
JSON Array of Booleans

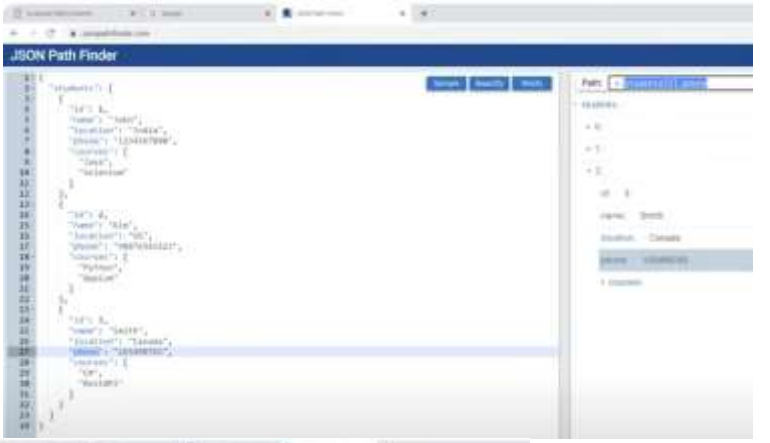
[true, true, false, false, true]

JSON Array of Objects

```
{"employees":[
   {"name":"Ram", "email":"ram@gmail.com", "age":23},
   {"name":"Shyam", "email":"shyam23@gmail.com", "age":28},
   {"name":"John", "email":"john@gmail.com", "age":33},
   {"name":"Bob", "email":"bob32@gmail.com", "age":41}
]}
```



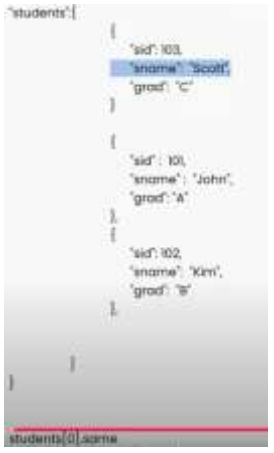


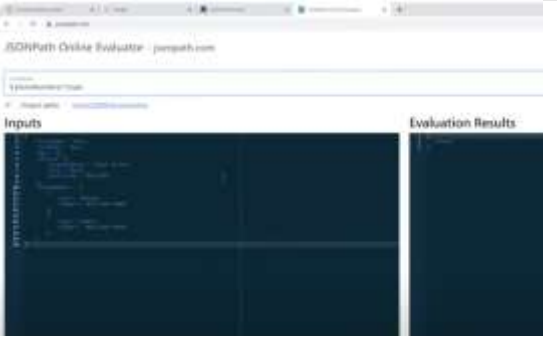


Capture & Validate JSON Path

https://jsonpathfinder.com/

https://jsonpath.com/





npm --version

step2) json-server

runthe below command in the cmd/terminal
npm install -g json-server

tests/validations

JSON - Java Script Object Notation

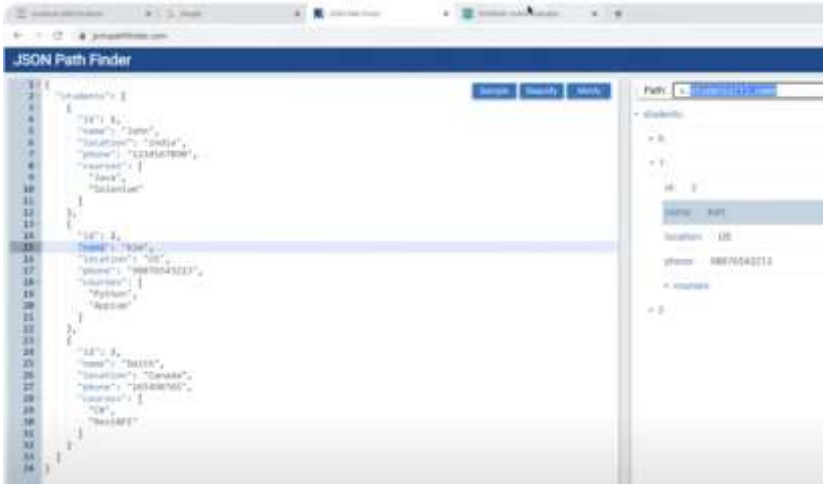
key values pais

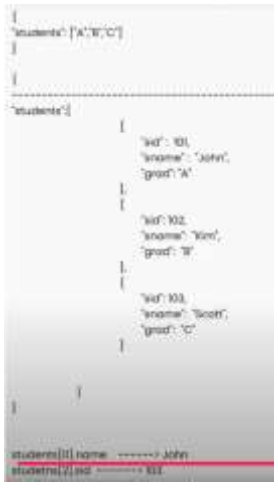
key: value

kery is always included in " " quotation

```
{
"firstname": "John",
"secondname": null
"age": 30,
"phone": [12345,678976 ]
"status": true
}
```

students data





Day-4

Response validations

Status code
Headers
Cookies
Response time
Response body

Assertion - validation

pm - library
functions
javascript

Status code
Headers
Cookies
Response time
Response body

Assertion - validation

pm - library
functions
javascript

Normal function
Arrow function

Response Validations

(Adding Tests)

- Status code
- Headers
- Cookies
- Response time
- Response body

Chai Assertion Library

```
pm.test("Test Name", function()
                    {
                      // assertion;
                    }
      );
```

```
pm.test("Test Name", () =>
                    {
                      // assertion;
                    }
      );
```

Testing status codes

Test for the response status code:

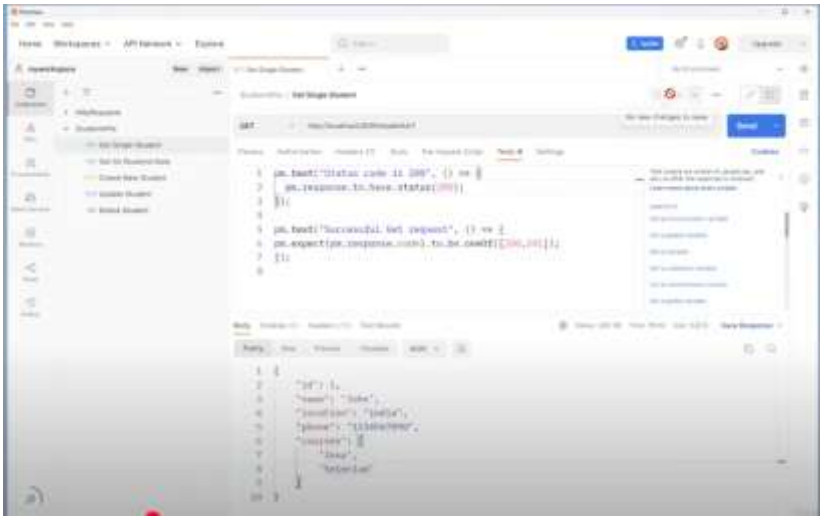
```
pm.test("Status code is 200", () => {
  pm.response.to.have.status(200);
});
```

If you want to test for the status code being one of a set, include them all in an array and use *one of*

```
pm.test("Successful POST request", () => {
pm.expect(pm.response.code).to.be.oneOf([201,202]);
});
```

Check the status code text:

```
pm.test("Status code name has string", () => {
  pm.response.to.have.status("Created");
});
```

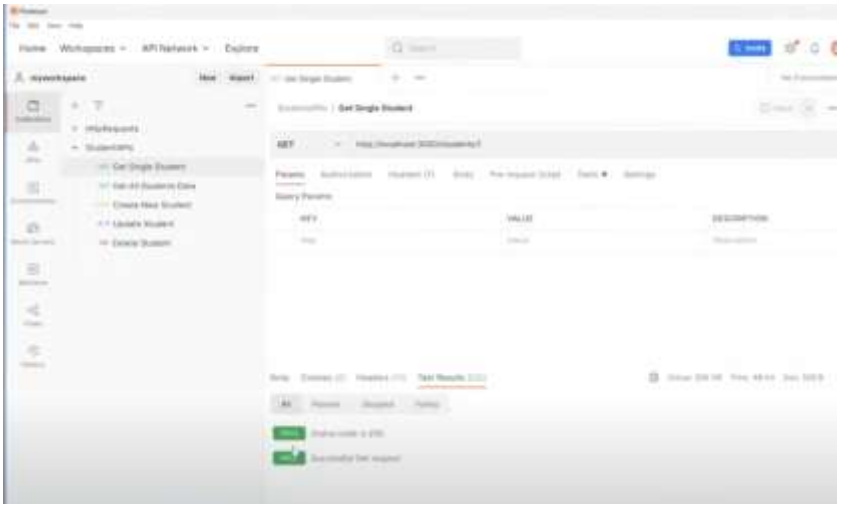


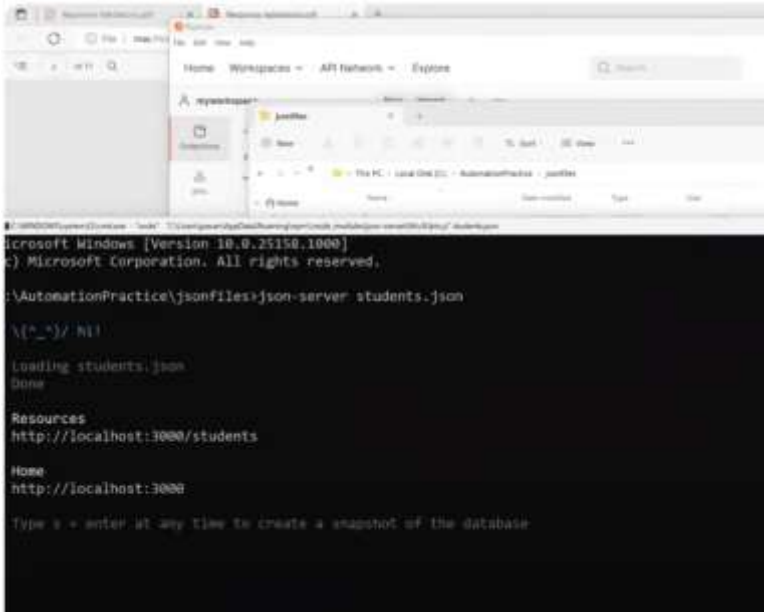
functions
javascript

Normal function
Arrow function

```
pm.test("Test Name", () =>
                        {
                        // assertion;
                        }
);
```

```
pm.test("Status code is 200", () => {
 pm.response.to.have.status(200);
});
```





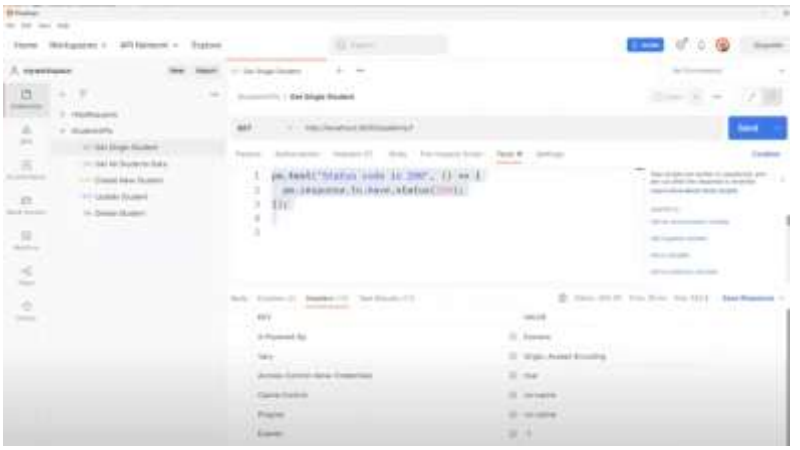
Testing headers

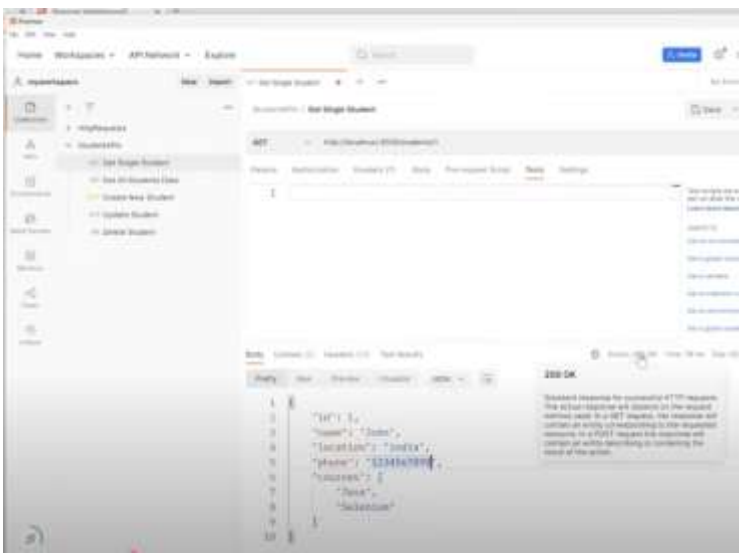
Check that a response header is present:

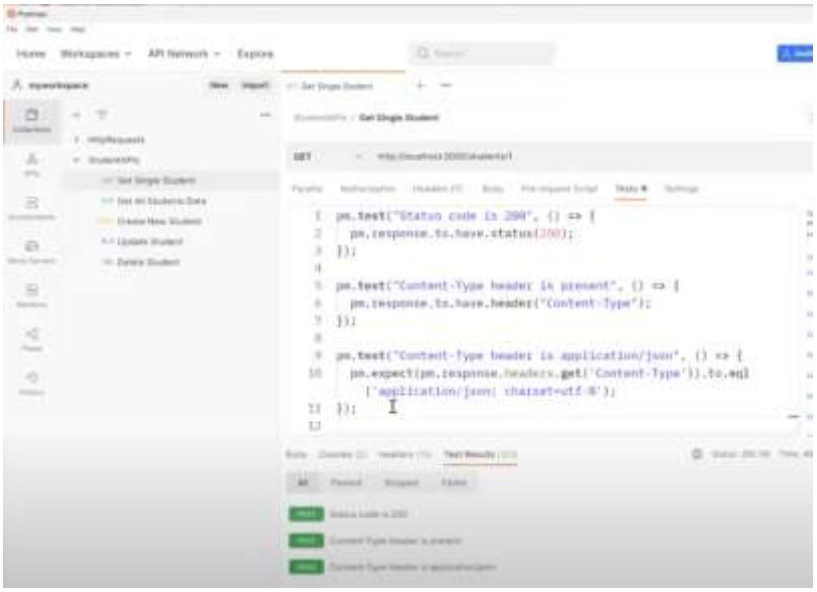
```
pm.test("Content-Type header is present", () => {
  pm.response.to.have.header("Content-Type");
});
```

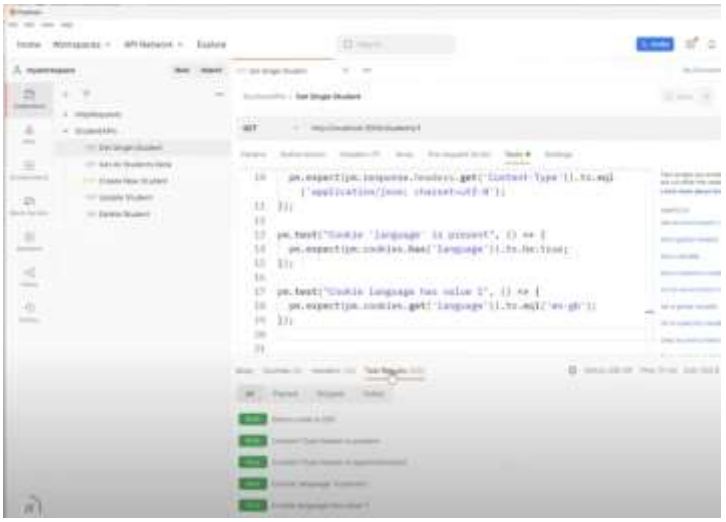
Test for a response header having a particular value:

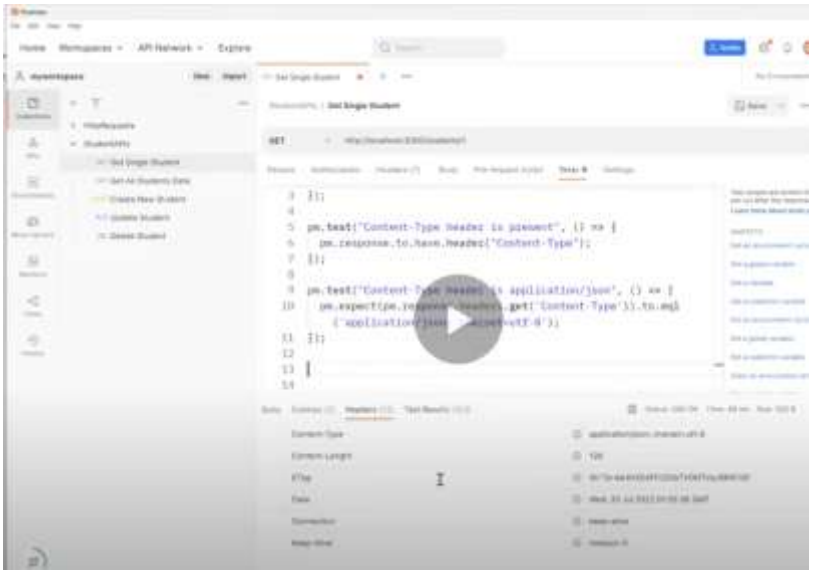
```
pm.test("Content-Type header is application/json", () => {
  pm.expect(pm.response.headers.get('Content-
Type')).to.eql('application/json; charset=utf-8');
});
```











Testing status codes

Test for the response status code:

```
pm.test("Status code is 200", () => {
  pm.response.to.have.status(200);
});
```

If you want to test for the status code being one of a set, include them all in an array and use *one of*

```
pm.test("Successful POST request", () => {
pm.expect(pm.response.code).to.be.oneOf([201,202]);
});
```

Check the status code text:

```
pm.test("Status code name has string", () => {
  pm.response.to.have.status("Created");
});
```

Testing cookies

Test if a cookie is present in the response:

```
pm.test("Cookie 'language' is present", () => {
  pm.expect(pm.cookies.has('language')).to.be.true;
});
```

Test for a particular cookie value:

```
pm.test("Cookie language has value 1", () => {
  pm.expect(pm.cookies.get('language')).to.eql('en-gb');
});
```

Testing response times

Test for the response time to be within a specified range:

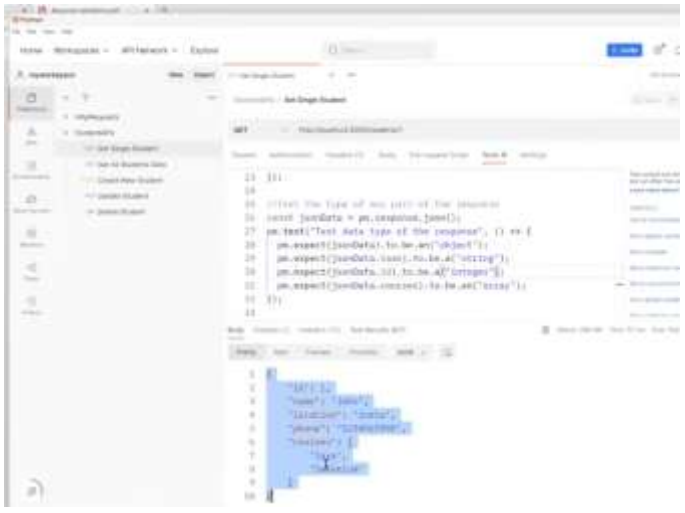
```
pm.test("Response time is less than 200ms", () => {
  pm.expect(pm.response.responseTime).to.be.below(200);
});
```

Asserting a value type

Test the type of any part of the response:

```
{
    "id": 1,
    "name": "John",
    "location": "india",
    "phone": "1234567890",
    "courses": [
        "Java",
        "Selenium"
    ]
}
```

```
const jsonData = pm.response.json();
pm.test("Test data type of the response", () => {
  pm.expect(jsonData).to.be.an("object");
  pm.expect(jsonData.name).to.be.a("string");
  pm.expect(jsonData.id).to.be.a("number");
  pm.expect(jsonData.courses).to.be.an("array");
});
```



Response Validations

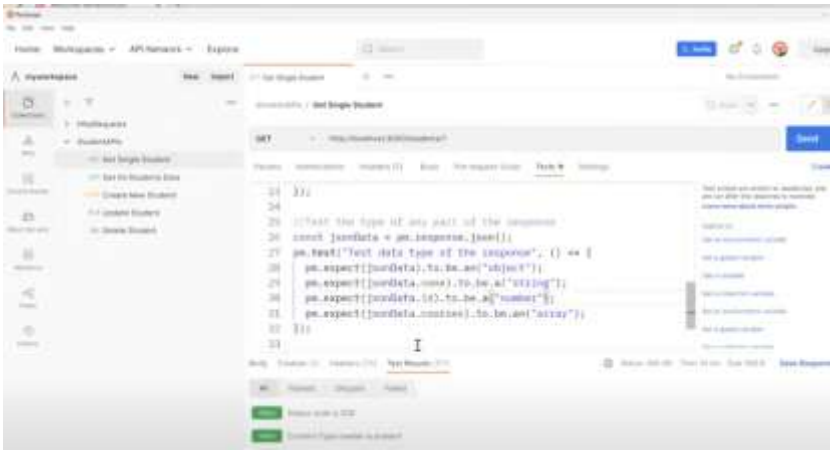
(Adding Tests)

- Status code
- Headers
- Cookies
- Response time
- Response body

Chai Assertion Library

```
pm.test("Test Name", function()
                {
                  // assertion;
                }
      );
```

```
pm.test("Test Name", () =>
                {
                  // assertion;
                }
      );
```



Asserting array properties

Check if an array is empty and if it contains particular items:

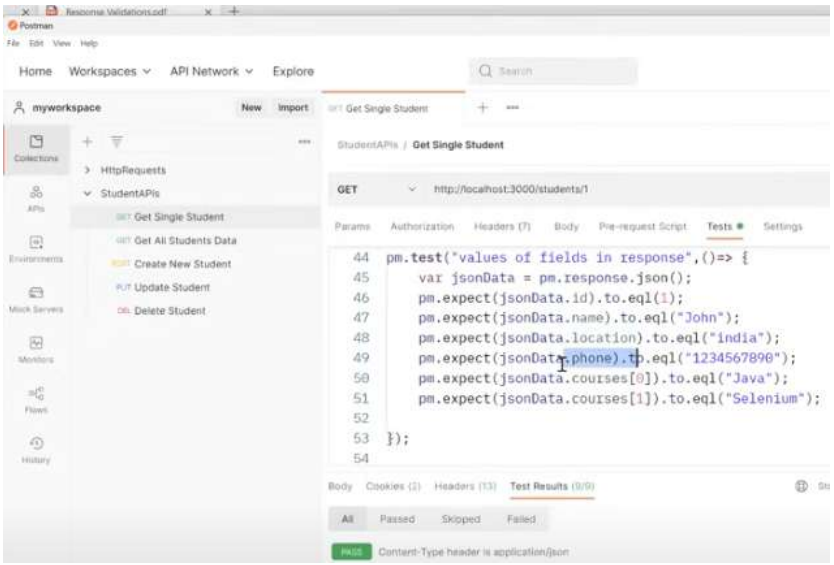
```
{
    "id": 1,
    "name": "John",
    "location": "india",
    "phone": "1234567890",
    "courses": [
        "Java",
        "Selenium"
    ]
}
```

```
pm.test("Test array properties", () => {
    //courses includes "Java"
  pm.expect(jsonData.courses).to.include("Java");
    //courses array must include all listed
  pm.expect(jsonData.courses)
    .to.have.members(["Java", "Selenium"]);
});
```

Validating JSON fields in Response

```
{
    "id": 1,
    "name": "John",
    "location": "india",
    "phone": "1234567890",
    "courses": [
        "Java",
        "Selenium"
    ]
}
```

```
pm.test("value of location field is India",()=> {
    var jsonData = pm.response.json();
    pm.expect(jsonData.id).to.eql(1);
    pm.expect(jsonData.name).to.eql("John");
    pm.expect(jsonData.location).to.eql("india");
    pm.expect(jsonData.phone).to.eql("1234567890");
    pm.expect(jsonData.courses[0]).to.eql("Java");
    pm.expect(jsonData.courses[1]).to.eql("Selenium");

});
```



Response

```
{
    "id": 1,
    "name": "John",
    "location": "india",
    "phone": "1234567890",
    "courses": [
        "Java",
        "Selenium"
    ]
}
```

JSON schema

```
var schema={
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "id": {
      "type": "integer"
    },
    "name": {
      "type": "string"
    },
    "location": {
      "type": "string"
    },
    "phone": {
      "type": "string"
    },
    "courses": {
      "type": "array",
      "items": [
        {
          "type": "string"
        },
        {
          "type": "string"
        }
      ]
    }
  },
  "required": [
    "id",
    "name",
    "location",
    "phone",
    "courses"
  ]
}
```

Validating JSON Schema

Response

```
{
    "id": 1,
    "name": "John",
    "location": "india",
    "phone": "1234567890",
    "courses": [
        "Java",
        "Selenium"
    ]
}
```

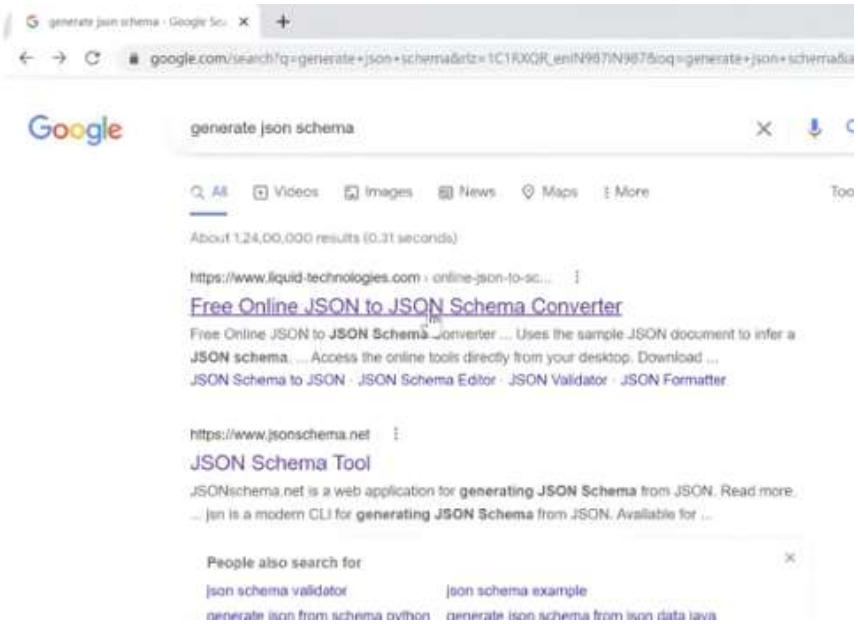
JSON schema

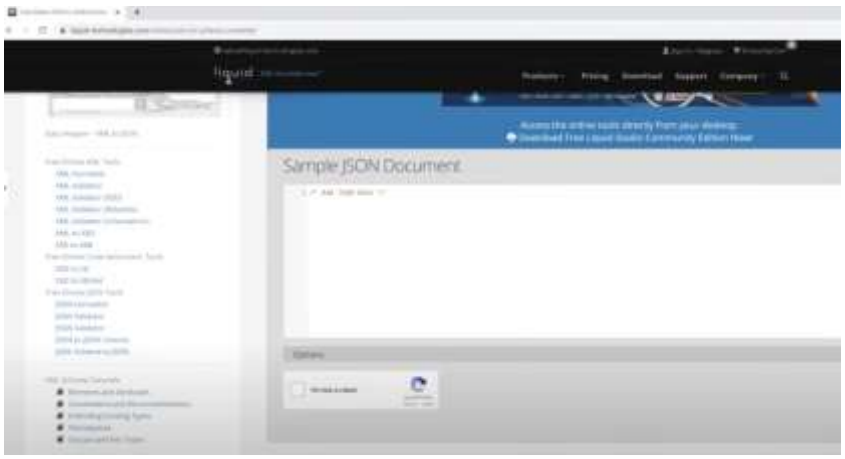
```
var schema={
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "id": {
```

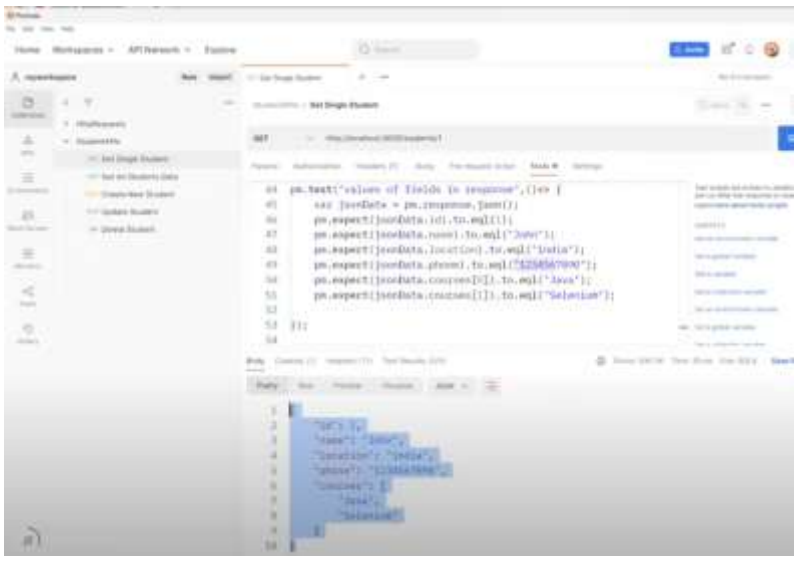
```
    "name",
    "location",
    "phone",
    "courses"
  ]
}
```

JSON schema Validation

```
pm.test('Schema is valid', function() {
  pm.expect(tv4.validate(jsonData, schema)).to.be.true;
});
```







```
    "type": "integer"
  },
  "name": {
    "type": "string"
  },
  "location": {
    "type": "string"
  },
  "phone": {
    "type": "string"
  },
  "courses": {
    "type": "array",
    "items": [
      {
        "type": "string"
      },
      {
        "type": "string"
      }
    ]
  }
},
"required": [
  "id",
```

Validating JSON fields in Response

```
{
    "id": 1,
    "name": "John",
    "location": "india",
    "phone": "1234567890",
    "courses": [
        "Java",
        "Selenium"
    ]
}

pm.test("value of location field is India",()=> {
    var jsonData = pm.response.json();
    pm.expect(jsonData.id).to.eql(1);
    pm.expect(jsonData.name).to.eql("John");
    pm.expect(jsonData.location).to.eql("india");
    pm.expect(jsonData.phone).to.eql("1234567890");
    pm.expect(jsonData.courses[0]).to.eql("Java");
    pm.expect(jsonData.courses[1]).to.eql("Selenium");

});
```

Validating JSON Schema

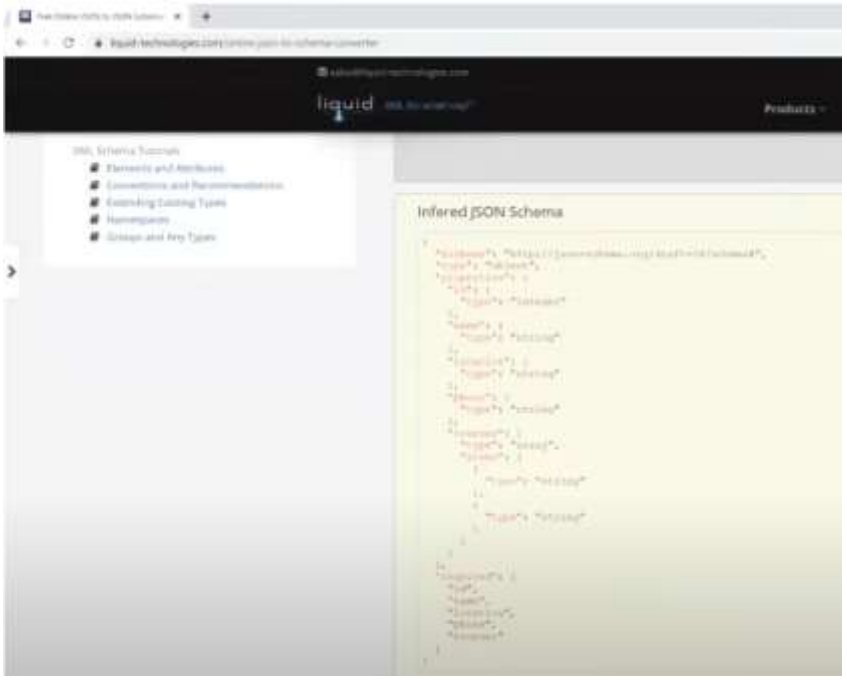
Response

```
{
    "id": 1,
    "name": "John",
    "location": "india",
    "phone": "1234567890",
    "courses": [
        "Java",
        "Selenium"
    ]
}
```

JSON schema

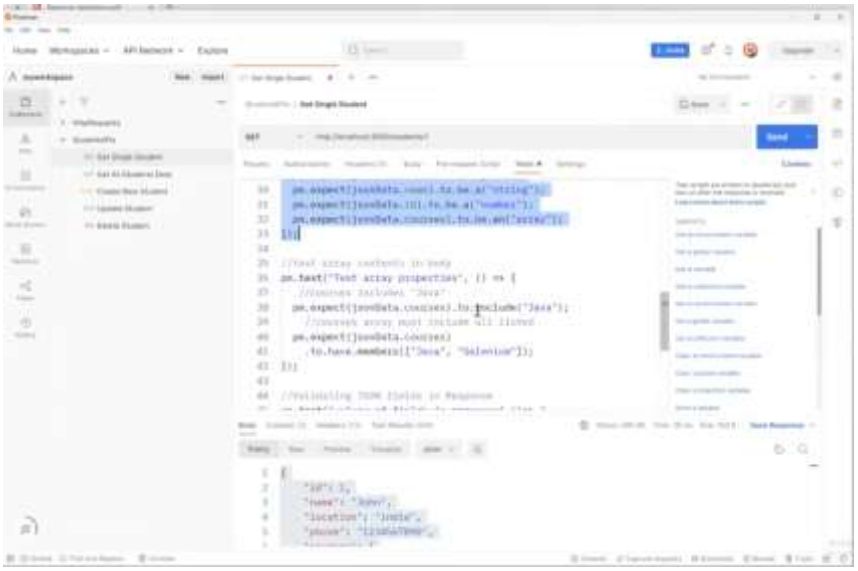
```
var schema={
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "id": {
```

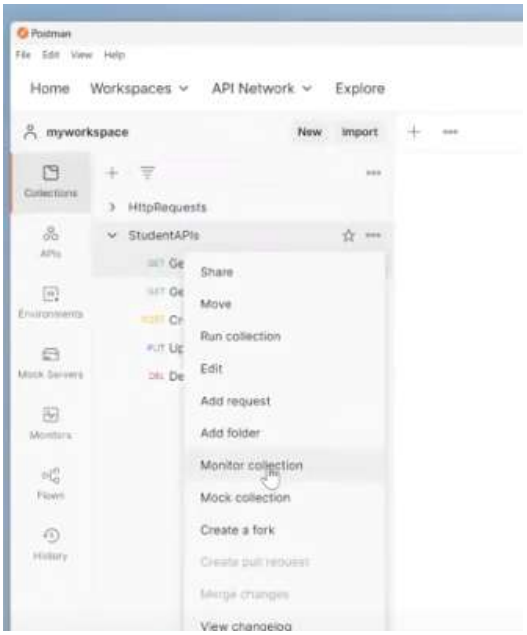
```
      "type": "integer"
    },
    "name": {
      "type": "string"
    },
    "location": {
      "type": "string"
    },
    "phone": {
      "type": "string"
    },
    "courses": {
      "type": "array",
      "items": [
        {
          "type": "string"
        },
        {
          "type": "string"
        }
      ]
    }
  },
  "required": [
    "id",
    "name",
    "location",
    "phone",
    "courses"
  ]
}
```

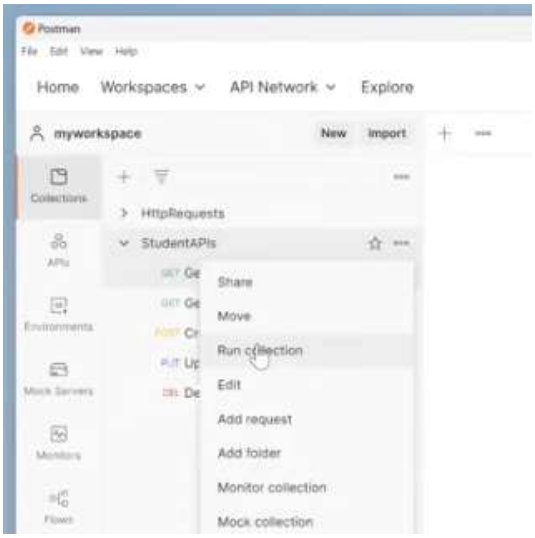


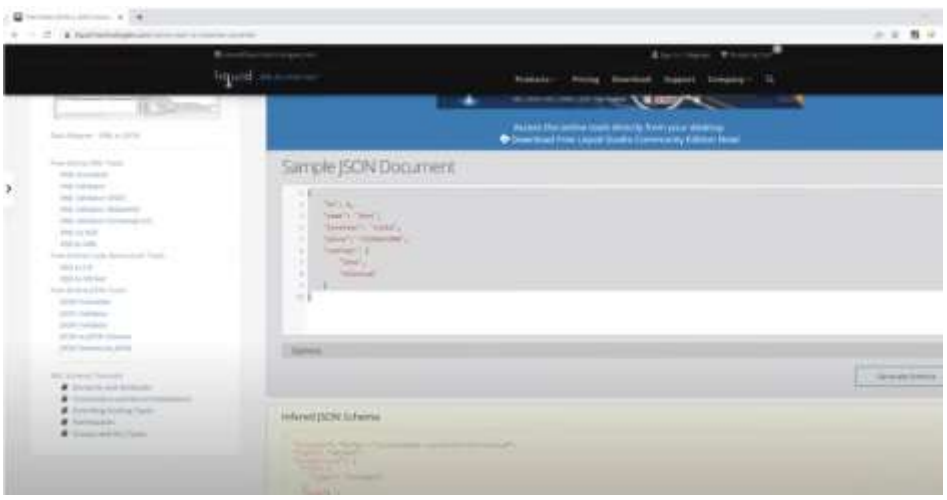
JSON schema Validation

```
pm.test('Schema is valid', function() {
  pm.expect(tv4.validate(jsonData, schema)).to.be.true;
});
```





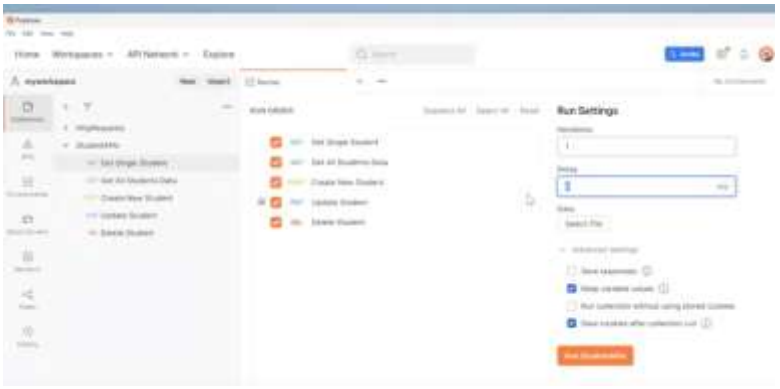


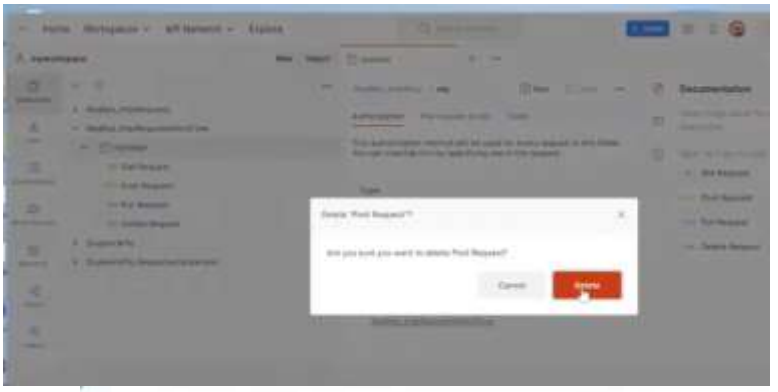


Day 5

Scripts

Pre-request scripts
Tests





Day 5

Scripts

Pre-request scripts

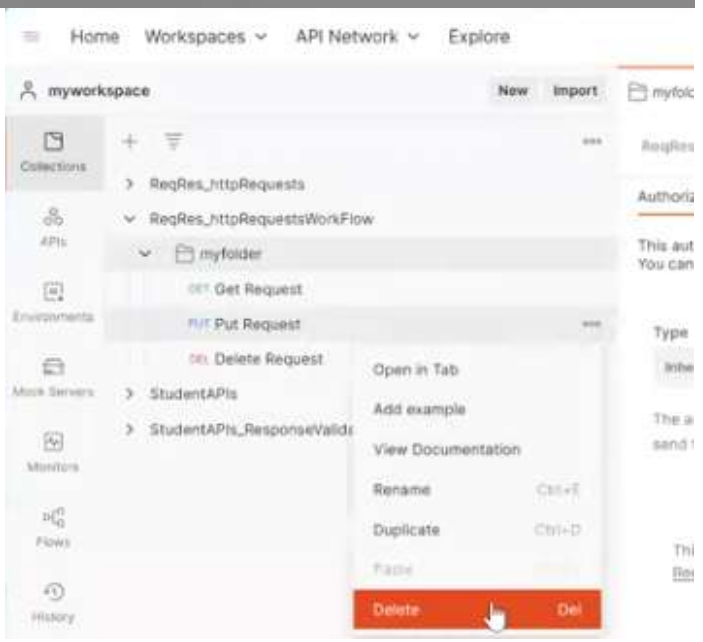
Tests

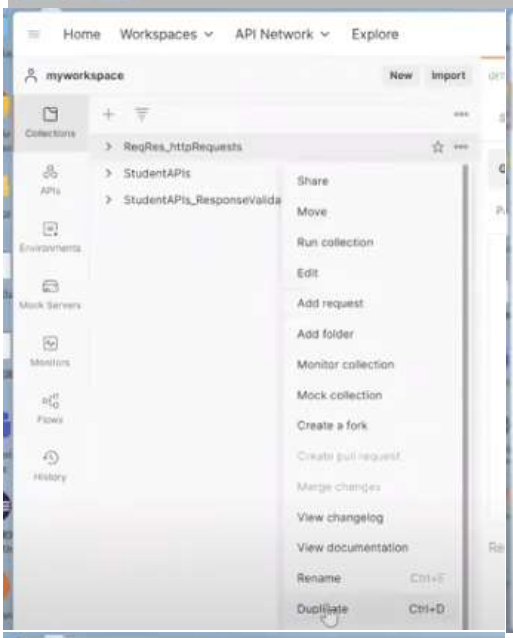
Pre-requestScript ---->Request---->Response-- Tests

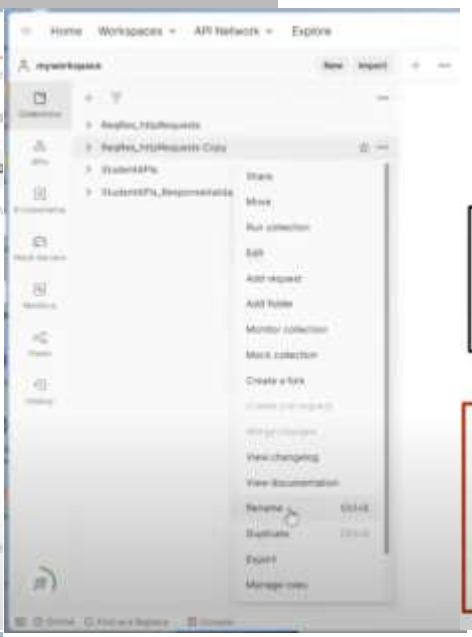
Collection

Folder

Request

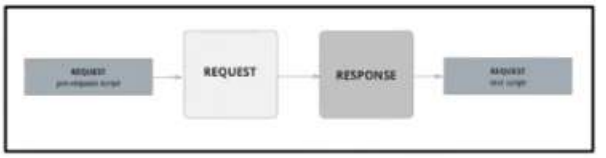


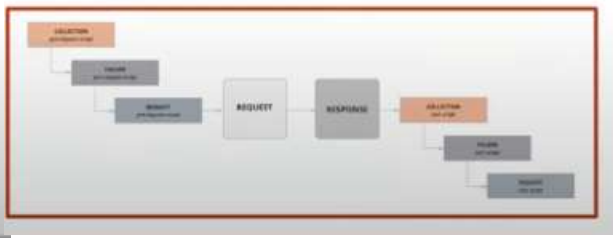


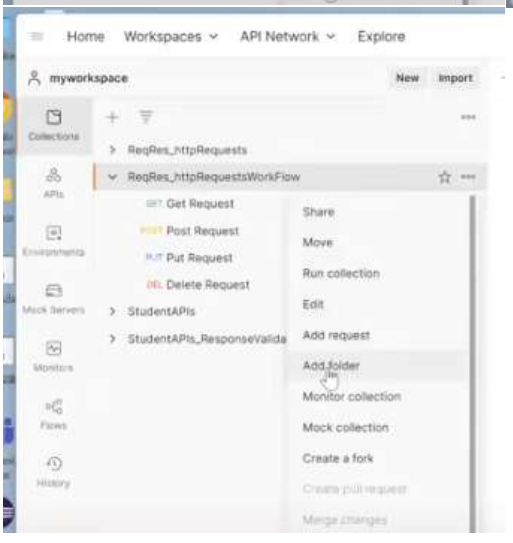


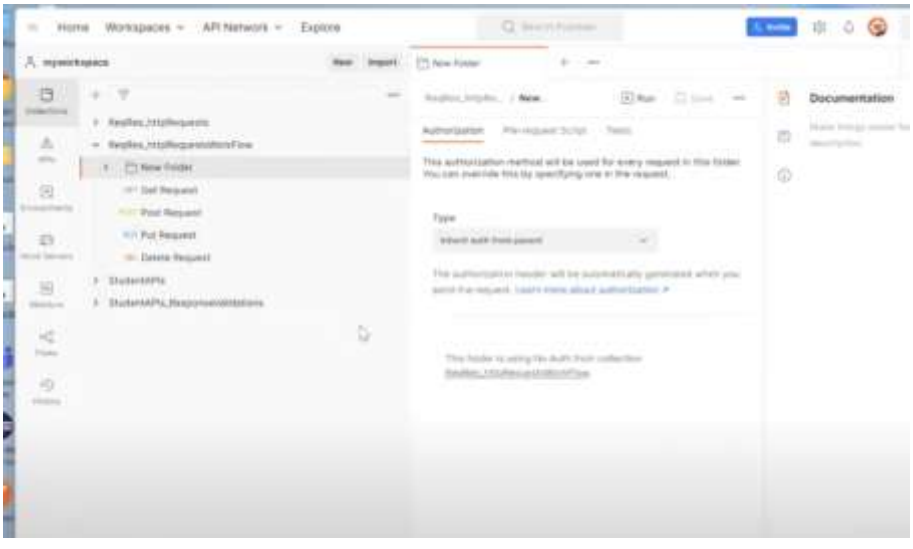
Scripts in Postman

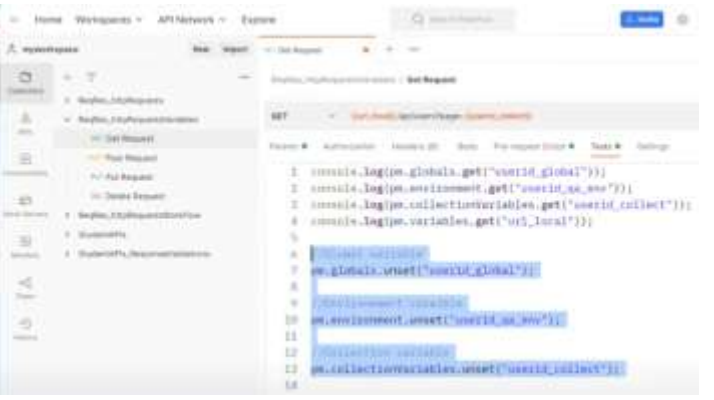
- Pre-request Scripts
- Tests Scripts

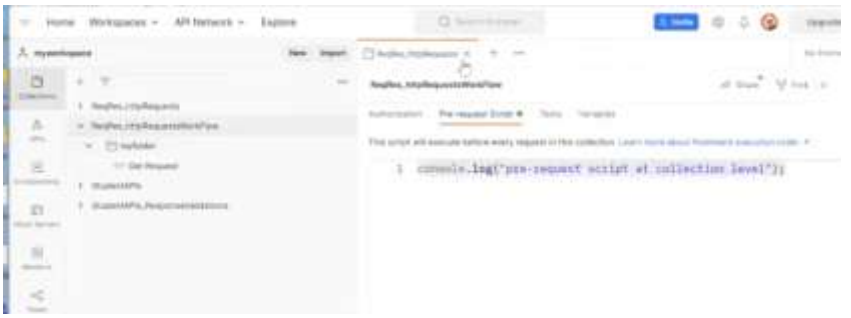


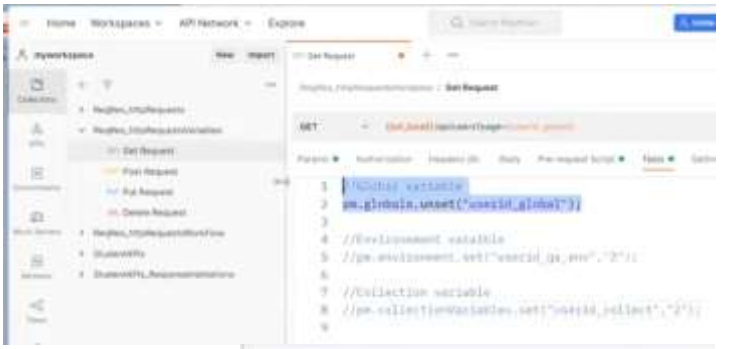












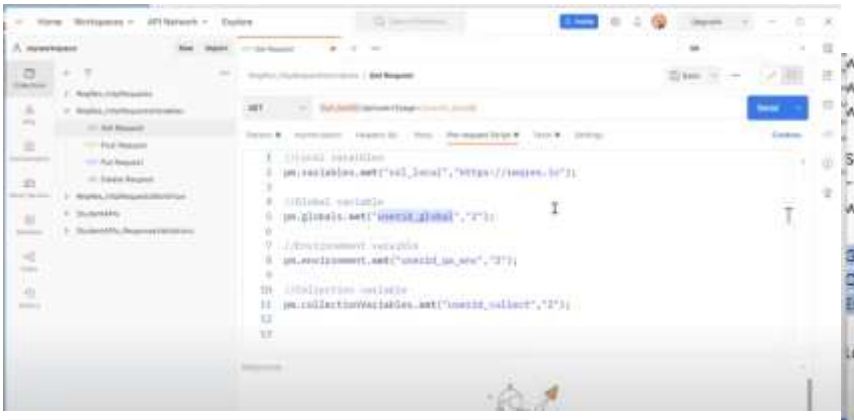
creating varaibles using pre-request scripts

//Local varaibles
pm.variables.set("url_local","https://reqres.in");

//Global variable
pm.globals.set("userid_global","2");

//Environement varaible
pm.environment.set("userid_qa_env","2");

//Collection variable
pm.collectionVariables.set("userid_collect","2");



what?
why?
where?

Scope
---
workspace--->collections--->requests

Global -- accessible in workspace
Collection -- accessible within collection
Environment -- accessible in all collections, but we need to switch to particulur enviroment

Local -- accessible only within request(specific to equest)

```
//Local varaibles
pm.variables.set("url_local","https://reqres.in");
```

```
//Global variable
//pm.globals.unset("userid_global");

//Environement varaible
//pm.environment.unset("userid_qa_env");

//Collection variable
pm.collectionVariables.unset("userid_collect");
```

capture the values from varaibles

```
console.log(pm.globals.get("userid_global"));
console.log(pm.environment.get("userid_qa_env"));
console.log(pm.collectionVariables.get("userid_collect"));
console.log(pm.variables.get("url_local"));
```

```
//Collection variable
pm.collectionVariables.set("userid_collect","2");
```

Unset/remove variables using script

```
//Global variable
//pm.globals.unset("userid_global");

//Environement varaible
//pm.environment.unset("userid_qa_env");

//Collection variable
pm.collectionVariables.unset("userid_collect");
```

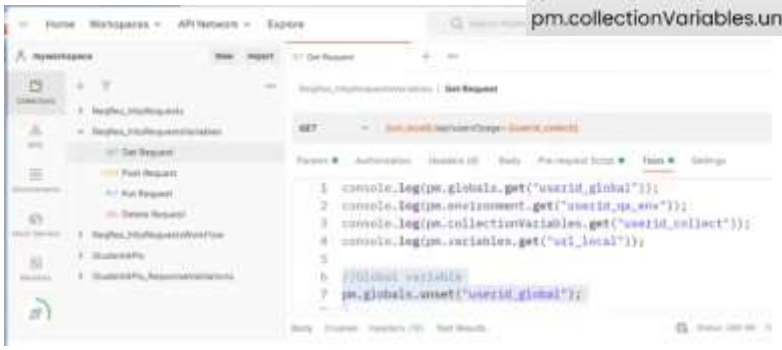
Local -- accessible only within request(specific to equest)

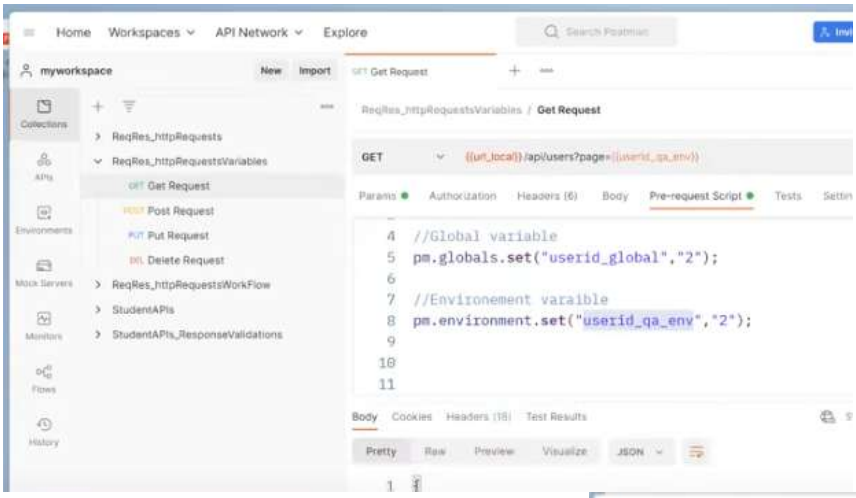
```
//Local varaibles
pm.variables.set("url_local","https://reqres.in");
```

data - external files csv/text...

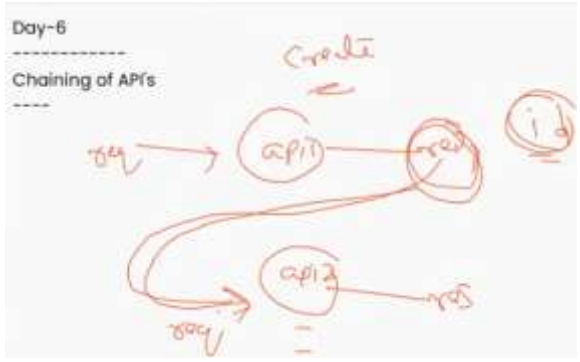
url_global
url_collect
url_qa_env

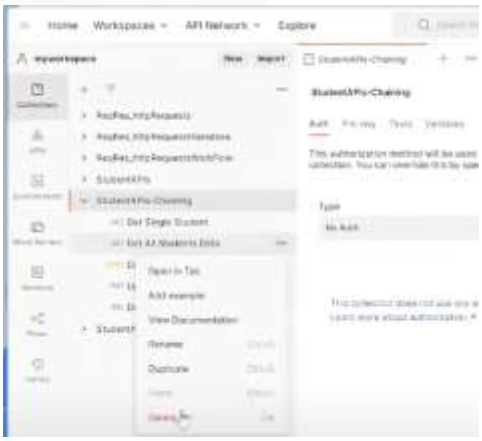
refering varaibles {{variable }}
creating varaibles uysing pre-request scripts

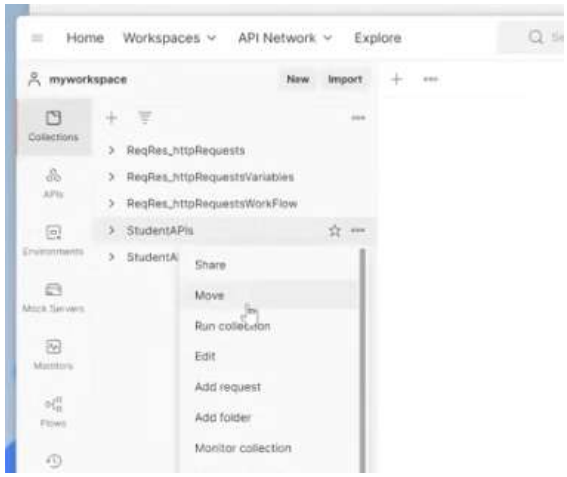


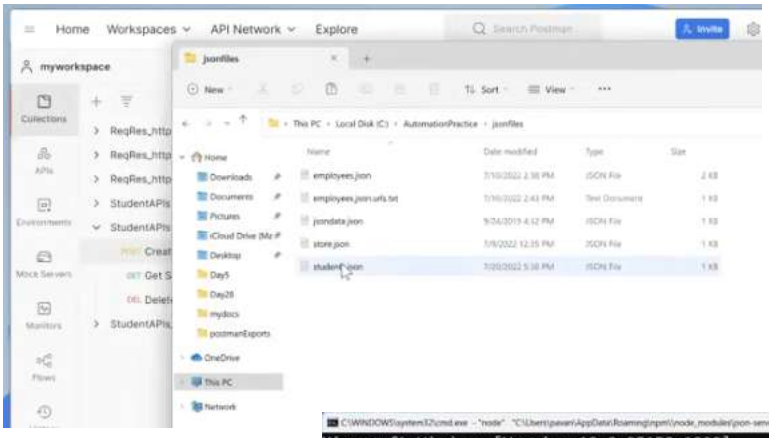


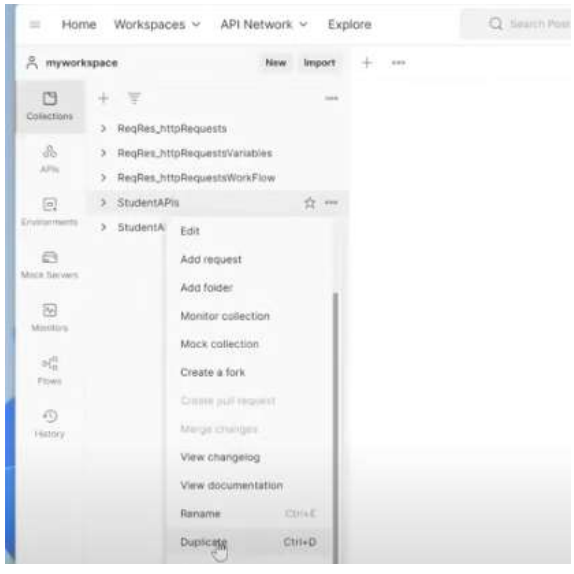

```
//pm.environment.unset("userid_qa_env");

//Collection variable
pm.collectionVariables.unset("userid_collect");

capture the values from varaibles
-----
console.log(pm.globals.get("userid_global"));
console.log(pm.environment.get("userid_qa_env"));
console.log(pm.collectionVariables.get("userid_collect"));
console.log(pm.variables.get("url_local"));


set - global, env, collection, local
unset -  global, env, collection, local
get -  global, env, collection, local
```

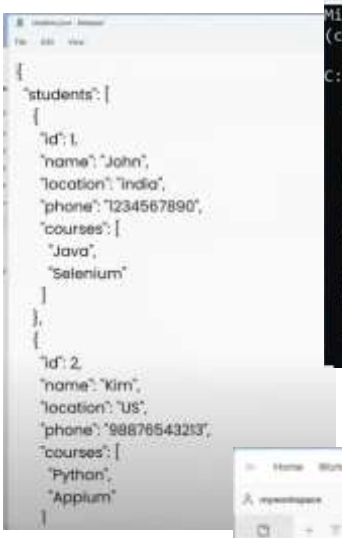




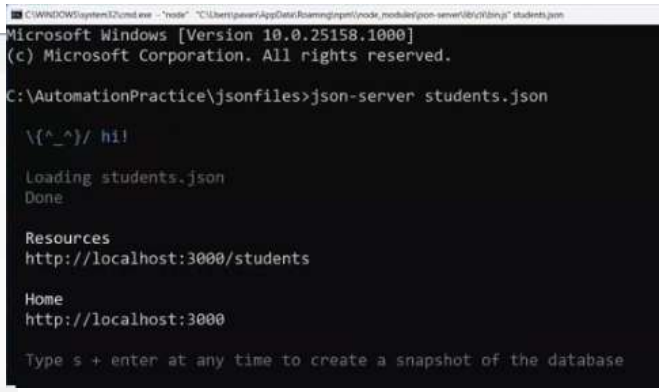




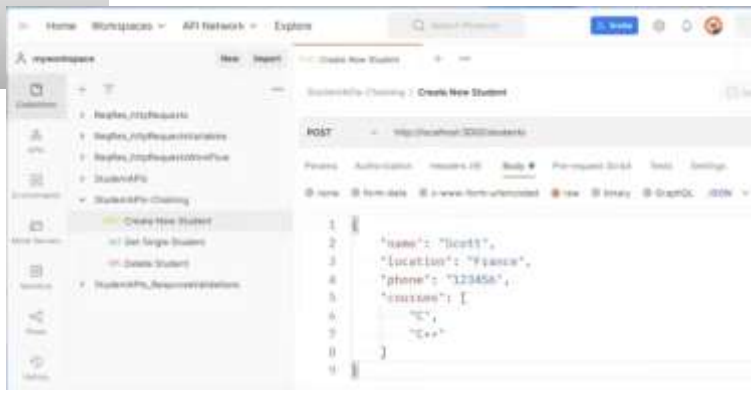


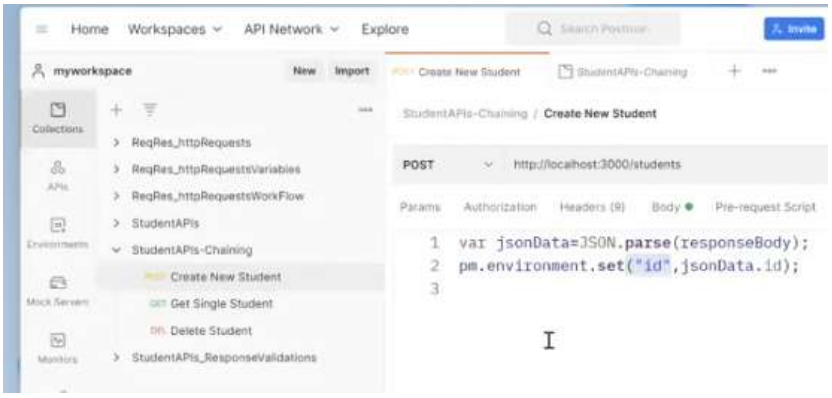


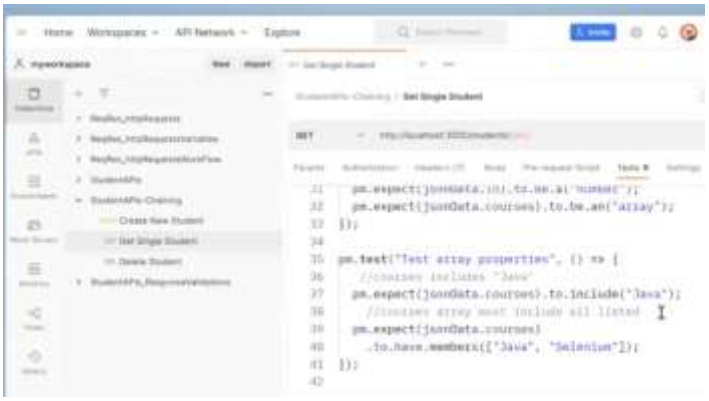
```
{
 "students": [
  {
   "id": 1,
   "name": "John",
   "location": "India",
   "phone": "1234567890",
   "courses": [
    "Java",
    "Selenium"
   ]
  },
  {
   "id": 2,
   "name": "Kim",
   "location": "US",
   "phone": "98876543213",
   "courses": [
    "Python",
    "Appium"
   ]
```

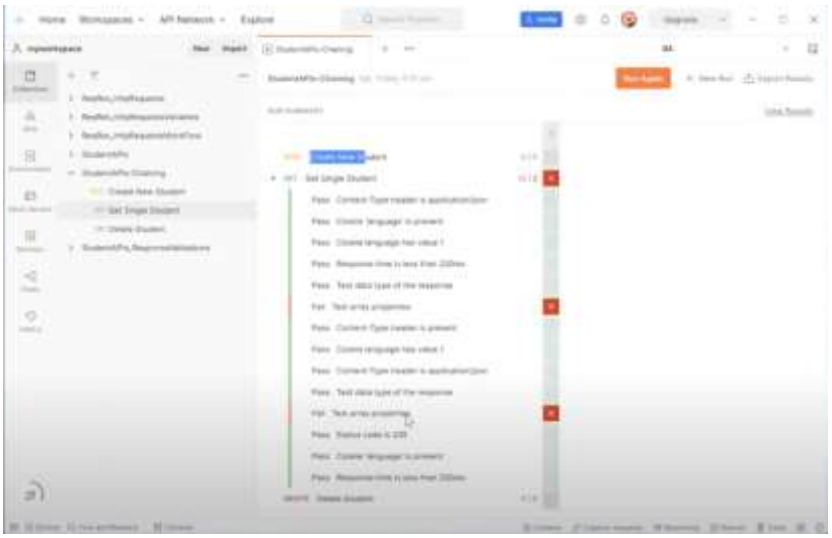


```
Microsoft Windows [Version 10.0.25158.1000]
(c) Microsoft Corporation. All rights reserved.

C:\AutomationPractice\jsonfiles>json-server students.json

  \{^_^}/ hi!

  Loading students.json
  Done

  Resources
  http://localhost:3000/students

  Home
  http://localhost:3000

  Type s + enter at any time to create a snapshot of the database
```

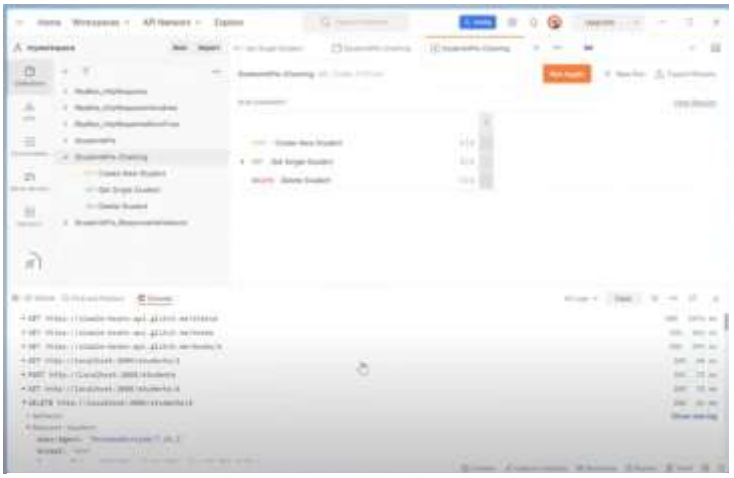


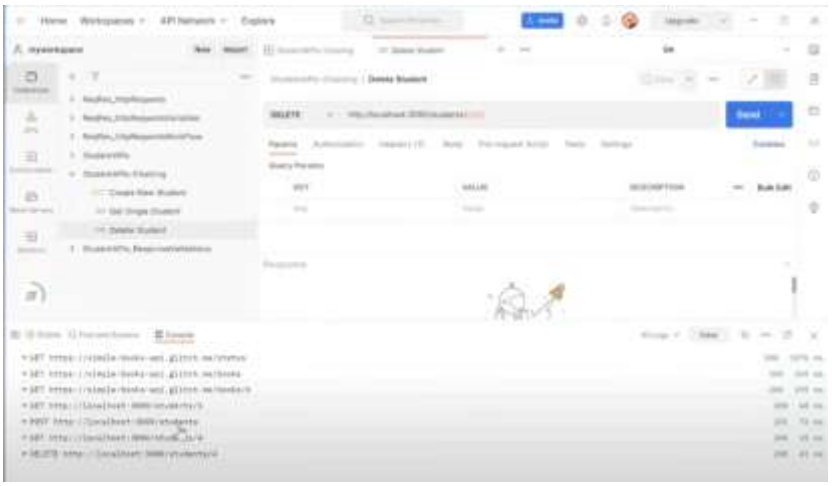

Home
Workspaces
API Network
Explore
myworkspace
New
Import
Create New Student
StudentAPIs-Chaining
Collections
ReqRes_httpRequests
ReqRes_httpRequestsVariables
ReqRes_httpRequestsWorkFlow
StudentAPIs
StudentAPIs-Chaining
Create New Student
Get Single Student
Delete Student
StudentAPIs_ResponseValidations
StudentAPIs-Chaining / Create New Student
POST
http://localhost:3000/students
Params
Authorization
Headers (9)
Body
Pre-request Script
1 var jsonData=JSON.parse(responseBody);
2 pm.environment.set("id",jsonData.id);
3

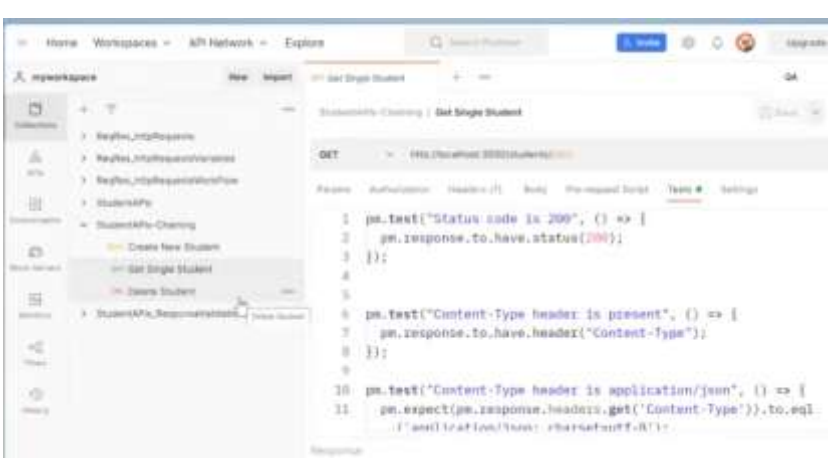

pm.expect(jsonData.id).to.be.a('number');
pm.expect(jsonData.courses).to.be.an('array');
});
pm.test("Test array properties", () => {
//courses includes "Java"
pm.expect(jsonData.courses).to.include("Java");
//courses array must include all listed
pm.expect(jsonData.courses)
.to.have.members(["Java", "Selenium"]);
});

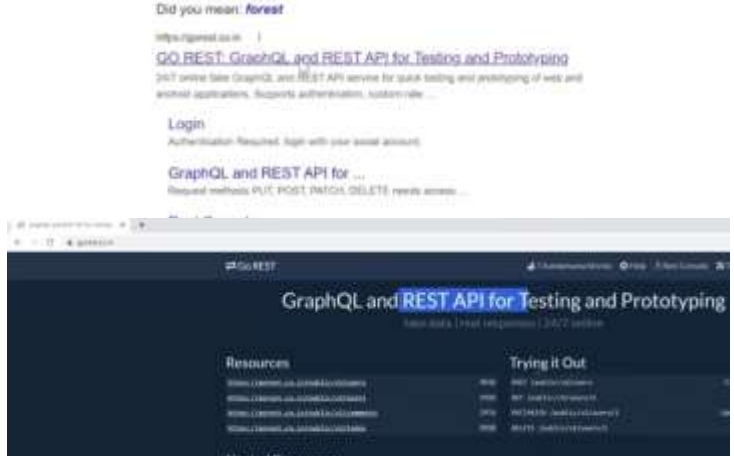



pm.test("Status code is 200", () => {
pm.response.to.have.status(200);
});

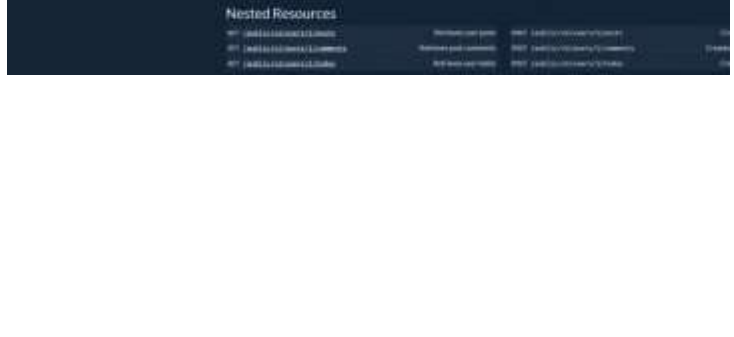







Google
gorest
Did you mean: forest
GO REST: GraphQL and REST API for Testing and Prototyping
Login
GraphQL and REST API for ...

Home
Workspaces
API Network
Explore
myworkspace
Get Single Student
Create New Student
Get Single Student
Delete Student
pm.test("Status code is 200", () => {
pm.response.to.have.status(200);
});
pm.test("Content-Type header is present", () => {
pm.response.to.have.header("Content-Type");
});
pm.test("Content-Type header is application/json", () => {
pm.expect(pm.response.headers.get('Content-Type')).to.eql


GraphQL and REST API for Testing and Prototyping
Resources
Trying it Out
Nested Resources

Day-6

Chaining of API's

Student API

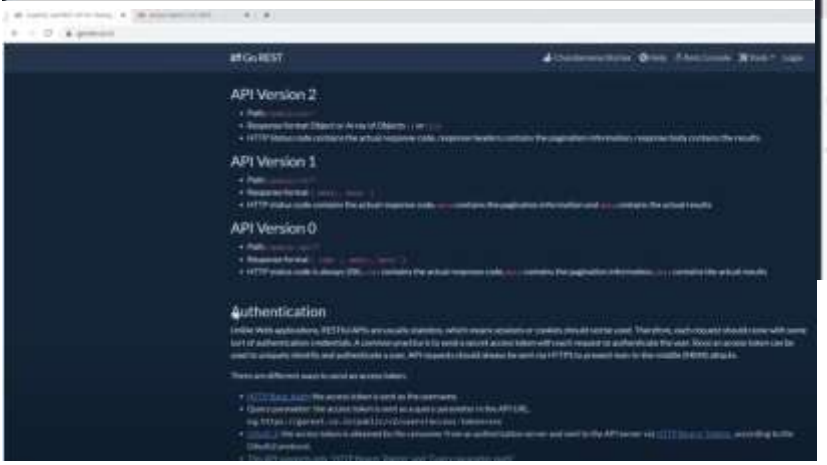
Gorest API

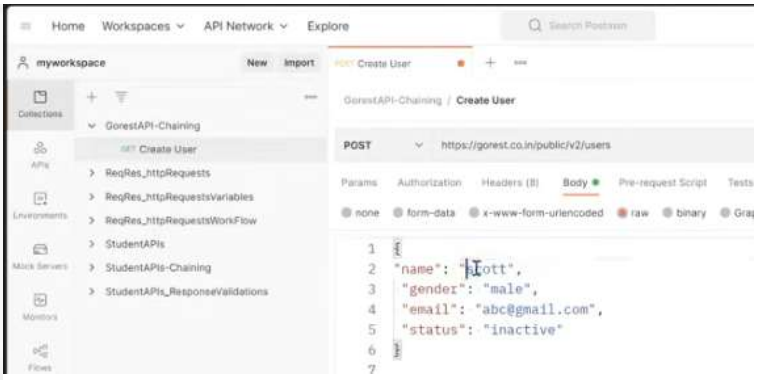
---

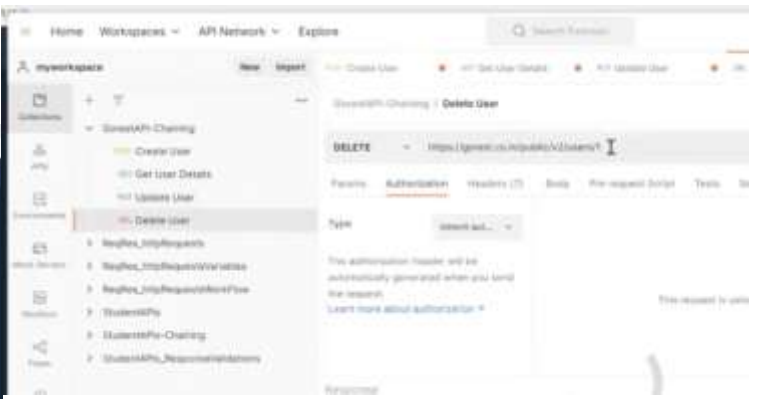
url: https://gorest.co.in

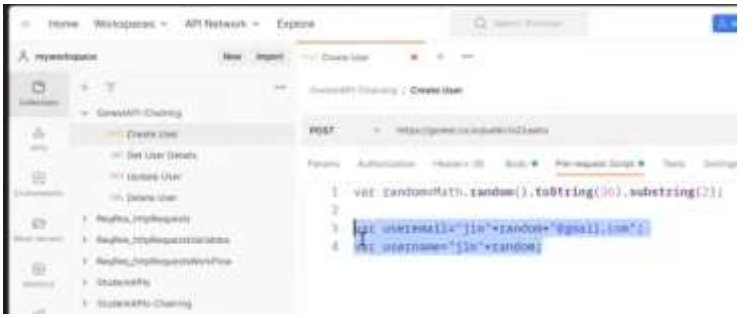
POST	/public/v2/users	Create a new user
GET	/public/v2/users/1	Get user details
PUT|PATCH	/public/v2/users/1	Update user details
DELETE	/public/v2/users/1	Delete user

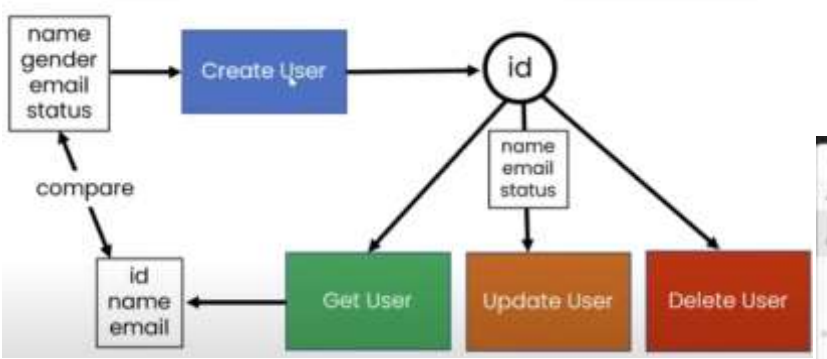
Token: c35e10e748c6f113775527bcef204e9929b4c9f4b995a8ee253eec46aed57b06

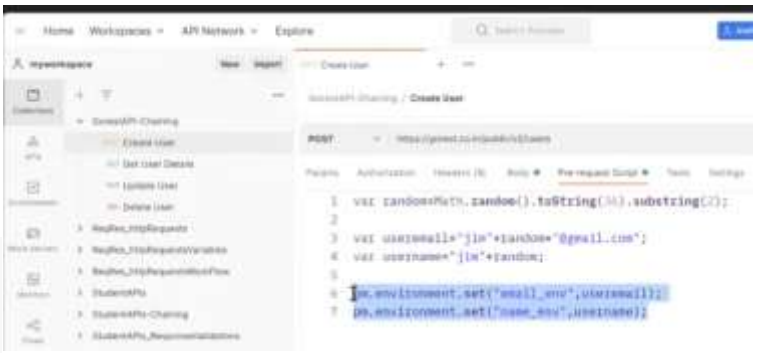












---

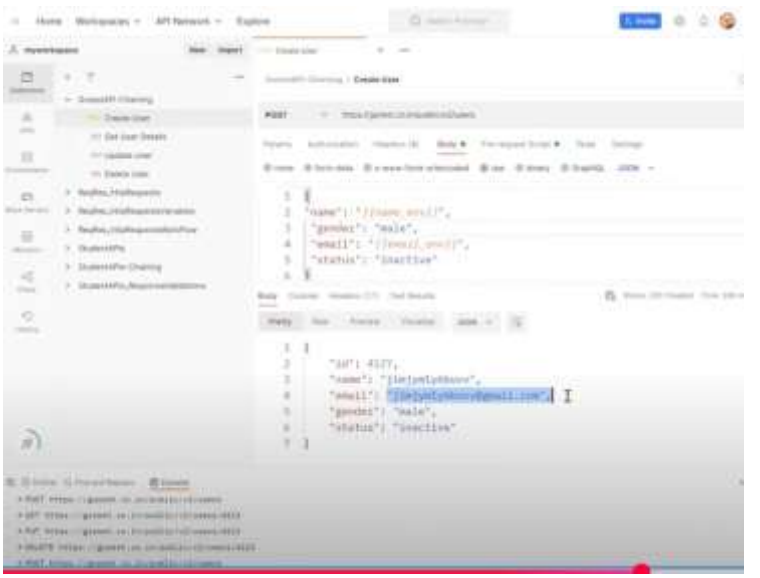
url: https://gorest.co.in

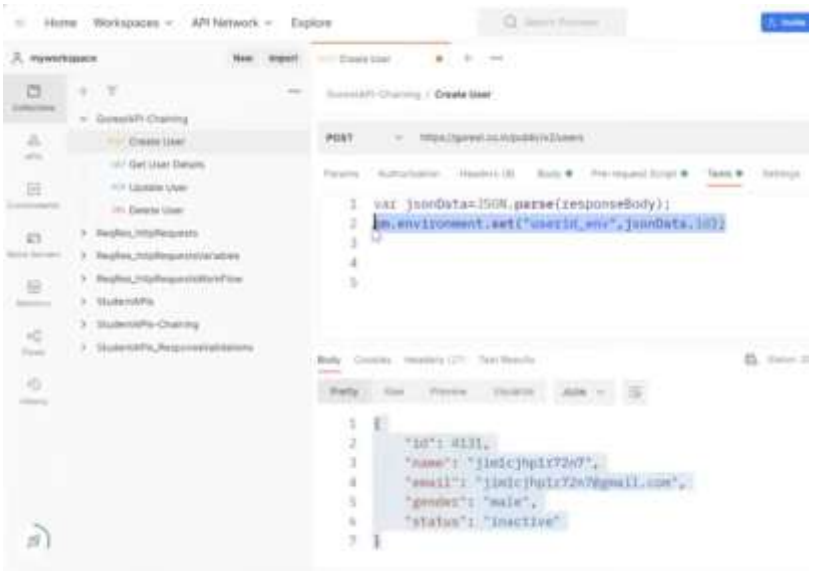
POST	/public/v2/users	Create a new user
GET	/public/v2/users/1	Get user details
PUT|PATCH	/public/v2/users/1	Update user details
DELETE	/public/v2/users/1	Delete user

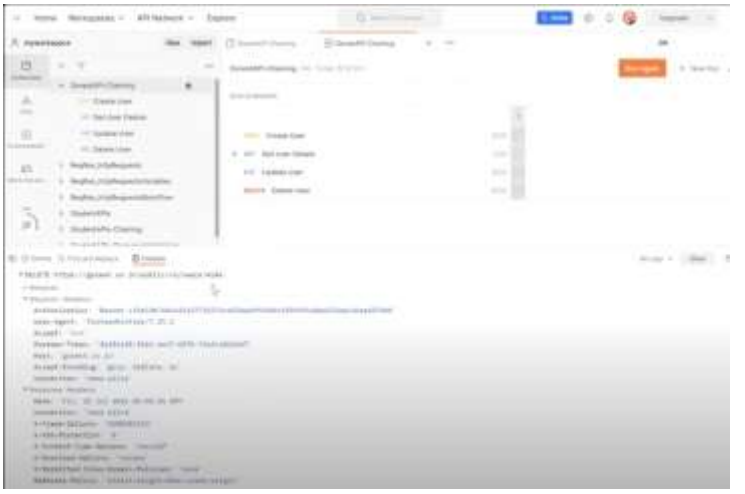
Token: c35e10e748c6f113775527bcef204e9929b4c9f4b995a8ee253eec46aed57b06

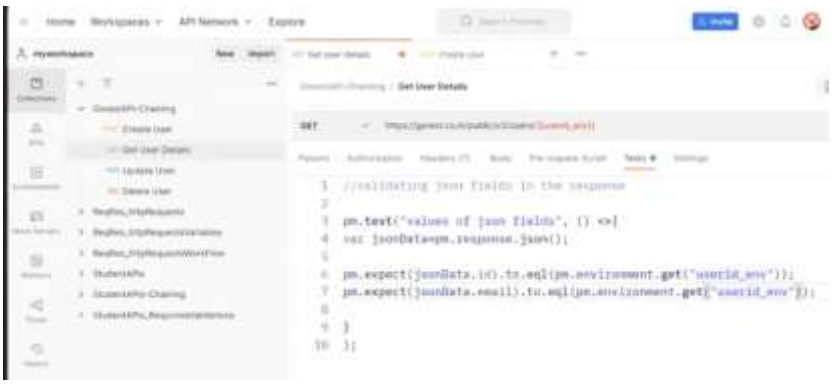
Request Body

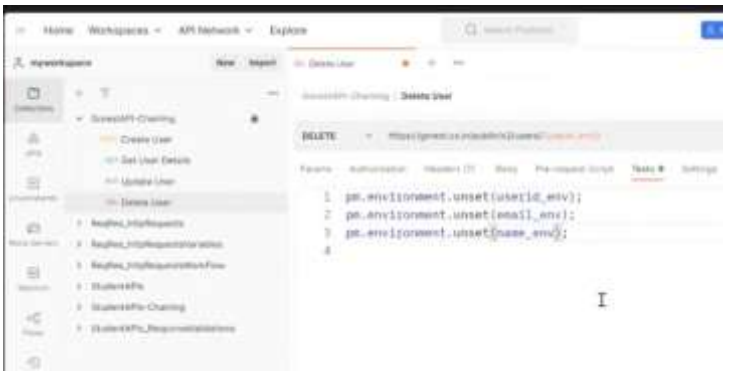
```
{
"name": "scott",
"gender": "male",
"email": "abc@gmail.com",
"status": "inactive"
}
```




var jsonData=JSON.parse(responseBody);
pm.environment.set("userid_env",jsonData.id)




//validating json fields in the response
pm.test("values of json fields", () =>{
var jsonData=pm.response.json();
pm.expect(jsonData.id).to.eql(pm.environment.get("userid_env"));
pm.expect(jsonData.email).to.eql(pm.environment.get("userid_env"));
});


pm.environment.unset(userid_env);
pm.environment.unset(email_env);
pm.environment.unset(name_env);


//validating json fields in the response
pm.test("values of json fields", () =>{
var jsonData=pm.response.json();
pm.expect(jsonData.id).to.eql(pm.environment.get("userid_env"));
pm.expect(jsonData.email).to.eql(pm.environment.get("userid_env"));
});


Run Settings


API Chaining
Example 1: Students API
Step1: Send Post request (create student) then collect "id" from response.
http://localhost:3000/students
Request body
{
"name": "Scott",
"location": "France",
"phone": "123454",
"courses": [
"C",
"C++"
]
}
Tests
Include below script in Tests to collect id as environment variable.
//Capturing id from response
var jsonData = JSON.parse(responseBody);
pm.environment.set("id",jsonData.id);
Step2: Send Get request (display student) using for "id" captured from previous request.
http://localhost:3000/students/{id}

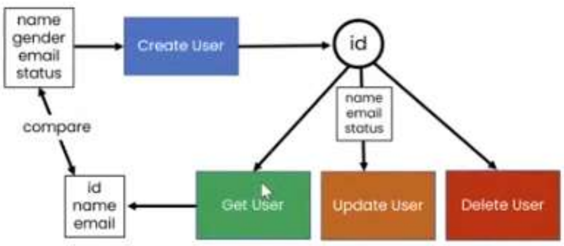
Example 2: GoRest API (Users)

https://gorest.co.in/

Access Token :
c35e10e748c6f113775527bcef204e9929b4c9f4b995a8ee25
3eec46aed57b06

https://gorest.co.in

POST	/public/v2/users	Create a new user
GET	/public/v2/users/23	Get user details
PUT|PATCH	/public/v2/users/23	Update user details
DELETE	/public/v2/users/23	Delete user



Create User (Post)

{{url}}/public/v2/users

Request Body

```
{
    "name": "{{name_env}}",
    "gender": "male",
    "email": "{{email_env}}",
    "status": "inactive"
}
```


Create User (Post)

{{url}}/public/v2/users

Request Body

```
{
    "name": "{{name_env}}",
    "gender": "male",
    "email": "{{email_env}}",
    "status": "inactive"
}
```


Pre-Request script

```
var random=Math.random().toString(36).substring(2);

var useremail="Jim"+random+"@gmail.com";
var username="Jim"+random;
```

```
//console.log(useremail);   // print to know value
//console.log(username);    // print to know value

pm.environment.set("email_env",useremail);
pm.environment.set("name_env",username)
```

Tests

```
//Capturing id from response & Set as environment var
var jsonData = JSON.parse(responseBody);
pm.environment.set("userid_env",jsonData.id);
```

Get User details (Get)

{{url}}/public/v2/users/{{userid_env}}

Tests

```
//Validating JSON fields in Response
pm.test("values of fields in response", () => {
    var jsonData = pm.response.json();

    pm.expect(jsonData.id).to.eql(pm.environment.get("userid_env"));
    pm.expect(jsonData.email).to.eql(pm.environment.get("email_env"));
    pm.expect(jsonData.name).to.eql(pm.environment.get("name_env"));
});
```

Update User details (Put)

{{url}}/public/v2/users/{{userid_env}}

Request Body

```
{
    "name": "{{name_env}}",
    "email": "{{email_env}}",
    "status": "active"
}
```

Pre-Request script

```
var random=Math.random().toString(36).substring(2);

var useremail="Jim"+random+"@gmail.com";
var username="Jim"+random;

pm.environment.set("email_env",useremail);
pm.environment.set("name_env",username)
```

Delete user (Delete)

{{url}}/public/v2/users/{{userid_env}}

Day-7

Books API
----

https://simple-books-api.glitch.me

Books API

This API allows you to reserve a book.

The API is available at https://simple-books-api.glitch.me

API Authentication

To **submit or view an order**, you need to register your API client.

POST /api-clients/

The request body needs to be in JSON format and include the following properties:

- clientName - String
- clientEmail - String

Example

```
{
  "clientName": "Postman",
  "clientEmail": "valentin@example.com"
}
```

The response body will contain the access token. The access token is valid for **7 days**.

Possible errors

Status code 409 - "API client already registered." Try changing the values for clientEmail and clientName to something else.

Endpoints

Status

GET /status
Returns the status of the API.

List of books

GET /books
Returns a list of books.

Optional query parameters:

- type: fiction or non-fiction
- limit: a number between 1 and 20.

Get a single book

GET /books/:bookId
Retrieve detailed information about a book.

Status

GET /status
Returns the status of the API.

List of books

GET /books
Returns a list of books.

Optional query parameters:

- type: fiction or non-fiction
- limit: a number between 1 and 20.

Get a single book

GET /books/:bookId
Retrieve detailed information about a book.

Submit an order

POST /orders
Allows you to submit a new order. Requires authentication.

The request body needs to be in JSON format and include the following properties:

- bookId - Integer - Required
- customerName - String - Required

Example

The response body will contain the order id.

Get all orders

GET /orders
Allows you to view all orders. Requires authentication.

Get an order

GET /orders/:orderId
Allows you to view an existing order. Requires authentication.

Update an order

PATCH /orders/:orderId
Update an existing order. Requires authentication.

The request body needs to be in JSON format and allows you to update the following properties:

- customerName - String

Example

Delete an order

DELETE /orders/:orderId

Bilingual

Allows you to view all orders. Requires authentication.

Get an order

GET `/orders/:orderId`

Allows you to view an existing order. Requires authentication.

Update an order

PATCH `/orders/:orderId`

Update an existing order. Requires authentication.

The request body needs to be in JSON format and allows you to update the following properties:

- `customerName` - String

Example

```
PATCH /orders/PF6MflPDcuHWobZcgm3y5
Authorization: Bearer <YOUR TOKEN>
```

```
{
  "customerName": "John"
}
```

Delete an order

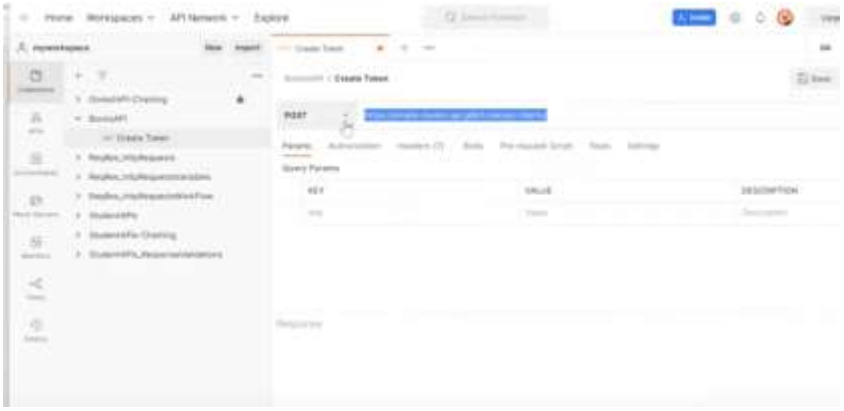
DELETE `/orders/:orderId`

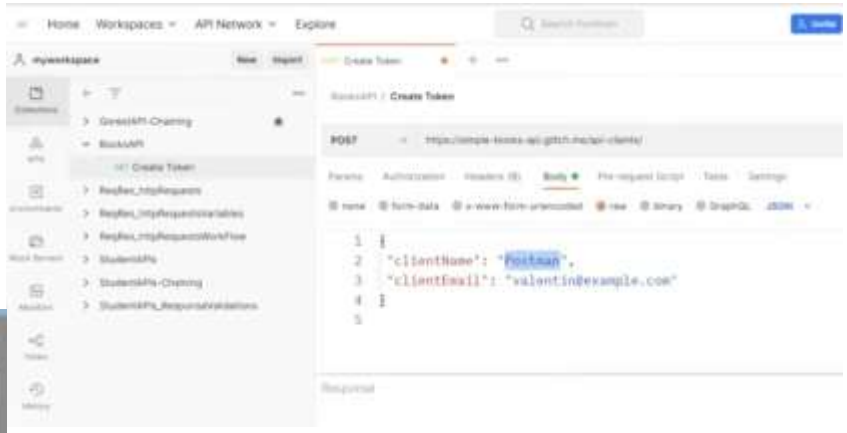
Delete an existing order. Requires authentication.

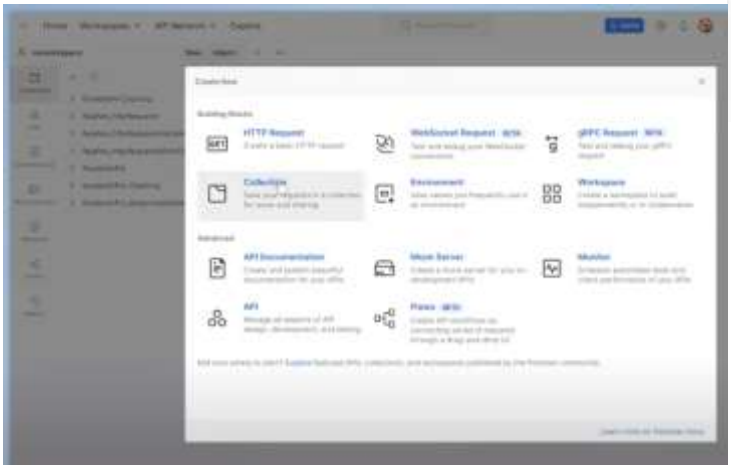
The request body needs to be empty.

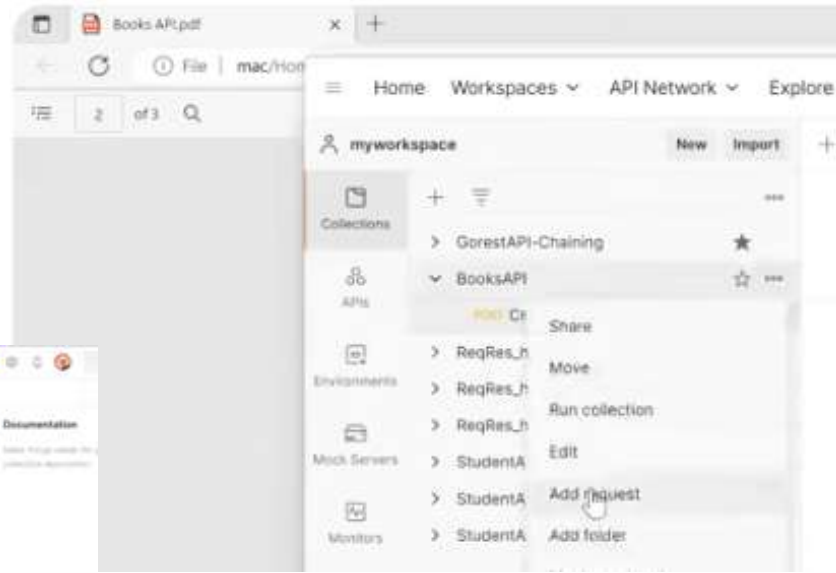
Example

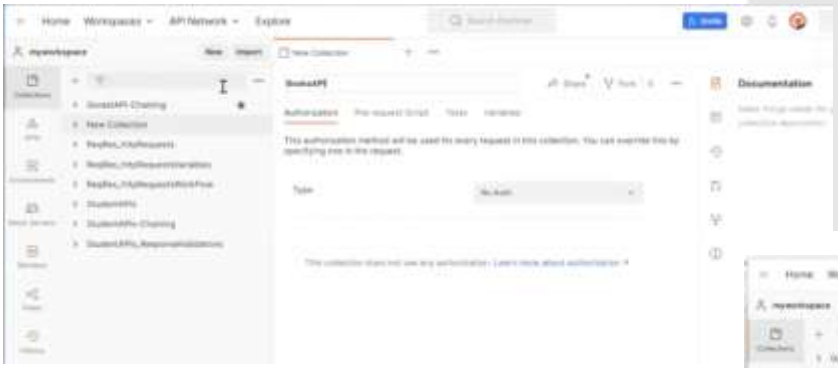
```
DELETE /orders/PF6MflPDcuHWobZcgm3y5
Authorization: Bearer <YOUR TOKEN>
```

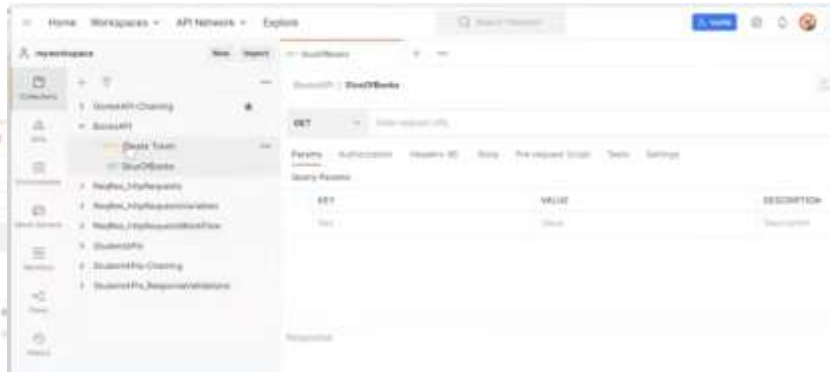


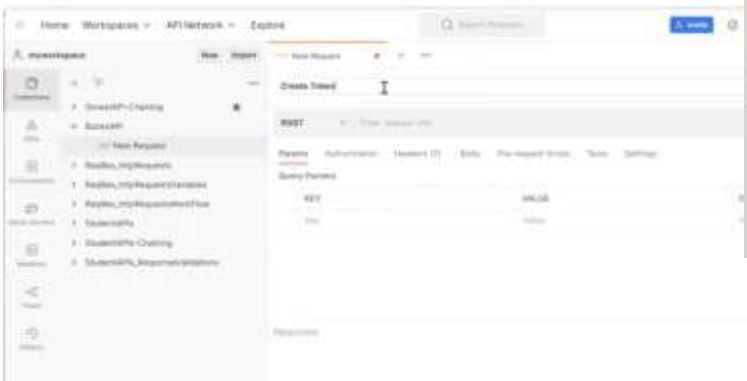


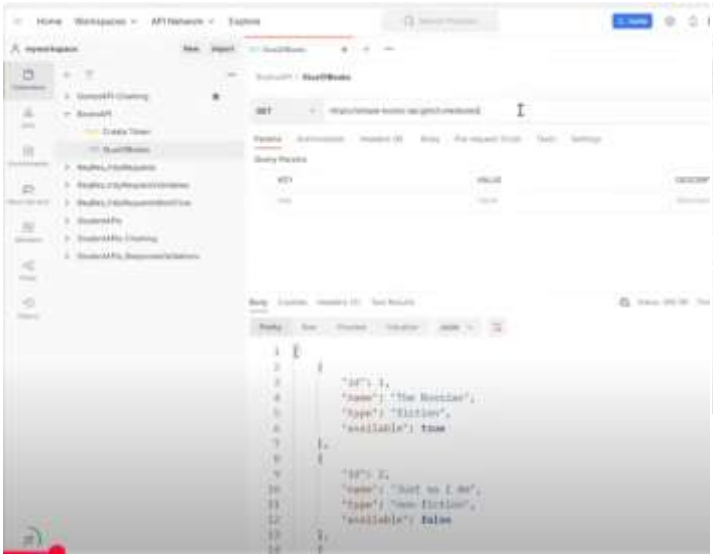


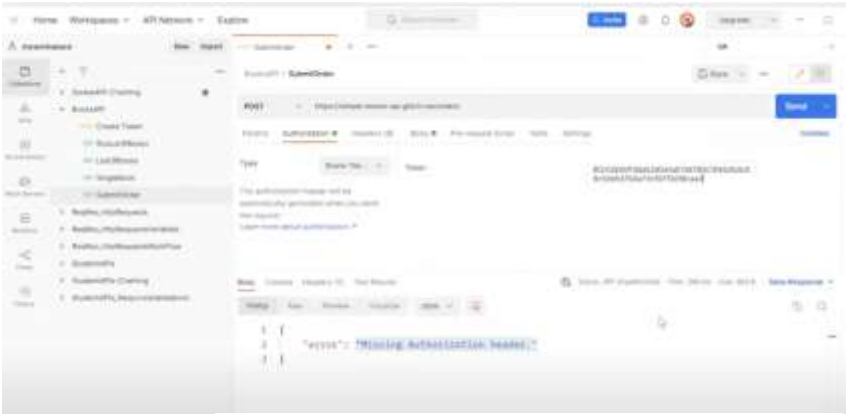


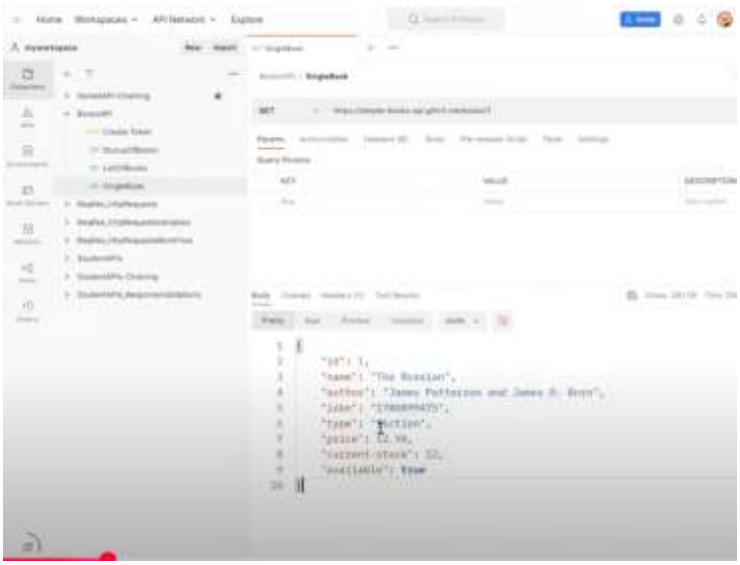












Books API Data Driven Test

(Data Parameters – JSON/csv)

Step1 :

Request Type: POST

URL: https://simple-books-api.glitch.me/orders/

Request Body:

```
{
  "bookId": "{{BookID}}",
  "customerName": "{{CustomerName}}"
}
```

Tests

```
pm.test("Status code is 201", () => {
  pm.response.to.have.status(201);
});

var jsonData=JSON.parse(responseBody);
pm.environment.set("orderid_env",jsonData.orderI
d);
```

Step 2:

Request Type: GET

URL: https://simple-books-api.glitch.me/orders/{{orderid_env}}

Tests

```
pm.test("Status code is 200", () => {
  pm.response.to.have.status(200);
});

pm.test("check orderID present in response body"
, () =>{
var jsonData=pm.response.json();
pm.expect(jsonData.id).to.eql(pm.environment.get
("orderid_env"));
});
```

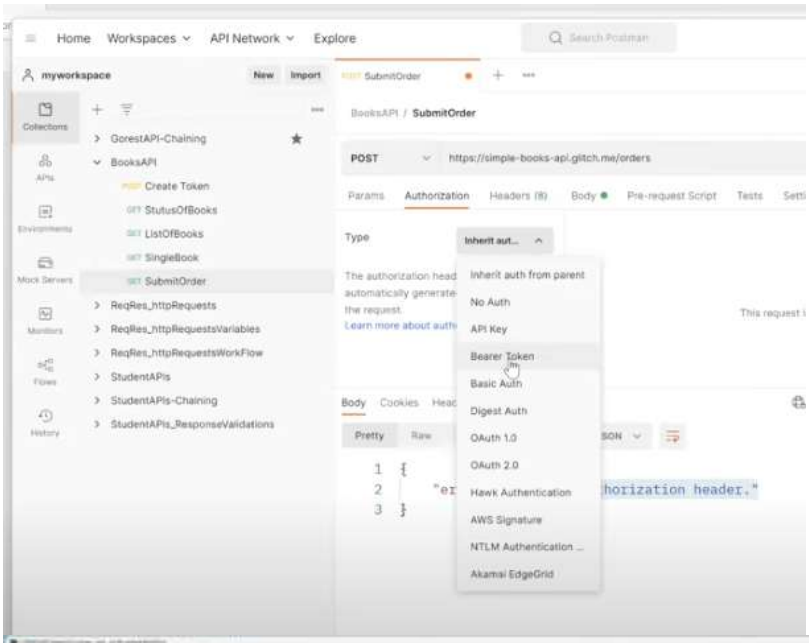
Step 3:

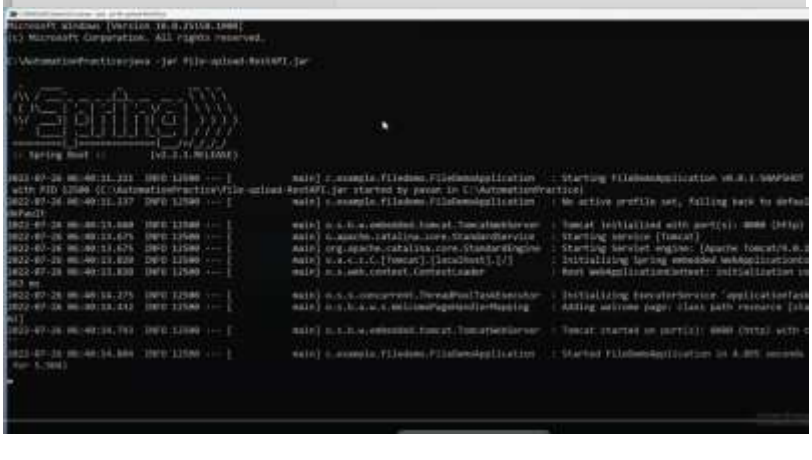
Request Type: DELETE

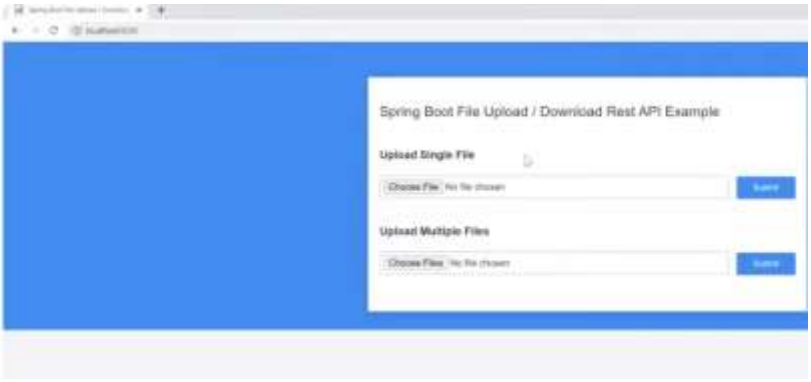
URL: https://simple-books-api.glitch.me/orders/{{orderid_env}}

Tests

```
pm.test("Status code is 204", () => {
  pm.response.to.have.status(204);
});

//Unset orderid environmetn var
pm.environment.unset("orderid_env");
```







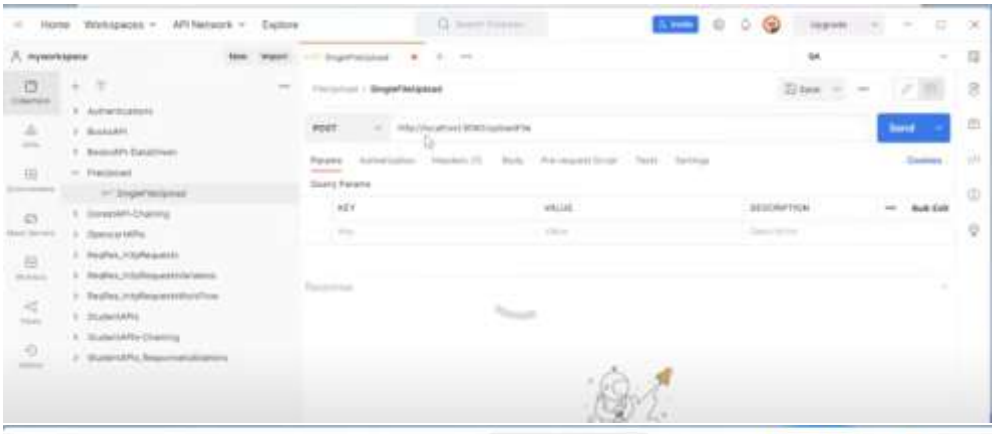
Day-8

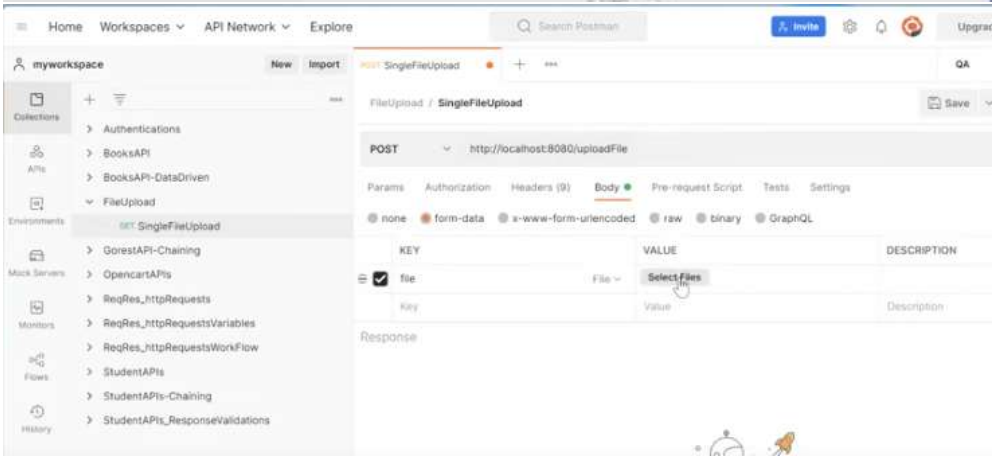
---

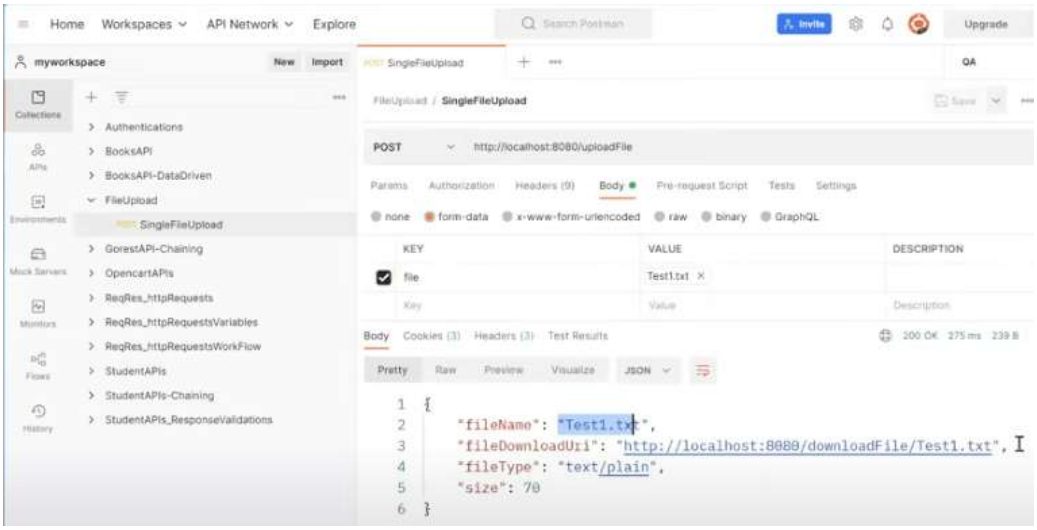
1) Upload file

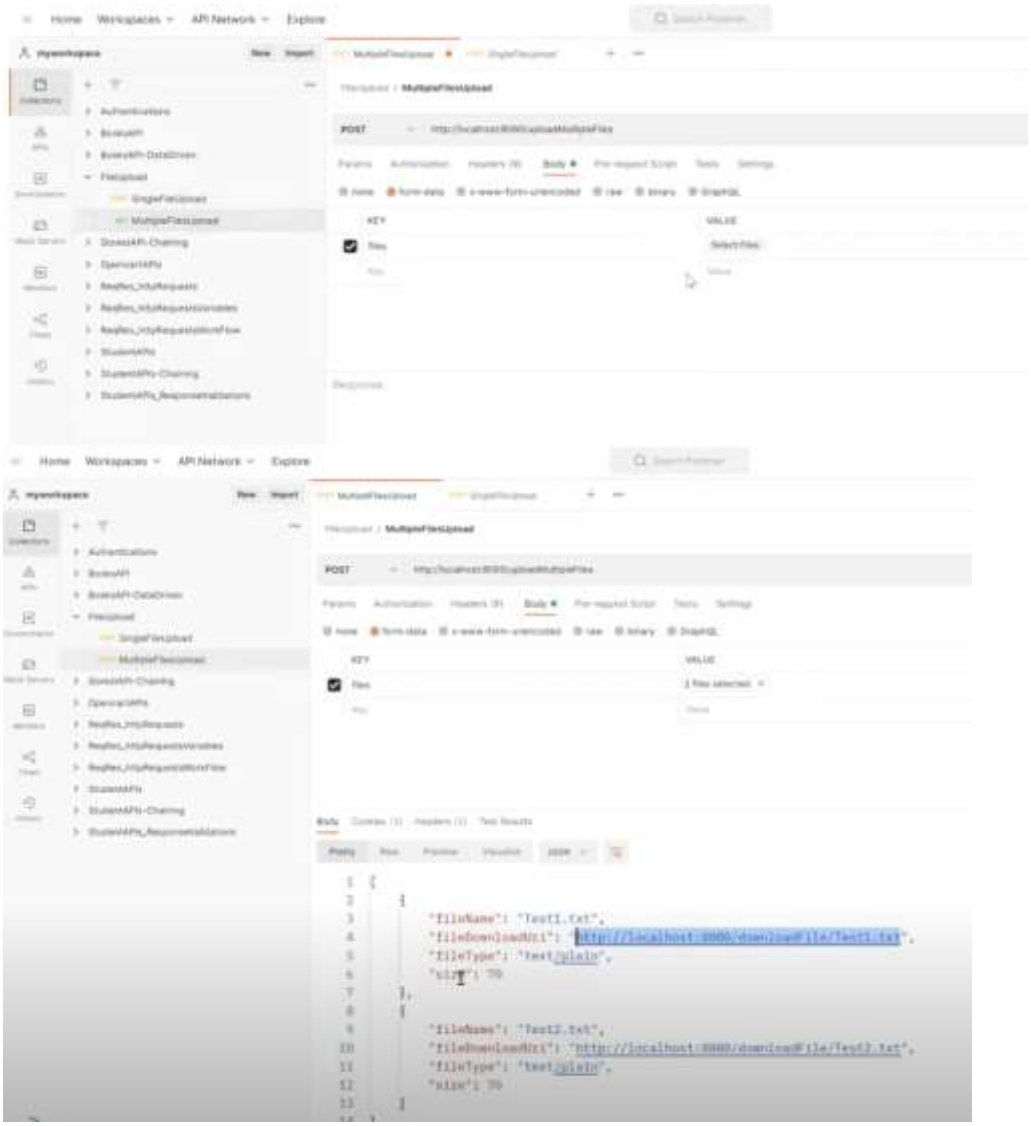
Post : http://localhost:8080/uploadFile

POST: http://localhost:8080/uploadMultipleFiles

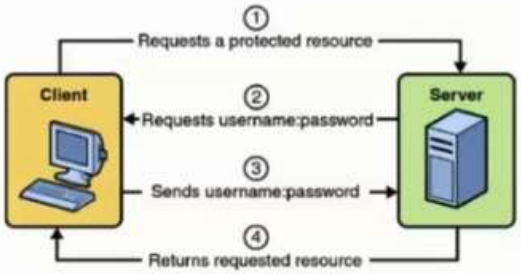


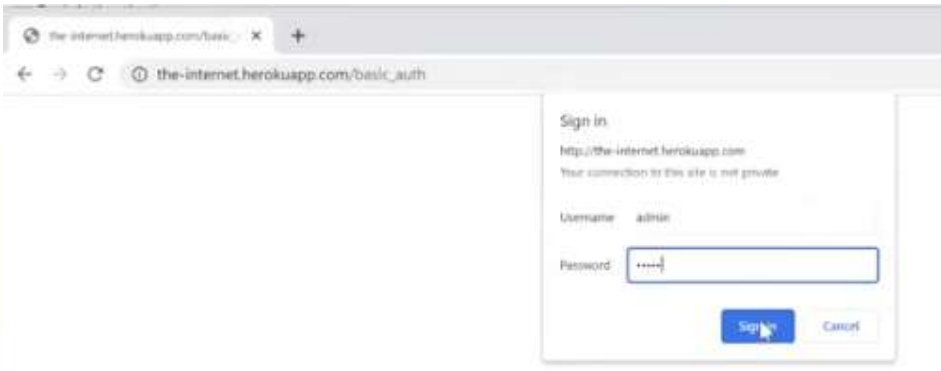


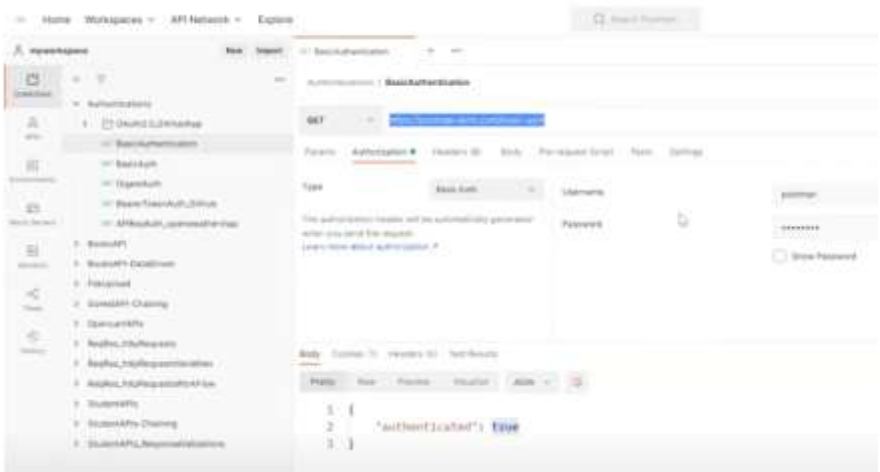


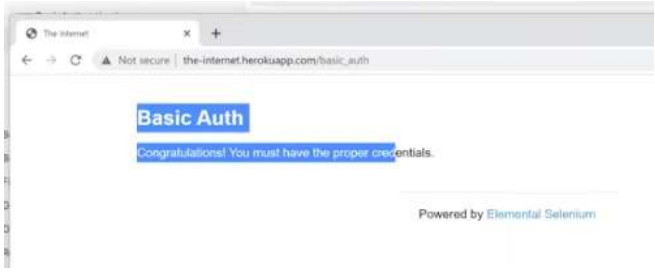


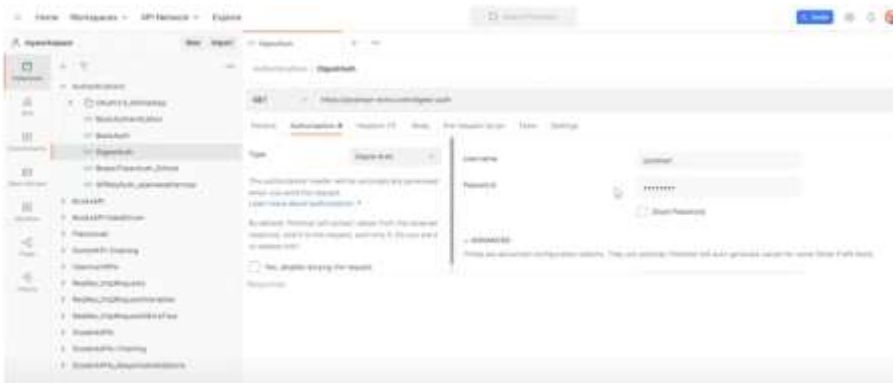
Basic Vs Digest authentication



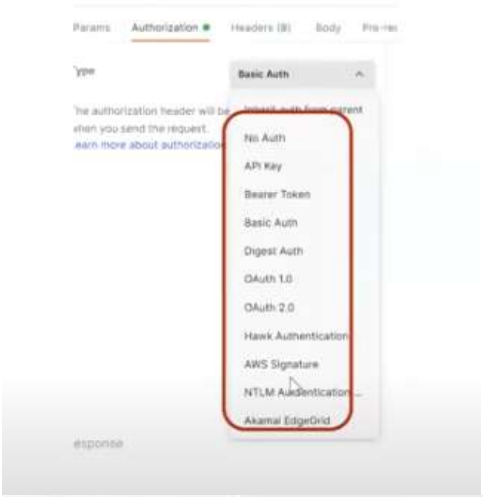


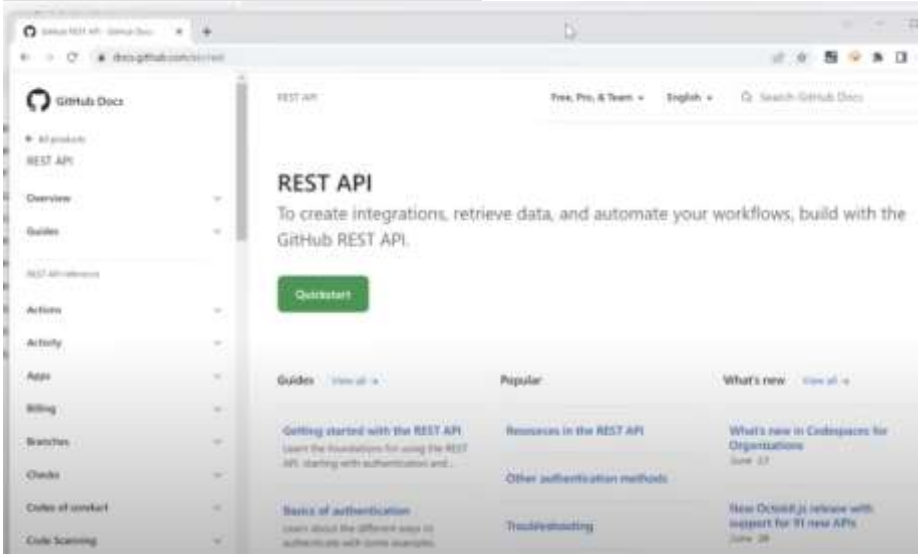


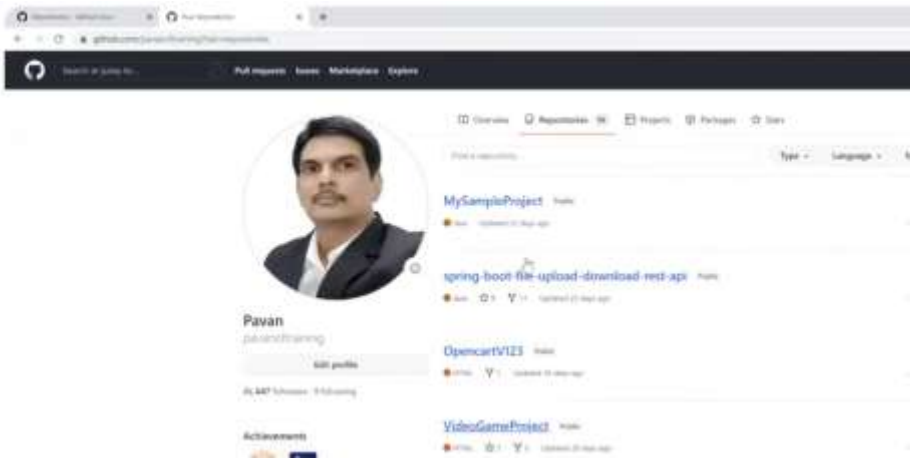


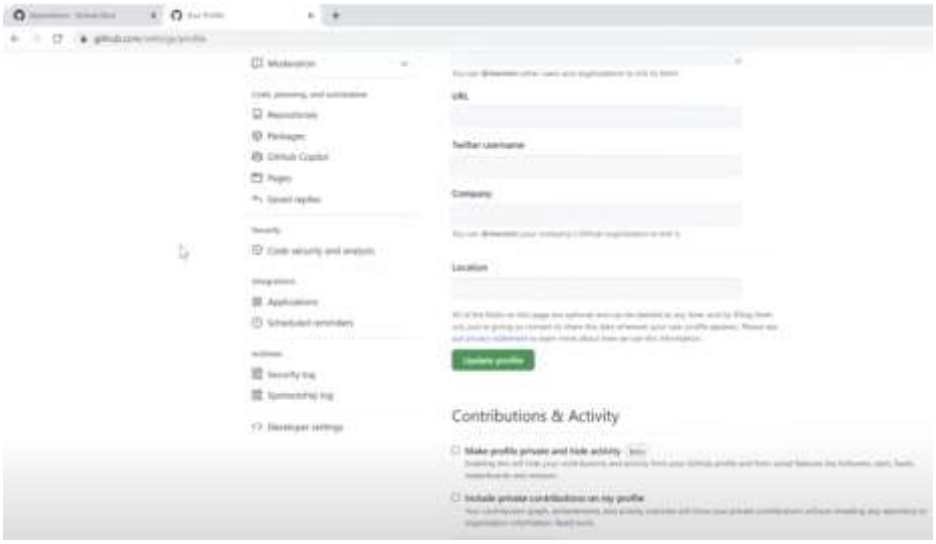


Postman Authorizing requests

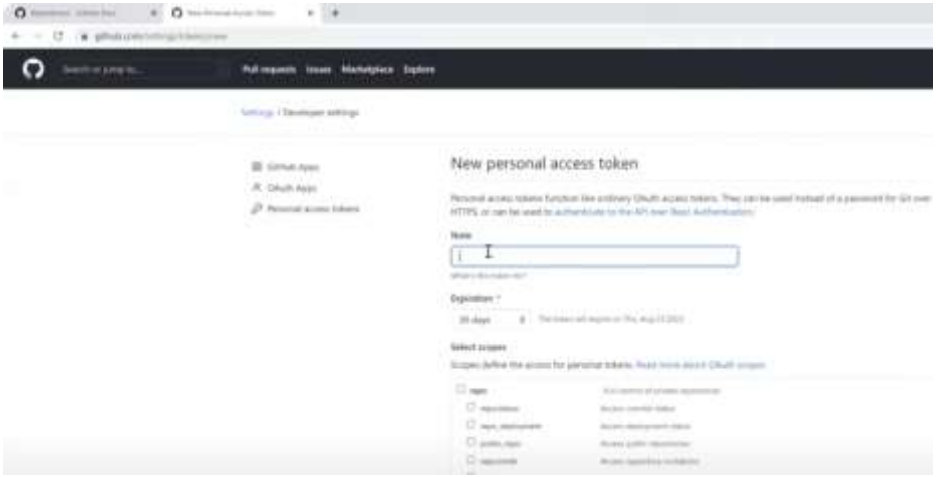


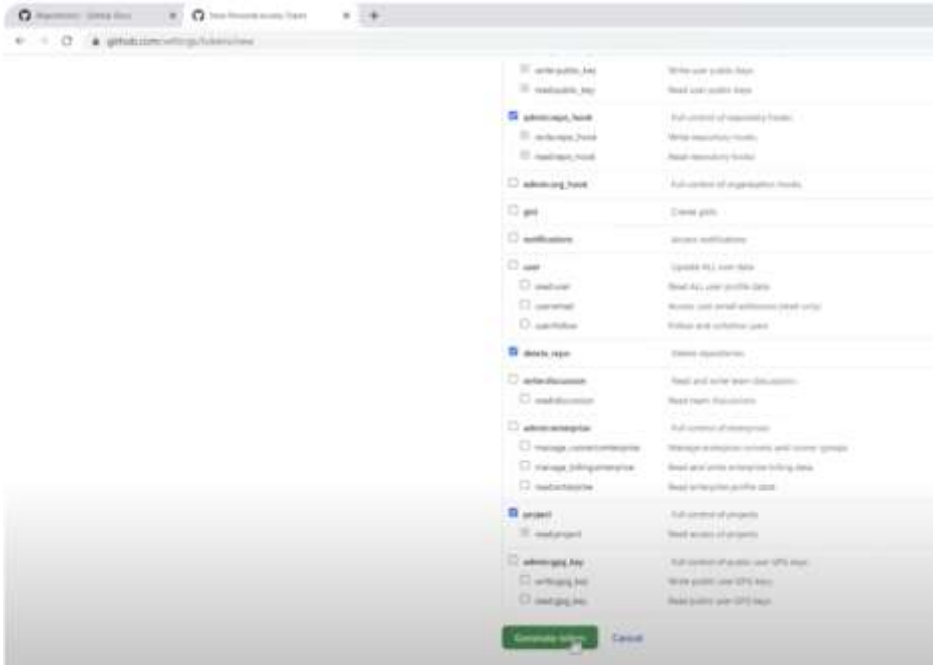


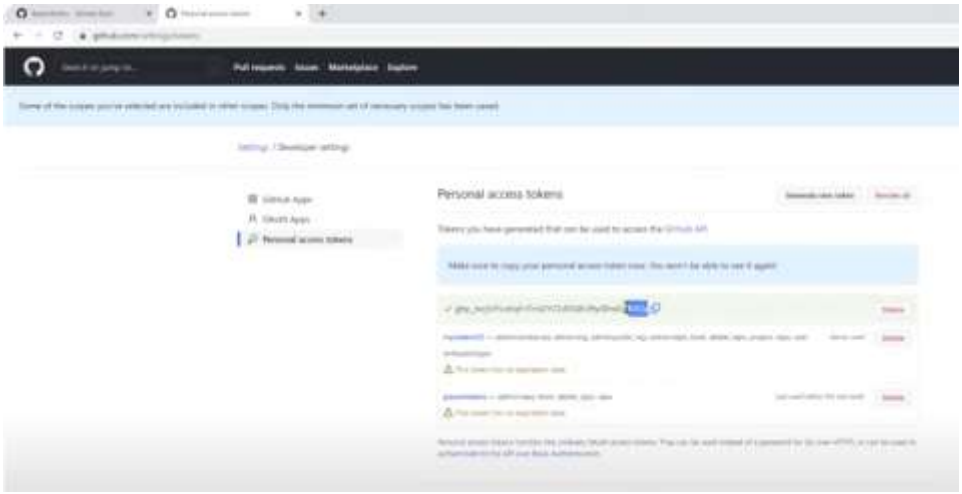


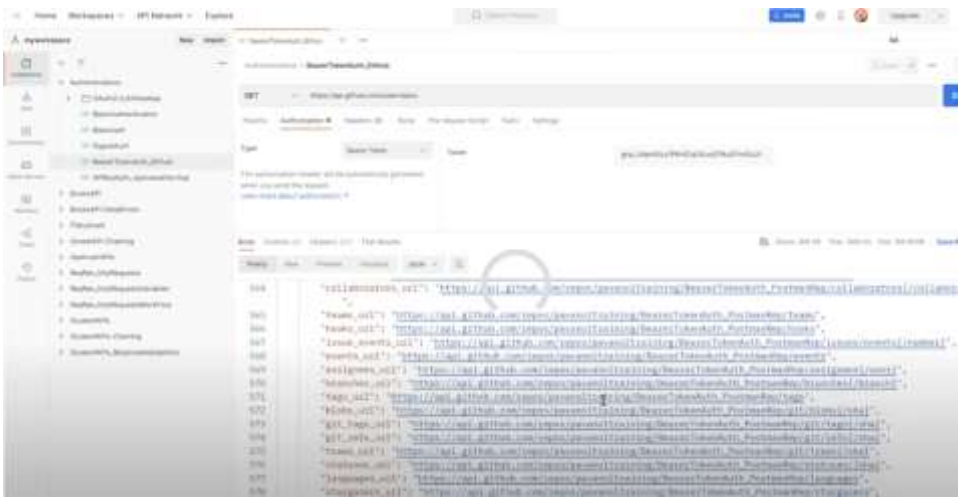

Contributions & Activity

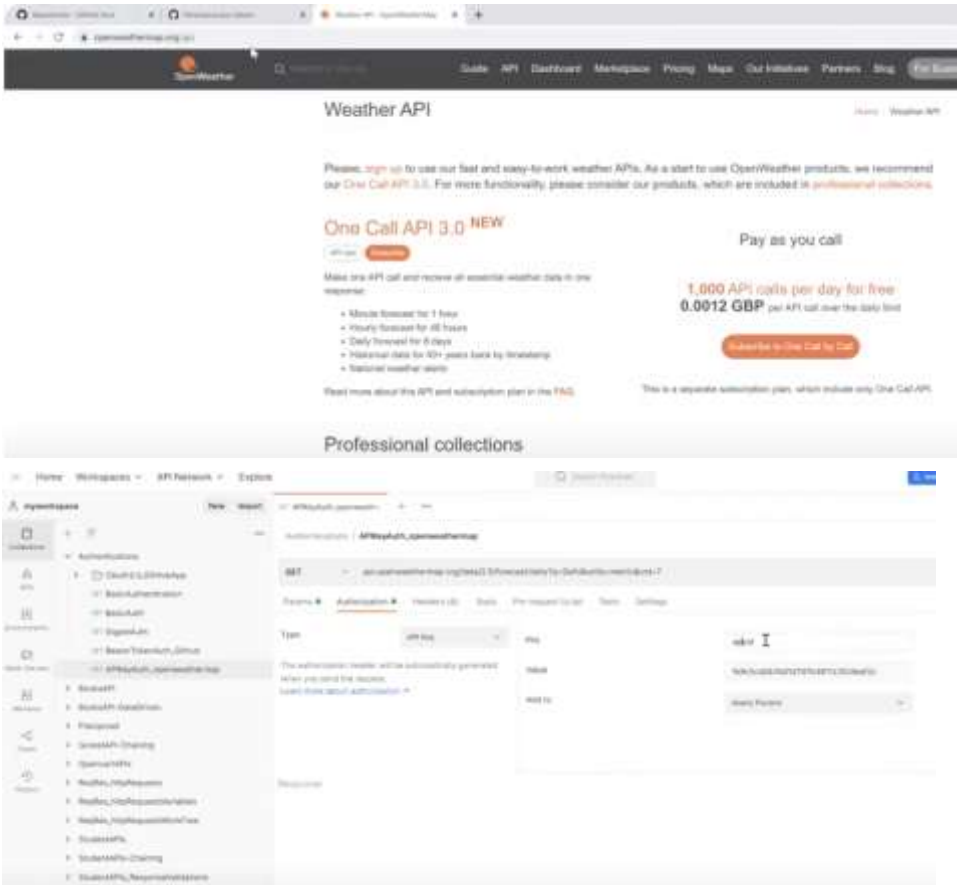

GitHub Apps


New personal access token



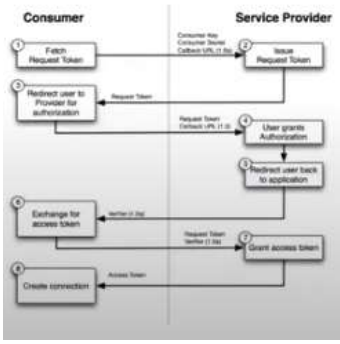

Personal access tokens



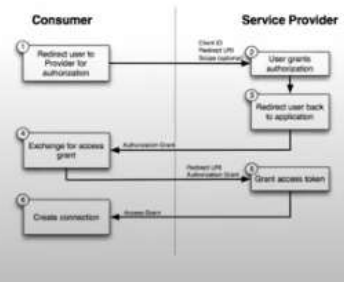

Weather API
Please, sign up to use our fast and easy-to-work weather APIs. As a start to use OpenWeather products, we recommend our One Call API 3.0. For more functionality, please consider our products, which are included in professional collections.
One Call API 3.0 NEW
Make one API call and receive all essential weather data in one response:
Minute forecast for 1 hour
Hourly forecast for 48 hours
Daily forecast for 8 days
Historical data for 40+ years back by timestamp
National weather alerts
Read more about this API and subscription plan in the FAQ.
Pay as you call
1,000 API calls per day for free
0.0012 GBP per API call over the daily limit
Subscribe to One Call by Call
This is a separate subscription plan, which include only One Call API.
Professional collections

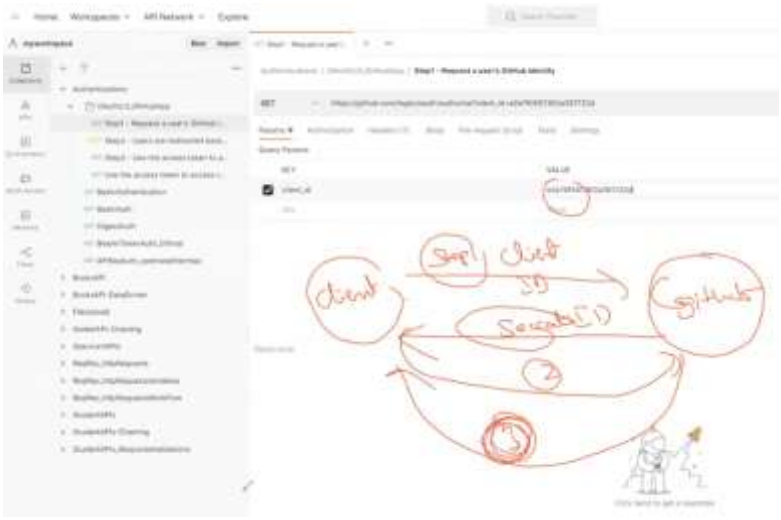
OAuth 1.0 Vs OAuth 2.0

OAuth 1.0



OAuth 2.0

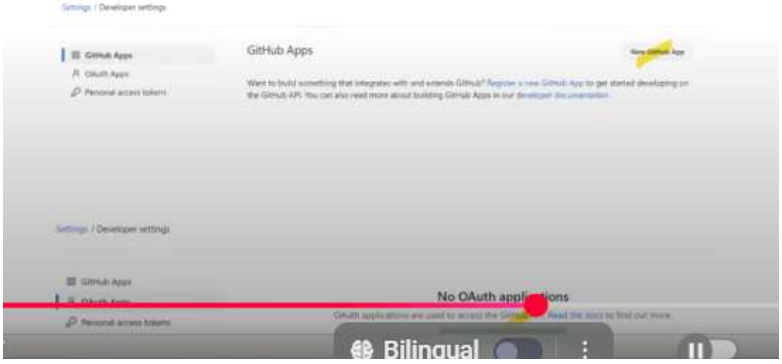


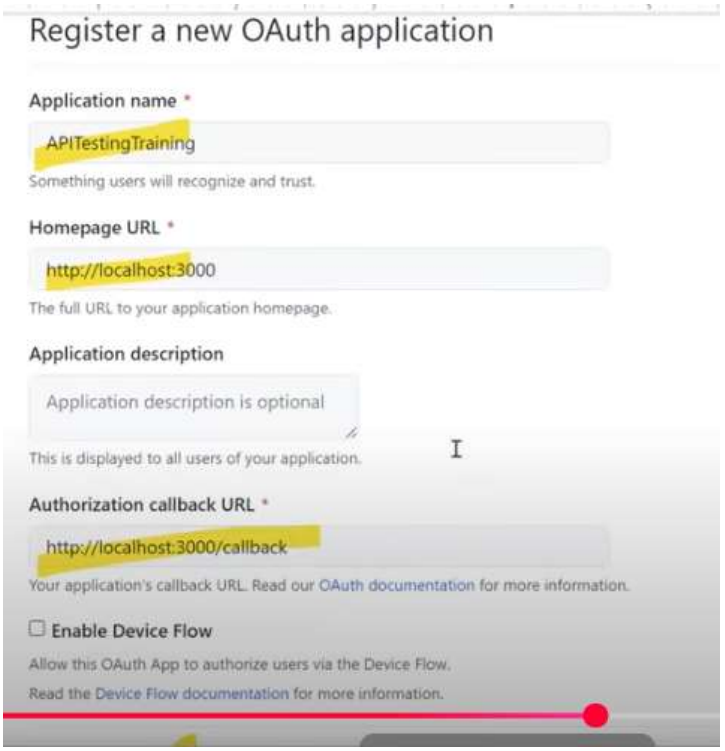


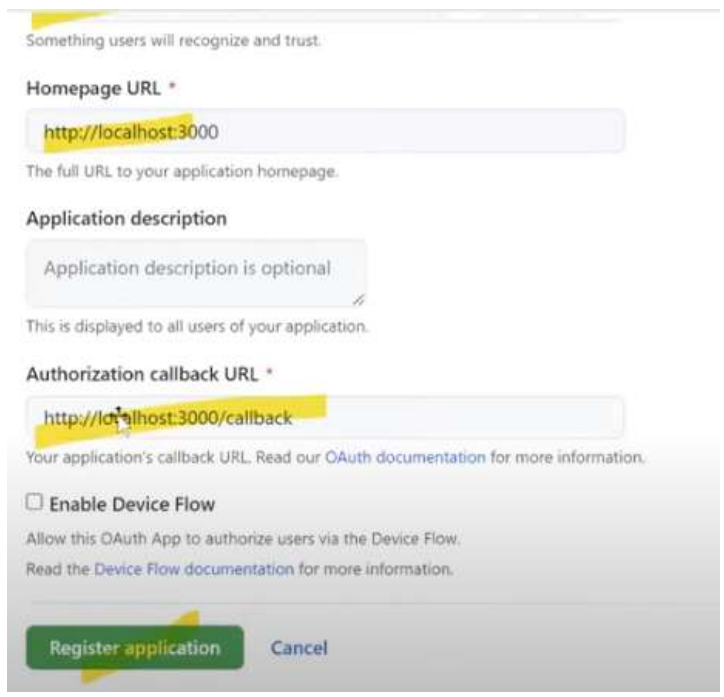
Setup OAuth2.0 – GitHub App

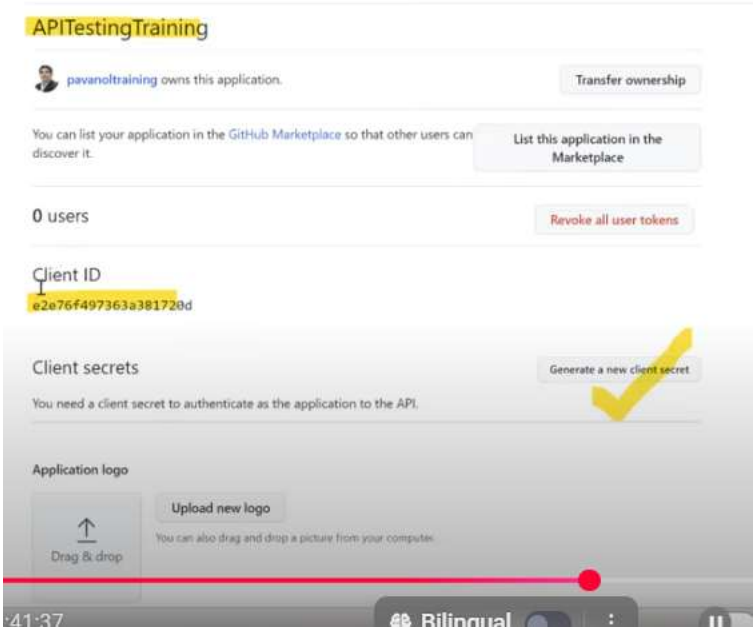
Login to GitHub→ Go to profile→ Settings→ Developer Settings

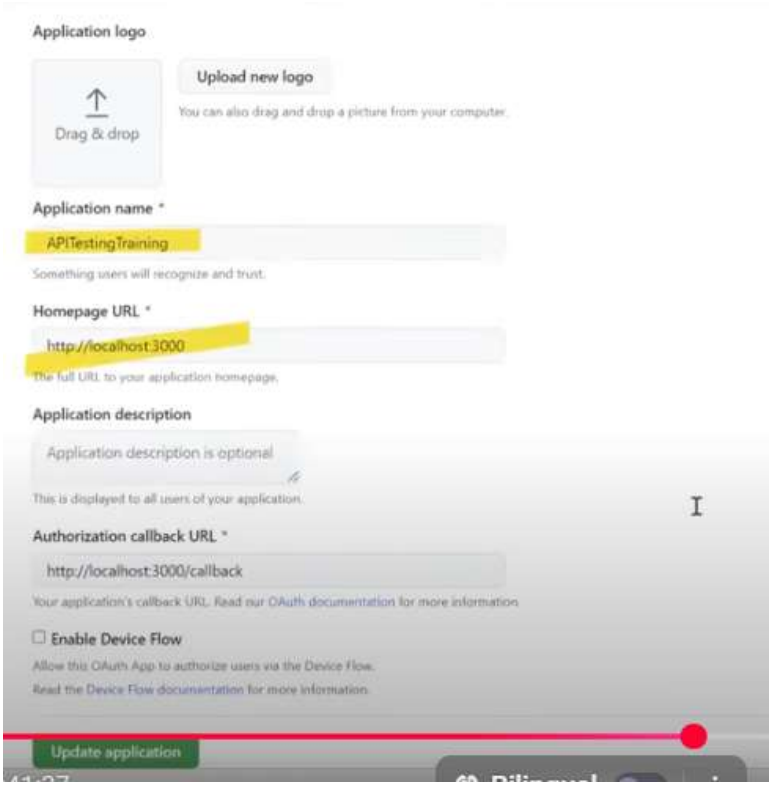
Create a new GitHub App:







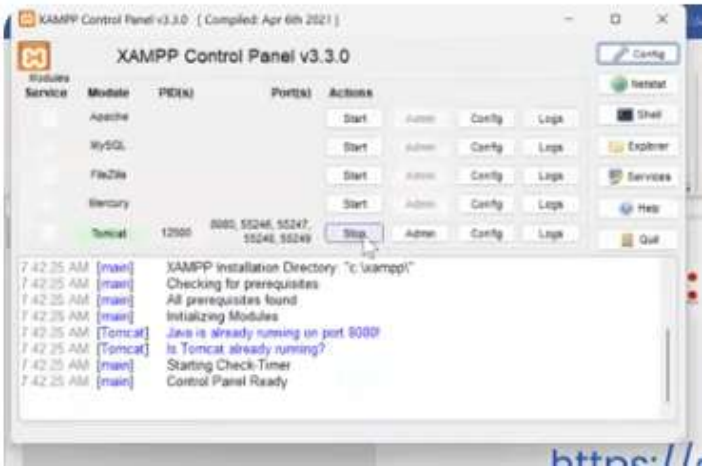




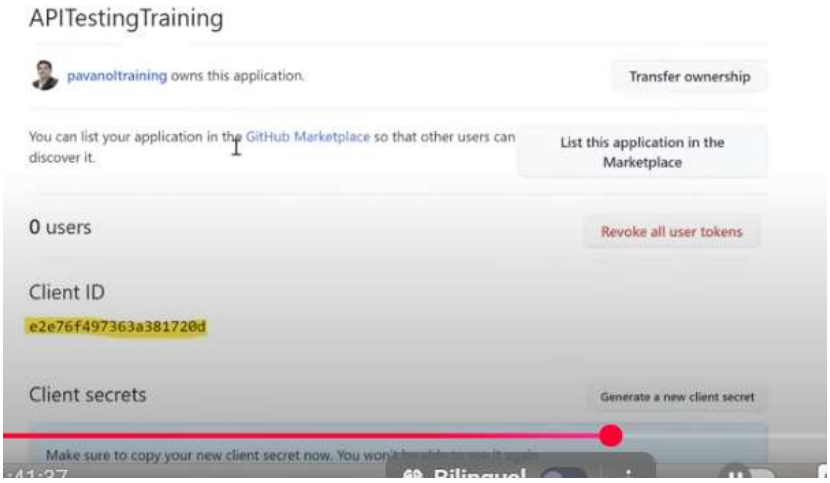
Step 1: Request a user's GitHub identity

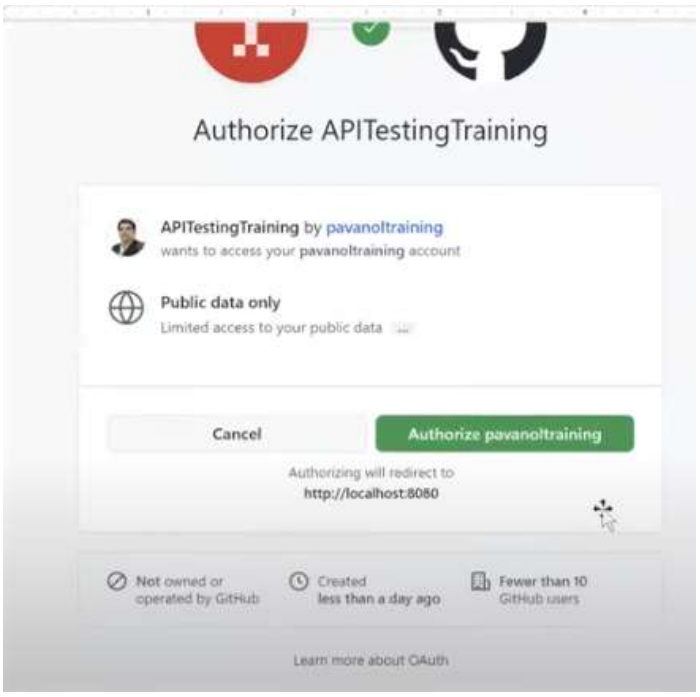
GET

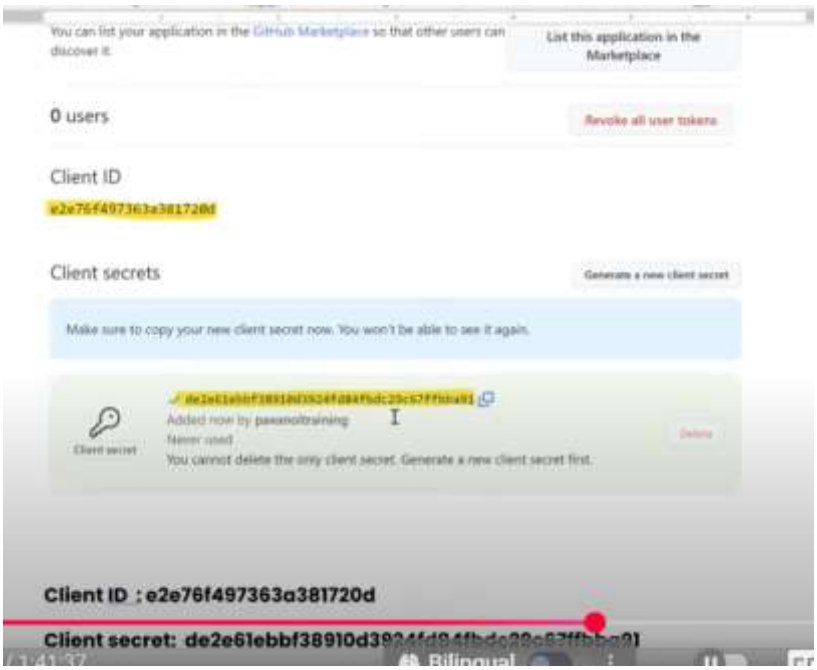
https://github.com/login/oauth/authorize?client_id=e2e76f497363a381720d

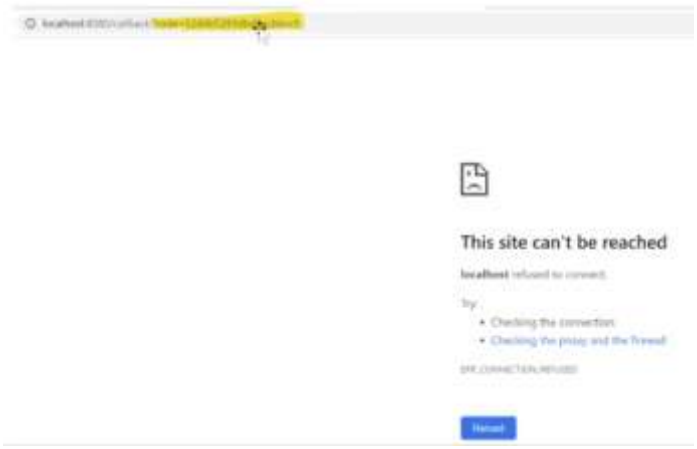


After generating new client secrete....

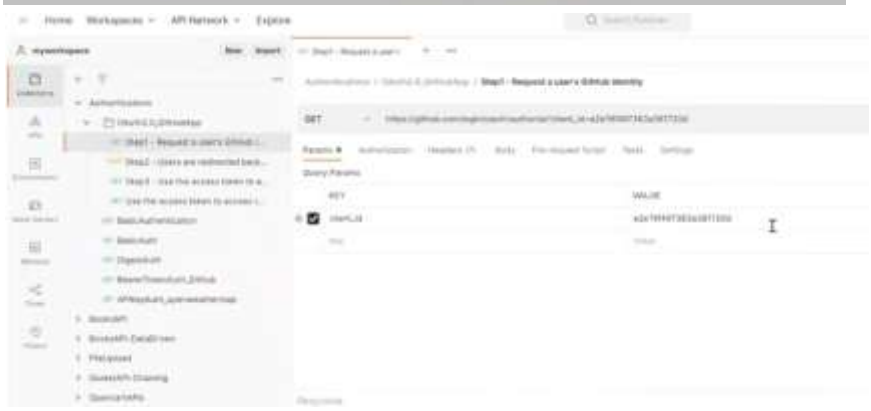


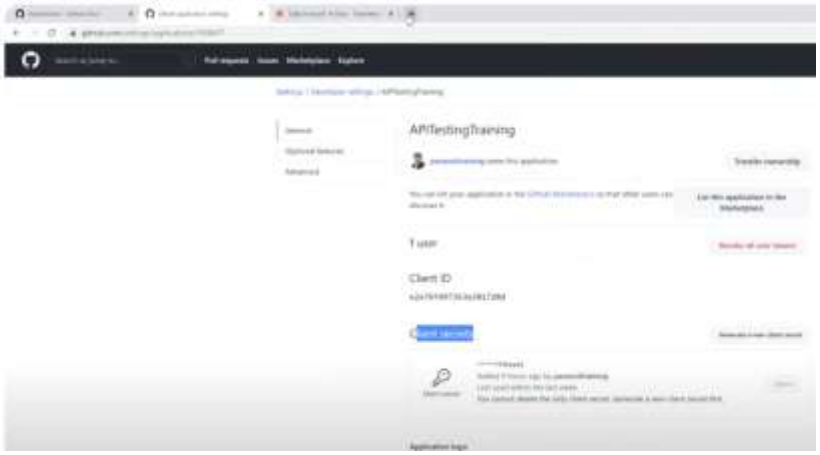






Step 2 : Users are redirected back to your site by GitHub

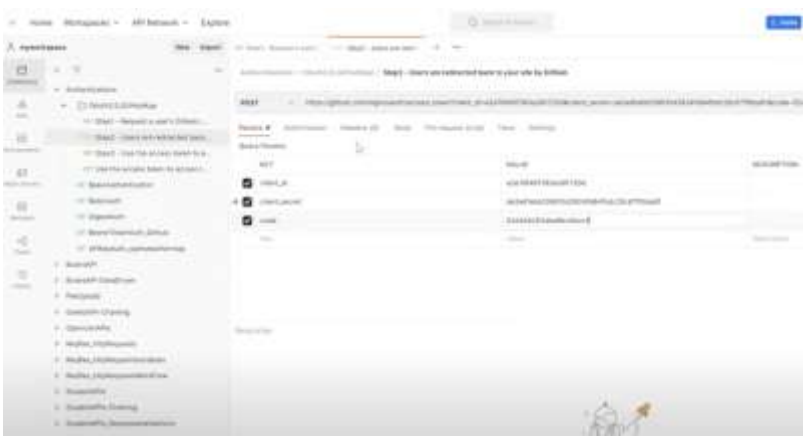


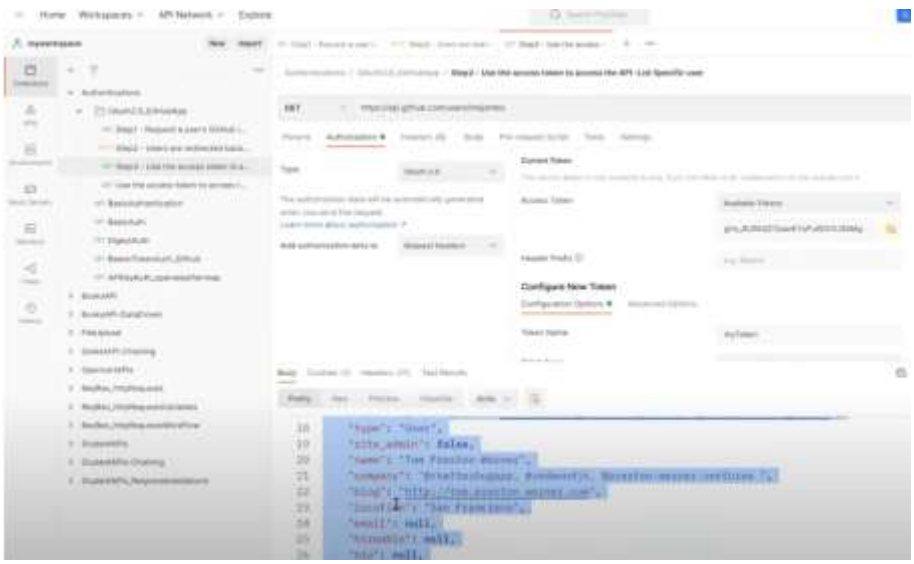


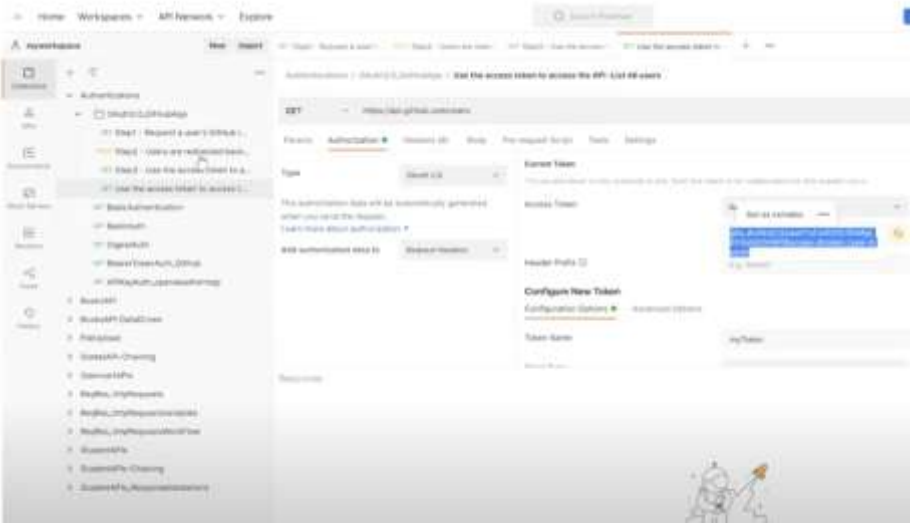
Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Tue Jul 26 07:45:25 IST 2022
There was an unexpected error (type=Not Found, status=404).
No message available



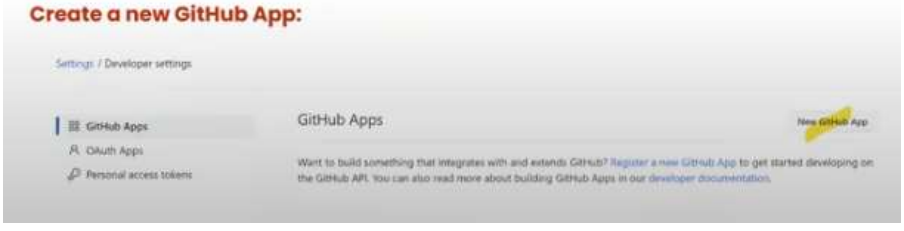




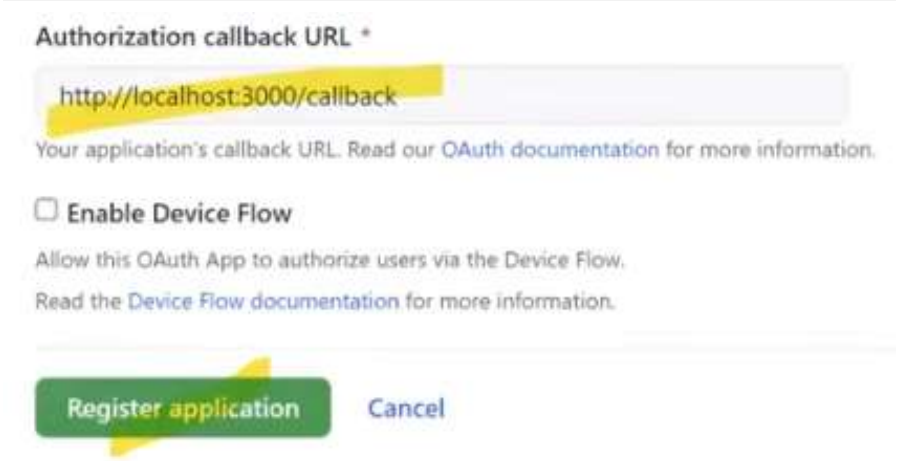
Setup OAuth2.0 – GitHub App

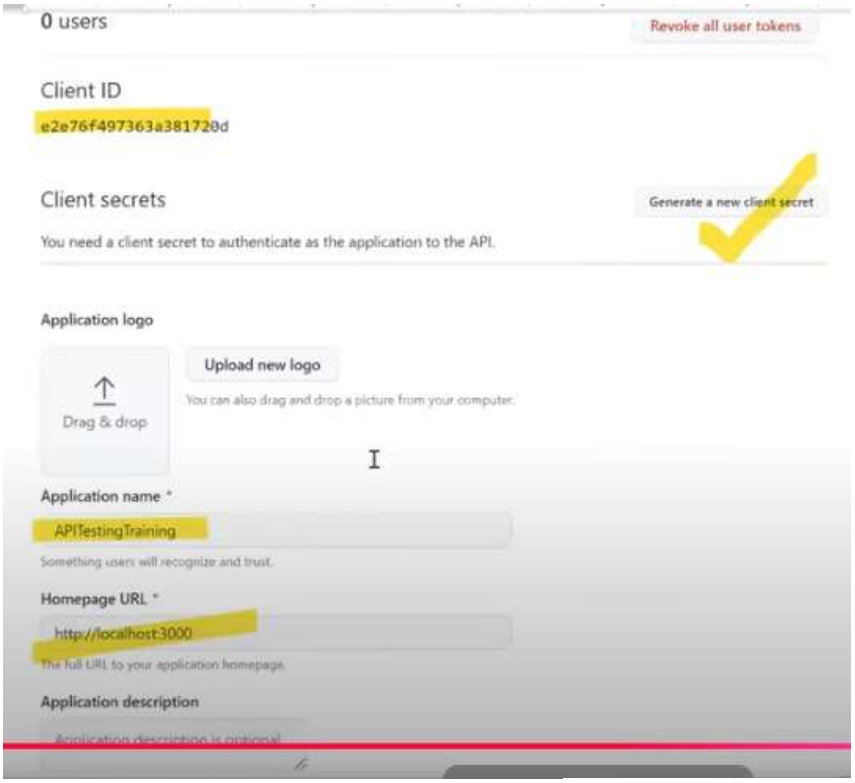
Login to GitHub→ Go to profile→ Settings→ Developer Settings

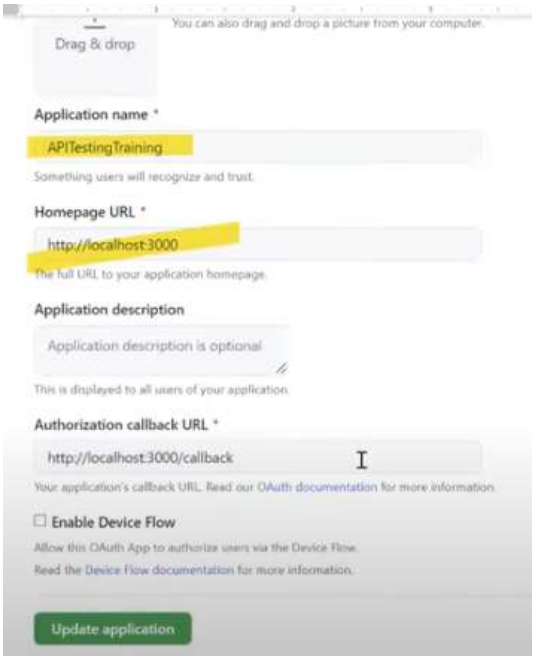
Create a new GitHub App:




Settings / Developer settings
GitHub Apps
OAuth Apps
Personal access tokens
No OAuth applications
OAuth applications are used to access the GitHub API. Read the docs to find out more.
Register a new application


Authorization callback URL *
http://localhost:3000/callback
Your application's callback URL. Read our OAuth documentation for more information.
Enable Device Flow
Allow this OAuth App to authorize users via the Device Flow.
Read the Device Flow documentation for more information.
Register application
Cancel


0 users
Revoke all user tokens
Client ID
e2e76f497363a381720d
Client secrets
Generate a new client secret
You need a client secret to authenticate as the application to the API.
Application logo
Upload new logo
Drag & drop
You can also drag and drop a picture from your computer.
Application name *
APITestingTraining
Something users will recognize and trust.
Homepage URL *
http://localhost:3000
The full URL to your application homepage.
Application description


Drag & drop
You can also drag and drop a picture from your computer.
Application name *
APITestingTraining
Something users will recognize and trust.
Homepage URL *
http://localhost:3000
The full URL to your application homepage.
Application description
Application description is optional
This is displayed to all users of your application.
Authorization callback URL *
http://localhost:3000/callback
Your application's callback URL. Read our OAuth documentation for more information.
Enable Device Flow
Allow this OAuth App to authorize users via the Device Flow.
Read the Device Flow documentation for more information.
Update application


Pull requests
Issues
Marketplace
Explore
GitHub Apps
OAuth Apps
Personal access tokens
GitHub Apps

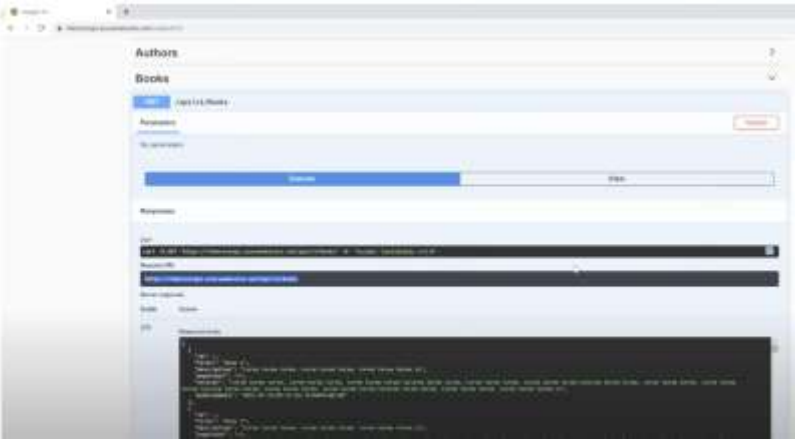
Day-8
---
1) Upload file

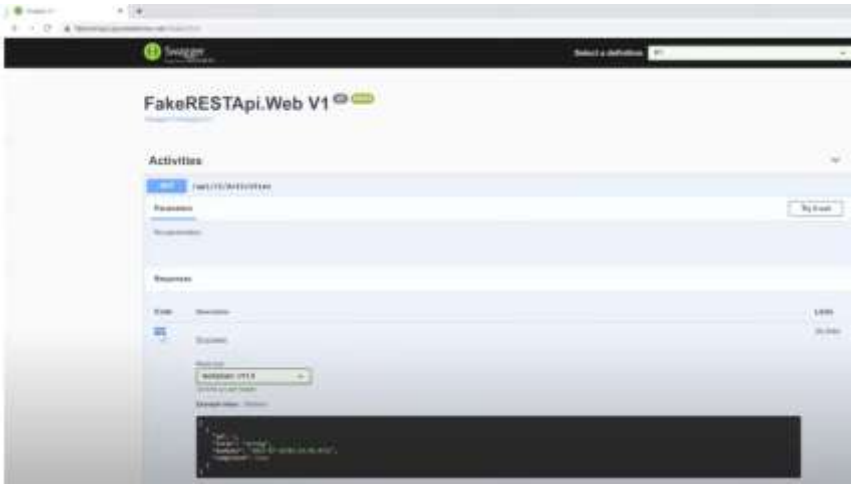
Post : http://localhost:8080/uploadFile
POST: http://localhost:8080/uploadMultipleFiles

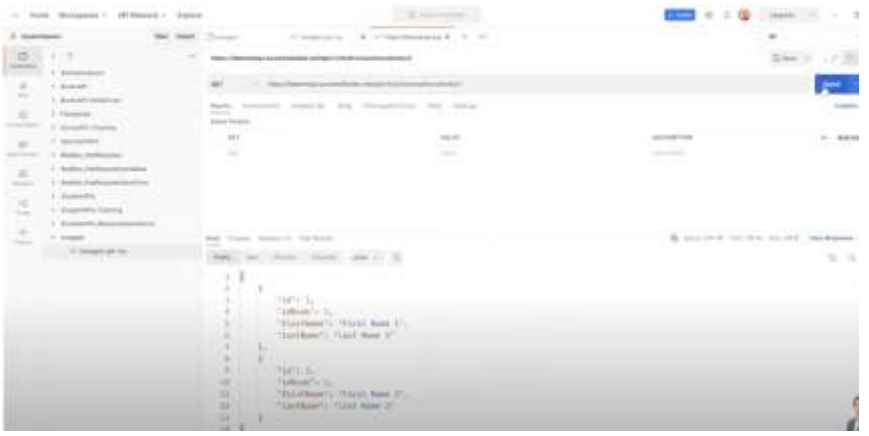
Authentication types

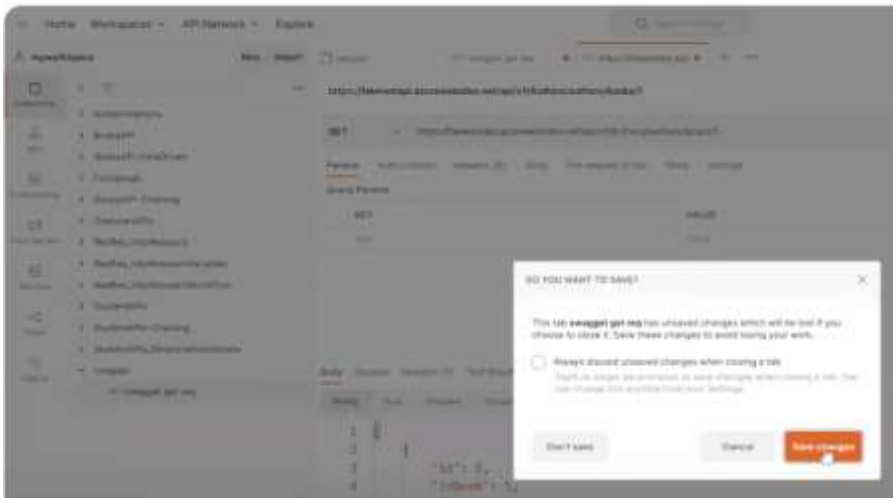
Swagger - interactive documentation

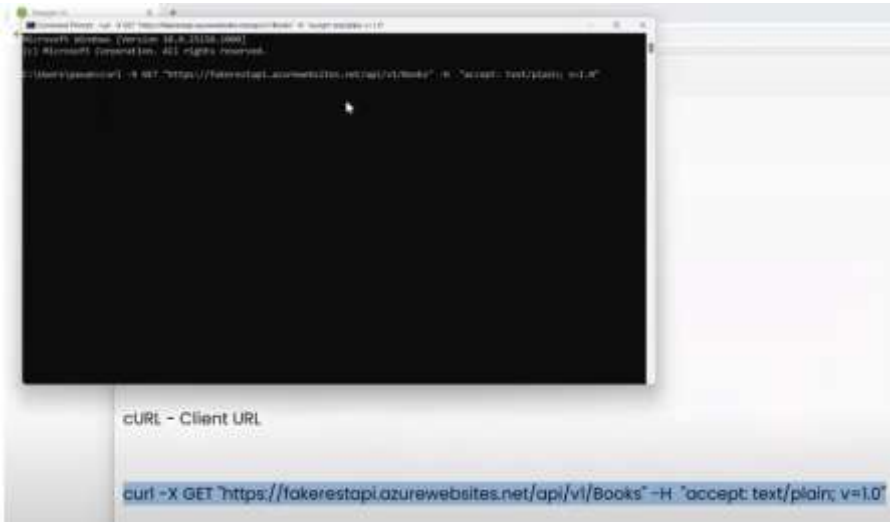
cURL

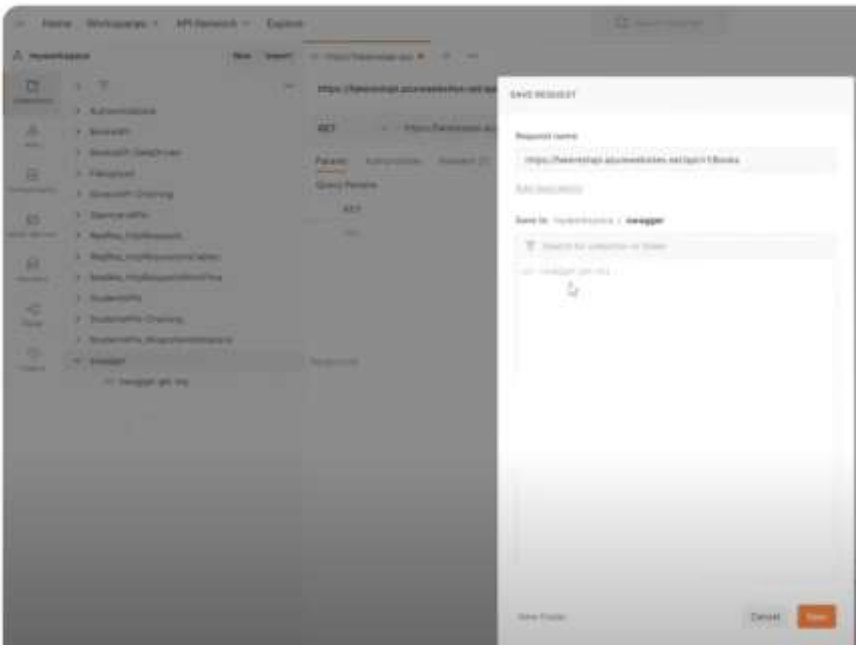


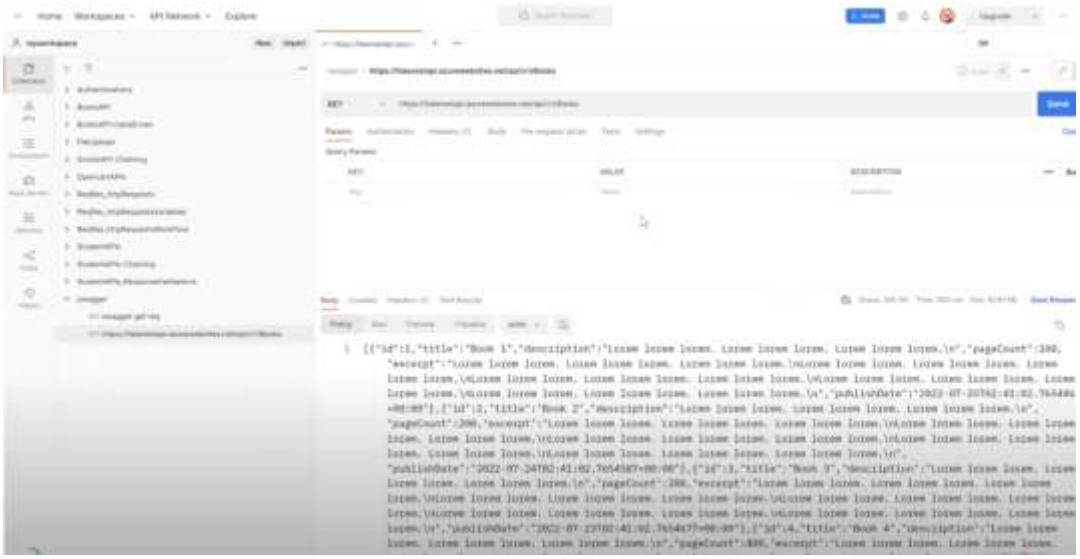


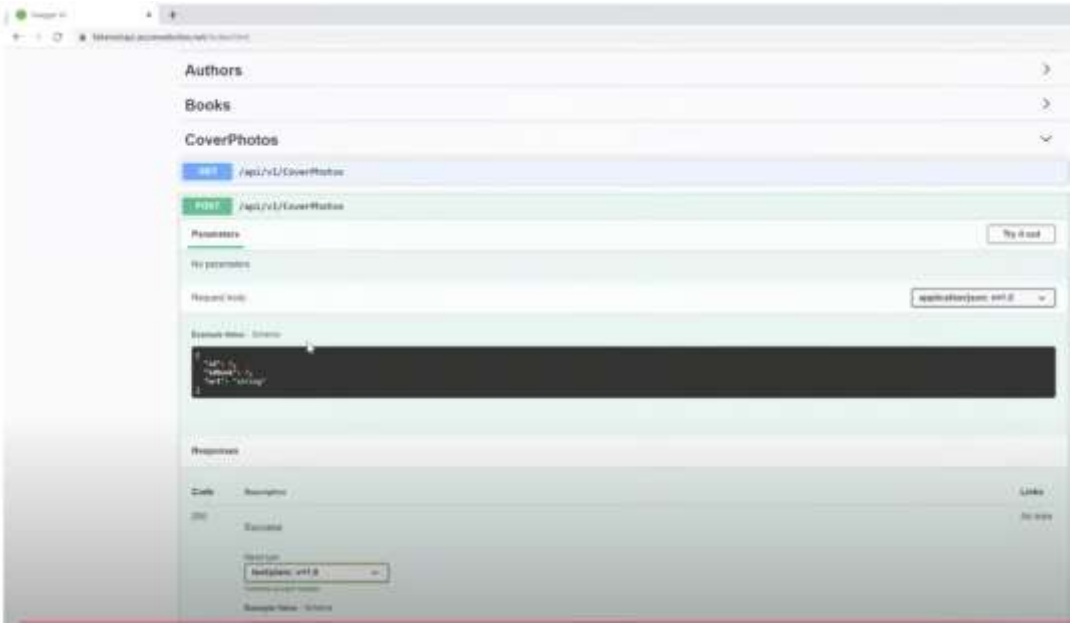


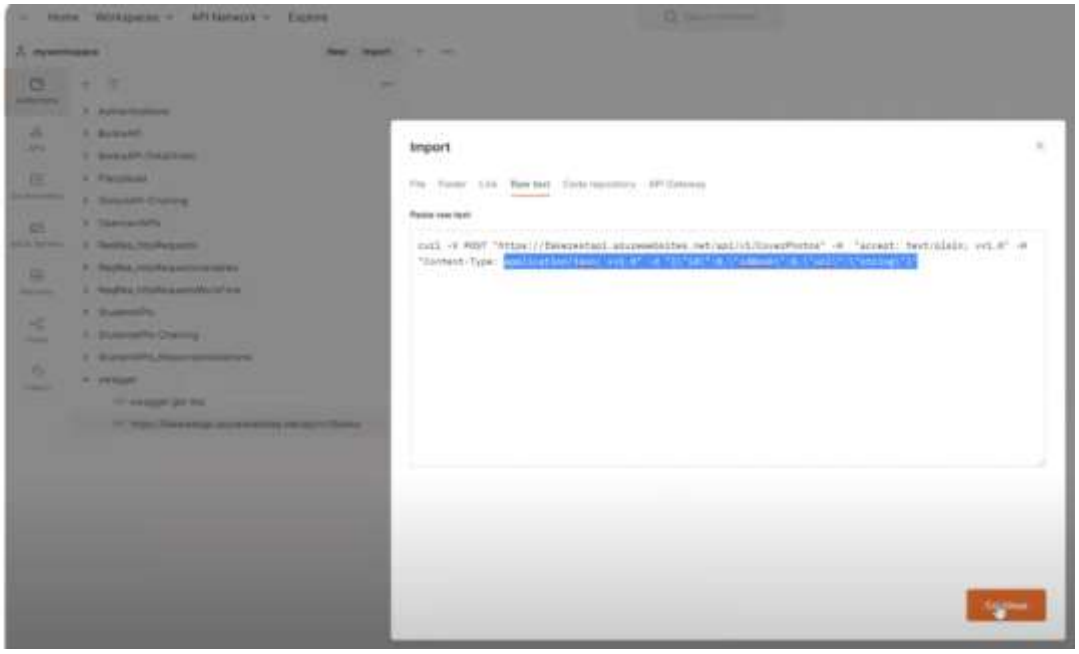


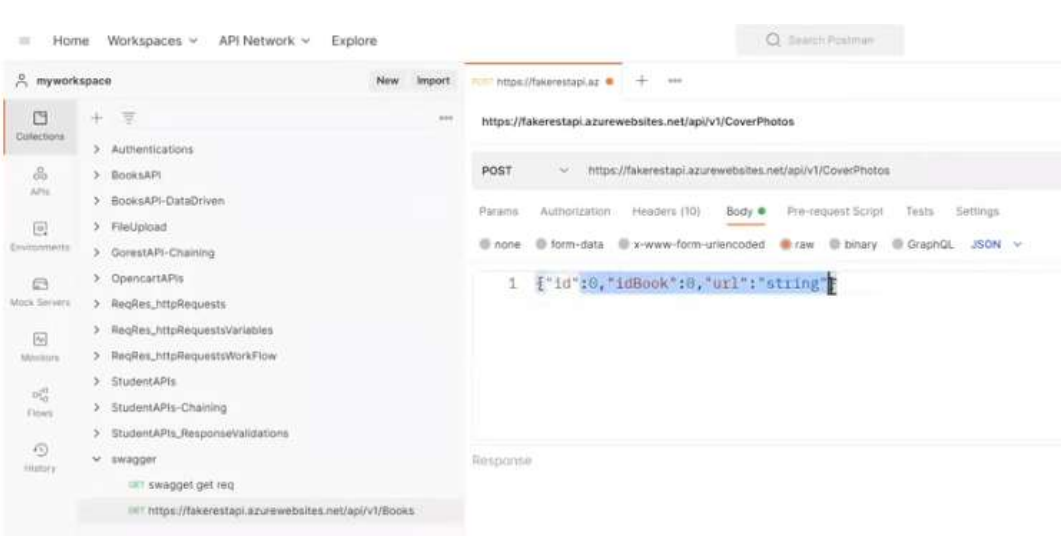

cURL - Client URL
curl -X GET "https://fakerestapi.azurewebsites.net/api/v1/Books" -H "accept: text/plain; v=1.0"

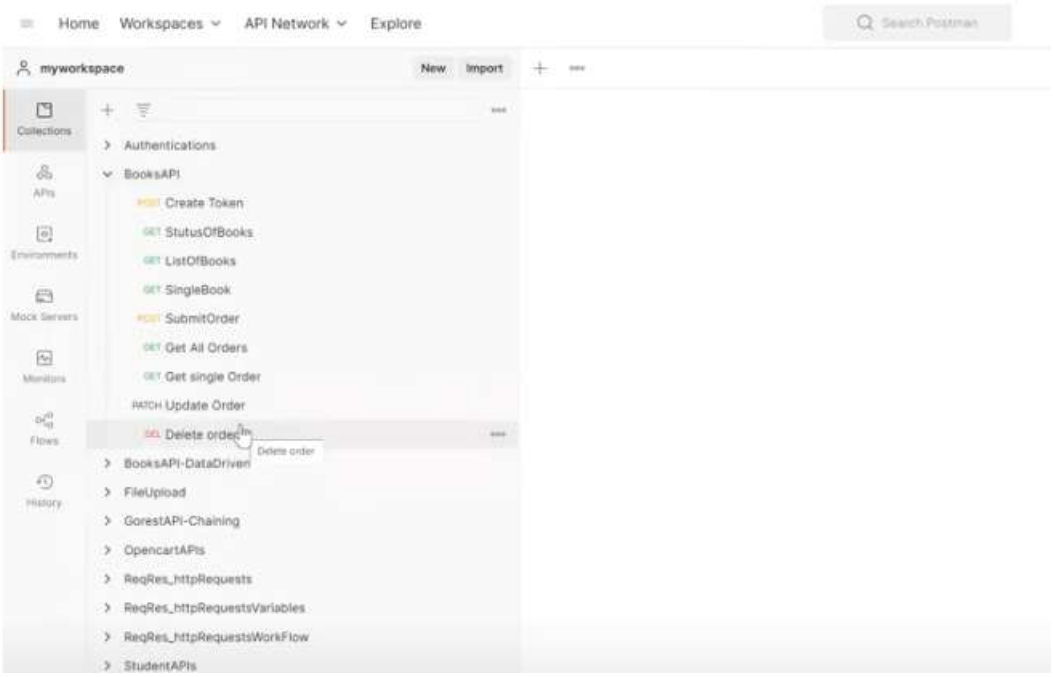


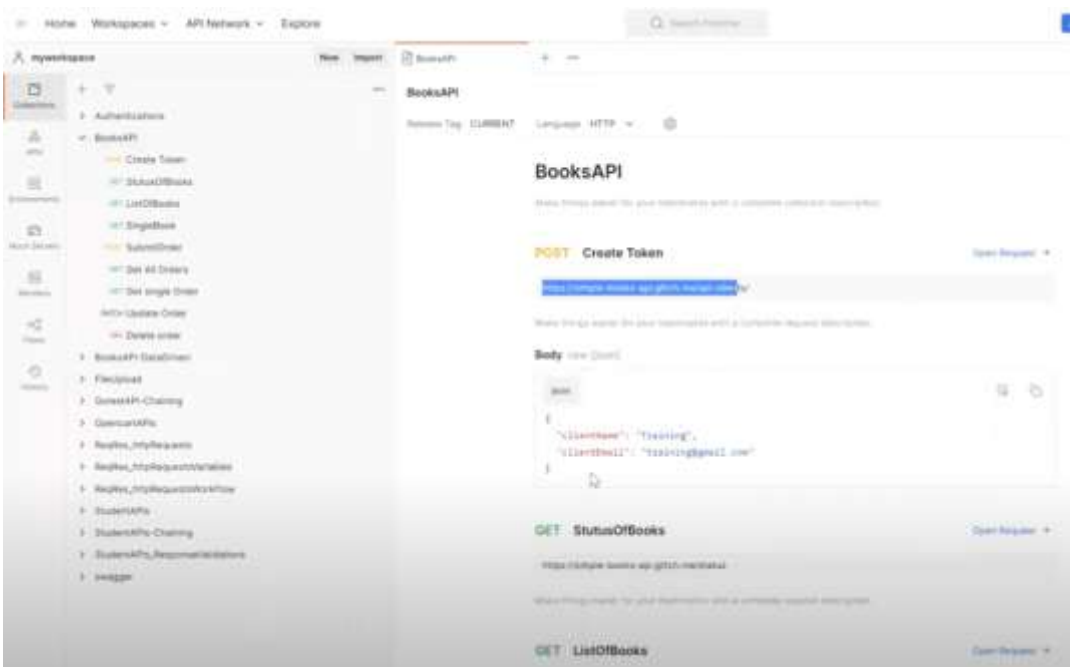


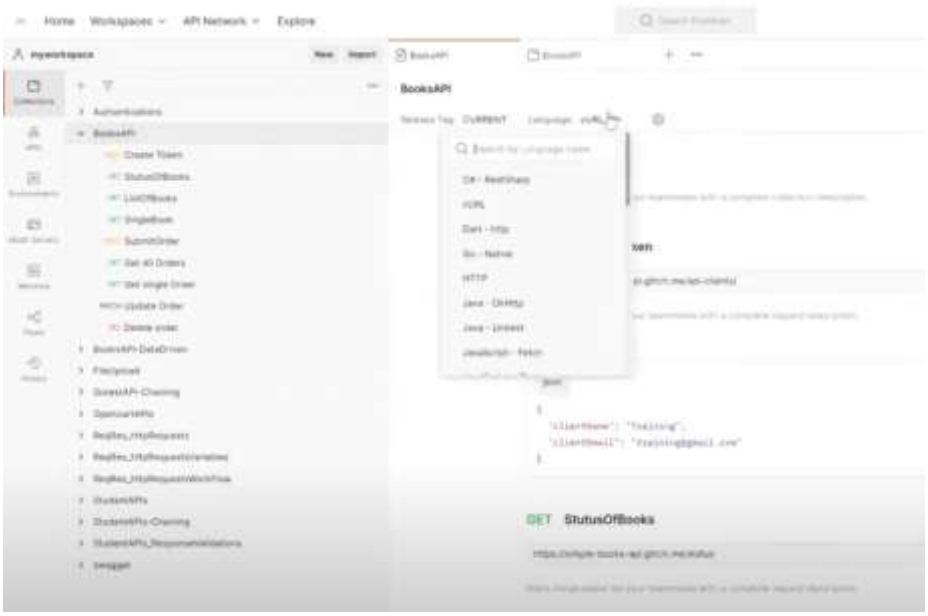

Authors
Books
CoverPhotos
GET /api/v1/CoverPhotos
POST /api/v1/CoverPhotos
Parameters
Try it out
No parameters
Request body
Responses
Code
Description
Links

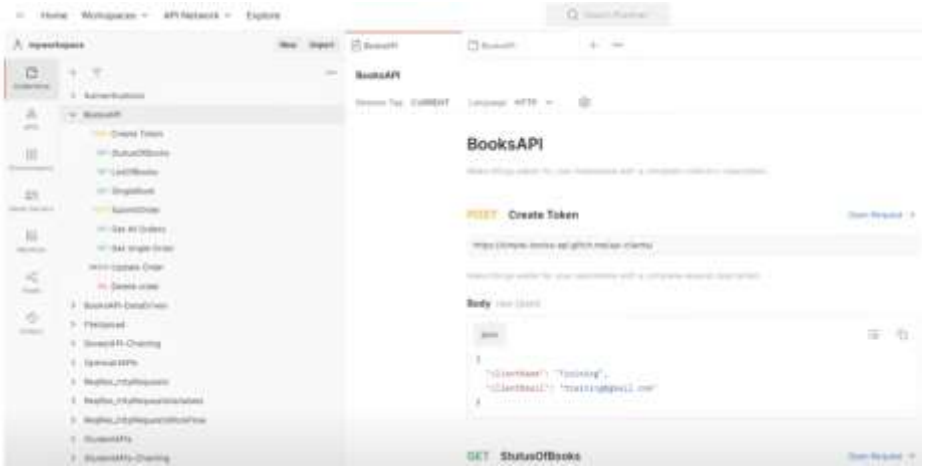

Import
File
Folder
Link
Raw text
API Gateway
Paste raw text


Home
Workspaces
API Network
Explore
myworkspace
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Authentications
BooksAPI
BooksAPI-DataDriven
FileUpload
GorestAPI-Chaining
OpencartAPIs
ReqRes_httpRequests
ReqRes_httpRequestsVariables
ReqRes_httpRequestsWorkFlow
StudentAPIs
StudentAPIs-Chaining
StudentAPIs_ResponseValidations
swagger
swagget get req
https://fakerestapi.azurewebsites.net/api/v1/Books
https://fakerestapi.azurewebsites.net/api/v1/CoverPhotos
POST
https://fakerestapi.azurewebsites.net/api/v1/CoverPhotos
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
1 {"id":0,"idBook":0,"url":"string"}
Response


Home
Workspaces
API Network
Explore
myworkspace
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Authentications
BooksAPI
Create Token
StutusOfBooks
ListOfBooks
SingleBook
SubmitOrder
Get All Orders
Get single Order
Update Order
Delete order
Delete order
BooksAPI-DataDriven
FileUpload
GorestAPI-Chaining
OpencartAPIs
ReqRes_httpRequests
ReqRes_httpRequestsVariables
ReqRes_httpRequestsWorkFlow
StudentAPIs


BooksAPI
POST
Create Token
Body
GET
StutusOfBooks
GET
ListOfBooks





Day-8
---
1) Upload file

Post : http://localhost:8080/uploadFile
POST: http://localhost:8080/uploadMultipleFiles

Authentication types

Swagger - interactive documentation
cURL - Client URL

curl -X GET "https://fakerestapi.azurewebsites.net/api/v1/Books" -H "accept: text/plain; v=1.0"

Day-8
---
1) Upload file

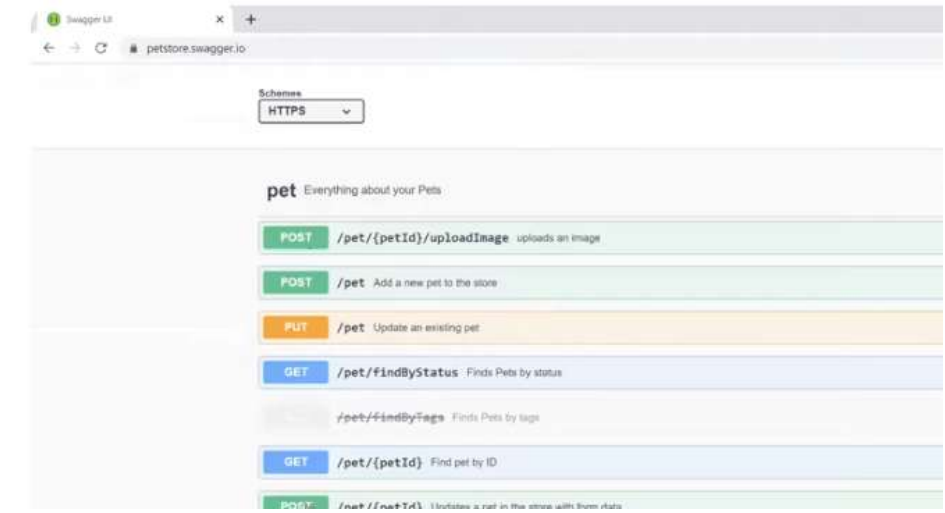
Post : http://localhost:8080/uploadFile
POST: http://localhost:8080/uploadMultipleFiles

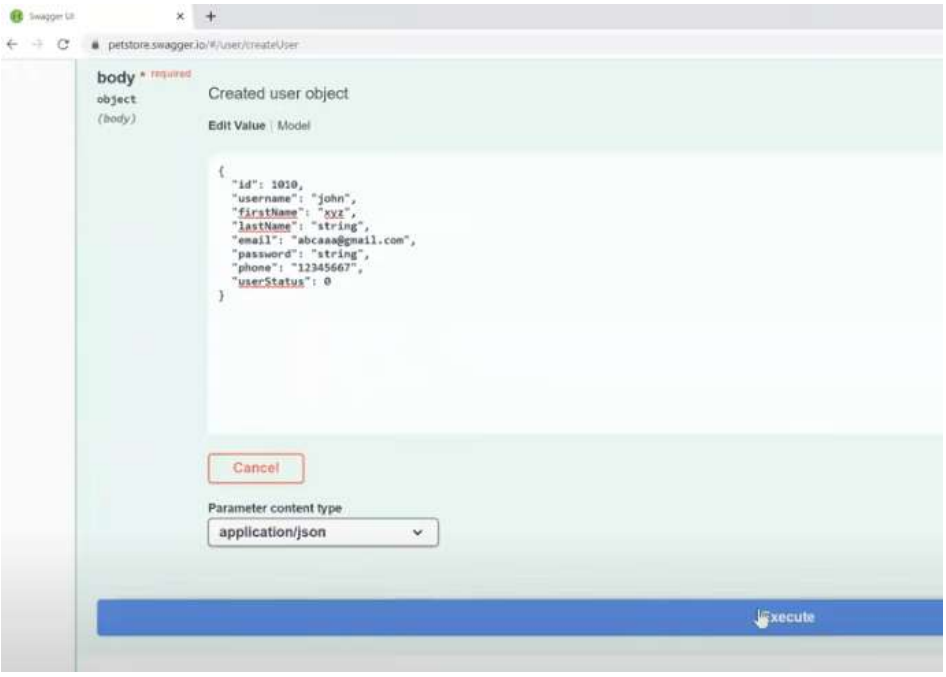
Authentication types

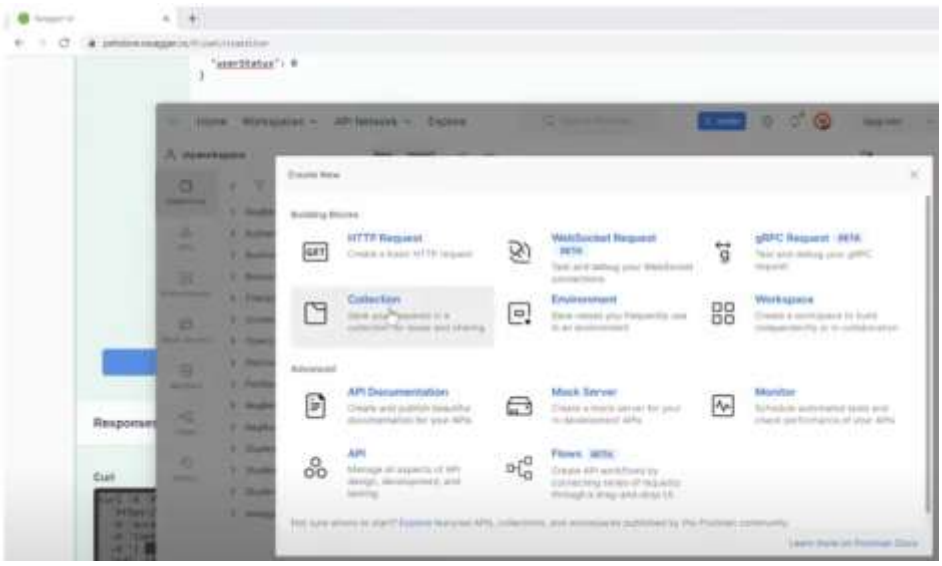
Swagger - interactive documentation
cURL - Client URL

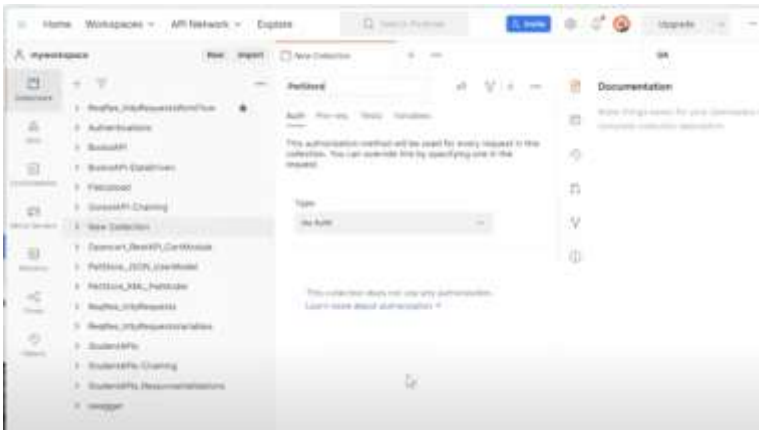
curl -X GET "https://fakerestapi.azurewebsites.net/api/v1/Books" -H "accept: text/plain; v=1.0"

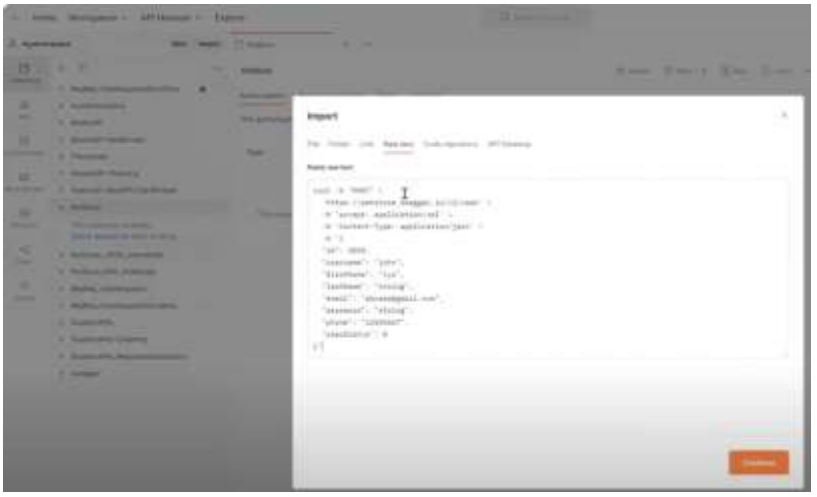
https://petstore.swagger.io/
https://httpbin.org/#/

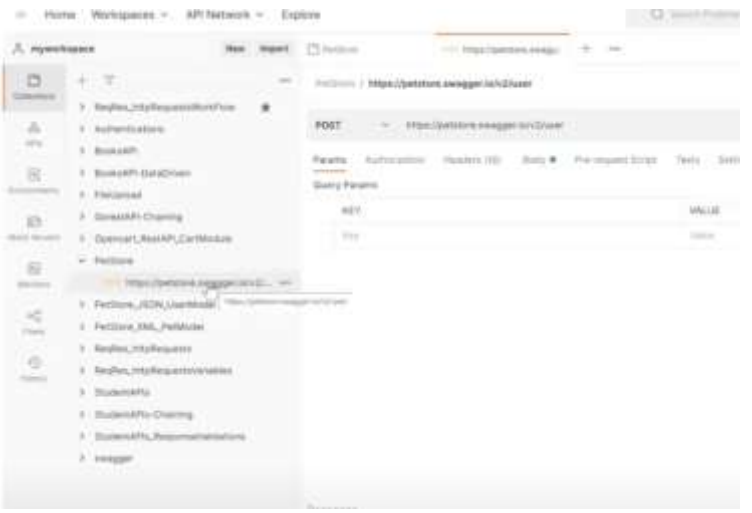

petstore.swagger.io
Schemes
HTTPS
pet Everything about your Pets
POST /pet/{petId}/uploadImage uploads an image
POST /pet Add a new pet to the store
PUT /pet Update an existing pet
GET /pet/findByStatus Finds Pets by status
GET /pet/{petId} Find pet by ID

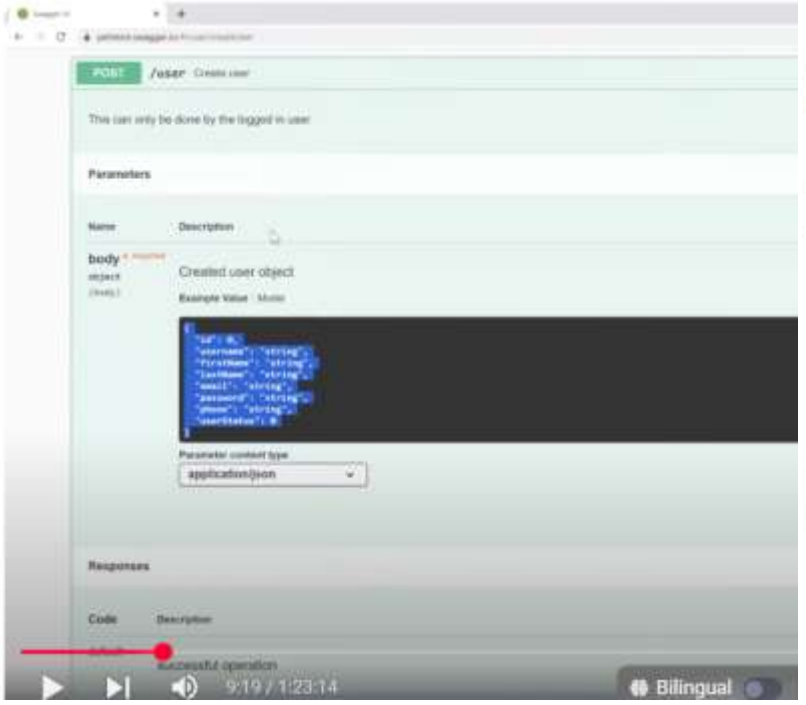

petstore.swagger.io/#/user/createUser
body * required
object
(body)
Created user object
Edit Value | Model
{
"id": 1010,
"username": "john",
"firstName": "xyz",
"lastName": "string",
"email": "abcaaa@gmail.com",
"password": "string",
"phone": "12345667",
"userStatus": 0
}
Cancel
Parameter content type
application/json
Execute

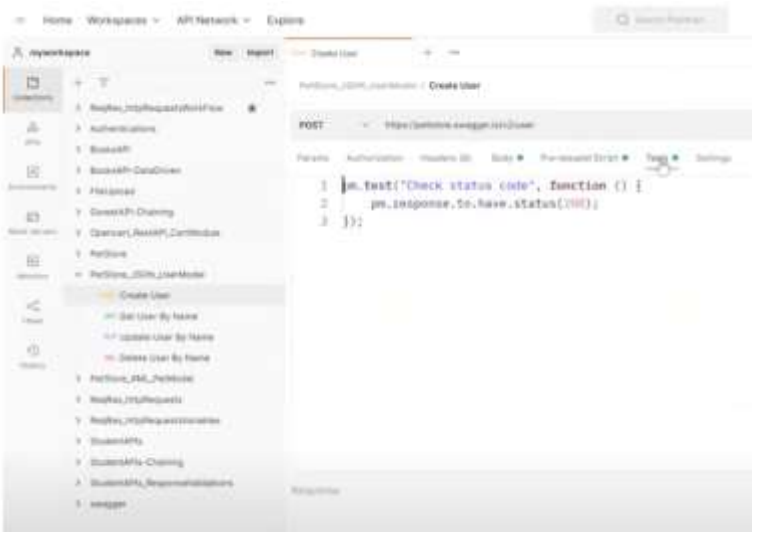

HTTP Request
WebSocket Request
gRPC Request
Collection
Environment
Workspace
API Documentation
Mock Server
Monitor
API
Flows

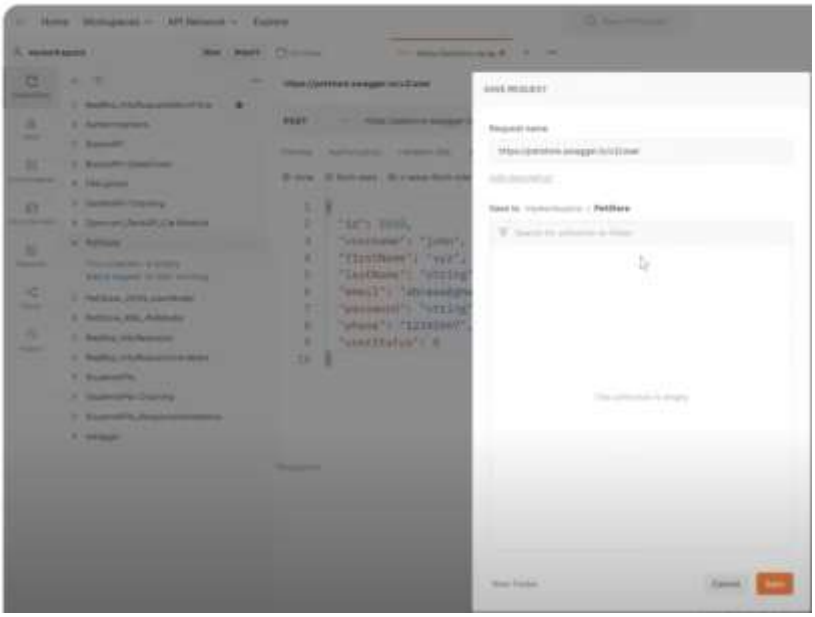

Home
Workspaces
API Network
Explore
Documentation

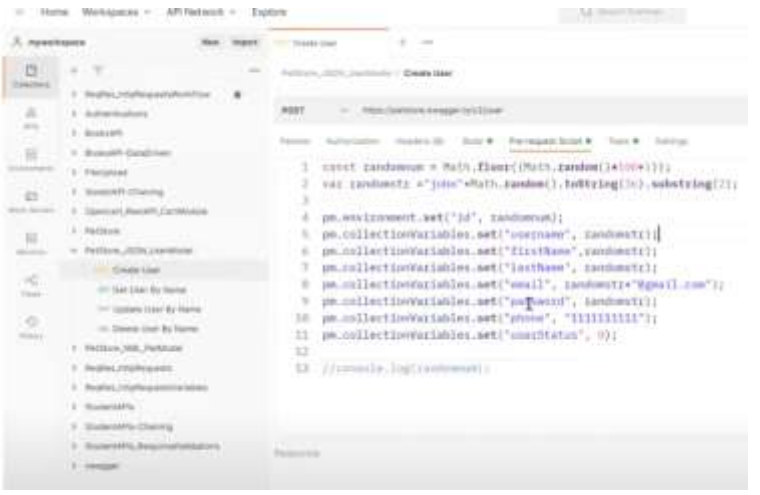


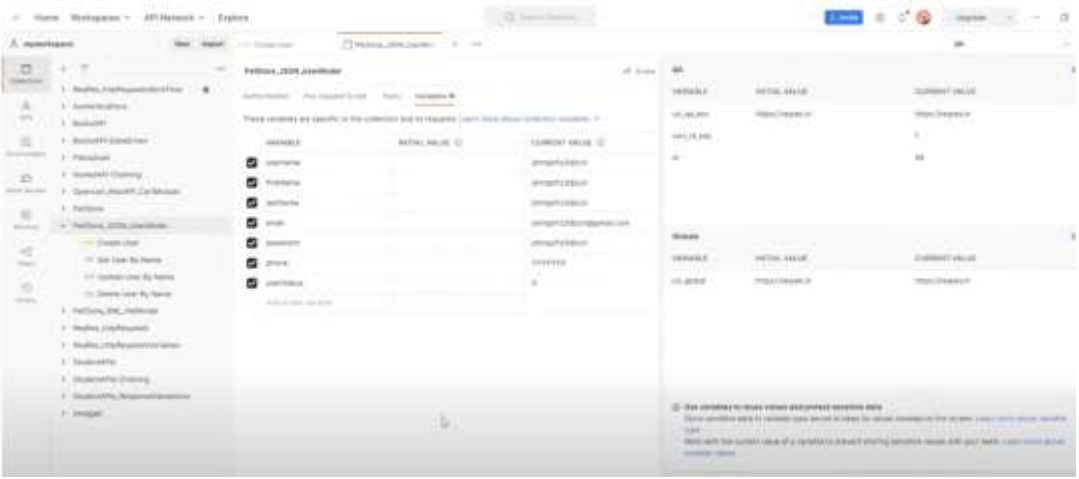


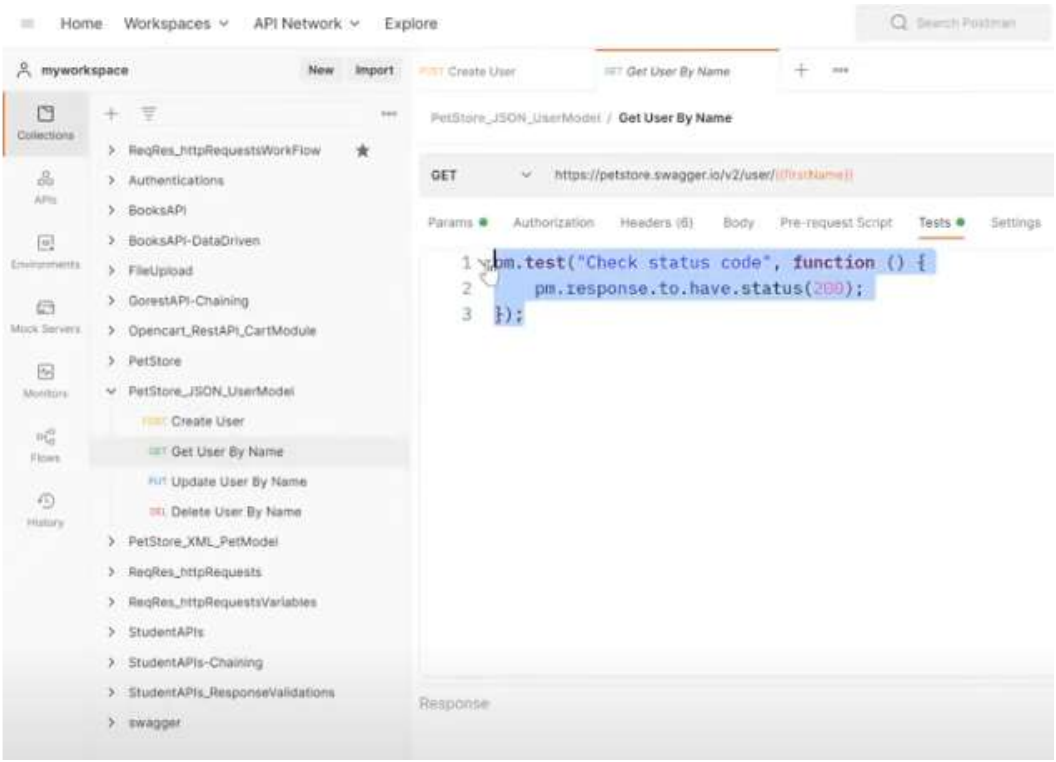

POST
/user
Create user
This can only be done by the logged in user
Parameters
Name
Description
body
Created user object
Example Value
Model
Parameter content type
application/json
Responses
Code
Description
successful operation
9:19 / 1:23:14
Bilingual

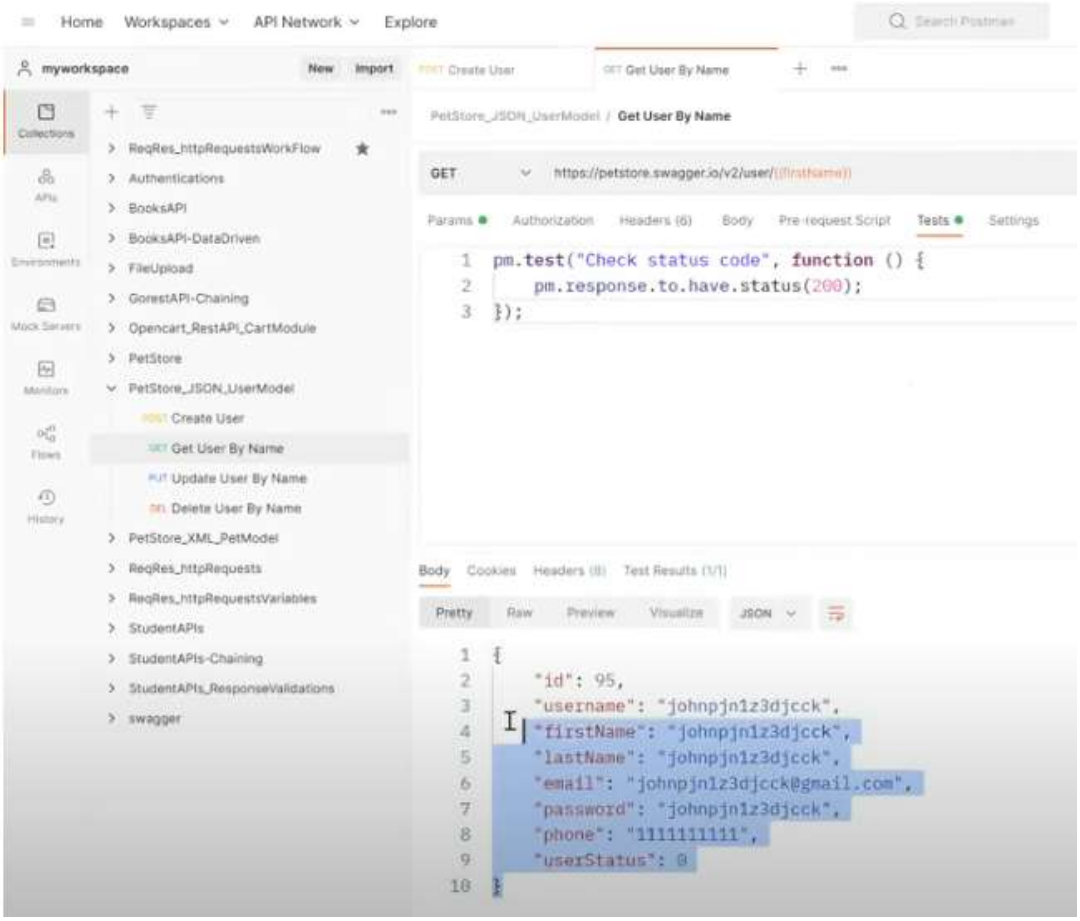


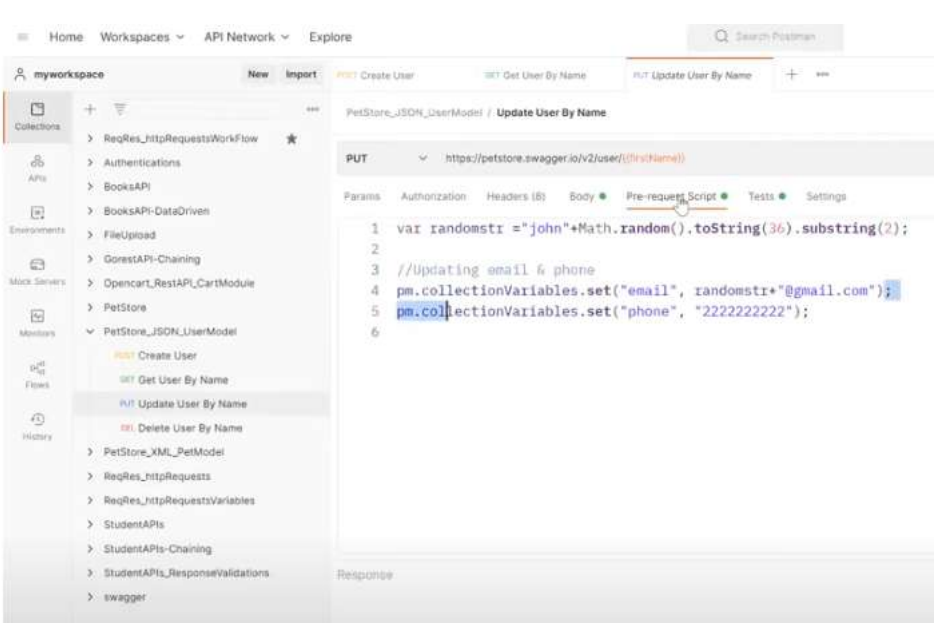


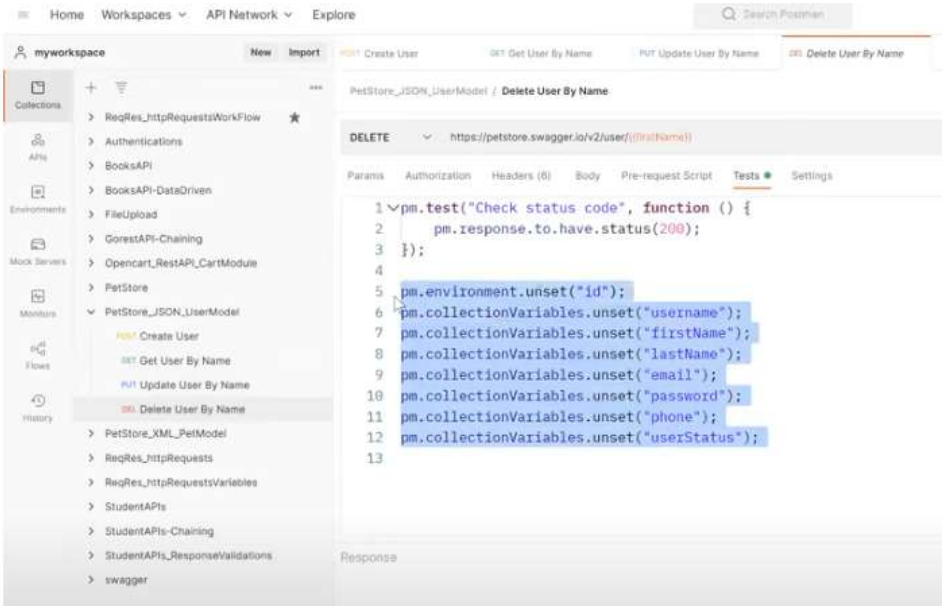

pm.test("Check status code", function () {
pm.response.to.have.status(200);
});

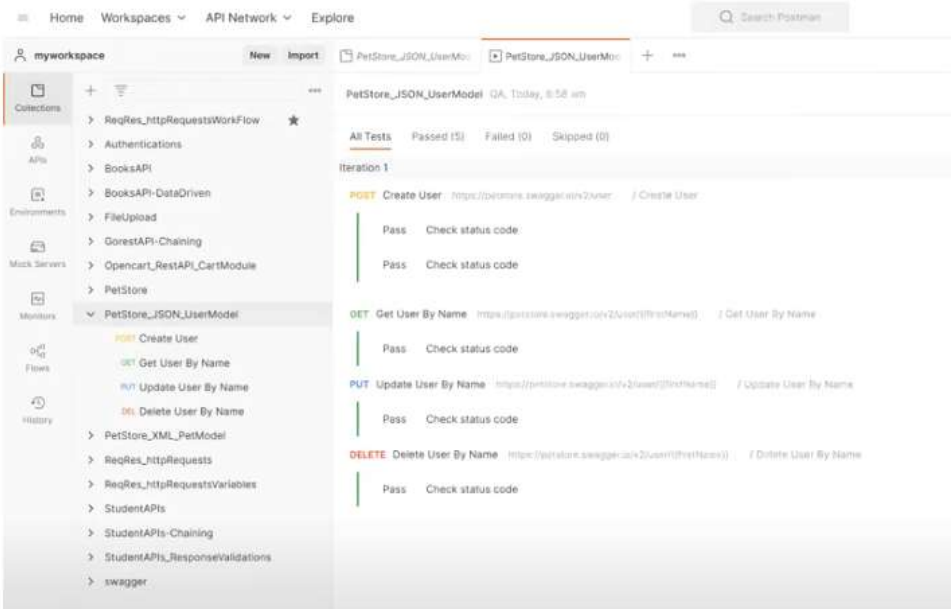



Home
Workspaces
API Network
Explore
myworkspace
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
ReqRes_httpRequestsWorkFlow
Authentications
BooksAPI
BooksAPI-DataDriven
FileUpload
GorestAPI-Chaining
Opencart_RestAPI_CartModule
PetStore
PetStore_JSON_UserModel
Create User
Get User By Name
Update User By Name
Delete User By Name
PetStore_XML_PetModel
ReqRes_httpRequests
ReqRes_httpRequestsVariables
StudentAPIs
StudentAPIs-Chaining
StudentAPIs_ResponseValidations
swagger
Create User
Get User By Name
PetStore_JSON_UserModel / Get User By Name
GET
https://petstore.swagger.io/v2/user/{{firstName}}
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
pm.test("Check status code", function () {
pm.response.to.have.status(200);
});
Response


Home
Workspaces
API Network
Explore
myworkspace
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
ReqRes_httpRequestsWorkFlow
Authentications
BooksAPI
BooksAPI-DataDriven
FileUpload
GorestAPI-Chaining
Opencart_RestAPI_CartModule
PetStore
PetStore_JSON_UserModel
Create User
Get User By Name
Update User By Name
Delete User By Name
PetStore_XML_PetModel
ReqRes_httpRequests
ReqRes_httpRequestsVariables
StudentAPIs
StudentAPIs-Chaining
StudentAPIs_ResponseValidations
swagger
Create User
Get User By Name
PetStore_JSON_UserModel / Get User By Name
GET
https://petstore.swagger.io/v2/user/{{firstName}}
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
pm.test("Check status code", function () {
pm.response.to.have.status(200);
});
Body
Cookies
Headers (8)
Test Results (1/1)
Pretty
Raw
Preview
Visualize
JSON
{
"id": 95,
"username": "johnpjn1z3djcck",
"firstName": "johnpjn1z3djcck",
"lastName": "johnpjn1z3djcck",
"email": "johnpjn1z3djcck@gmail.com",
"password": "johnpjn1z3djcck",
"phone": "1111111111",
"userStatus": 0
}


Home
Workspaces
API Network
Explore
myworkspace
New
Import
Create User
Get User By Name
Update User By Name
PetStore_JSON_UserModel / Update User By Name
PUT
https://petstore.swagger.io/v2/user/{{firstName}}
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
var randomstr ="john"+Math.random().toString(36).substring(2);
//Updating email & phone
pm.collectionVariables.set("email", randomstr+"@gmail.com");
pm.collectionVariables.set("phone", "2222222222");
Response
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
ReqRes_httpRequestsWorkFlow
Authentications
BooksAPI
BooksAPI-DataDriven
FileUpload
GorestAPI-Chaining
Opencart_RestAPI_CartModule
PetStore
PetStore_JSON_UserModel
Create User
Get User By Name
Update User By Name
Delete User By Name
PetStore_XML_PetModel
ReqRes_httpRequests
ReqRes_httpRequestsVariables
StudentAPIs
StudentAPIs-Chaining
StudentAPIs_ResponseValidations
swagger


Home
Workspaces
API Network
Explore
myworkspace
New
Import
Create User
Get User By Name
Update User By Name
Delete User By Name
PetStore_JSON_UserModel / Delete User By Name
DELETE
https://petstore.swagger.io/v2/user/{{firstName}}
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
pm.test("Check status code", function () {
pm.response.to.have.status(200);
});
pm.environment.unset("id");
pm.collectionVariables.unset("username");
pm.collectionVariables.unset("firstName");
pm.collectionVariables.unset("lastName");
pm.collectionVariables.unset("email");
pm.collectionVariables.unset("password");
pm.collectionVariables.unset("phone");
pm.collectionVariables.unset("userStatus");
Response
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
ReqRes_httpRequestsWorkFlow
Authentications
BooksAPI
BooksAPI-DataDriven
FileUpload
GorestAPI-Chaining
Opencart_RestAPI_CartModule
PetStore
PetStore_JSON_UserModel
Create User
Get User By Name
Update User By Name
Delete User By Name
PetStore_XML_PetModel
ReqRes_httpRequests
ReqRes_httpRequestsVariables
StudentAPIs
StudentAPIs-Chaining
StudentAPIs_ResponseValidations
swagger


Home
Workspaces
API Network
Explore
myworkspace
New
Import
PetStore_JSON_UserModel
All Tests
Passed (5)
Failed (0)
Skipped (0)
Iteration 1
POST Create User
Pass
Check status code
Pass
Check status code
GET Get User By Name
Pass
Check status code
PUT Update User By Name
Pass
Check status code
DELETE Delete User By Name
Pass
Check status code
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
ReqRes_httpRequestsWorkFlow
Authentications
BooksAPI
BooksAPI-DataDriven
FileUpload
GorestAPI-Chaining
Opencart_RestAPI_CartModule
PetStore
PetStore_JSON_UserModel
Create User
Get User By Name
Update User By Name
Delete User By Name
PetStore_XML_PetModel
ReqRes_httpRequests
ReqRes_httpRequestsVariables
StudentAPIs
StudentAPIs-Chaining
StudentAPIs_ResponseValidations
swagger

Day-9

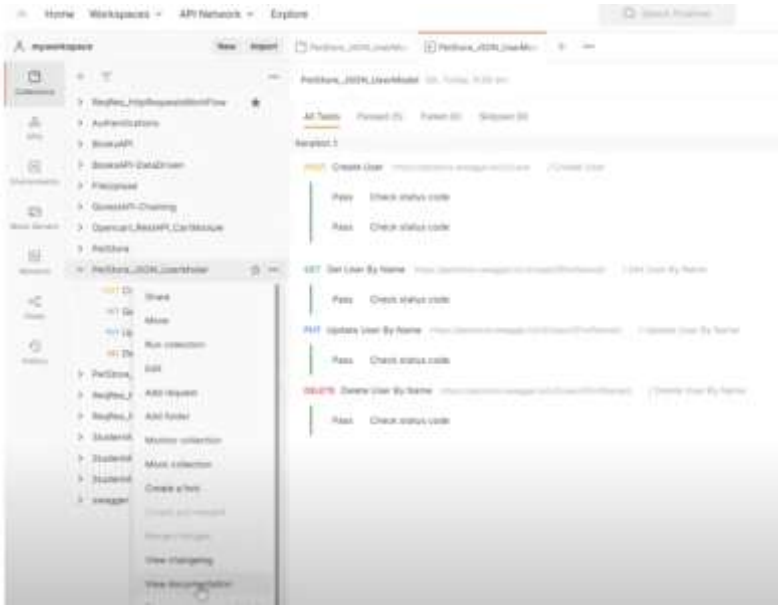
Petstore -

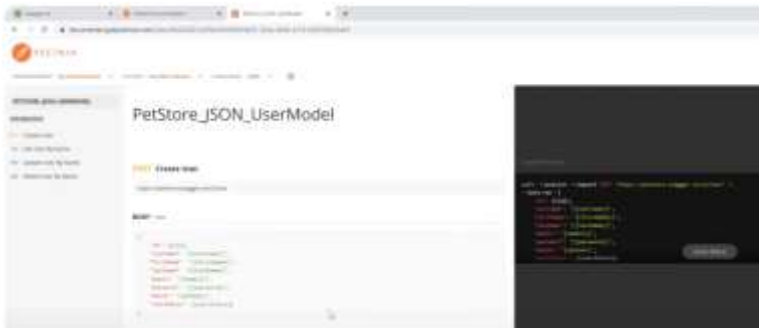
Json

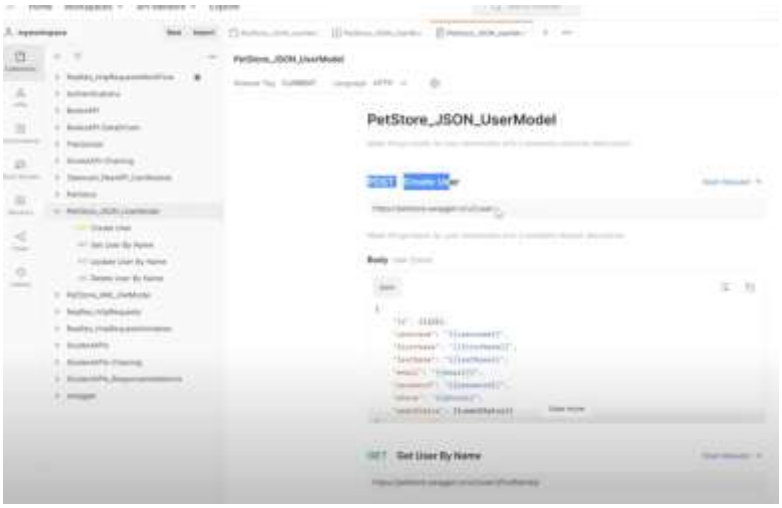
XML

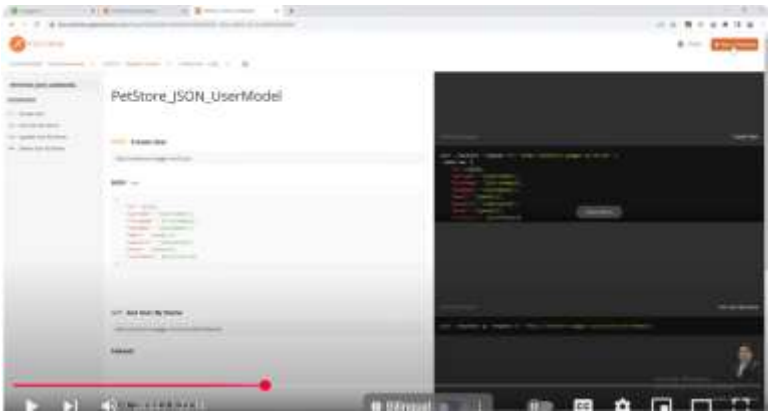
https://petstore.swagger.io/

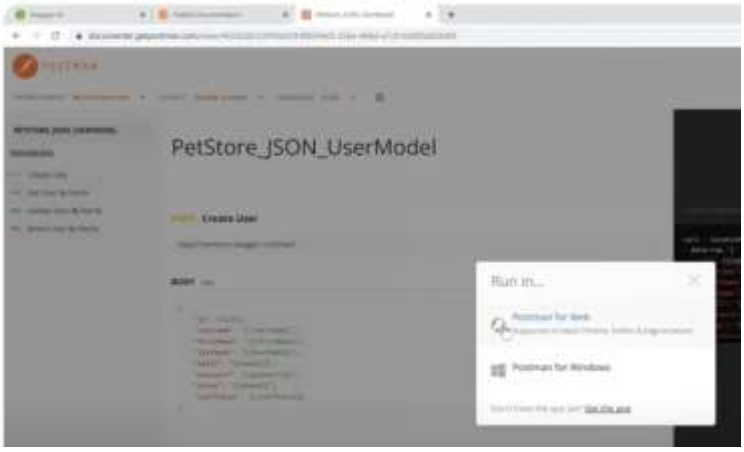
published PetStore API : https://documenter.getpostman.com/view/4924228/UzXNVdJS

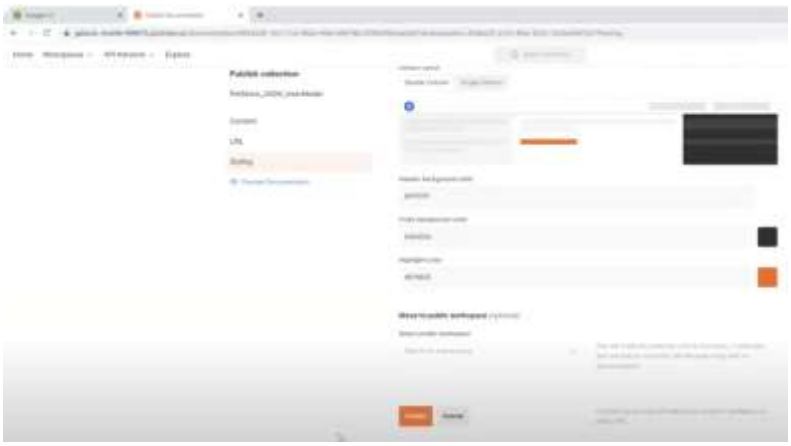


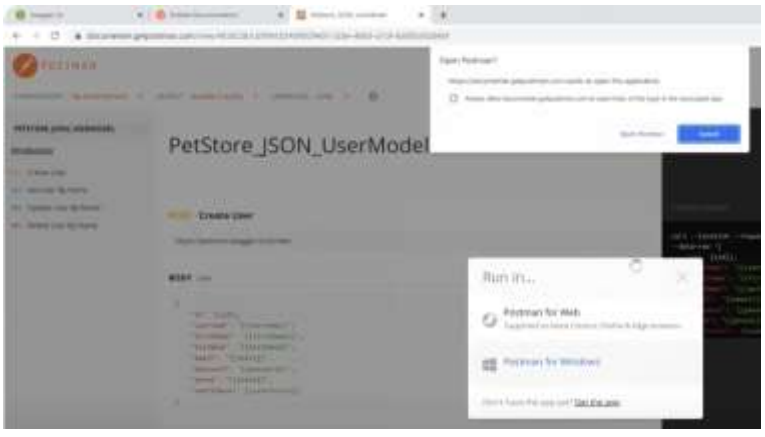


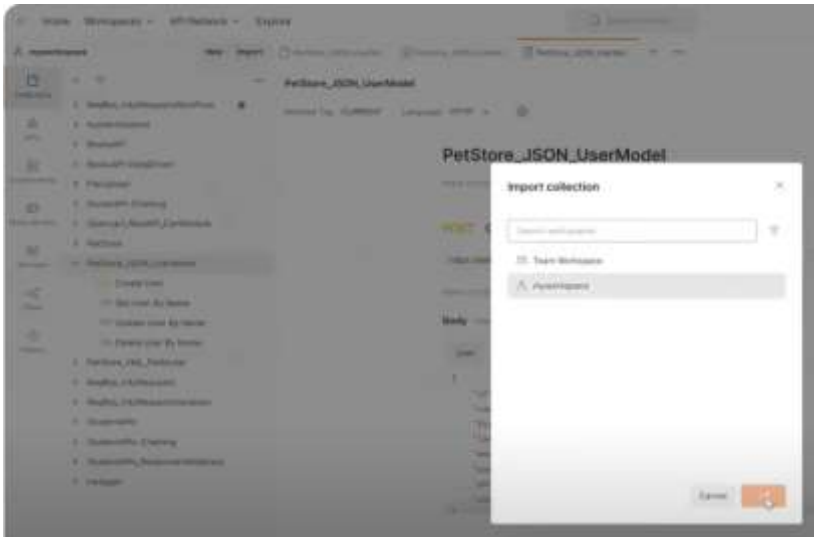


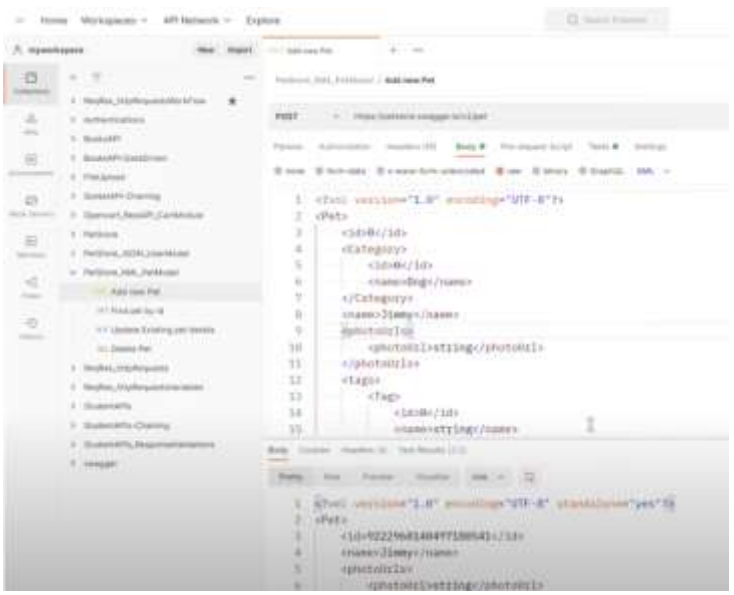












Day-9

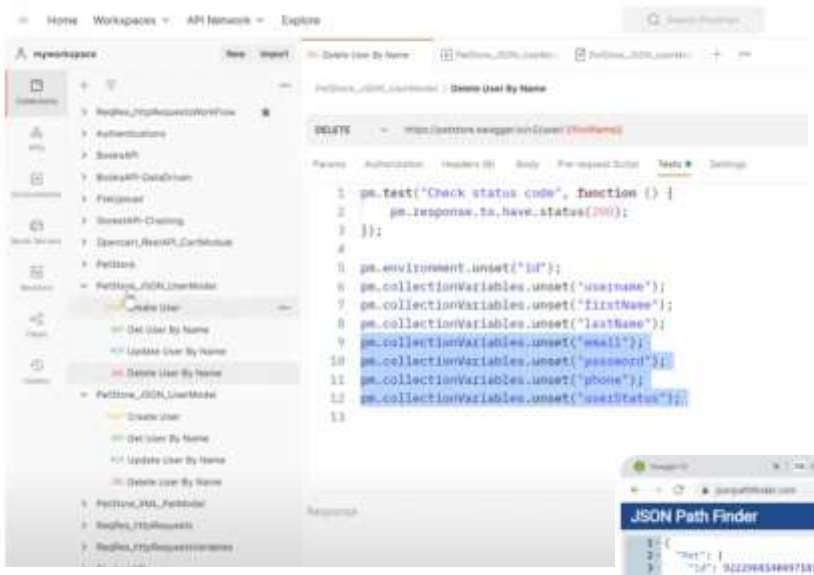
Petstore -
Json
XML

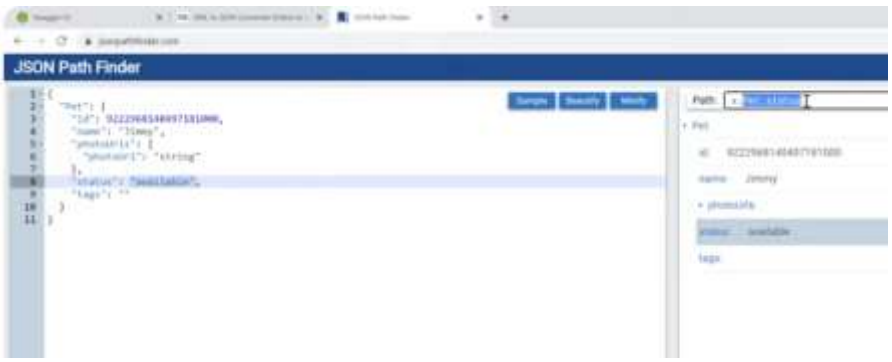
https://petstore.swagger.io/

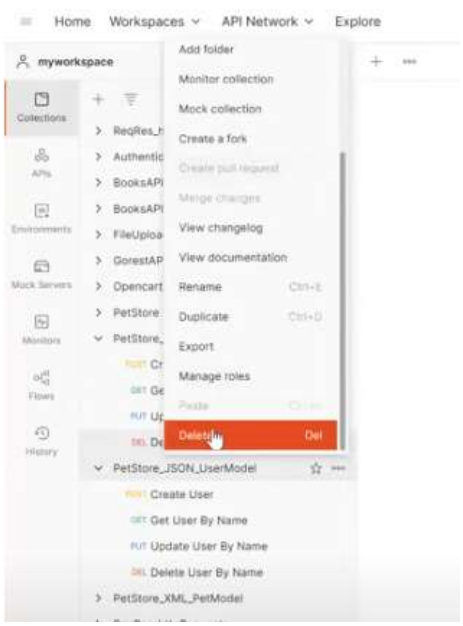
Swagger - Exploring API request
interactive document

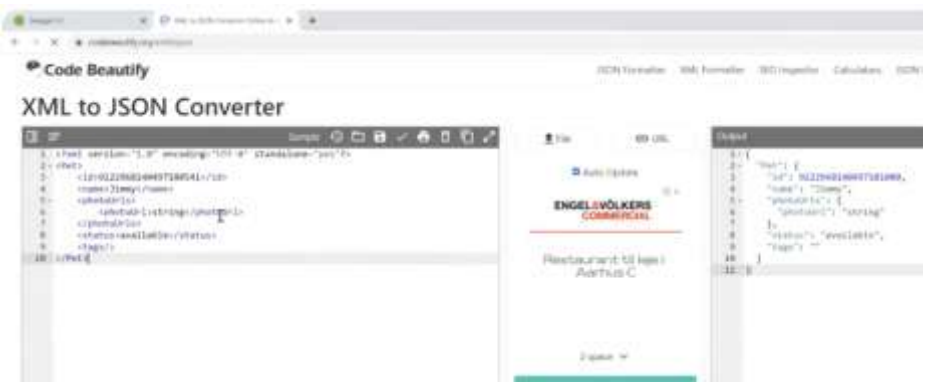
create documentation
publish it

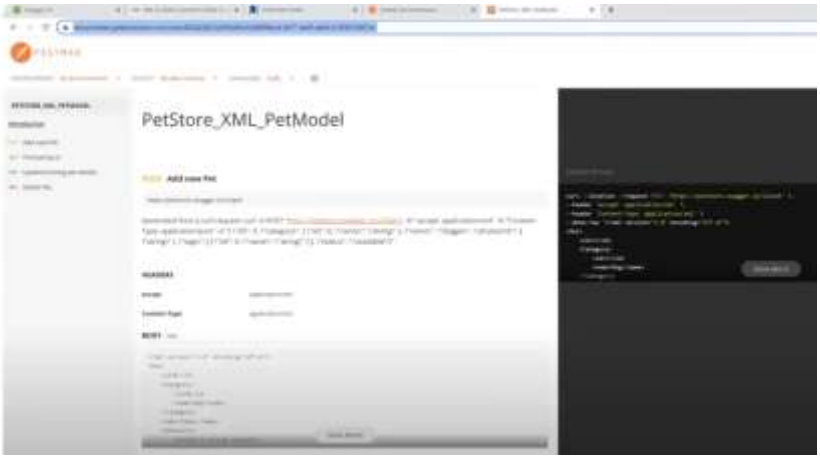
published PetStore API : https://documenter.getpostman.com/view/4924228/UzXNVdJS











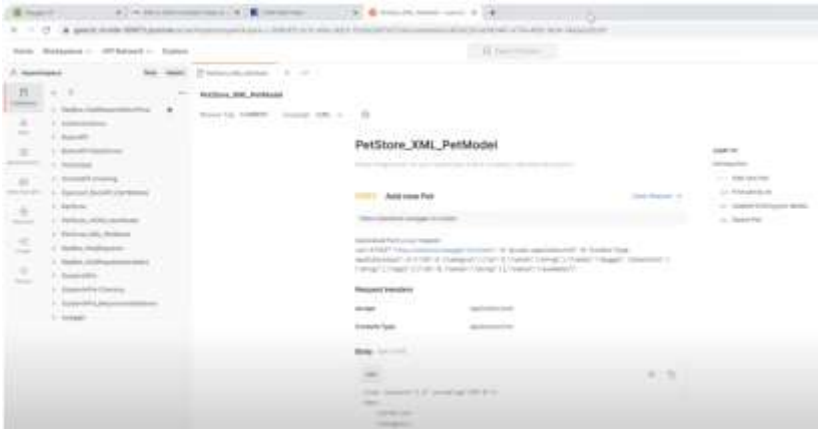
Json
XML

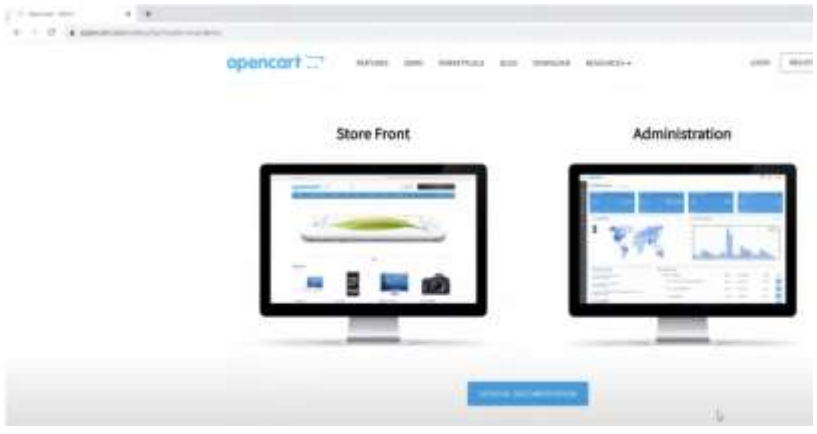
https://petstore.swagger.io/

Swagger - Exploring API request
interactive document

create documentation
publish it

published PetStore API's :
---
https://documenter.getpostman.com/view/4924228/UzXNVdJS
https://documenter.getpostman.com/view/4924228/UzXNVdNo



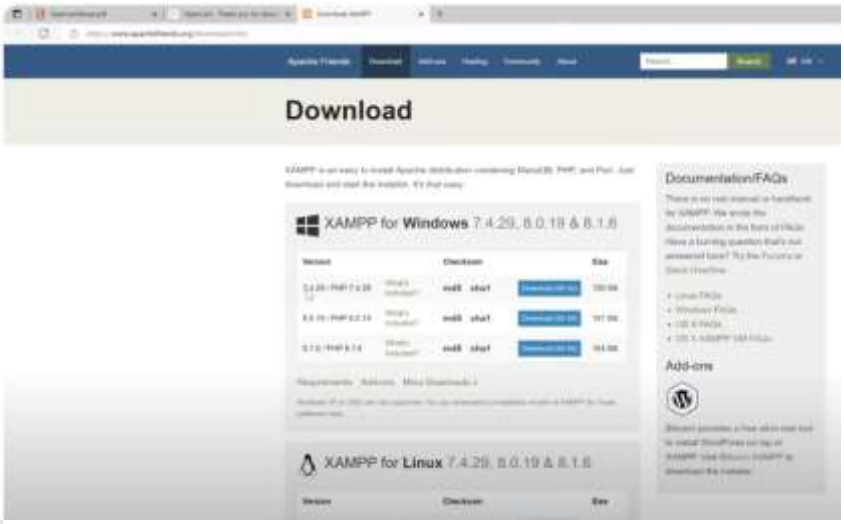


Opencart Build Deployment/Installation

1) **Opencart download location:**
   https://www.opencart.com/index.php?route=cms/download

Click on previous releases

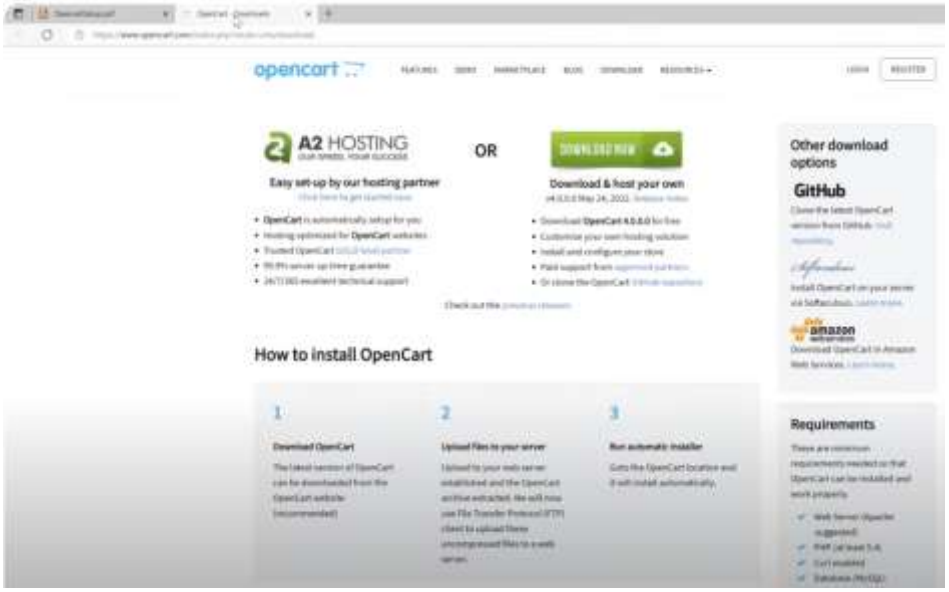
Recommended version: opencart-3.0.3.8

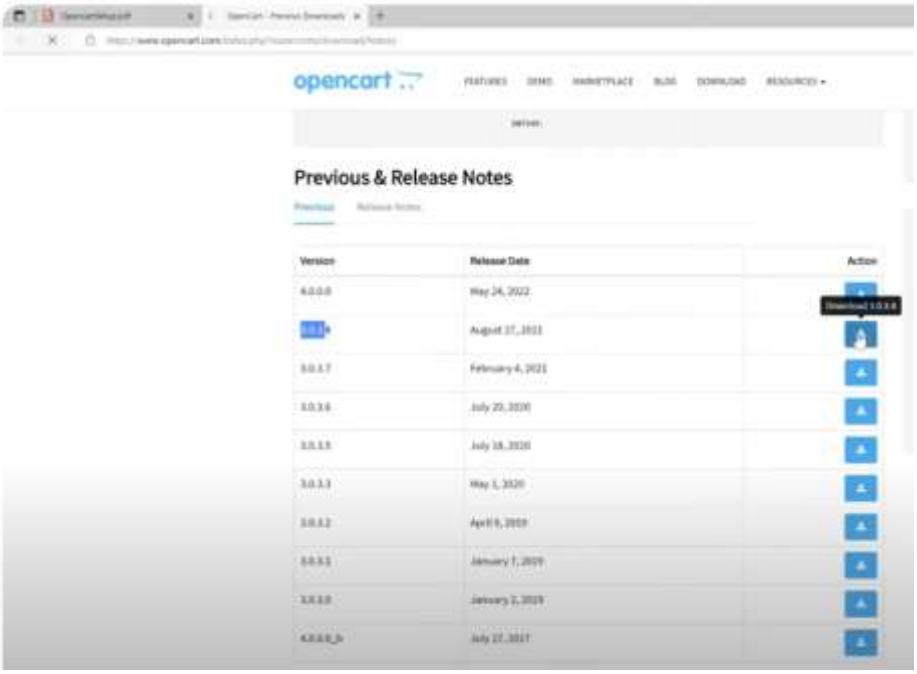


2) **XAMPP for apache, mysql, and php installation.**

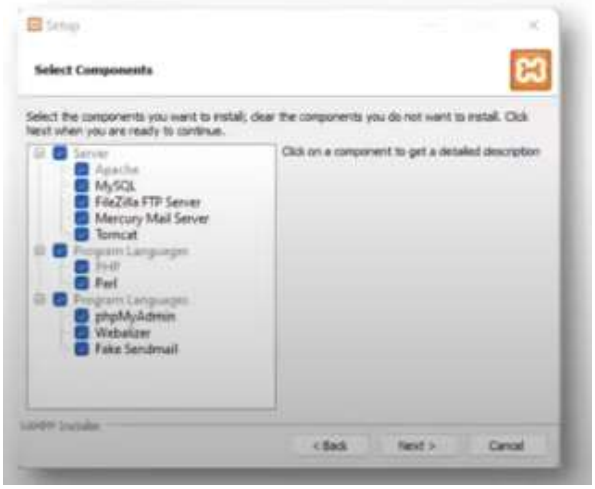
Download Link: https://www.apachefriends.org/download.html

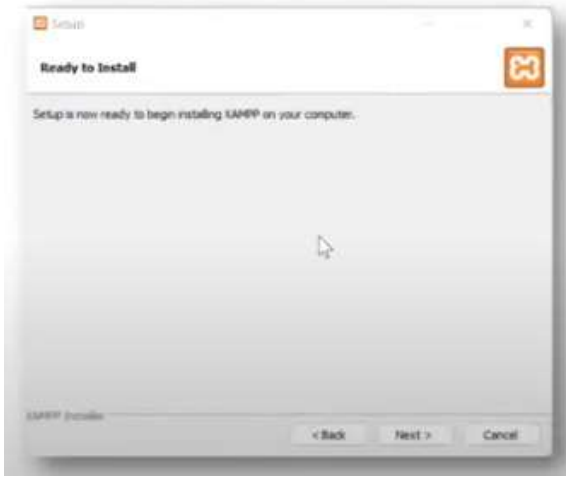
Recommended version: 7.4.29/PHP 7.4.29

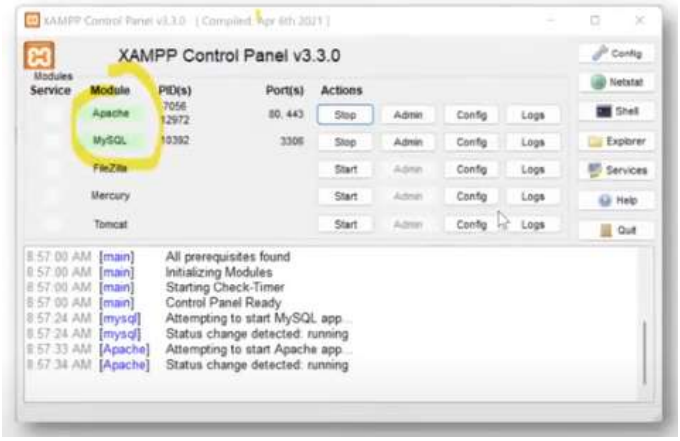




Step by step XAMPP Installation





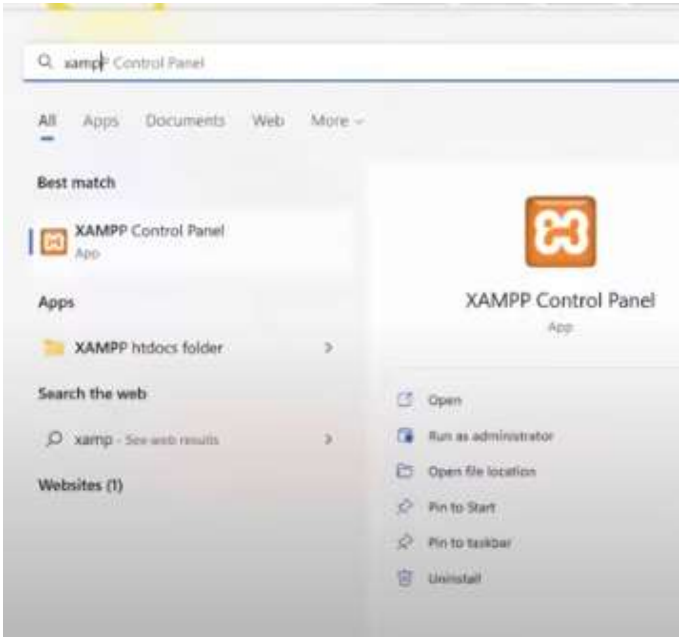


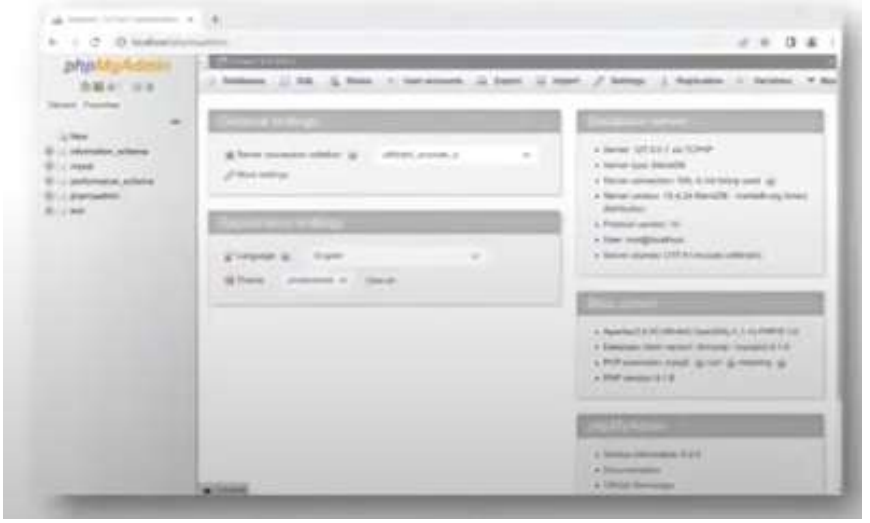
5) **Rename files**
Go to C:\xampp\htdocs\opencart\upload
Rename file 'config-dist.php' to 'config.php'

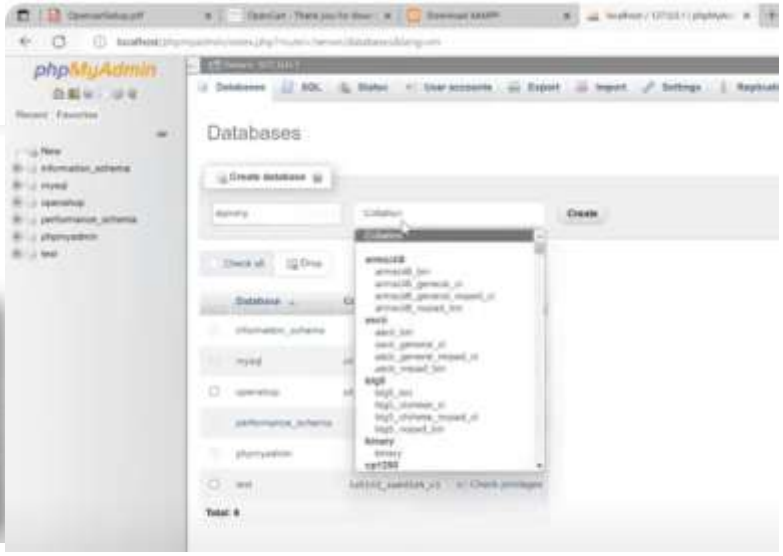
Go to C:\xampp\htdocs\opencart\upload\admin
Rename file 'config-dist.php' to 'config.php'

6) **Connect to the database and create DB.**
DB Access URL: http://localhost/phpmyadmin/

Create new database 'openshop' (Refer the screenshots below to create new Database in MySQL)



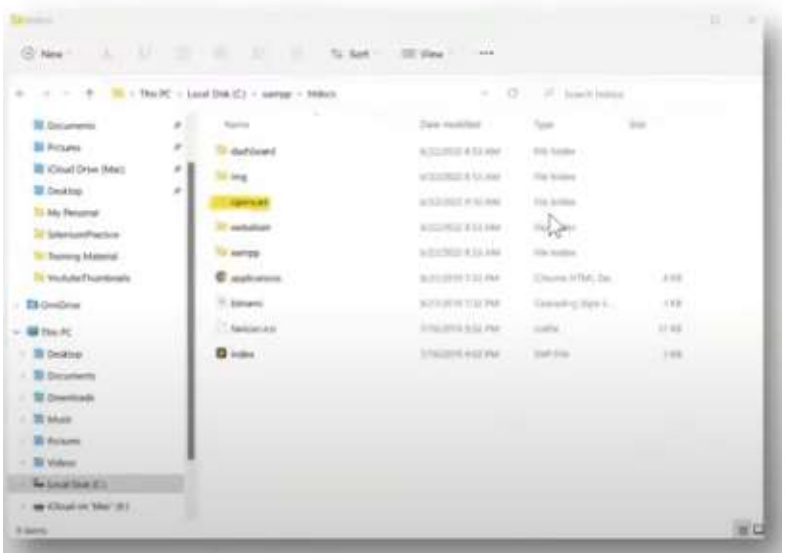


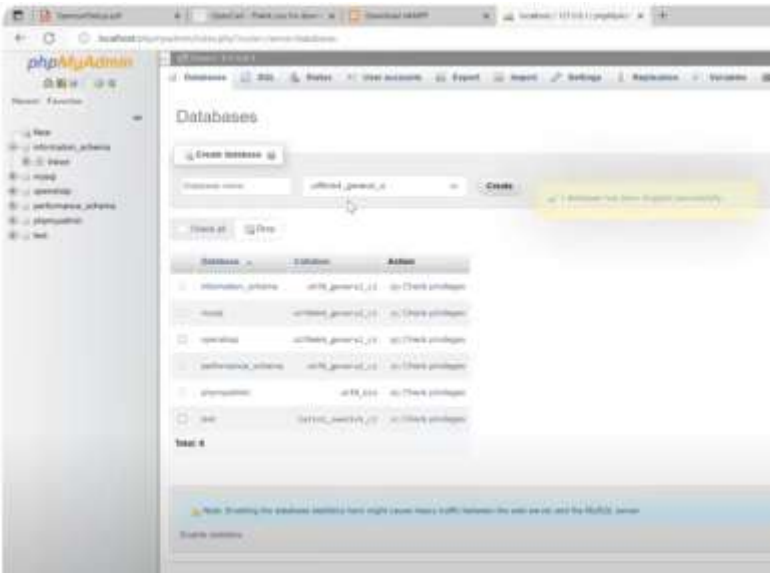


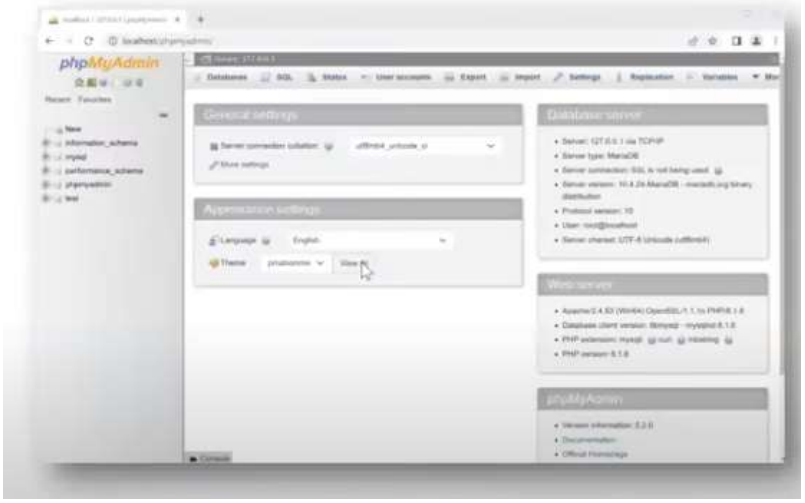
Step by step OpenCart Installation

3) **Copy opencart folder(Step1) in to below location. C:\xampp\htdocs**

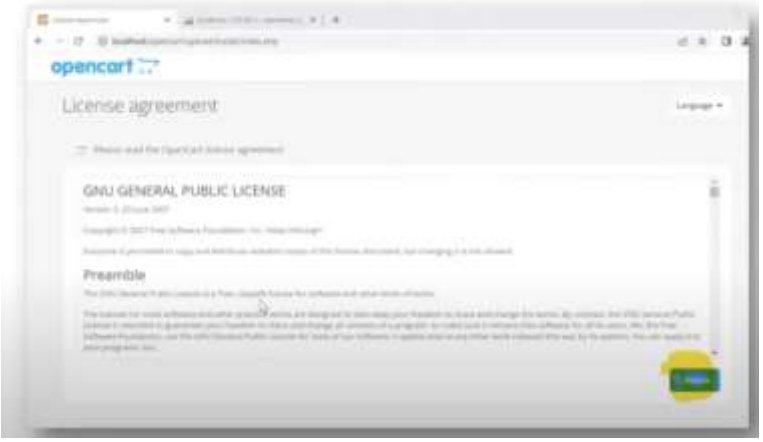
4) **After copying rename the folder " opencart "**

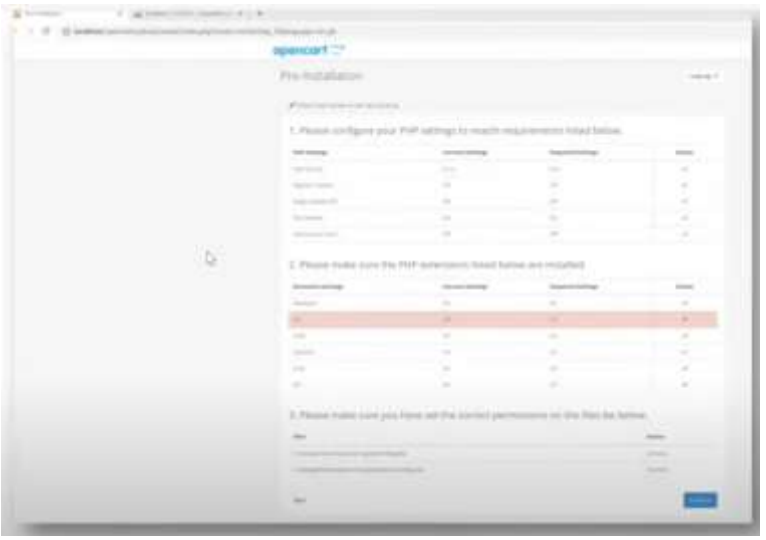


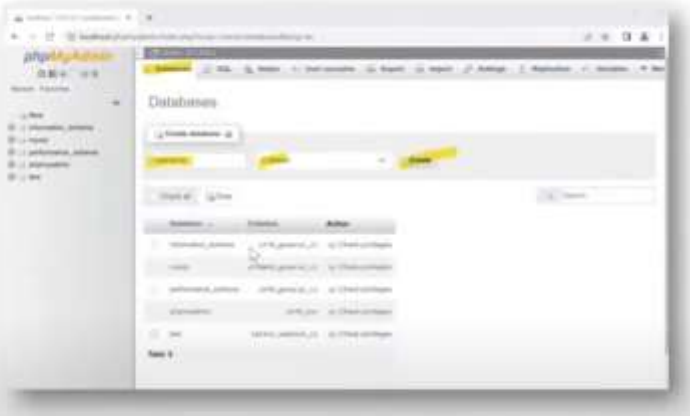




Click on CONTINUE Button....



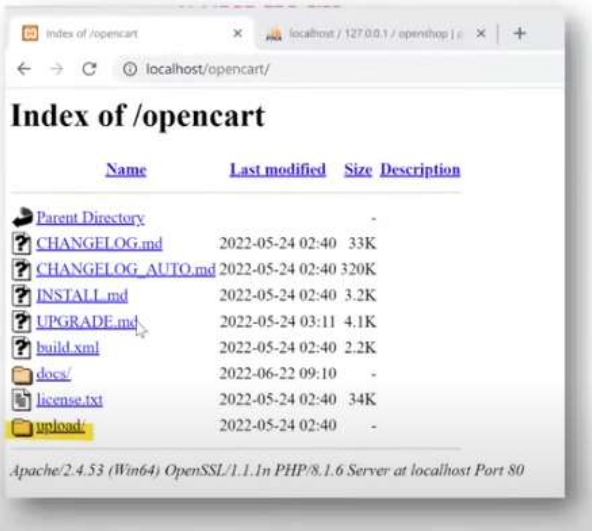




7) Open the site http://localhost/opencart/
Click on upload

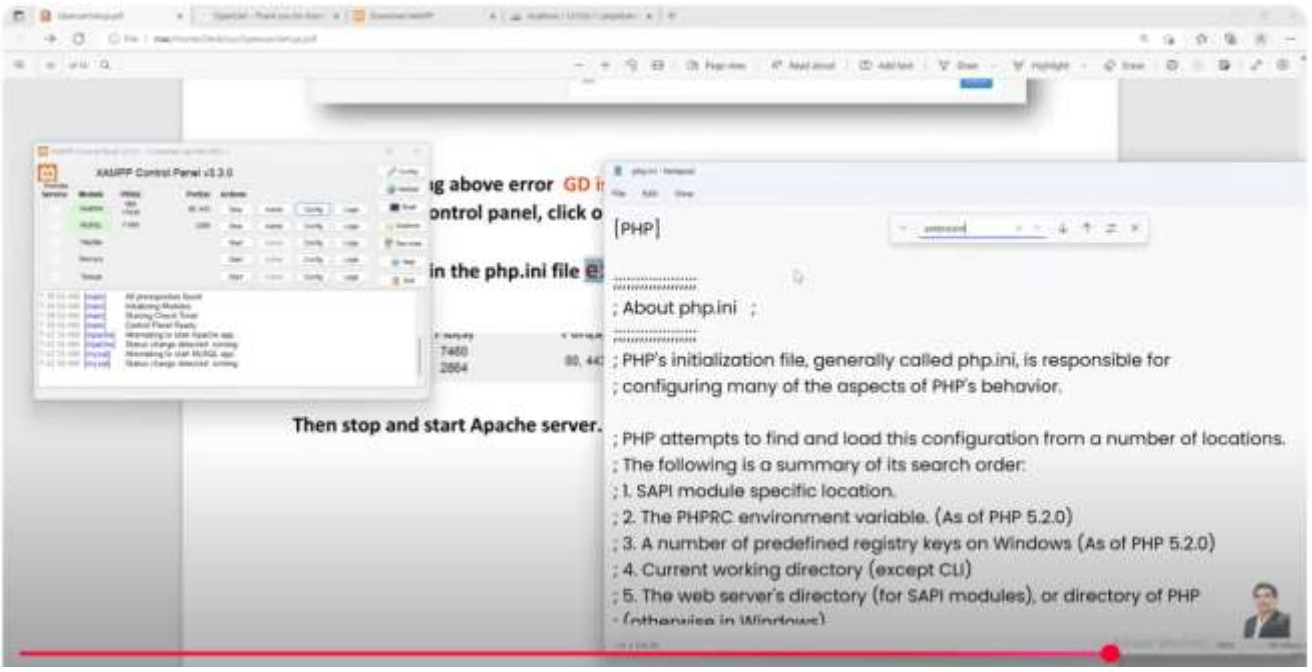
If you are getting above error GD is off then you have to do below steps.
Open XAMPP Control panel, click on config button for Apache then open PHP(php.ini) file
Add new entry in the php.ini file `extension=gd`

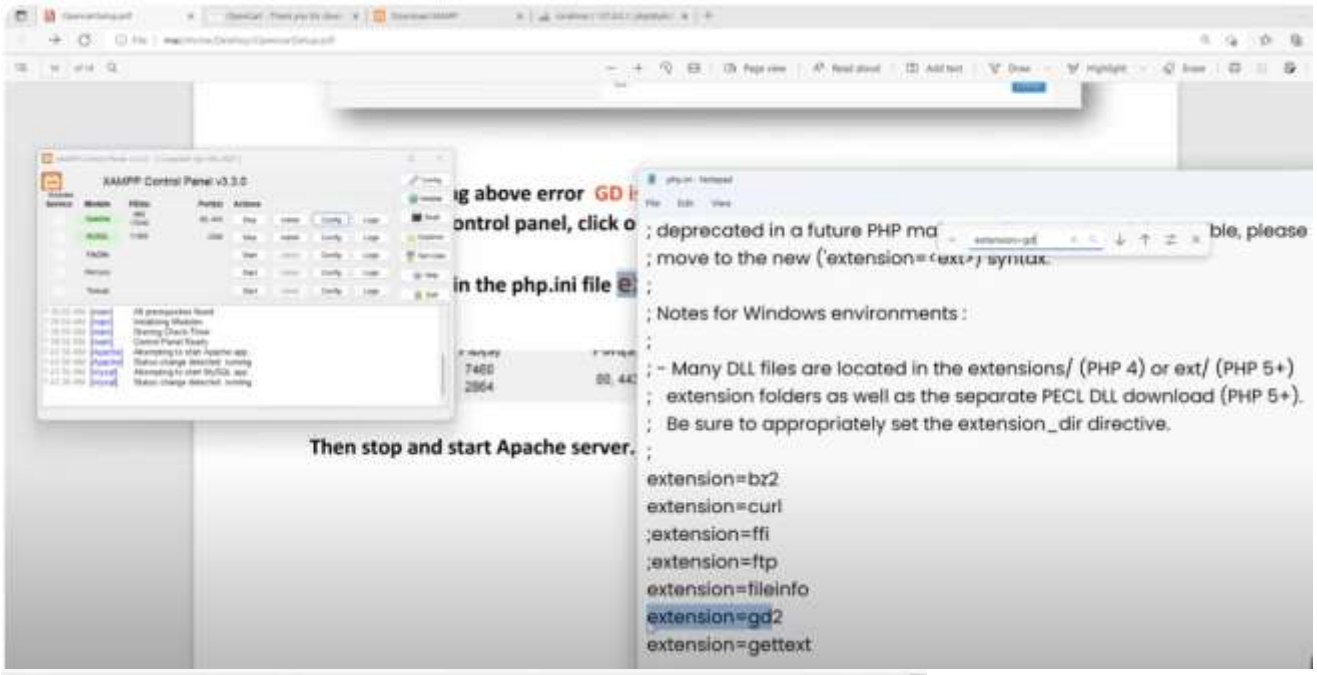
Then stop and start Apache server. Then you will see below screen.

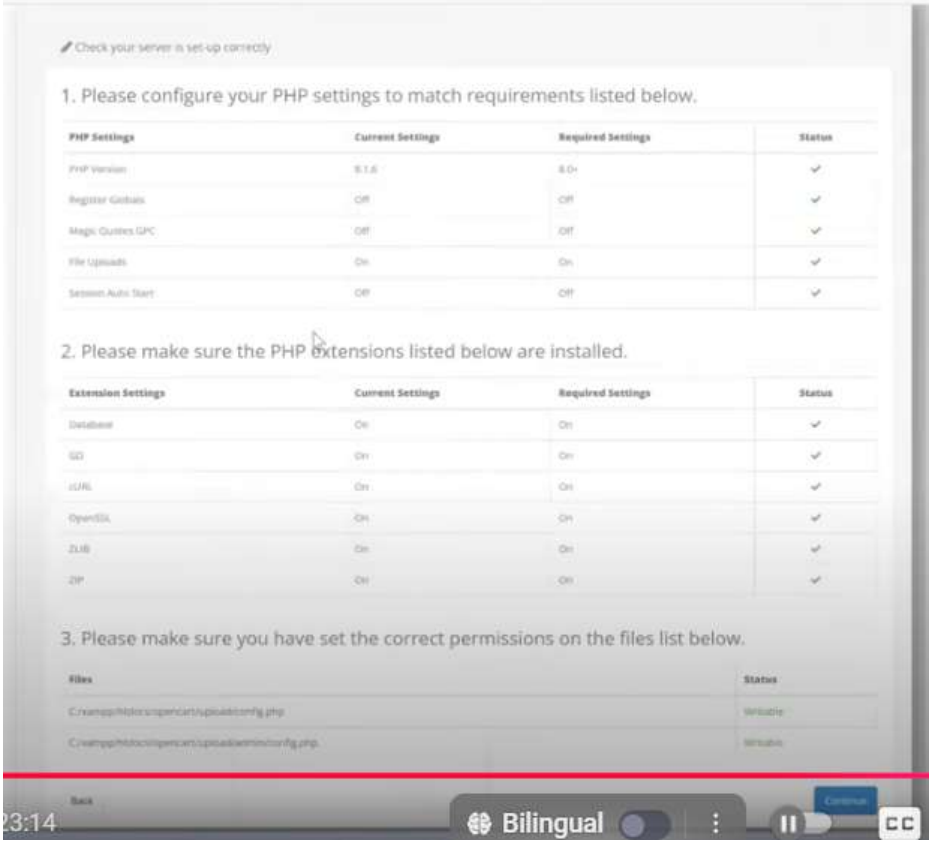


Navigate to http://localhost/opencart/upload/install/index.php

Click on CONTINUE Button....

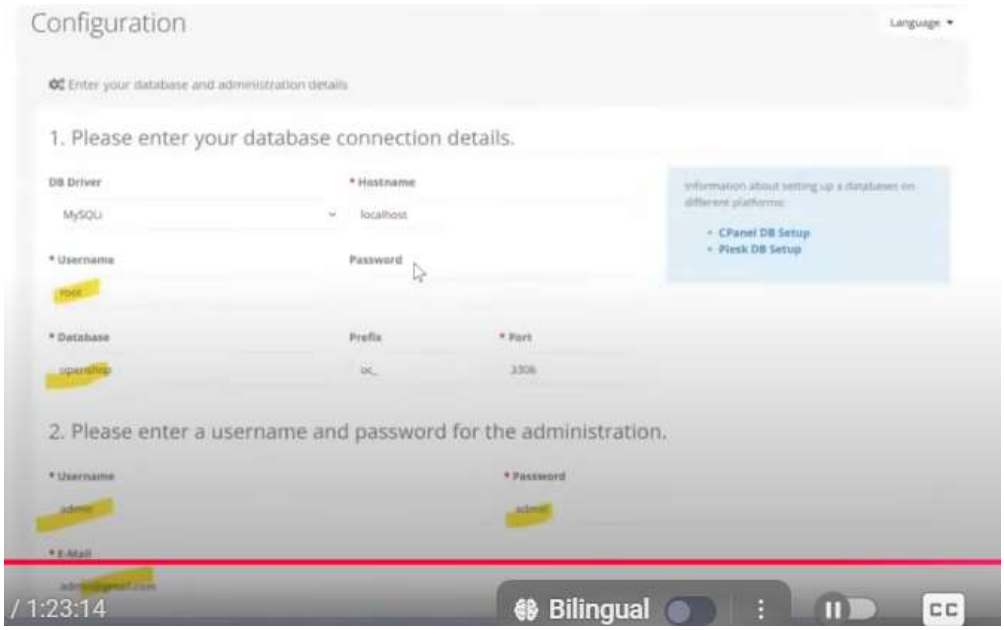

XAMPP Control Panel v3.3.0
[PHP]
; About php.ini ;
; PHP's initialization file, generally called php.ini, is responsible for
; configuring many of the aspects of PHP's behavior.
; PHP attempts to find and load this configuration from a number of locations.
; The following is a summary of its search order:
; 1. SAPI module specific location.
; 2. The PHPRC environment variable. (As of PHP 5.2.0)
; 3. A number of predefined registry keys on Windows (As of PHP 5.2.0)
; 4. Current working directory (except CLI)
; 5. The web server's directory (for SAPI modules), or directory of PHP
Then stop and start Apache server.

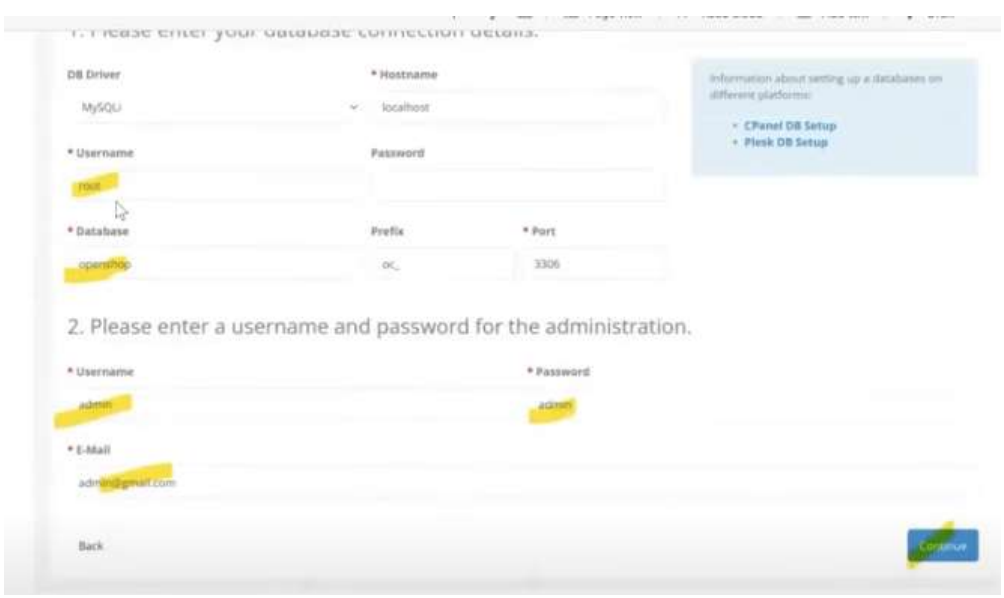

XAMPP Control Panel v3.3.0
; deprecated in a future PHP ma
ble, please
; move to the new ('extension=ext/') syntax.
;
; Notes for Windows environments :
;
; - Many DLL files are located in the extensions/ (PHP 4) or ext/ (PHP 5+)
; extension folders as well as the separate PECL DLL download (PHP 5+).
; Be sure to appropriately set the extension_dir directive.
;
extension=bz2
extension=curl
;extension=ffi
;extension=ftp
extension=fileinfo
extension=gd2
extension=gettext
Then stop and start Apache server.

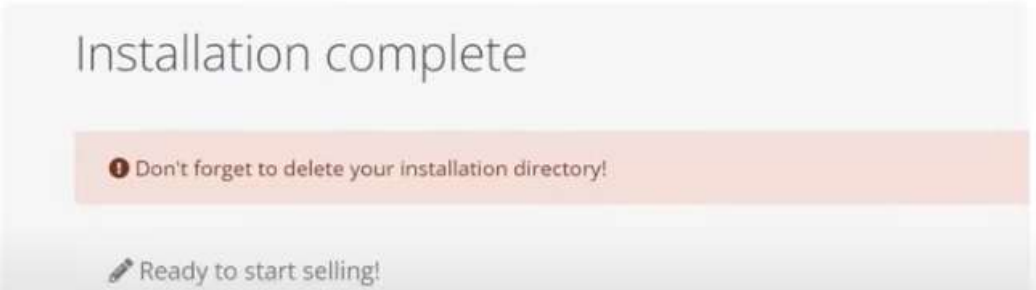

Check your server is set-up correctly
1. Please configure your PHP settings to match requirements listed below.
PHP Settings
Current Settings
Required Settings
Status
2. Please make sure the PHP extensions listed below are installed.
Extension Settings
Current Settings
Required Settings
Status
3. Please make sure you have set the correct permissions on the files list below.
Files
Status
Bilingual
23:14

Click on continue.....

8) Provide Database connection details

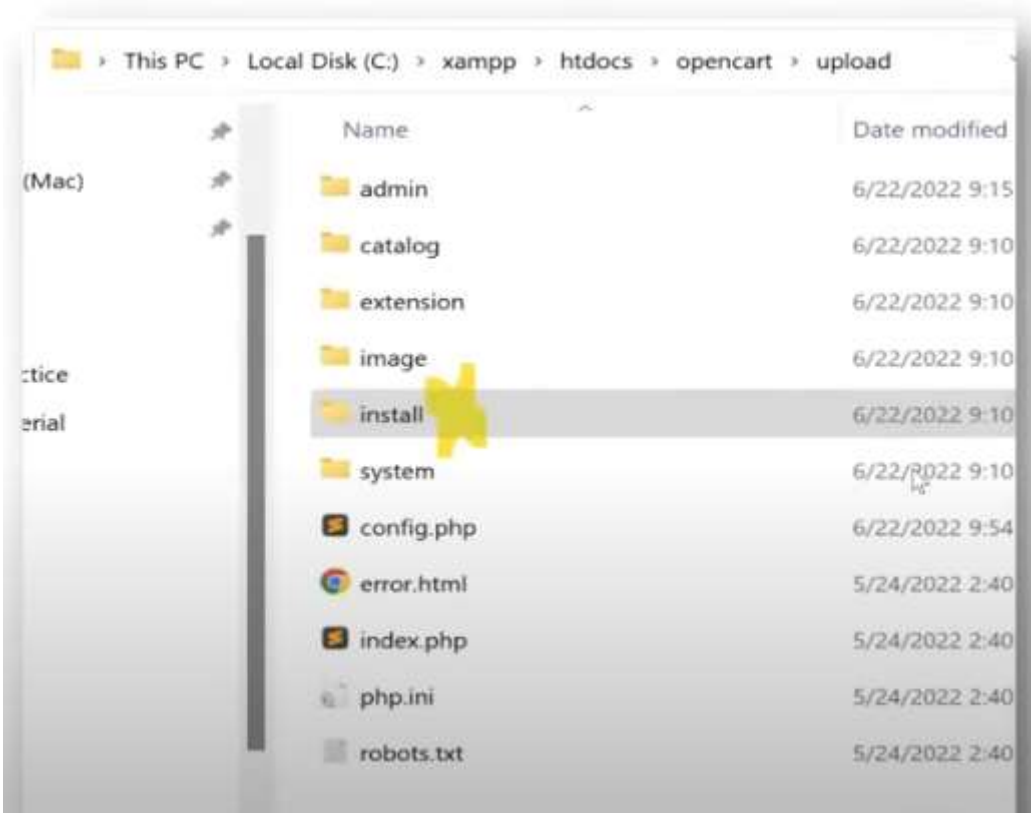






9) Go to location C:\xampp\htdocs\opencart\upload

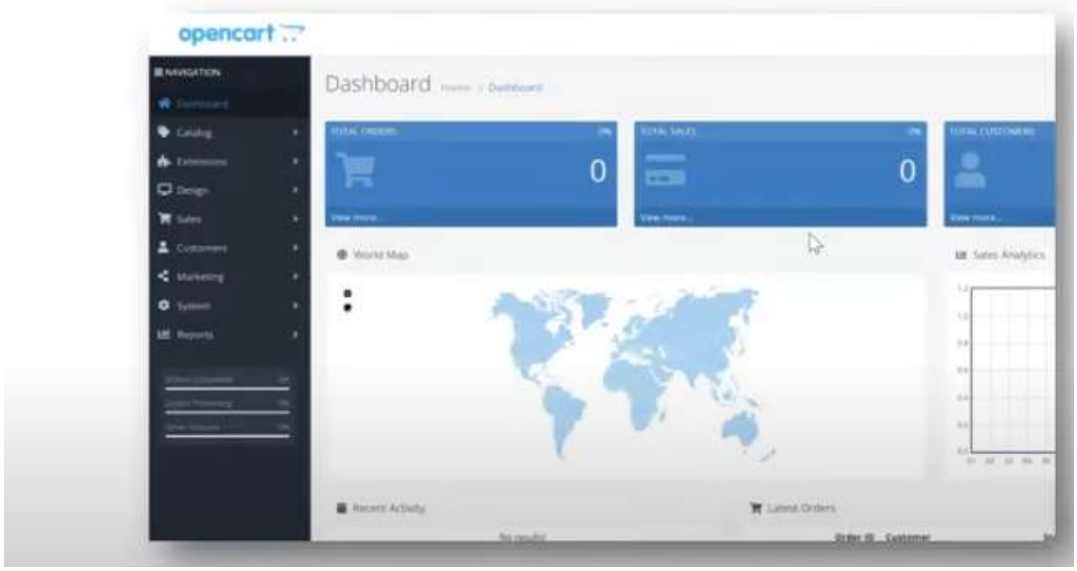
Delete install folder



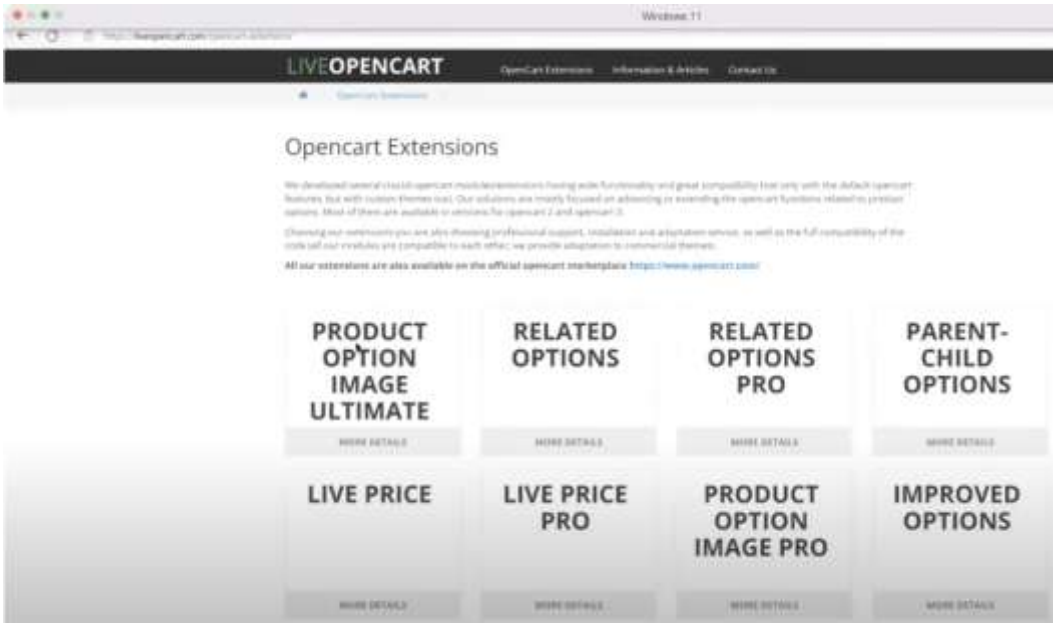
10) Go to Application URL

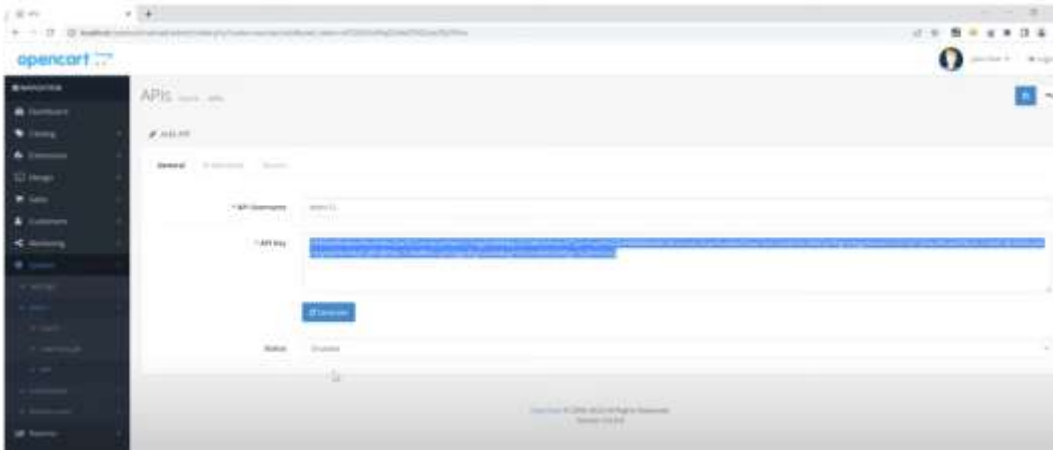
Frontend application URL: http://localhost/opencart/upload/

Backend Application url: http://localhost/opencart/upload/admin/




Windows 11
opencart
Please enter your login details.
Username
Password
Login


Windows 11
LIVEOPENCART
OpenCart Extensions
Information & Articles
Contact Us
Opencart Extensions
PRODUCT OPTION IMAGE ULTIMATE
RELATED OPTIONS
RELATED OPTIONS PRO
PARENT-CHILD OPTIONS
LIVE PRICE
LIVE PRICE PRO
PRODUCT OPTION IMAGE PRO
IMPROVED OPTIONS
MORE DETAILS


opencart
APIs
Dashboard
Catalog
Extensions
Design
Sales
Customers
Marketing
System
Reports

XML

https://petstore.swagger.io/

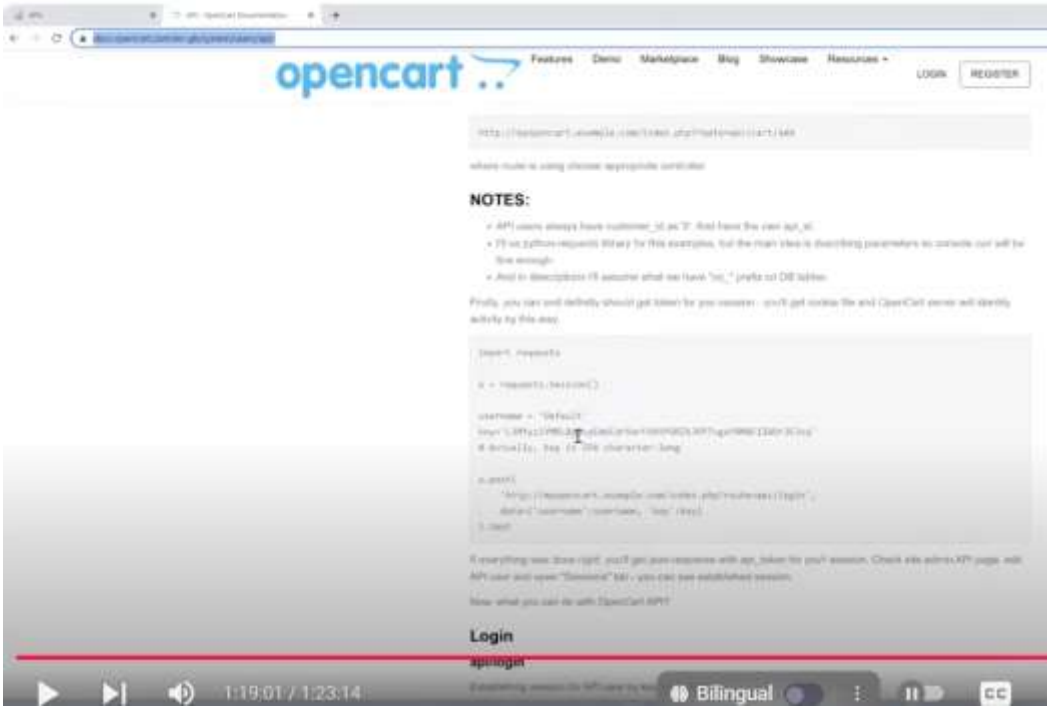
Swagger - Exploring API request
interactive document

create documentation
publish it

published PetStore API's :
---
https://documenter.getpostman.com/view/4924228/UzXNVdJS
https://documenter.getpostman.com/view/4924228/UzXNVdNo

E-Commerce application
---
opencart

Opencart API Documentation:
https://docs.opencart.com/en-gb/system/users/api/



XML

https://petstore.swagger.io/

Swagger - Exploring API request
interactive document

create documentation
publish it

published PetStore API's :
---
https://documenter.getpostman.com/view/4924228/UzXNVdJS
https://documenter.getpostman.com/view/4924228/UzXNVdNo

E-Commerce application

Install opencart app
API user

Opencart API Documentation: https://docs.opencart.com/en-gb/system/users/api/

Day-10

Opencart Application API

Opencart Build Deployment/Installation

1) **Opencart download location:**
   https://www.opencart.com/index.php?route=cms/download

Click on previous releases

Recommended version: opencart-3.0.3.8

2) **XAMPP for apache, mysql, and php installation.**

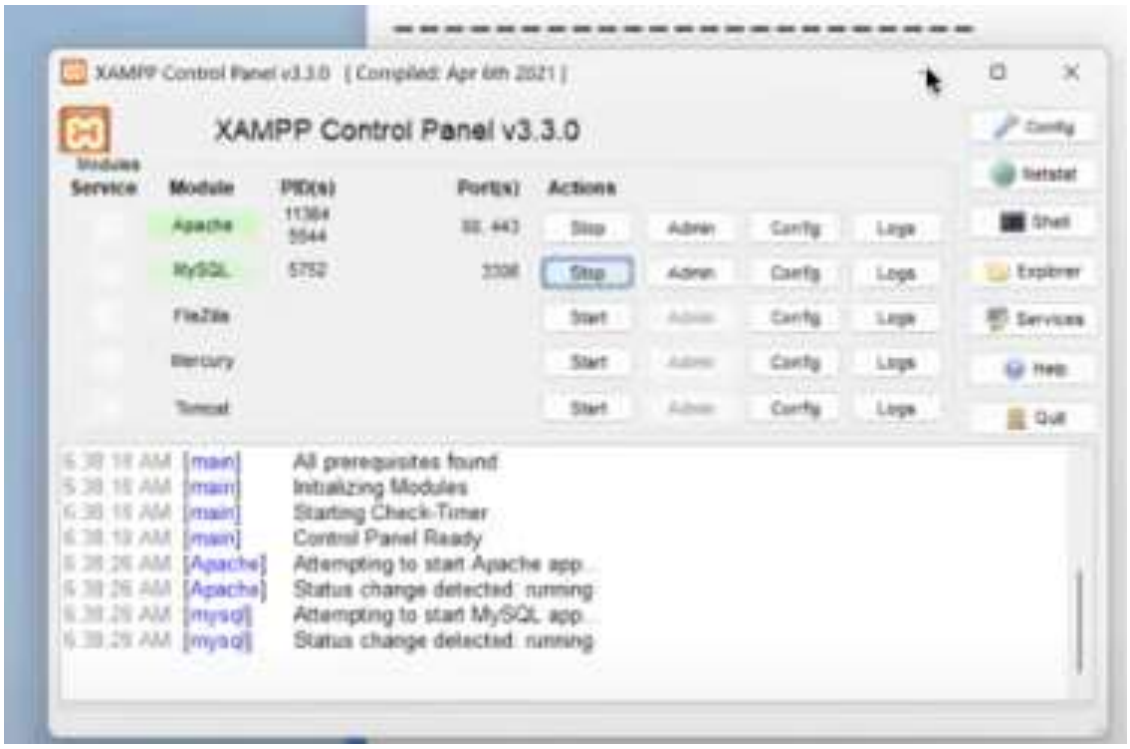
Download Link: https://www.apachefriends.org/download.html

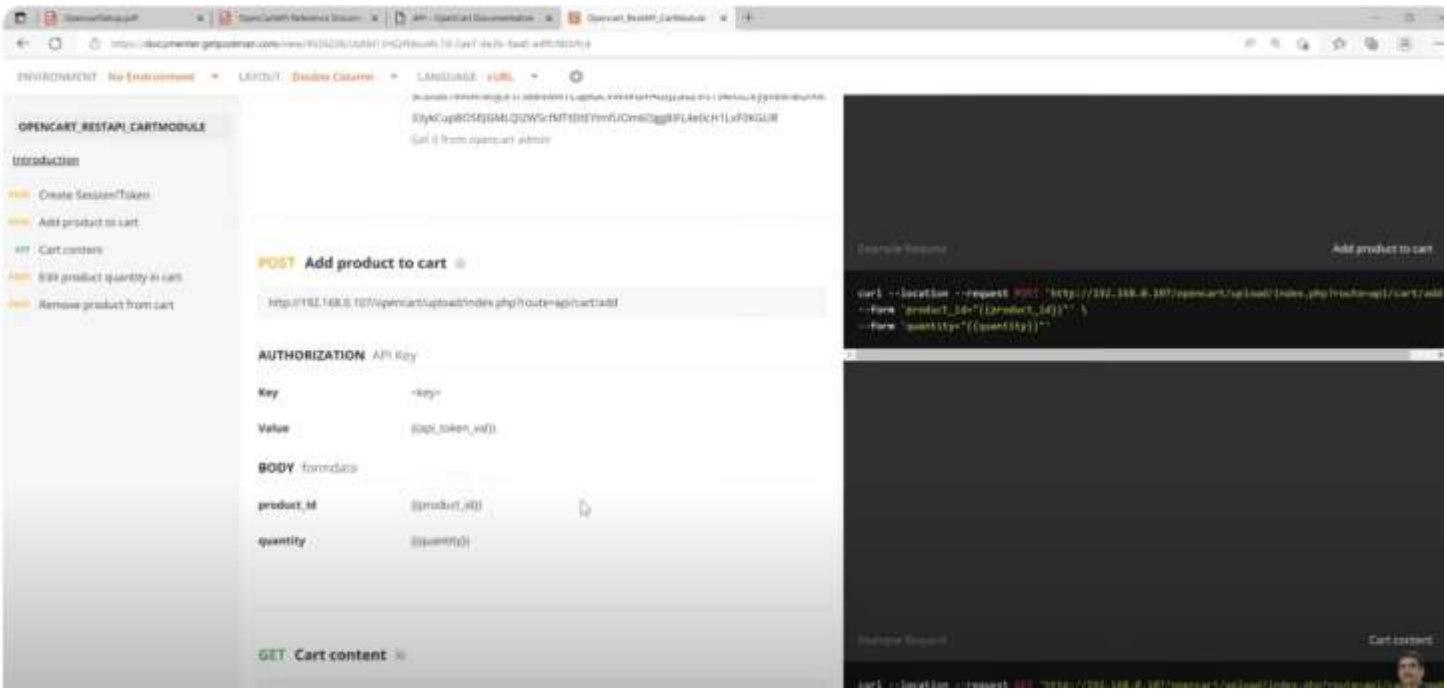
Recommended version: 7.4.29/PHP 7.4.29

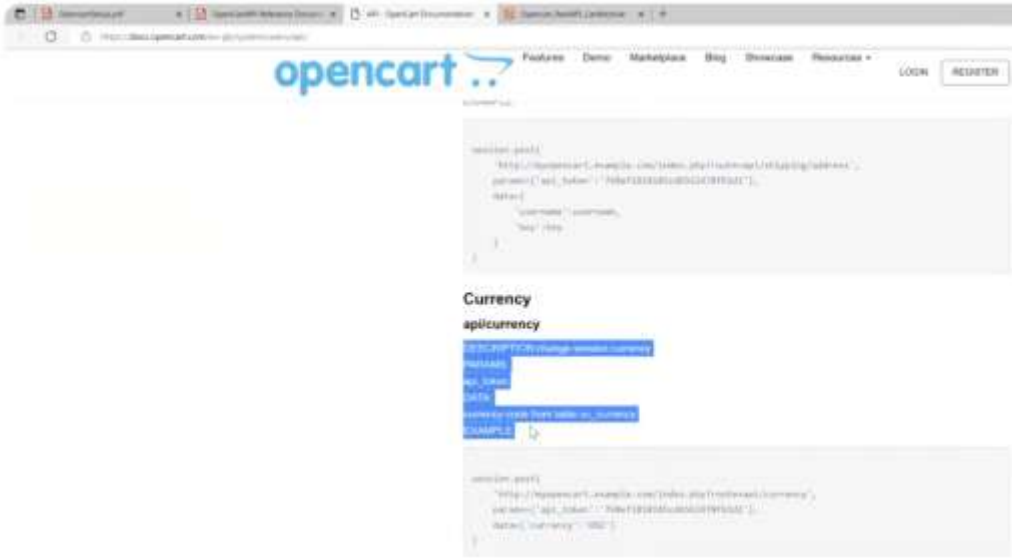
2) **XAMPP for apache, mysql, and php installation.**

Download Link: https://www.apachefriends.org/download.html

Recommended version: 7.4.29/PHP 7.4.29







Open Cart Rest API (Cart Module)

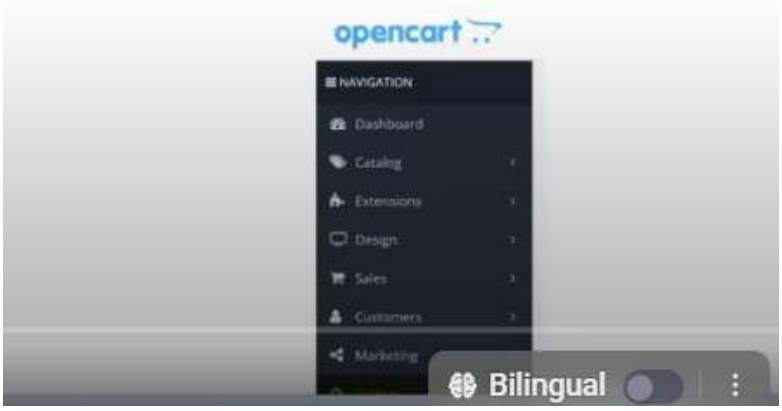
Reference Documents:

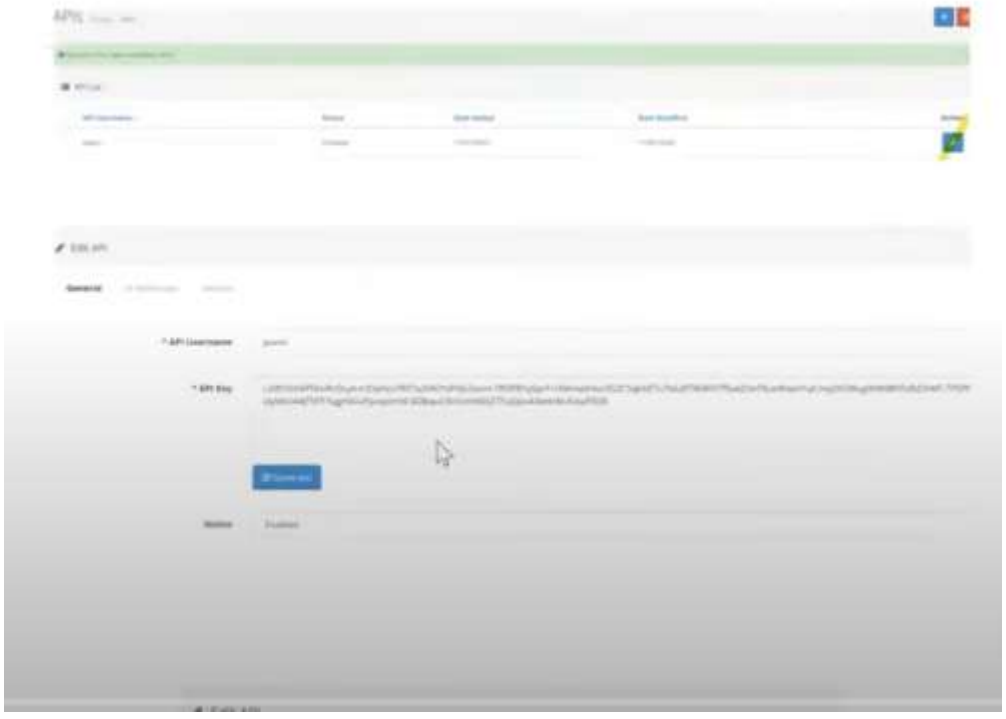
https://documenter.getpostman.com/view/4924228/UzXM1JmQ

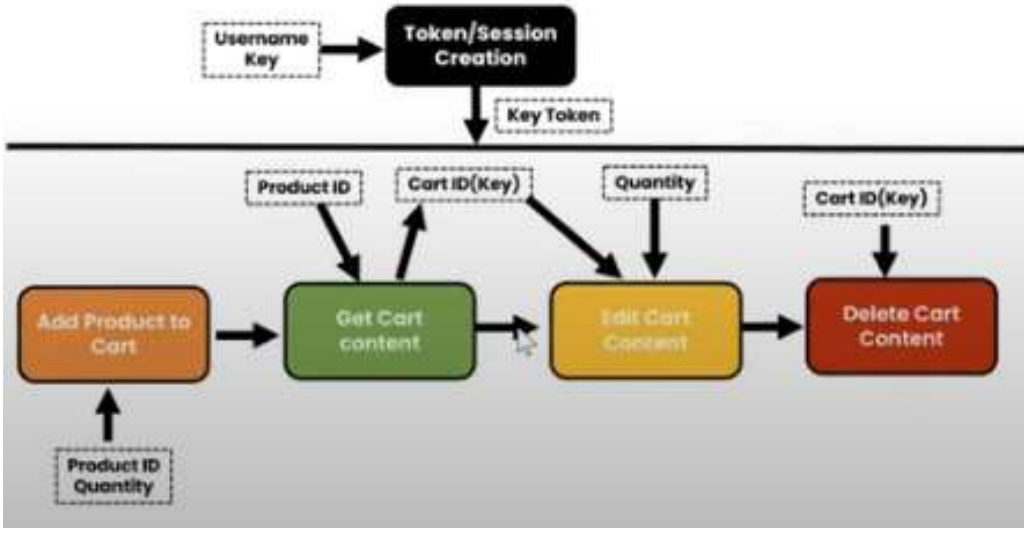
https://docs.opencart.com/en-gb/system/users/api/

Pre-Requisite: Get info from OpenCart Admin application.

Login to Admin → System → Users → API → API Username
Edit → API Key
Edit → IP Address







Day-10

Opencart Application API

---

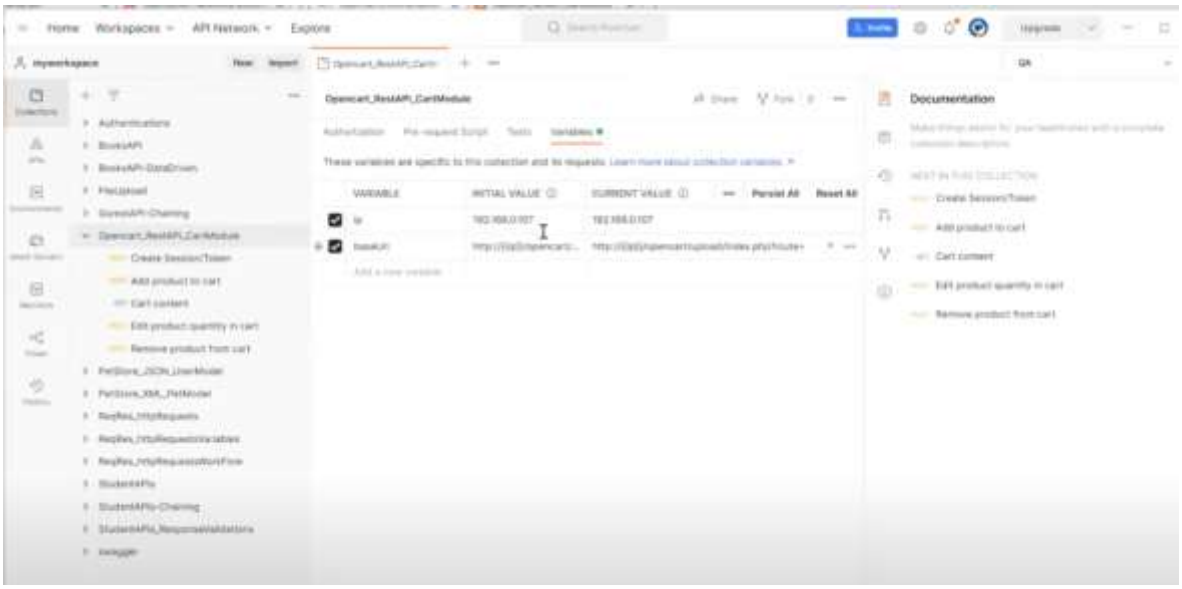
ip = 134.56.78.90

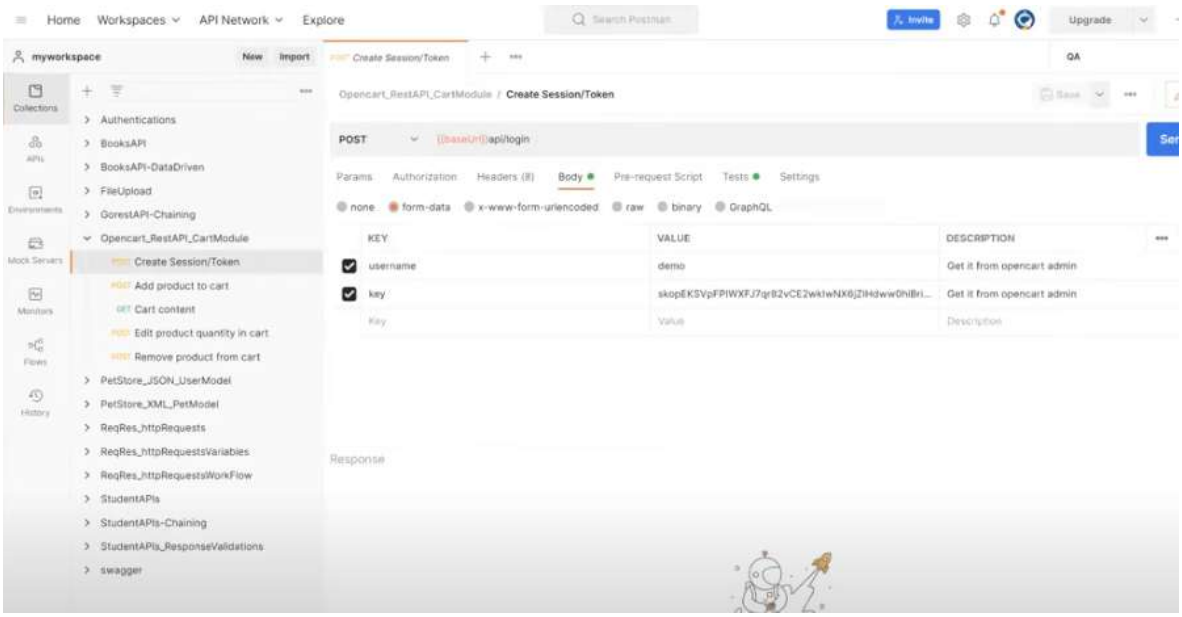
baseURL = http://{{ip}}/index.php?route=

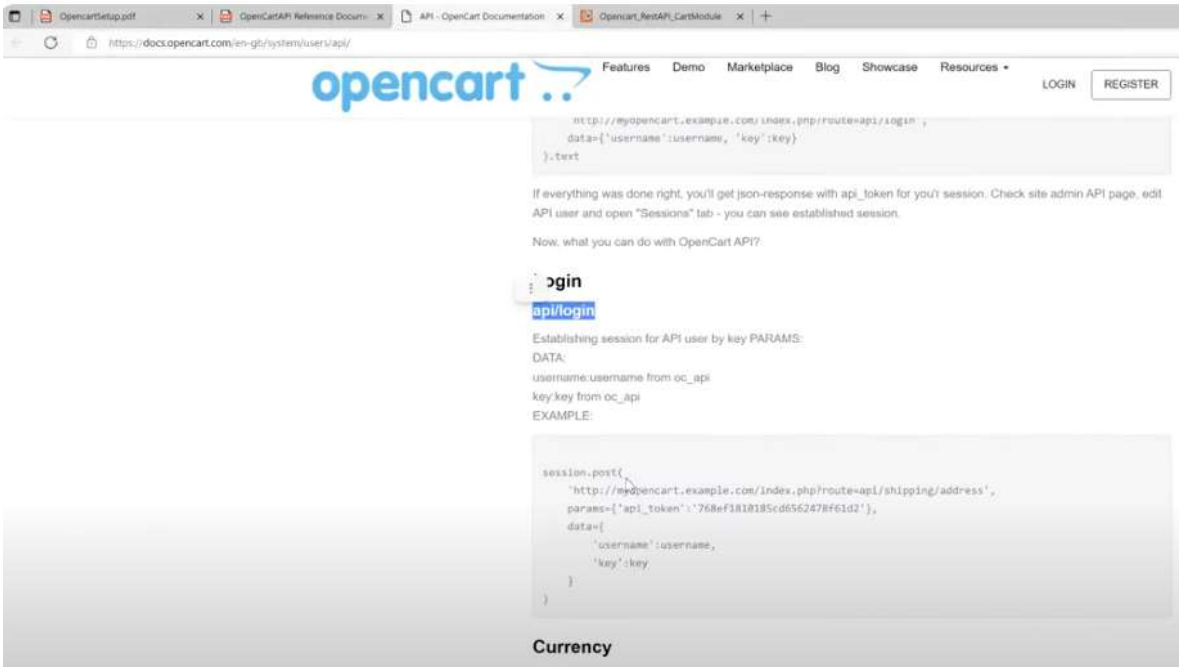
{{baseURL}}api/login

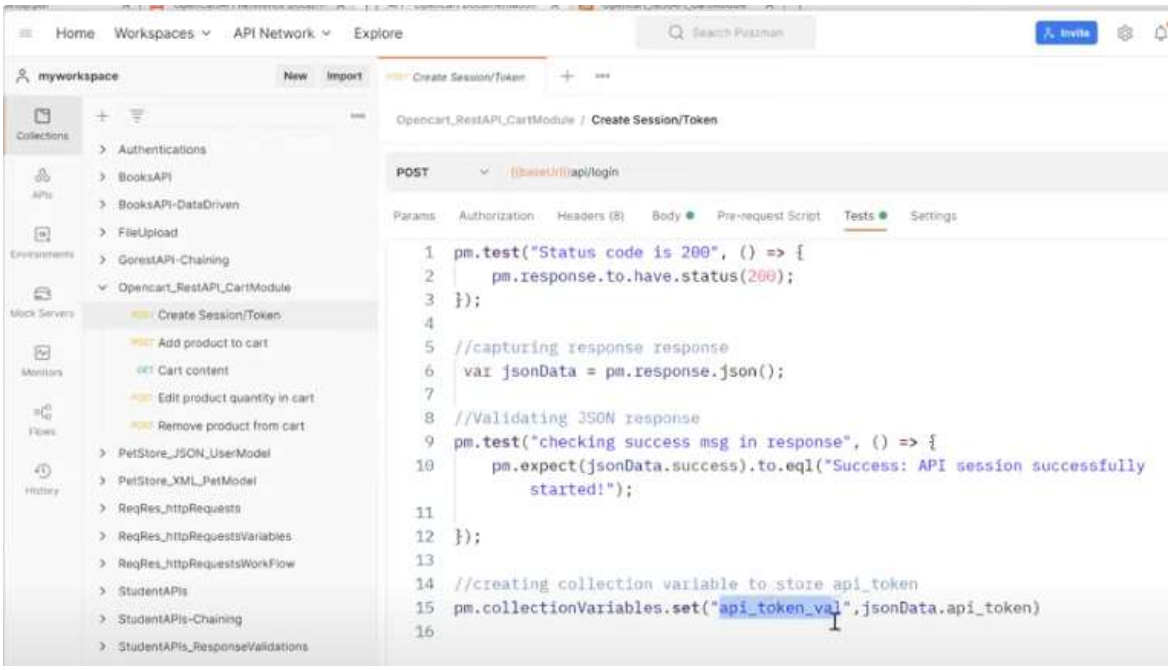
{{baseURL}}api/shipping/address

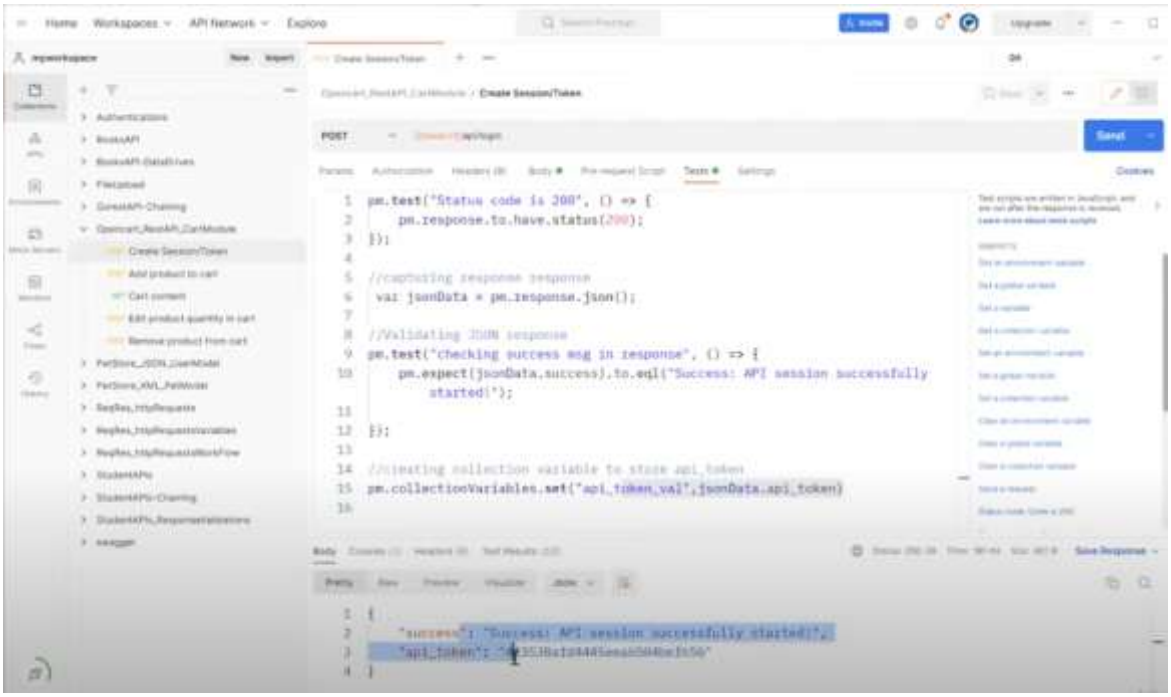
{{baseURL}}api/cart/add

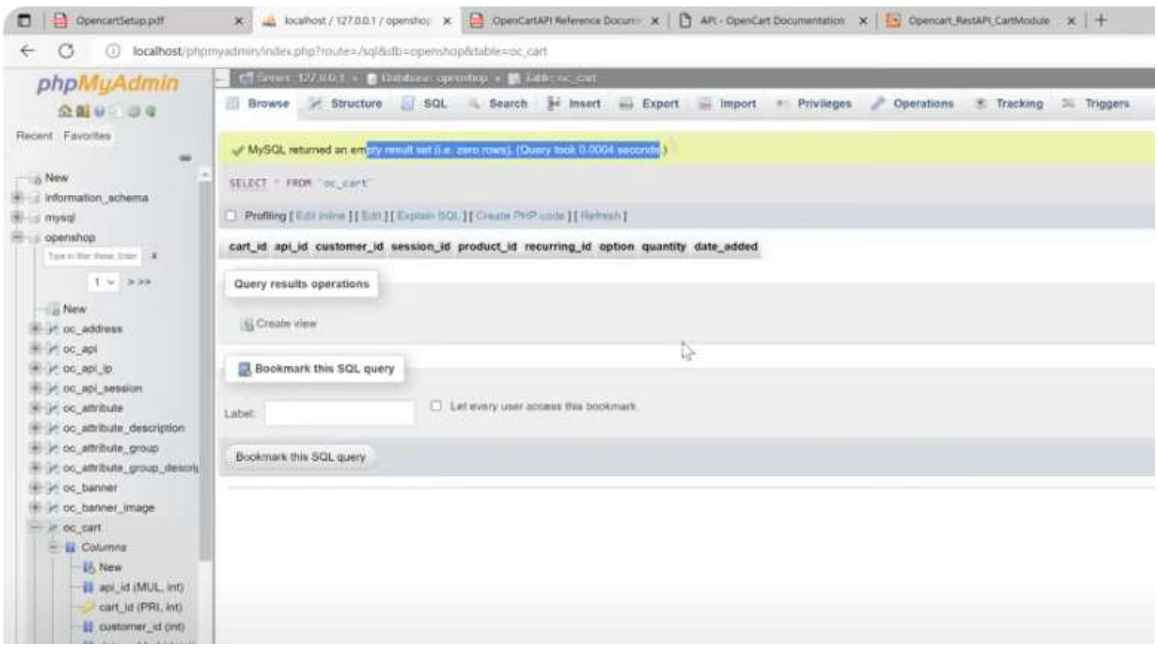


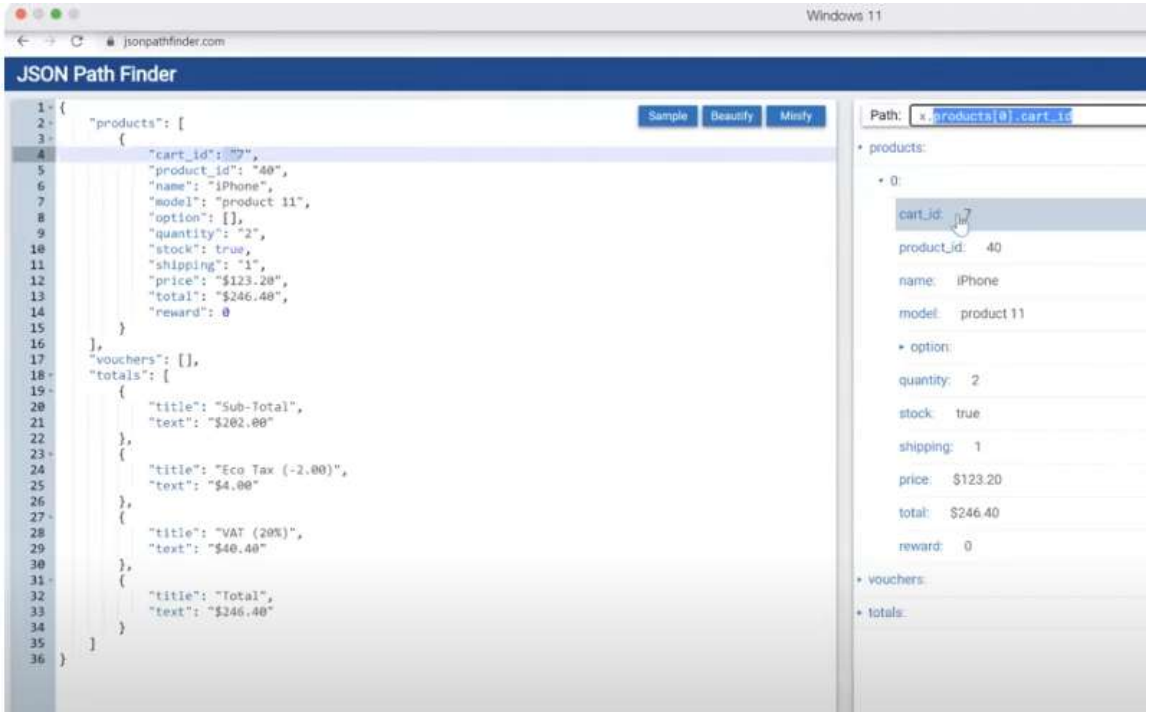

Home
Workspaces
API Network
Explore
myworkspace
New
Import
Create Session/Token
QA
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Authentications
BooksAPI
BooksAPI-DataDriven
FileUpload
GorestAPI-Chaining
Opencart_RestAPI_CartModule
Create Session/Token
Add product to cart
Cart content
Edit product quantity in cart
Remove product from cart
PetStore_JSON_UserModel
PetStore_XML_PetModel
ReqRes_HttpRequests
ReqRes_HttpRequestsVariables
ReqRes_HttpRequestsWorkFlow
StudentAPIs
StudentAPIs-Chaining
StudentAPIs_ResponseValidations
swagger
Opencart_RestAPI_CartModule / Create Session/Token
POST
{{baseUrl}}/api/login
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
KEY
VALUE
DESCRIPTION
username
demo
Get it from opencart admin
key
Get it from opencart admin
Response

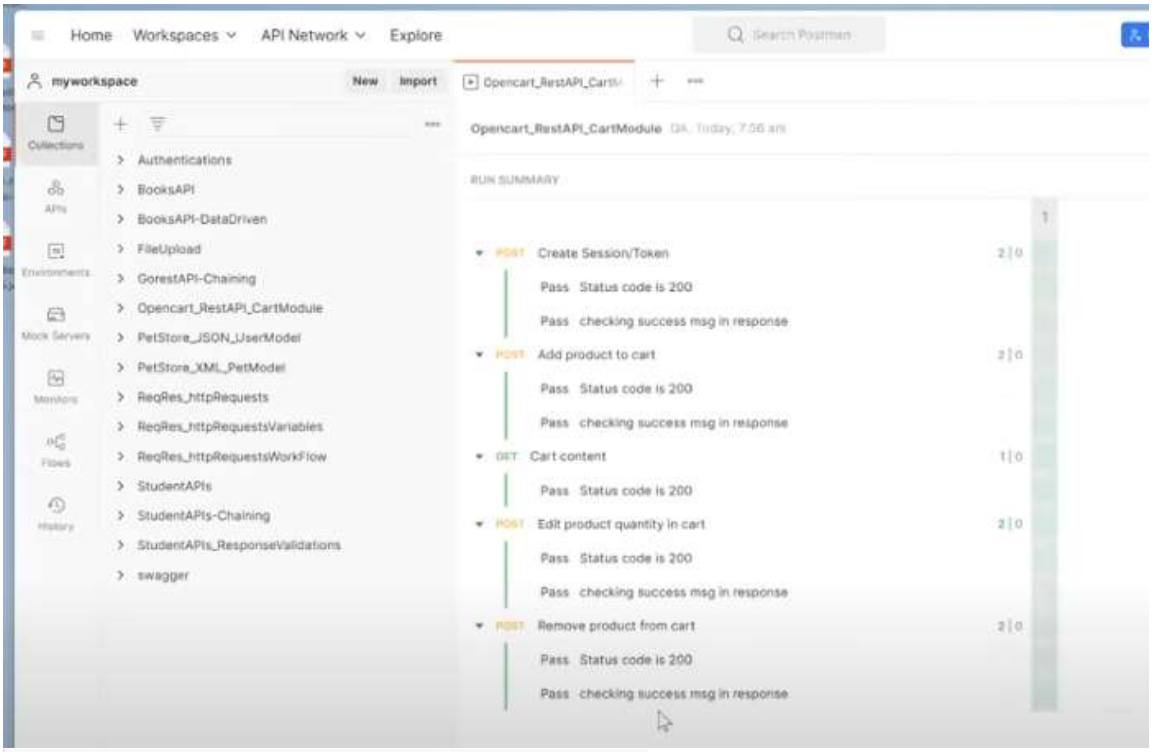

https://docs.opencart.com/en-gb/system/users/api/
opencart
Features
Demo
Marketplace
Blog
Showcase
Resources
LOGIN
REGISTER
data={'username':username, 'key':key}
).text
If everything was done right, you'll get json-response with api_token for you'r session. Check site admin API page, edit API user and open "Sessions" tab - you can see established session.
Now, what you can do with OpenCart API?
ogin
api/login
Establishing session for API user by key PARAMS:
DATA:
username:username from oc_api
key:key from oc_api
EXAMPLE:
session.post(
'http://myopencart.example.com/index.php?route=api/shipping/address',
params={'api_token':'768ef1810185cd656247ff61d2'},
data={
'username':username,
'key':key
}
)
Currency


Home
Workspaces
API Network
Explore
myworkspace
New
Import
Collections
Authentications
BooksAPI
BooksAPI-DataDriven
FileUpload
GorestAPI-Chaining
Opencart_RestAPI_CartModule
Create Session/Token
Add product to cart
Cart content
Edit product quantity in cart
Remove product from cart
PetStore_JSON_UserModel
PetStore_XML_PetModel
ReqRes_httpRequests
ReqRes_httpRequestsVariables
ReqRes_httpRequestsWorkFlow
StudentAPIs
StudentAPIs-Chaining
StudentAPIs_ResponseValidations
Opencart_RestAPI_CartModule / Create Session/Token
POST
{{baseUrl}}api/login
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
pm.test("Status code is 200", () => {
pm.response.to.have.status(200);
});
//capturing response response
var jsonData = pm.response.json();
//Validating JSON response
pm.test("checking success msg in response", () => {
pm.expect(jsonData.success).to.eql("Success: API session successfully started!");
});
//creating collection variable to store api_token
pm.collectionVariables.set("api_token_val",jsonData.api_token)


pm.test("Status code is 200", () => {
pm.response.to.have.status(200);
});
//capturing response response
var jsonData = pm.response.json();
//Validating JSON response
pm.test("checking success msg in response", () => {
pm.expect(jsonData.success).to.eql("Success: API session successfully started!");
});
//creating collection variable to store api_token
pm.collectionVariables.set("api_token_val",jsonData.api_token)
{
"success": "Success: API session successfully started!",
"api_token": "


localhost/phpmyadmin/index.php?route=/sql&db=openshop&table=oc_cart
phpMyAdmin
Recent
Favorites
New
information_schema
mysql
openshop
oc_address
oc_api
oc_api_ip
oc_api_session
oc_attribute
oc_attribute_description
oc_attribute_group
oc_banner
oc_banner_image
oc_cart
Columns
api_id (MUL, int)
cart_id (PRI, int)
customer_id (int)
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)
SELECT * FROM `oc_cart`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
cart_id api_id customer_id session_id product_id recurring_id option quantity date_added
Query results operations
Create view
Bookmark this SQL query
Label:
Let every user access this bookmark
Bookmark this SQL query





How many ways we can run collection

Pre-requisite: nodejs + npm

newman

newman-reporter-html

How to run postman collections

Pre-requisites

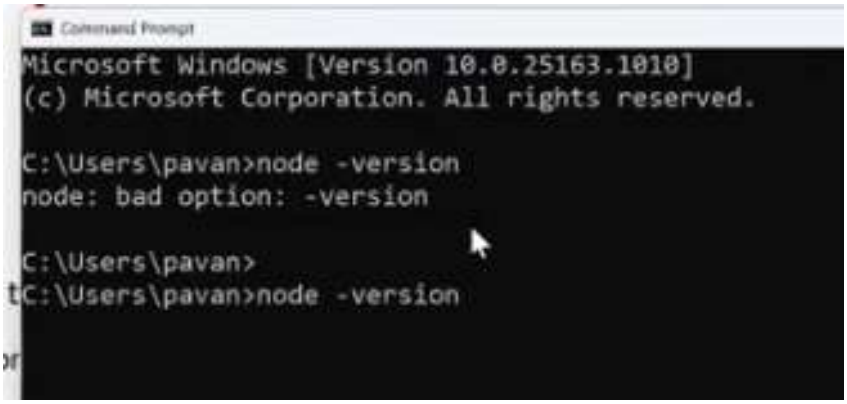
1. Install Node.js (npm)
2. Install newman
3. Export collection

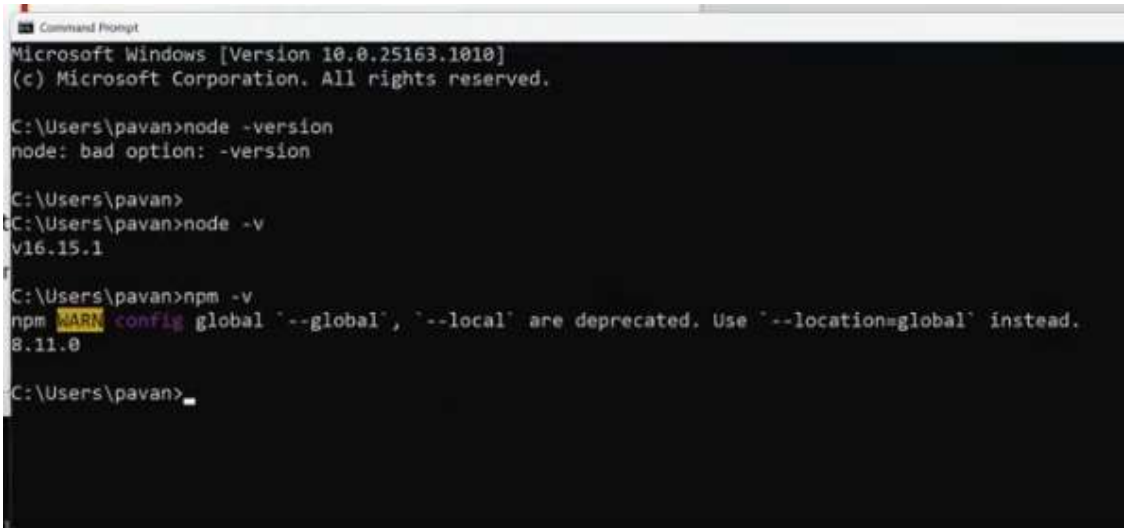
Step1: Install node.js (Refer the NodeJS installation document)

After installation, open command prompt and check if **node** and **npm** are installed

node -v

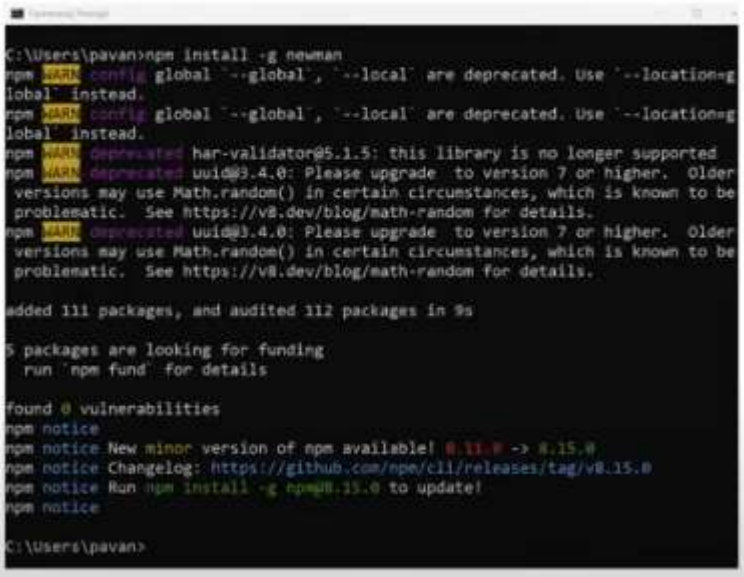
npm -v





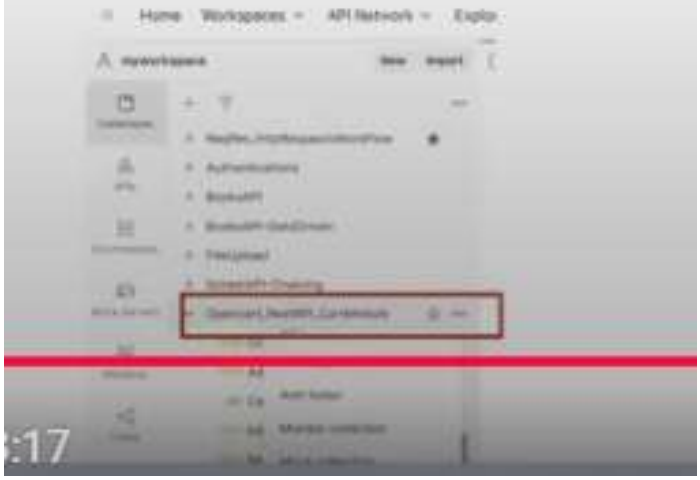
Step2: Install Newman

npm install -g newman



Step3: Export collection

Follow below steps to export collection.



How many ways we can run collection

Pre-requisite: nodejs + npm

2 components are required to run collection in command prompt and generate report

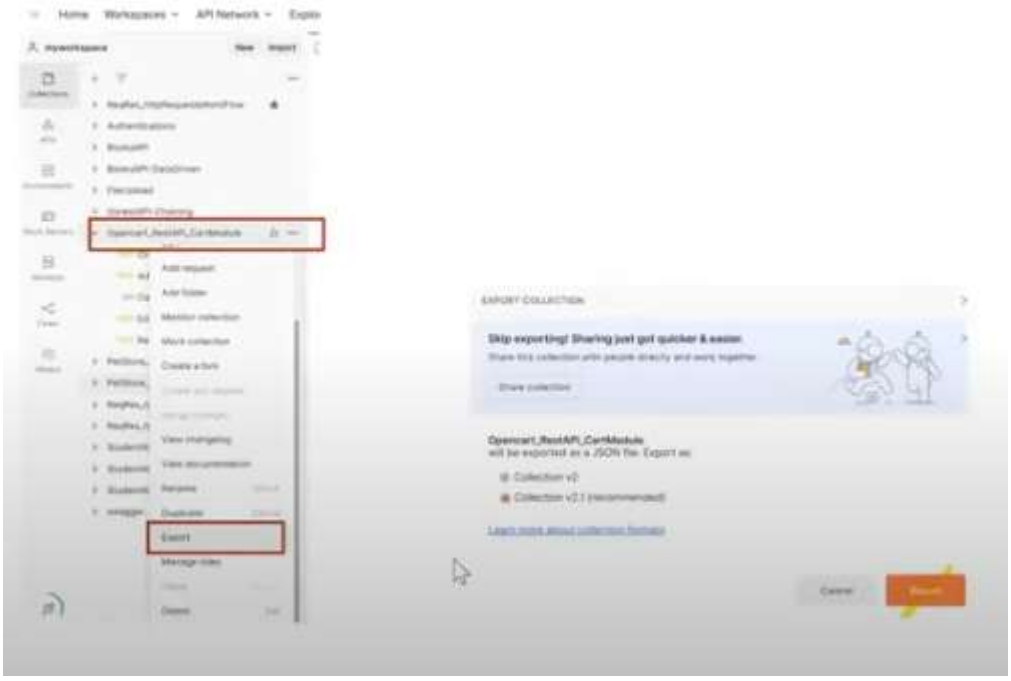
---

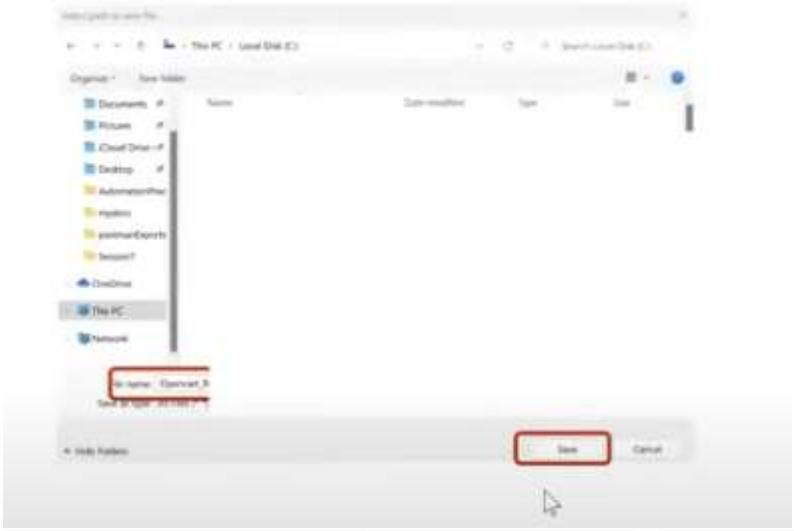
1) newman
2) newman-reporter-html

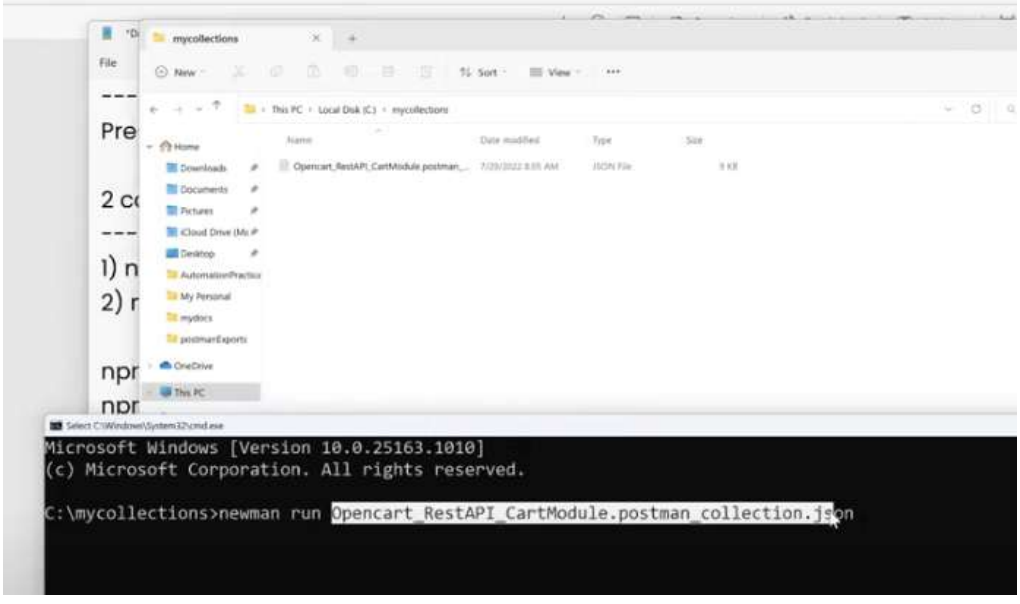
npm install -g newman
npm install -g newman-reporter-html

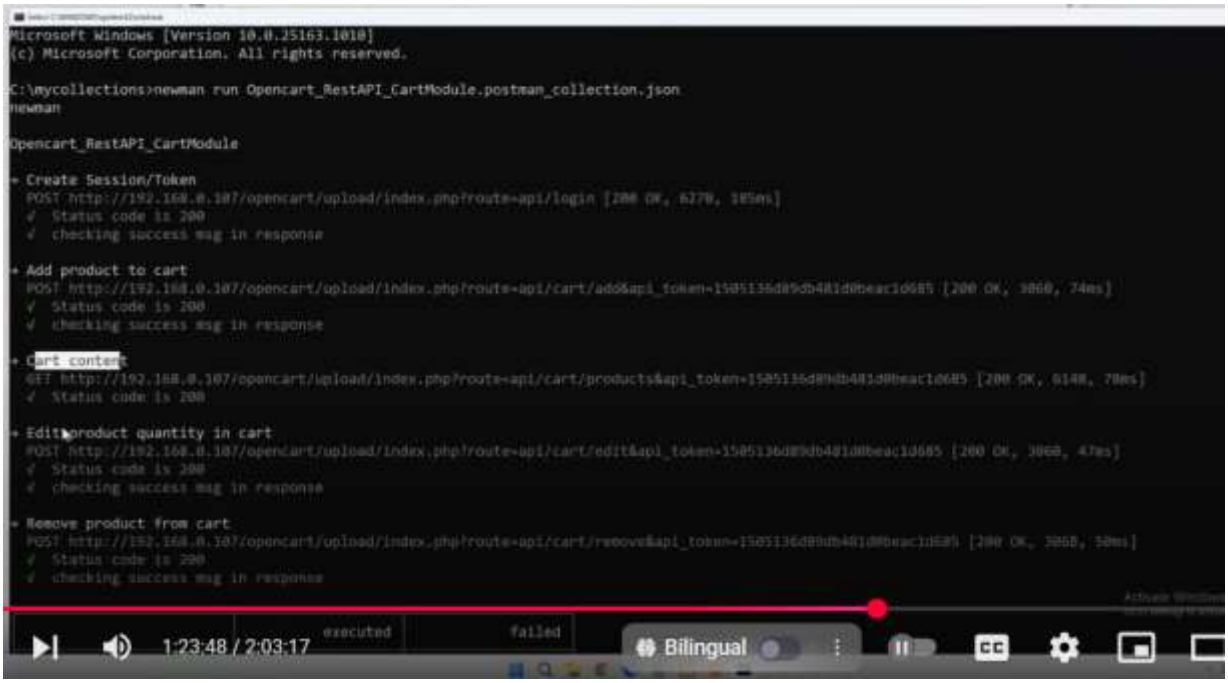
Step3: Export collection

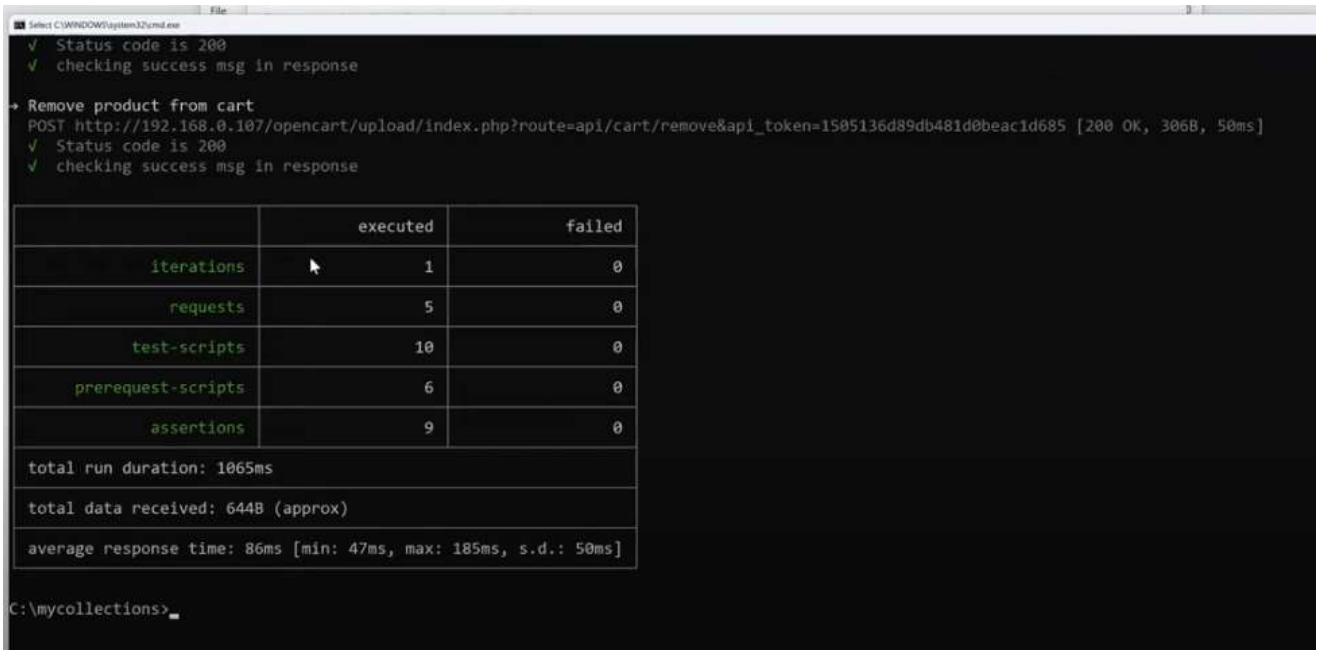
Follow below steps to export collection.

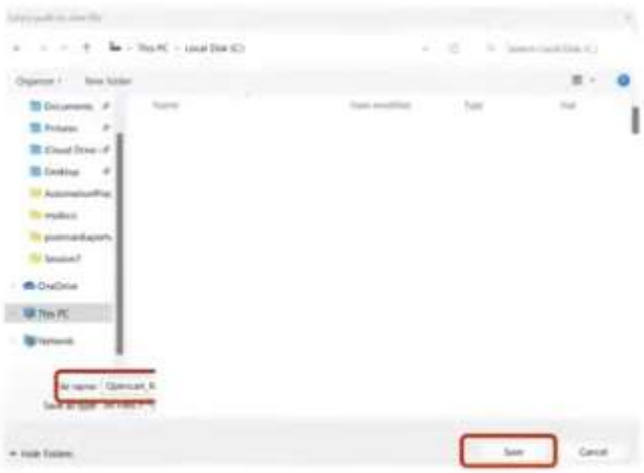




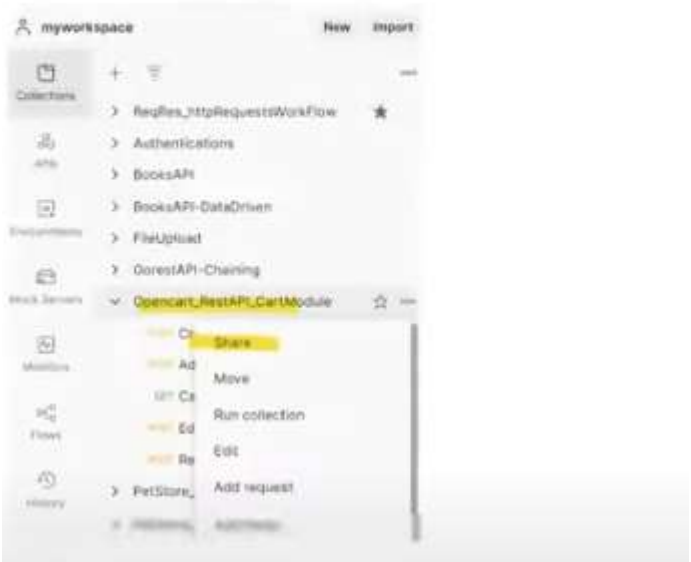


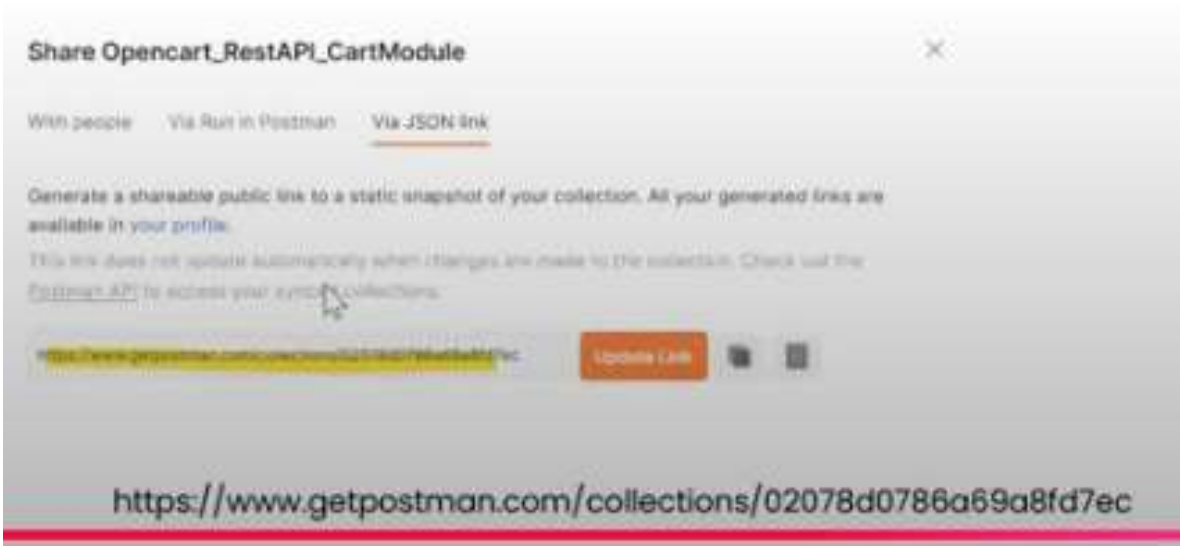


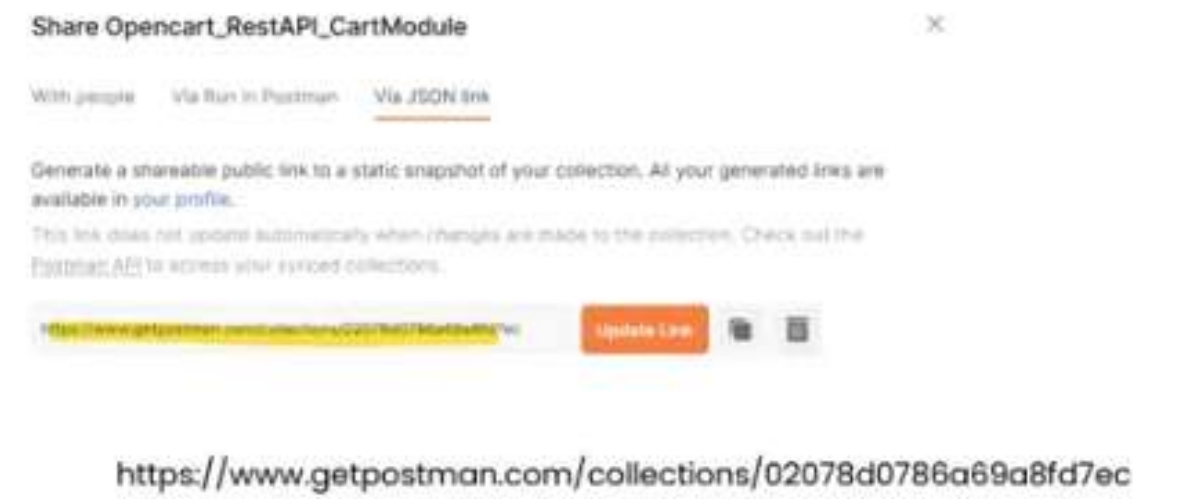




Share Collection Link (For remote execution)







Multiple ways to Run Collection

Method 1 : Run collection locally

newman run Opencart_RestAPI_CartModule.postman_collection.json

Method 2 : Run shared collection remotely

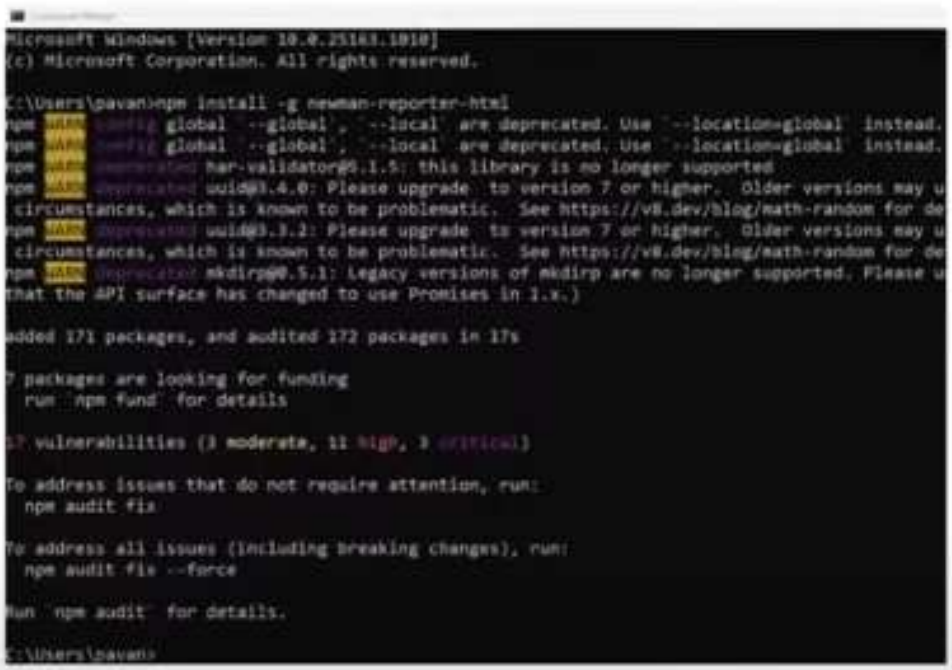
newman run https://www.getpostman.com/collections/02078d0786a69a8fd7ec

Method 3 : Run collections & generate html report

Note: We need to install **newman-reporter-html** to generate html report.

Open command prompt then run below command

npm install -g newman-reporter-html



newman run Opencart_RestAPI_CartModule.postman_collection.json **-r html**

newman run https://www.getpostman.com/collections/02078d0786a69a8fd7ec **-r html**

{{baseURL}}api/login
{{baseURL}}api/shipping/address
{{baseURL}}api/cart/add

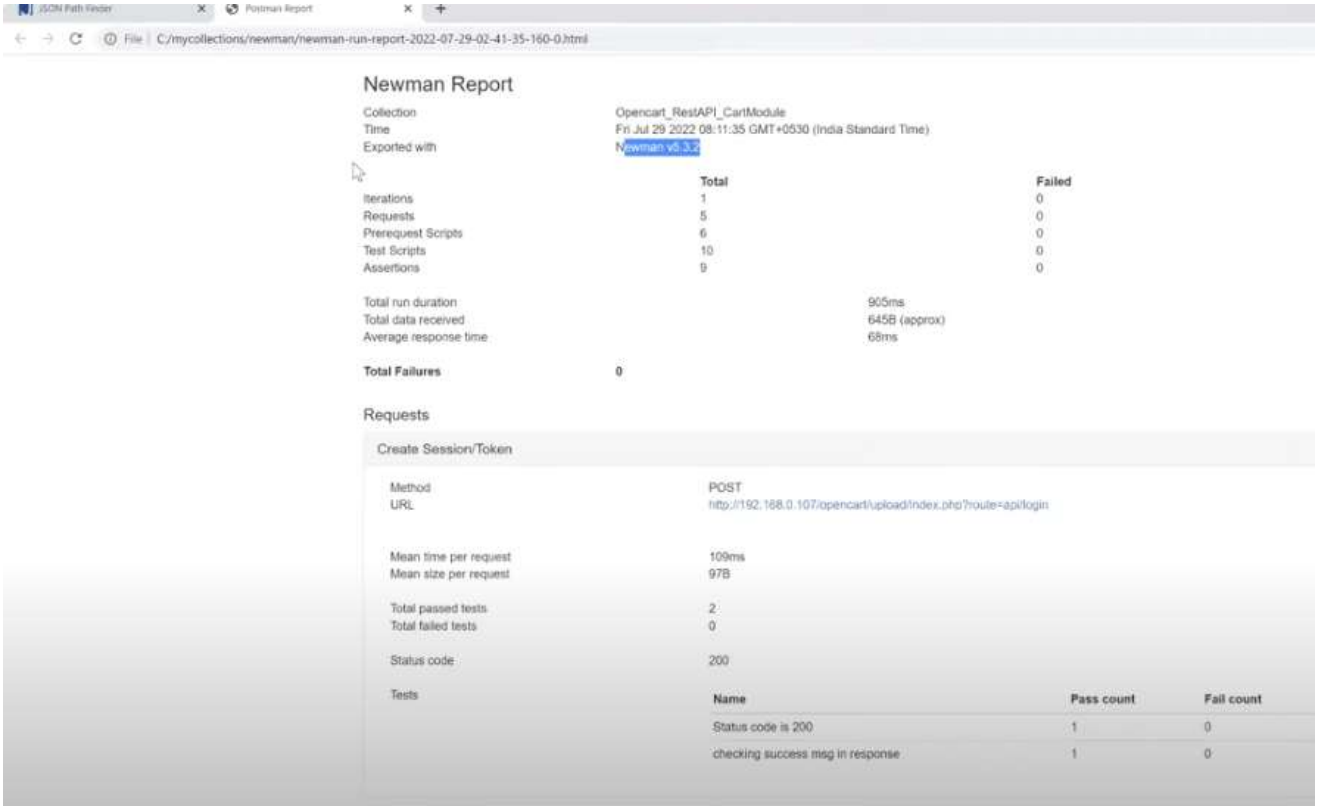
How many ways we can run collection

Pre-requisite: nodejs + npm

2 components are required to run collection in command prompt and generate report
---
1) newman
2) newman-reporter-html

npm install -g newman
npm install -g newman-reporter-html

Appraoch 1) run collection in command prompt
---
newman run Opencart_RestAPI_CartModule.postman_collection.json
newman run Opencart_RestAPI_CartModule.postman_collection.json



{{baseURL}}api/login
{{baseURL}}api/shipping/address
{{baseURL}}api/cart/add

How many ways we can run collection

Pre-requisite: nodejs + npm

2 components are required to run collection in command prompt and generate report

---

1) newman
2) newman-reporter-html

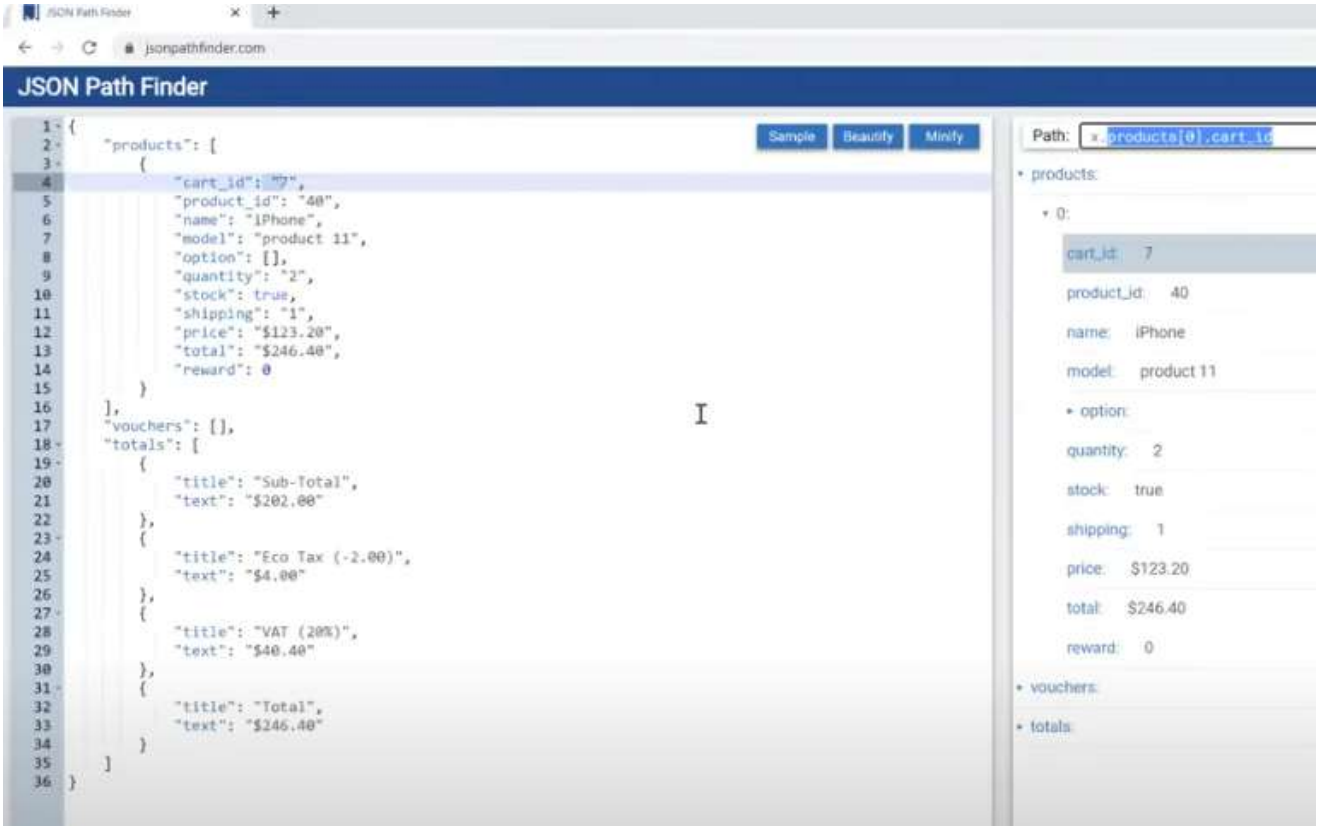
commands to install above components..

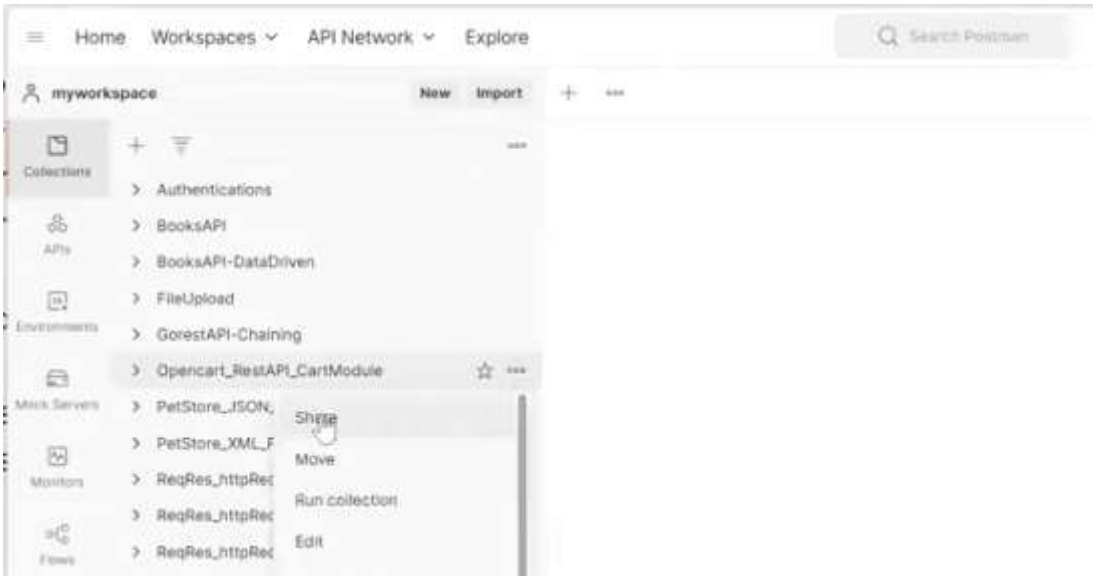
npm install -g newman
npm install -g newman-reporter-html

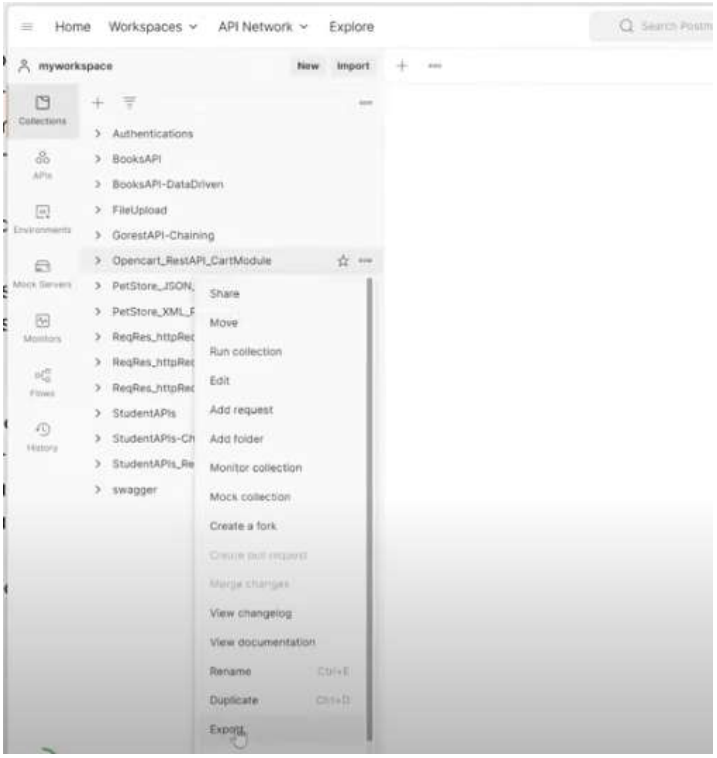
Appraoch 1) run collection in command prompt

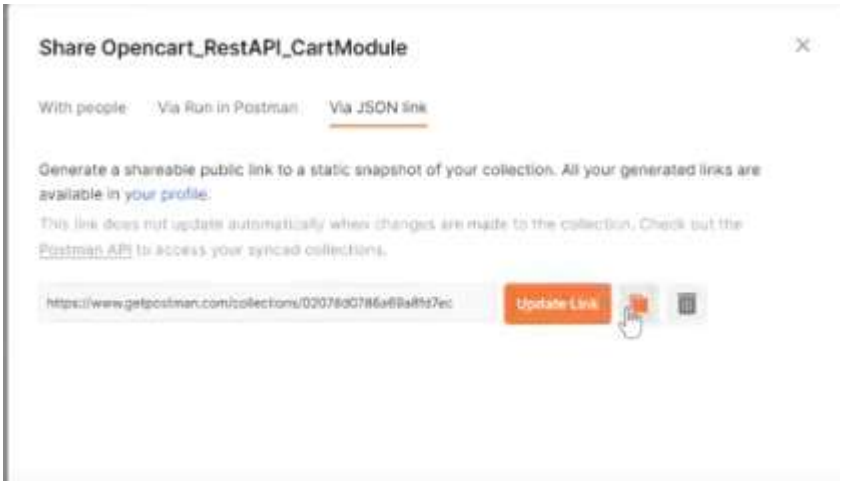
---

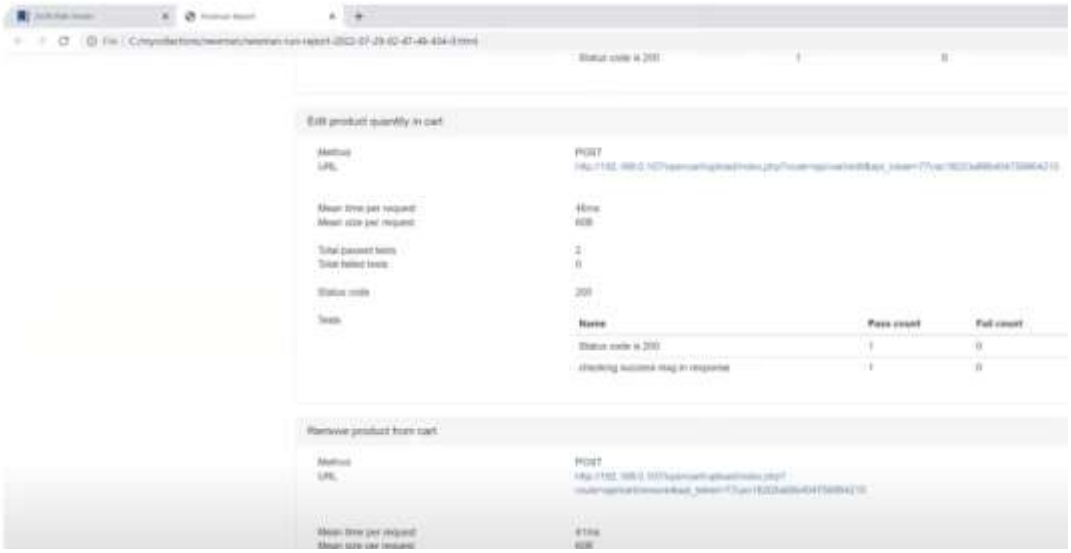
newman run Opencart_RestAPI_CartModule.postman_collection.json
newman run Opencart_RestAPI_CartModule.postman_collection.json -r html


JSON Path Finder
jsonpathfinder.com
JSON Path Finder
Sample
Beautify
Minify
Path: $.products[0].cart_id
products:
0:
cart_id: 7
product_id: 40
name: iPhone
model: product 11
option:
quantity: 2
stock: true
shipping: 1
price: $123.20
total: $246.40
reward: 0
vouchers:
totals:


Home
Workspaces
API Network
Explore
myworkspace
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
Flows
Authentications
BooksAPI
BooksAPI-DataDriven
FileUpload
GorestAPI-Chaining
Opencart_RestAPI_CartModule
Share
Move
Run collection
Edit


Home
Workspaces
API Network
Explore
myworkspace
New
Import
Collections
Authentications
BooksAPI
BooksAPI-DataDriven
FileUpload
GorestAPI-Chaining
Opencart_RestAPI_CartModule
Share
Move
Run collection
Edit
Add request
Add folder
Monitor collection
Mock collection
Create a fork
View changelog
View documentation
Rename
Duplicate
Export





2 components are required to run collection in command prompt and generate report

---

1) newman
2) newman-reporter-html

commands to install above components..

npm install -g newman
npm install -g newman-reporter-html

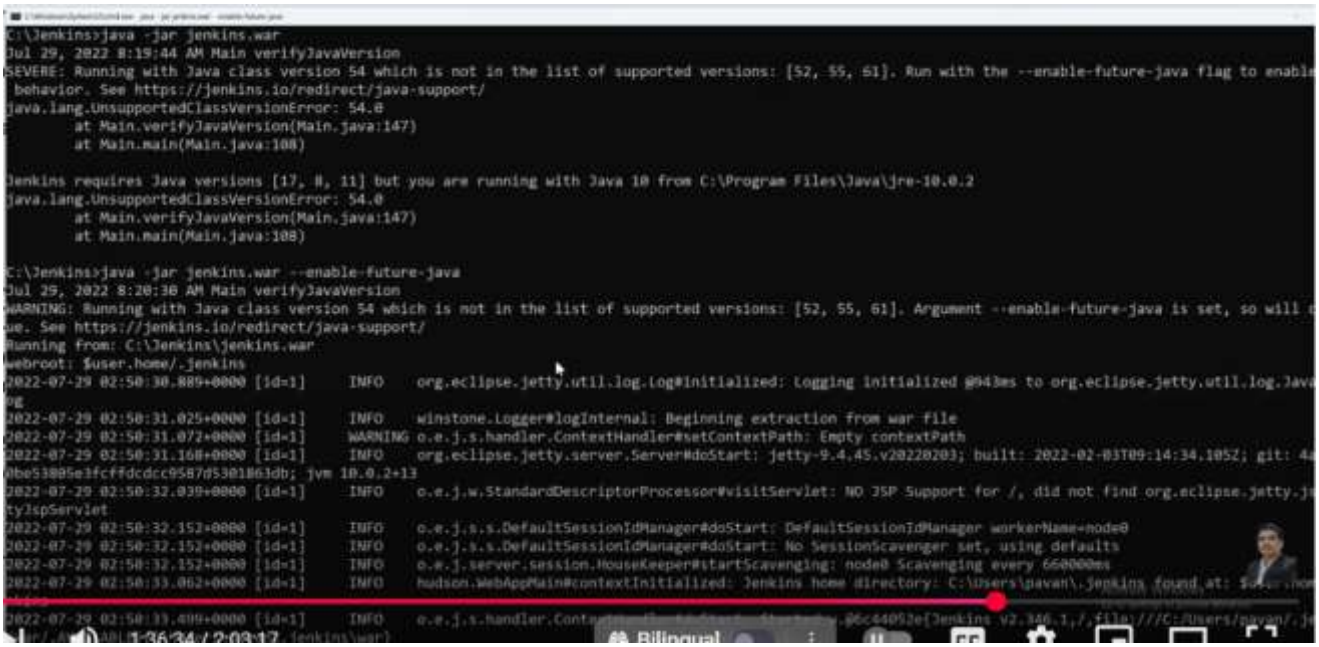
Appraoch 1) run collection in command prompt (local)

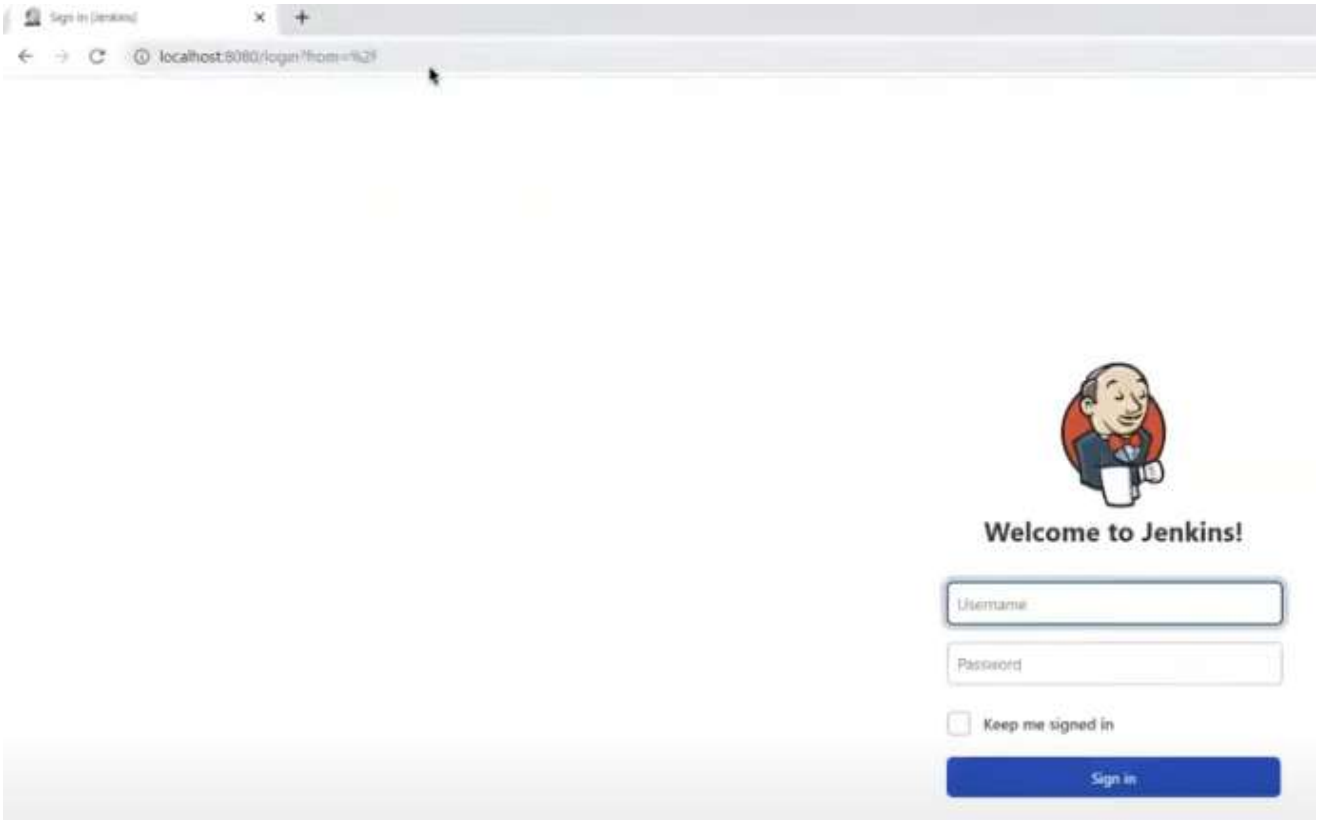
---

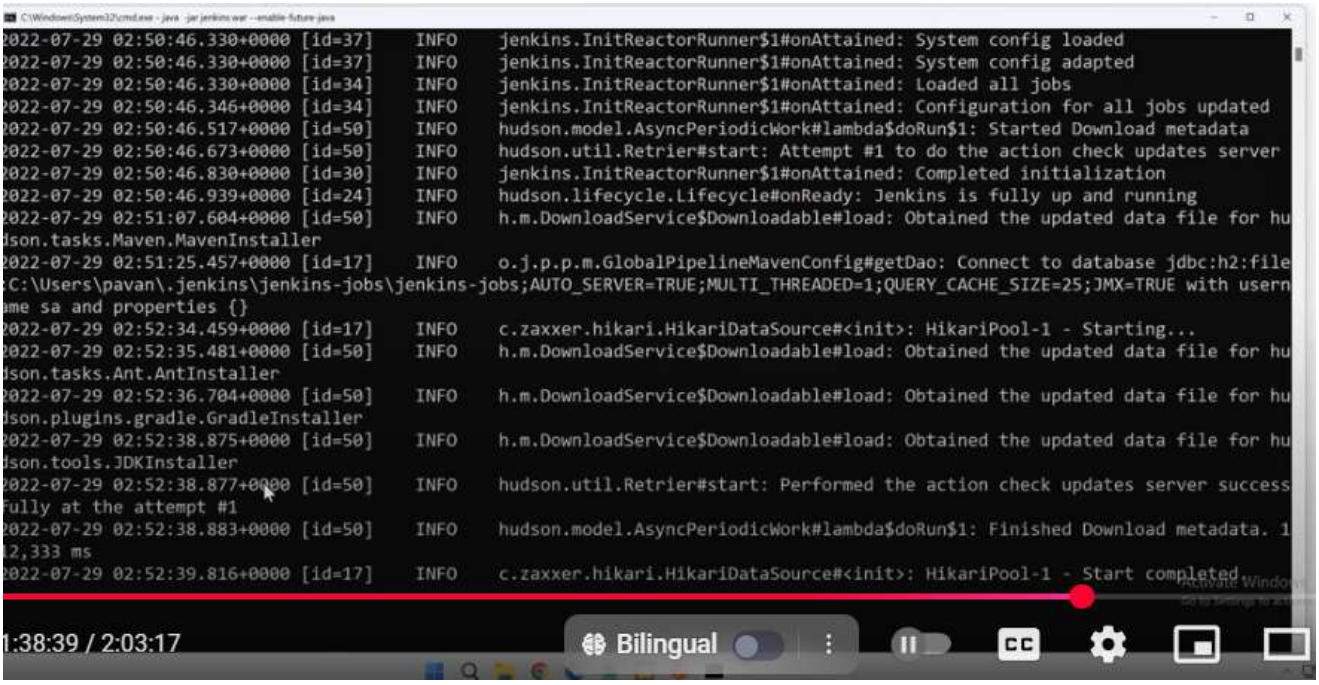
newman run Opencart_RestAPI_CartModule.postman_collection.json
newman run Opencart_RestAPI_CartModule.postman_collection.json -r html

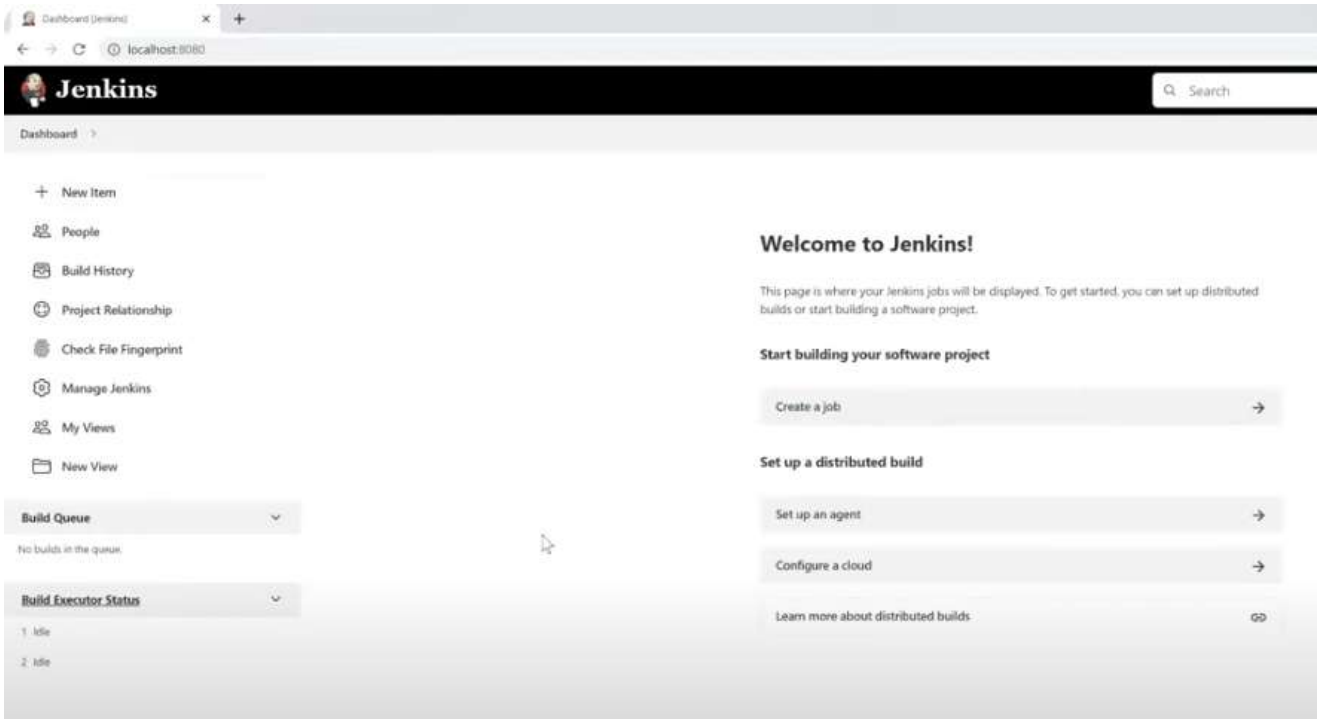
Appraoch 2) run the collection remotely using url.(remote)
https://www.getpostman.com/collections/02078d0786a69a8fd7ec

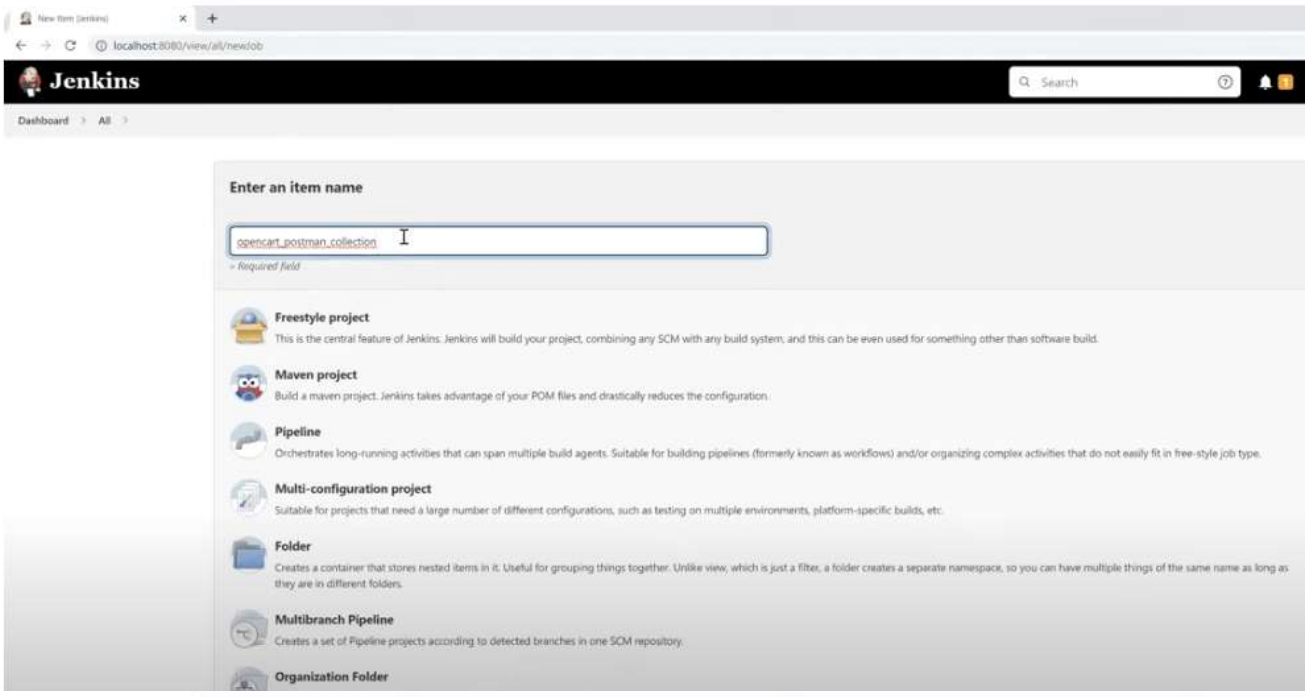
How to run collection in jenkis

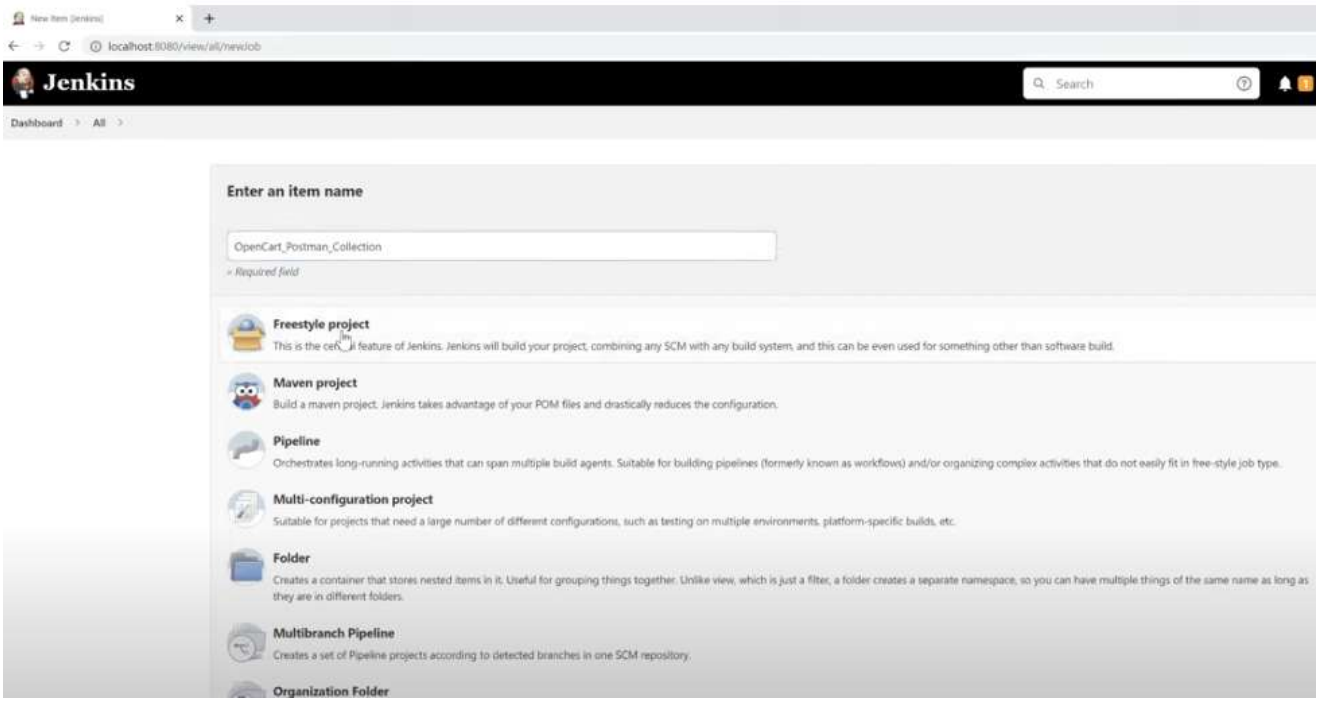

C:\Jenkins>java -jar jenkins.war
Jul 29, 2022 8:19:44 AM Main verifyJavaVersion
SEVERE: Running with Java class version 54 which is not in the list of supported versions: [52, 55, 61]. Run with the --enable-future-java flag to enable
behavior. See https://jenkins.io/redirect/java-support/
java.lang.UnsupportedClassVersionError: 54.0
at Main.verifyJavaVersion(Main.java:147)
at Main.main(Main.java:108)
Jenkins requires Java versions [17, 8, 11] but you are running with Java 10 from C:\Program Files\Java\jre-10.0.2
java.lang.UnsupportedClassVersionError: 54.0
at Main.verifyJavaVersion(Main.java:147)
at Main.main(Main.java:108)
C:\Jenkins>java -jar jenkins.war --enable-future-java
Jul 29, 2022 8:20:38 AM Main verifyJavaVersion
WARNING: Running with Java class version 54 which is not in the list of supported versions: [52, 55, 61]. Argument --enable-future-java is set, so will
ue. See https://jenkins.io/redirect/java-support/
Running from: C:\Jenkins\jenkins.war
webroot: $user.home/.jenkins
2022-07-29 02:50:30.889+0000 [id=1] INFO org.eclipse.jetty.util.log.Log#initialized: Logging initialized @943ms to org.eclipse.jetty.util.log.Jav
og
2022-07-29 02:50:31.025+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war file
2022-07-29 02:50:31.072+0000 [id=1] WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
2022-07-29 02:50:31.168+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-9.4.45.v20220203; built: 2022-02-03T09:14:34.105Z; git: 4
0be53805e3fcffdcdcc9587d5301863db; jvm 10.0.2+13
2022-07-29 02:50:32.039+0000 [id=1] INFO o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did not find org.eclipse.jetty.j
tylspServlet
2022-07-29 02:50:32.152+0000 [id=1] INFO o.e.j.s.s.DefaultSessionIdManager#doStart: DefaultSessionIdManager workerName=node0
2022-07-29 02:50:32.152+0000 [id=1] INFO o.e.j.s.s.DefaultSessionIdManager#doStart: No SessionScavenger set, using defaults
2022-07-29 02:50:32.152+0000 [id=1] INFO o.e.j.server.session.HouseKeeper#startScavenging: node0 Scavenging every 660000ms
2022-07-29 02:50:33.062+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory: C:\Users\pavan\.jenkins found at:
2022-07-29 02:50:33.499+0000 [id=1] INFO o.e.j.s.handler.ContextHandler.
1:36:34 / 2:03:17
Bilingual

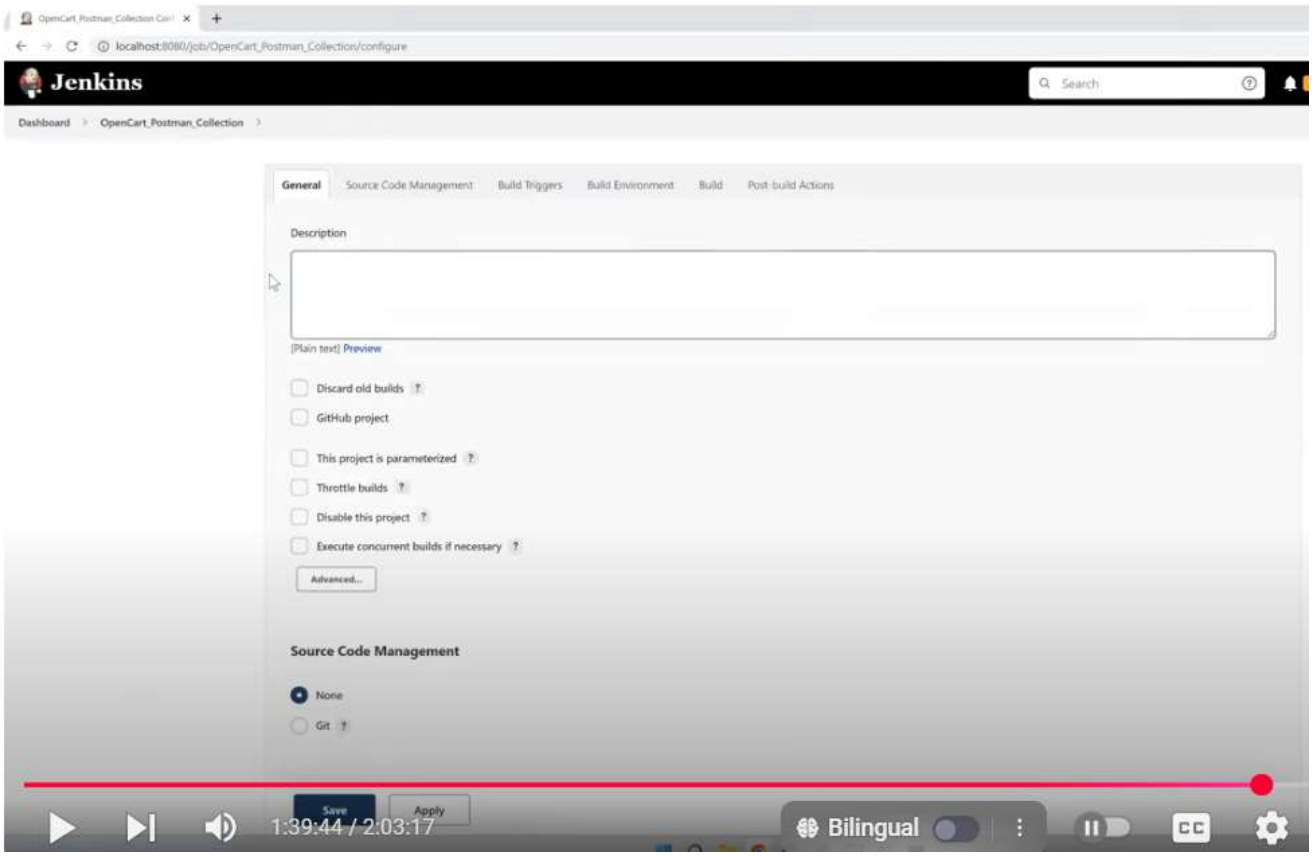

localhost:8080/login?from=%2F
Welcome to Jenkins!
Username
Password
Keep me signed in
Sign in

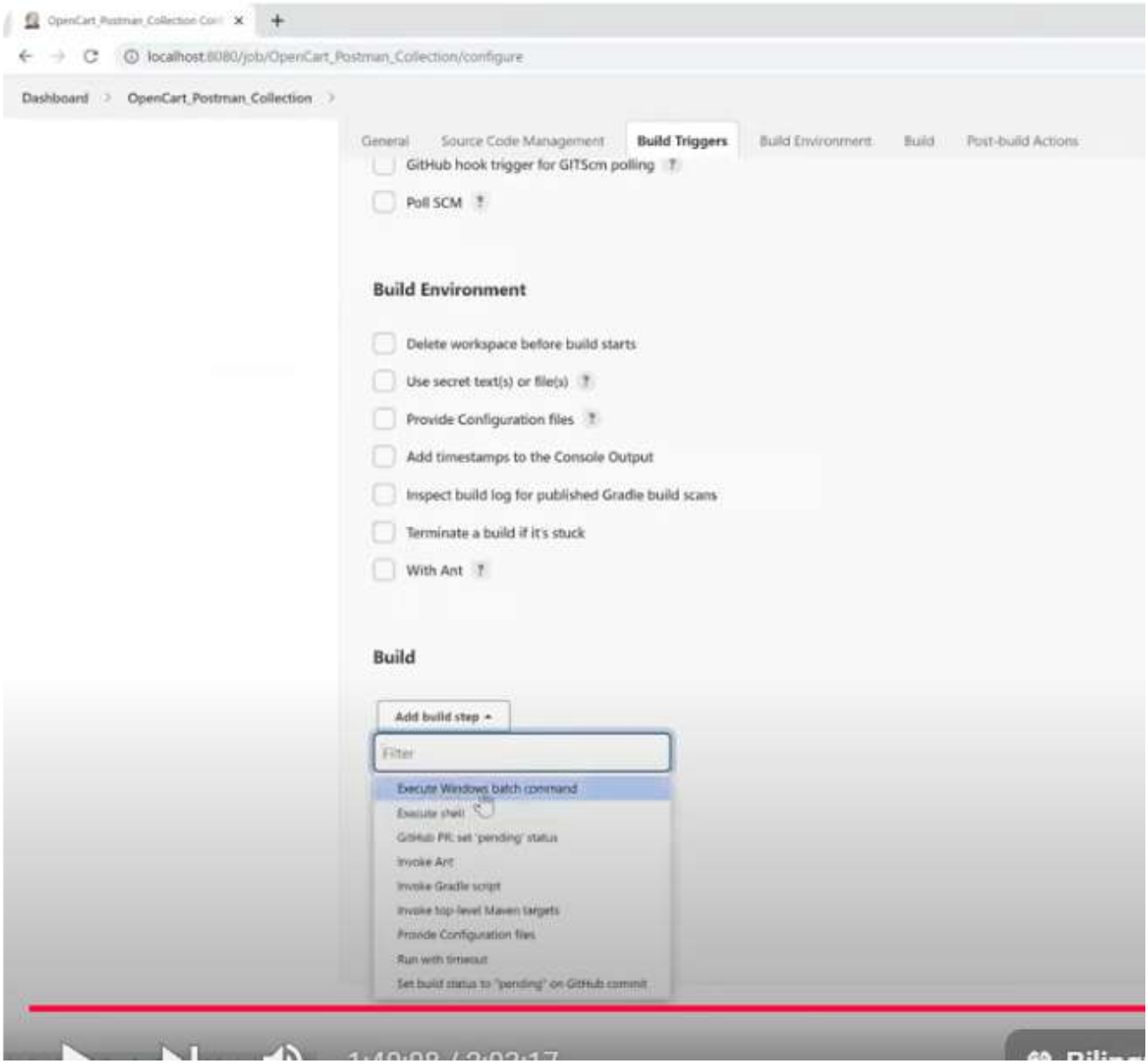

2022-07-29 02:50:46.330+0000 [id=37] INFO jenkins.InitReactorRunner$1#onAttained: System config loaded
2022-07-29 02:50:46.330+0000 [id=37] INFO jenkins.InitReactorRunner$1#onAttained: System config adapted
2022-07-29 02:50:46.330+0000 [id=34] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2022-07-29 02:50:46.346+0000 [id=34] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2022-07-29 02:50:46.517+0000 [id=50] INFO hudson.model.AsyncPeriodicWork#lambda$doRun$1: Started Download metadata
2022-07-29 02:50:46.673+0000 [id=50] INFO hudson.util.Retrier#start: Attempt #1 to do the action check updates server
2022-07-29 02:50:46.830+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2022-07-29 02:50:46.939+0000 [id=24] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
2022-07-29 02:51:07.604+0000 [id=50] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hu
dson.tasks.Maven.MavenInstaller
2022-07-29 02:51:25.457+0000 [id=17] INFO o.j.p.p.m.GlobalPipelineMavenConfig#getDao: Connect to database jdbc:h2:file
:C:\Users\pavan\.jenkins\jenkins-jobs\jenkins-jobs;AUTO_SERVER=TRUE;MULTI_THREADED=1;QUERY_CACHE_SIZE=25;JMX=TRUE with usern
me sa and properties {}
2022-07-29 02:52:34.459+0000 [id=17] INFO c.zaxxer.hikari.HikariDataSource#<init>: HikariPool-1 - Starting...
2022-07-29 02:52:35.481+0000 [id=50] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hu
dson.tasks.Ant.AntInstaller
2022-07-29 02:52:36.704+0000 [id=50] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hu
dson.plugins.gradle.GradleInstaller
2022-07-29 02:52:38.875+0000 [id=50] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hu
dson.tools.JDKInstaller
2022-07-29 02:52:38.877+0000 [id=50] INFO hudson.util.Retrier#start: Performed the action check updates server success
fully at the attempt #1
2022-07-29 02:52:38.883+0000 [id=50] INFO hudson.model.AsyncPeriodicWork#lambda$doRun$1: Finished Download metadata. 1
12,333 ms
2022-07-29 02:52:39.816+0000 [id=17] INFO c.zaxxer.hikari.HikariDataSource#<init>: HikariPool-1 - Start completed.
1:38:39 / 2:03:17
Bilingual

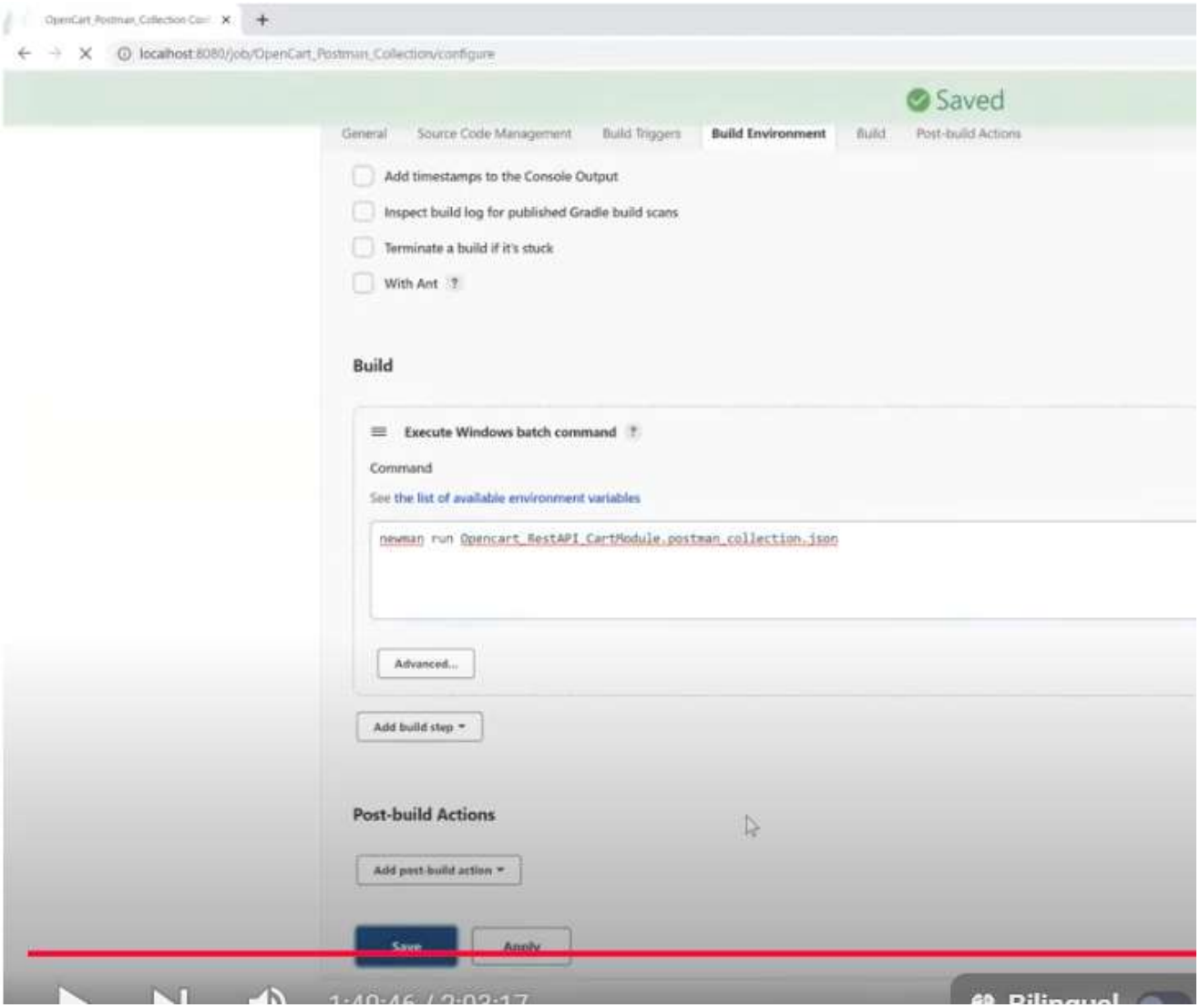

Dashboard [Jenkins]
localhost:8080
Jenkins
Search
Dashboard
New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
New View
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Welcome to Jenkins!
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.
Start building your software project
Create a job
Set up a distributed build
Set up an agent
Configure a cloud
Learn more about distributed builds

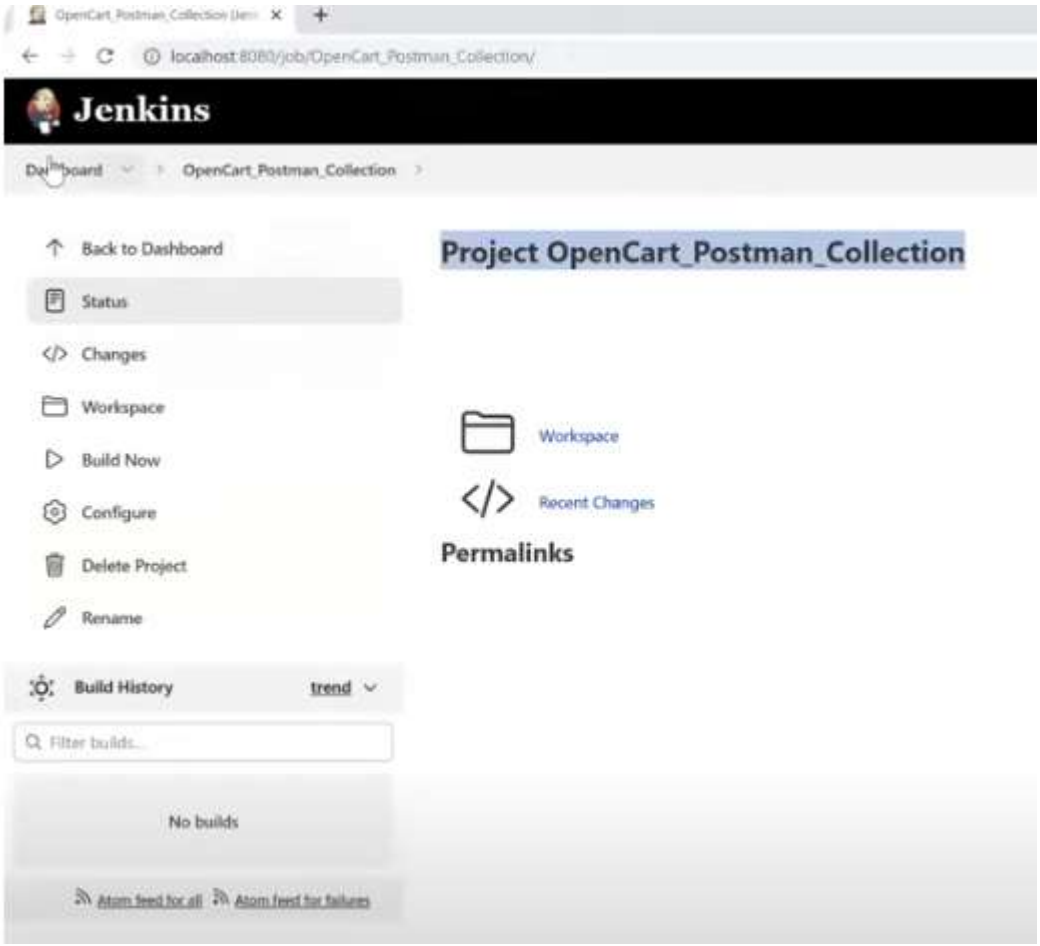

New Item [Jenkins]
localhost:8080/view/all/newJob
Jenkins
Search
Dashboard > All >
Enter an item name
opencart_postman_collection
» Required field
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.
Organization Folder

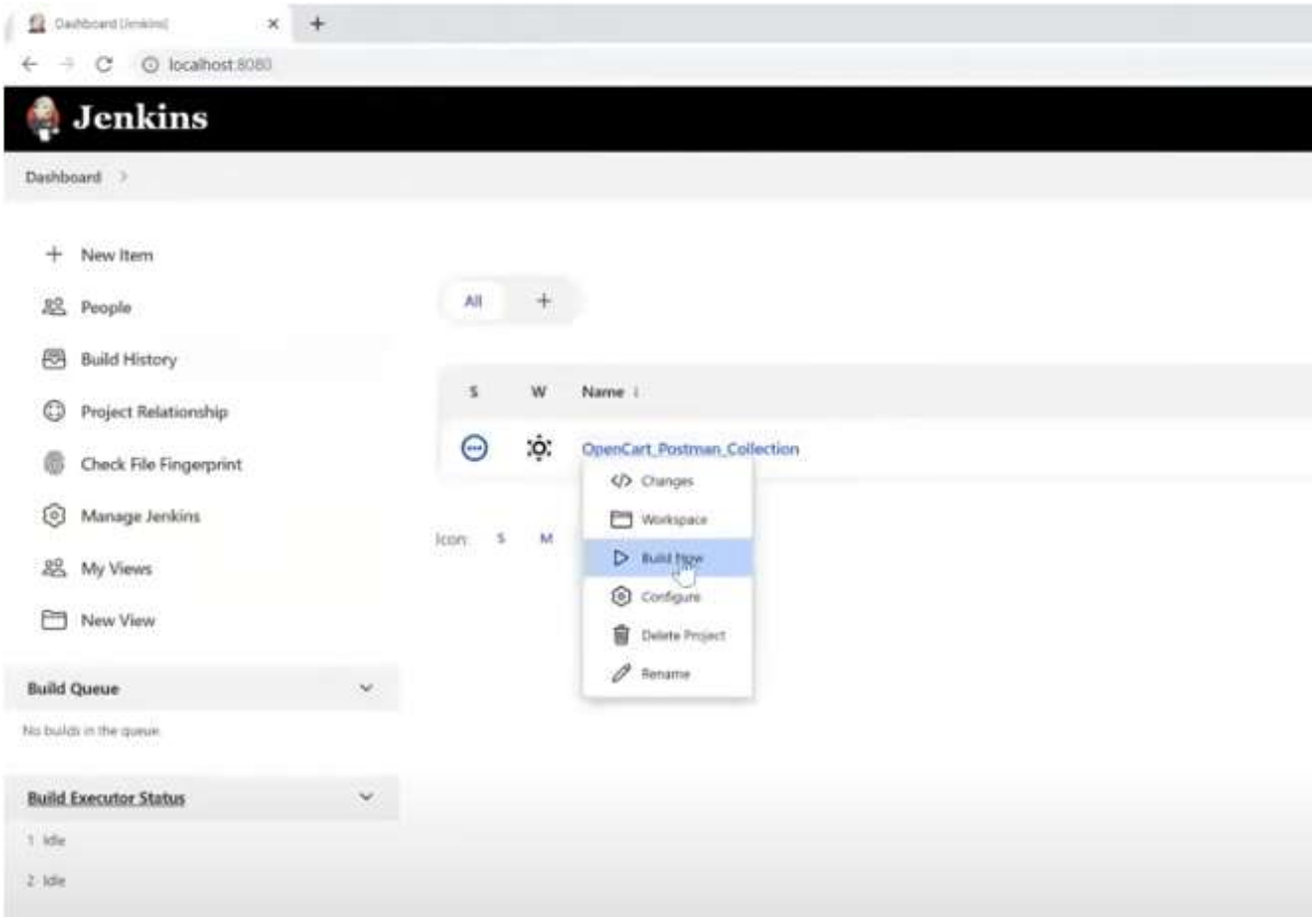

New Item [Jenkins]
localhost:8080/view/all/newJob
Jenkins
Search
Dashboard > All >
Enter an item name
OpenCart_Postman_Collection
» Required field
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.
Organization Folder

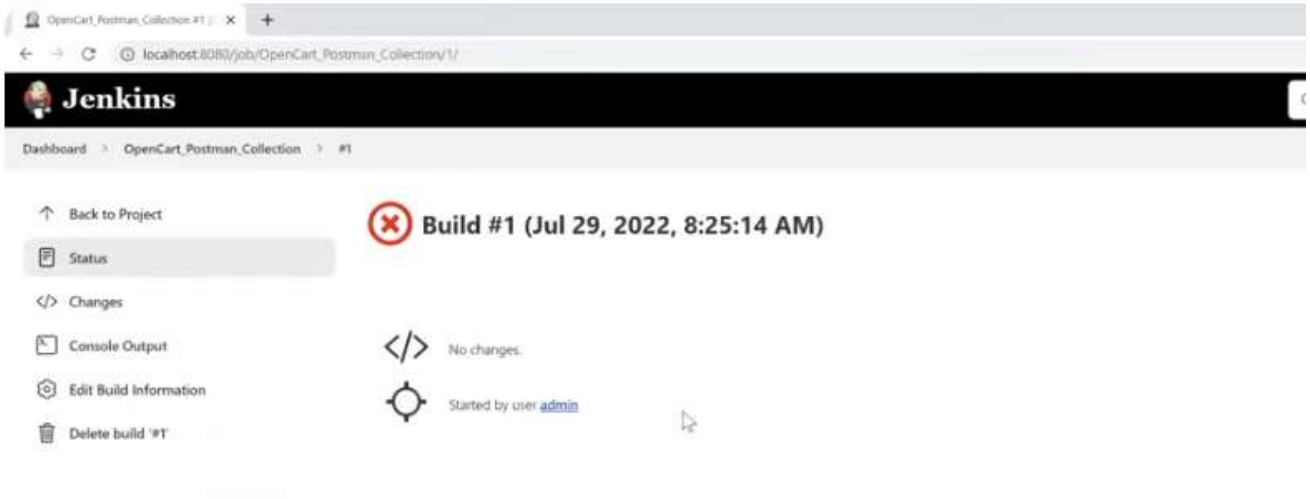

Jenkins
Search
Dashboard
OpenCart_Postman_Collection
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
Description
[Plain text] Preview
Discard old builds
GitHub project
This project is parameterized
Throttle builds
Disable this project
Execute concurrent builds if necessary
Advanced...
Source Code Management
None
Git
Save
Apply
1:39:44 / 2:03:17
Bilingual

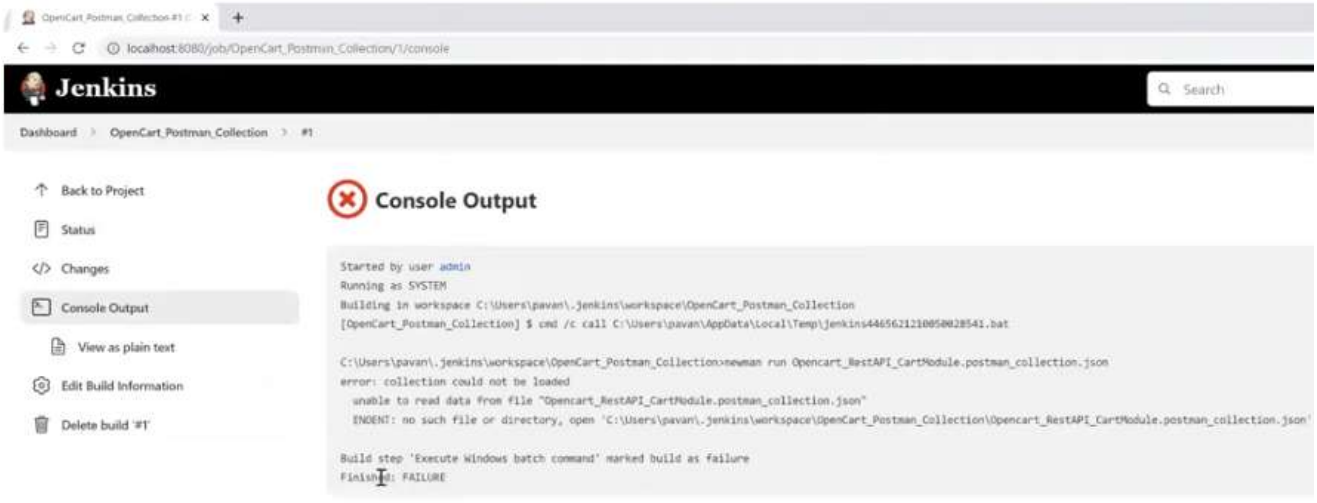

localhost:8080/job/OpenCart_Postman_Collection/configure
Dashboard
OpenCart_Postman_Collection
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
GitHub hook trigger for GITScm polling
Poll SCM
Build Environment
Delete workspace before build starts
Use secret text(s) or file(s)
Provide Configuration files
Add timestamps to the Console Output
Inspect build log for published Gradle build scans
Terminate a build if it's stuck
With Ant
Build
Add build step
Filter
Execute Windows batch command
Execute shell
GitHub PR: set 'pending' status
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Provide Configuration files
Run with timeout
Set build status to "pending" on GitHub commit

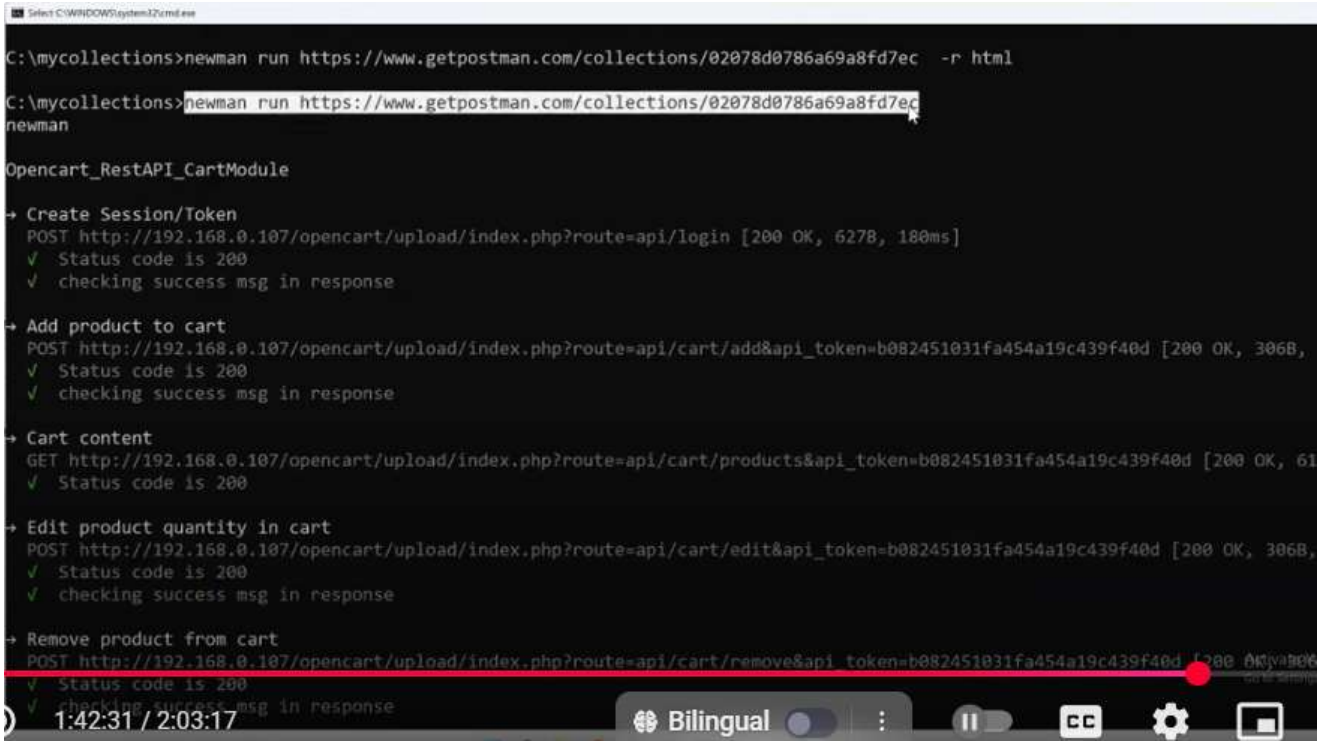

localhost:8080/job/OpenCart_Postman_Collection/configure
Saved
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
Add timestamps to the Console Output
Inspect build log for published Gradle build scans
Terminate a build if it's stuck
With Ant
Build
Execute Windows batch command
Command
See the list of available environment variables
newman run Opencart_RestAPI_CartModule.postman_collection.json
Advanced...
Add build step
Post-build Actions
Add post-build action
Save
Apply

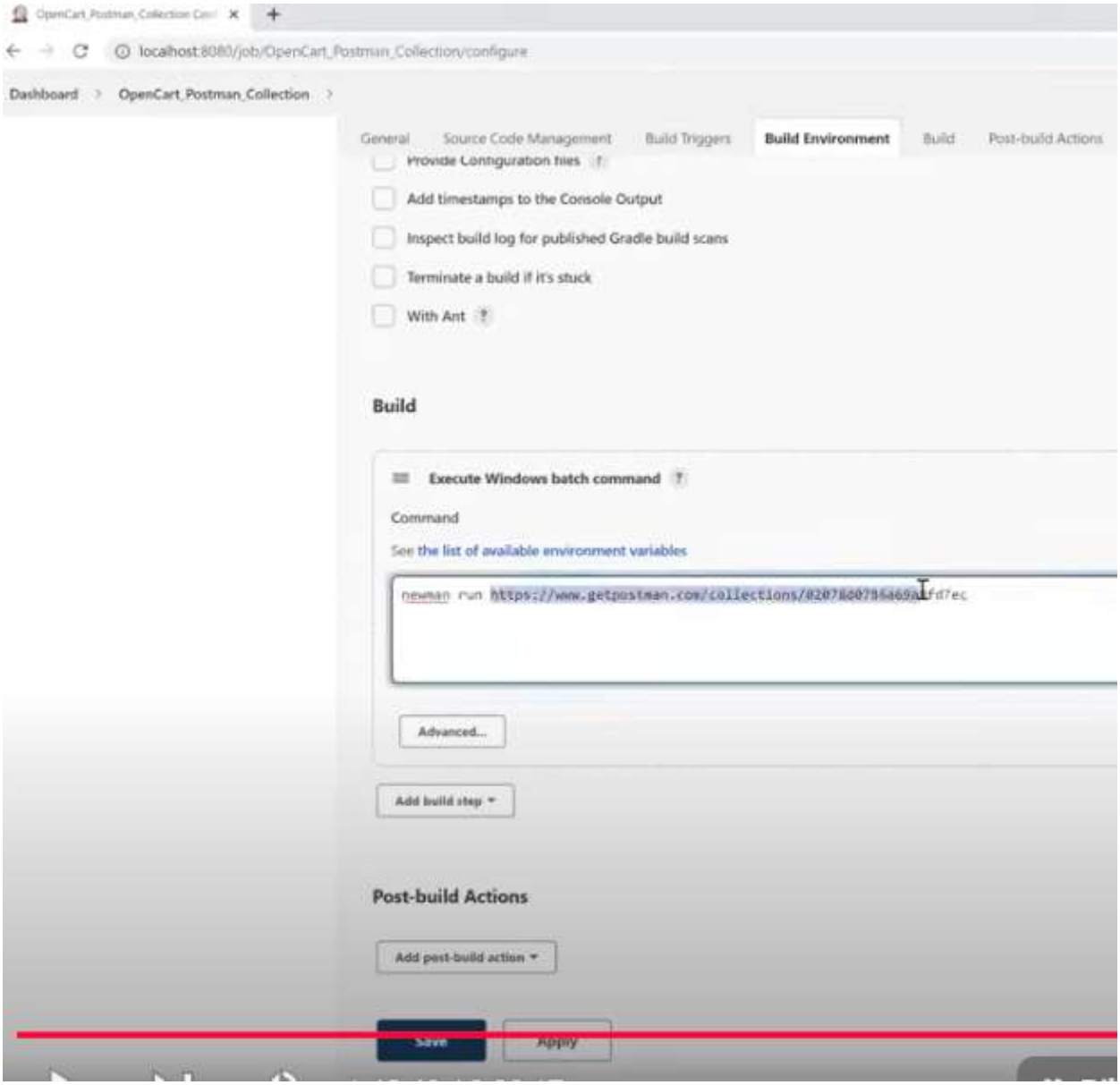

localhost:8080/job/OpenCart_Postman_Collection/
Jenkins
OpenCart_Postman_Collection
Back to Dashboard
Status
Changes
Workspace
Build Now
Configure
Delete Project
Rename
Project OpenCart_Postman_Collection
Workspace
Recent Changes
Permalinks
Build History
trend
Filter builds...
No builds
Atom feed for all
Atom feed for failures

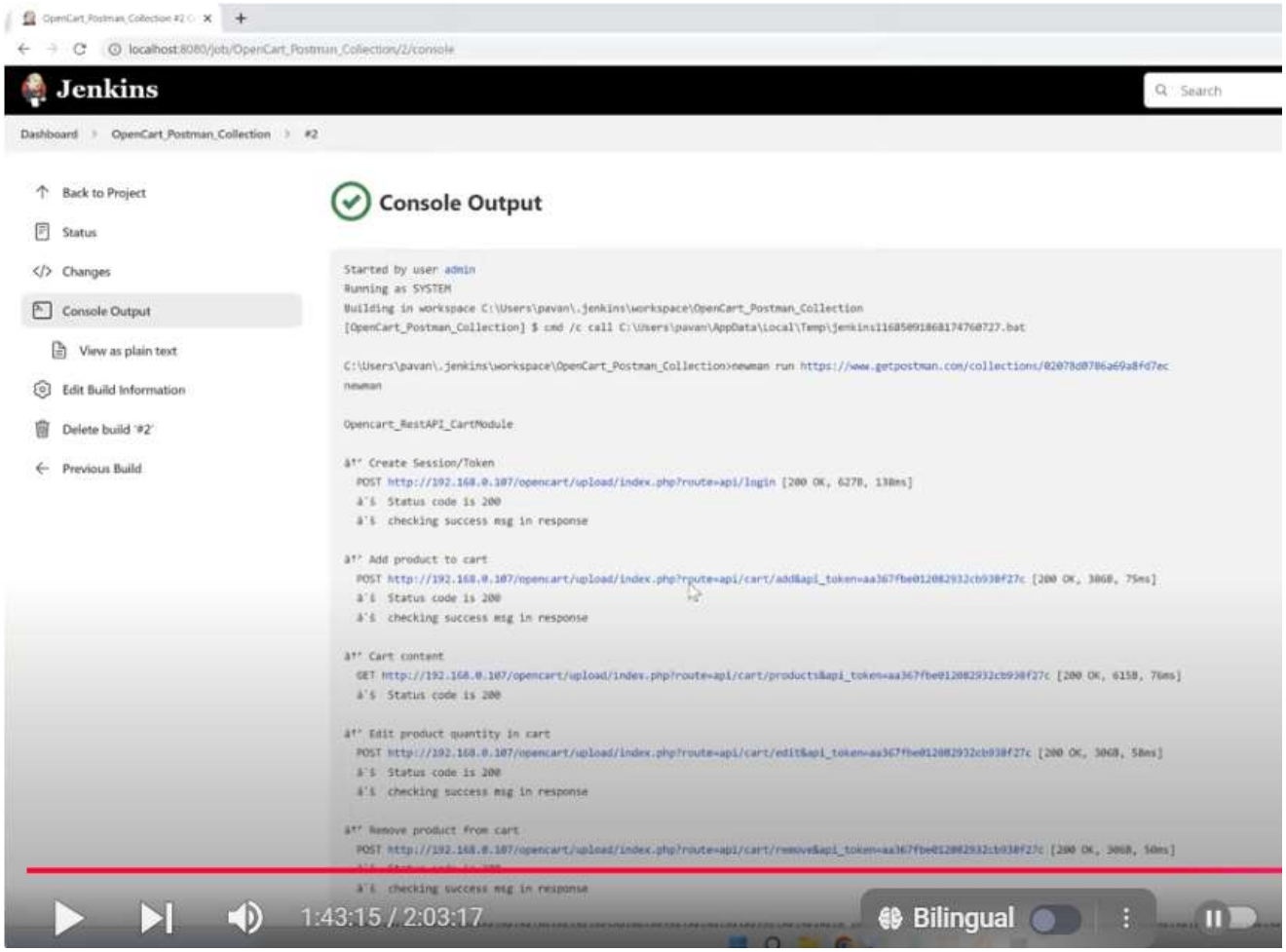

localhost:8080
Jenkins
Dashboard
New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
New View
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
All
S
W
Name
OpenCart_Postman_Collection
Changes
Workspace
Build Now
Configure
Delete Project
Rename

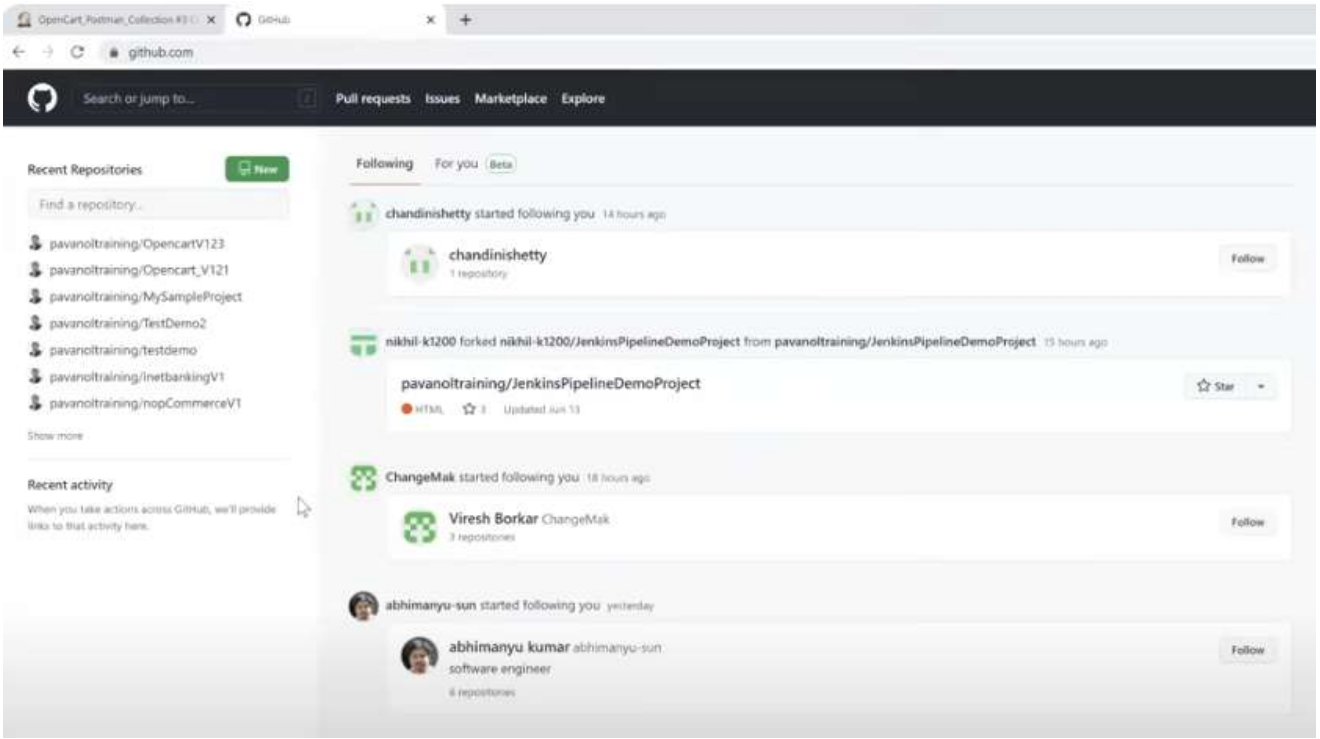

localhost:8080/job/OpenCart_Postman_Collection/1/
Jenkins
Dashboard
OpenCart_Postman_Collection
#1
Back to Project
Status
Changes
Console Output
Edit Build Information
Delete build '#1'
Build #1 (Jul 29, 2022, 8:25:14 AM)
No changes.
Started by user admin

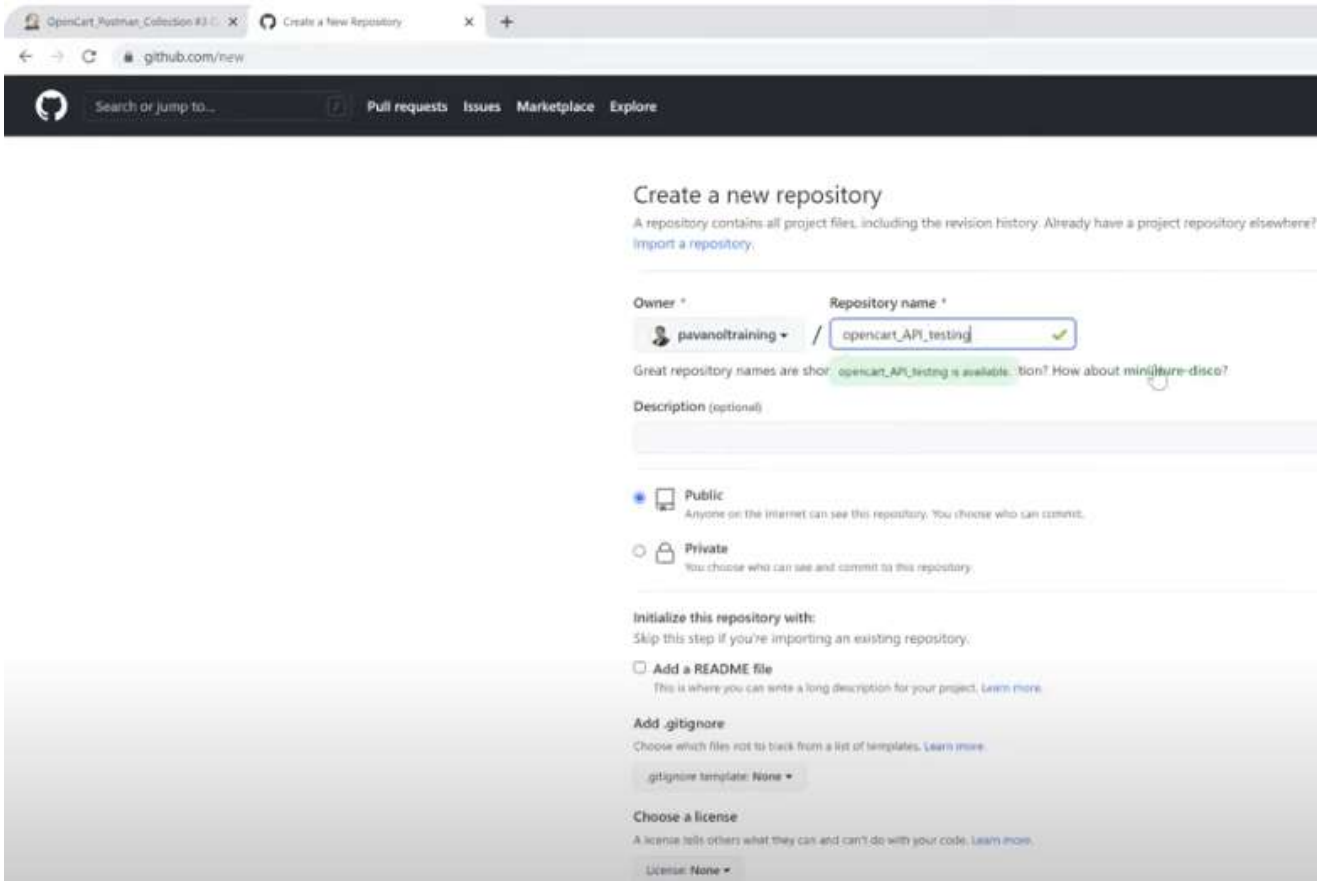

localhost:8080/job/OpenCart_Postman_Collection/1/console
Jenkins
Search
Dashboard
OpenCart_Postman_Collection
#1
Back to Project
Status
Changes
Console Output
View as plain text
Edit Build Information
Delete build '#1'
Console Output
Started by user admin
Running as SYSTEM
Building in workspace C:\Users\pavan\.jenkins\workspace\OpenCart_Postman_Collection
[OpenCart_Postman_Collection] $ cmd /c call C:\Users\pavan\AppData\Local\Temp\jenkins4465621210050028541.bat
C:\Users\pavan\.jenkins\workspace\OpenCart_Postman_Collection>newman run Opencart_RestAPI_CartModule.postman_collection.json
error: collection could not be loaded
unable to read data from file "Opencart_RestAPI_CartModule.postman_collection.json"
ENOENT: no such file or directory, open 'C:\Users\pavan\.jenkins\workspace\OpenCart_Postman_Collection\Opencart_RestAPI_CartModule.postman_collection.json'
Build step 'Execute Windows batch command' marked build as failure
Finished: FAILURE

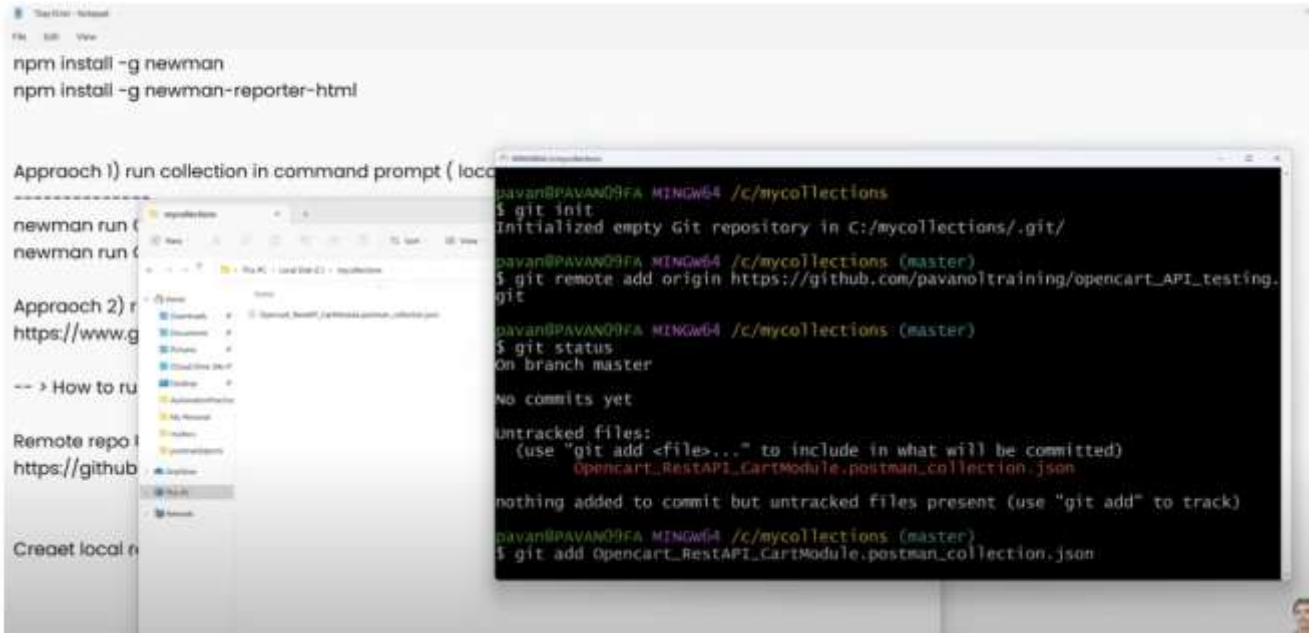

C:\mycollections>newman run https://www.getpostman.com/collections/02078d0786a69a8fd7ec -r html
C:\mycollections>newman run https://www.getpostman.com/collections/02078d0786a69a8fd7ec
newman
Opencart_RestAPI_CartModule
→ Create Session/Token
POST http://192.168.0.107/opencart/upload/index.php?route=api/login [200 OK, 627B, 180ms]
√ Status code is 200
√ checking success msg in response
→ Add product to cart
POST http://192.168.0.107/opencart/upload/index.php?route=api/cart/add&api_token=b082451031fa454a19c439f40d [200 OK, 306B,
√ Status code is 200
√ checking success msg in response
→ Cart content
GET http://192.168.0.107/opencart/upload/index.php?route=api/cart/products&api_token=b082451031fa454a19c439f40d [200 OK, 61
√ Status code is 200
→ Edit product quantity in cart
POST http://192.168.0.107/opencart/upload/index.php?route=api/cart/edit&api_token=b082451031fa454a19c439f40d [200 OK, 306B,
√ Status code is 200
√ checking success msg in response
→ Remove product from cart
√ Status code is 200
1:42:31 / 2:03:17
Bilingual

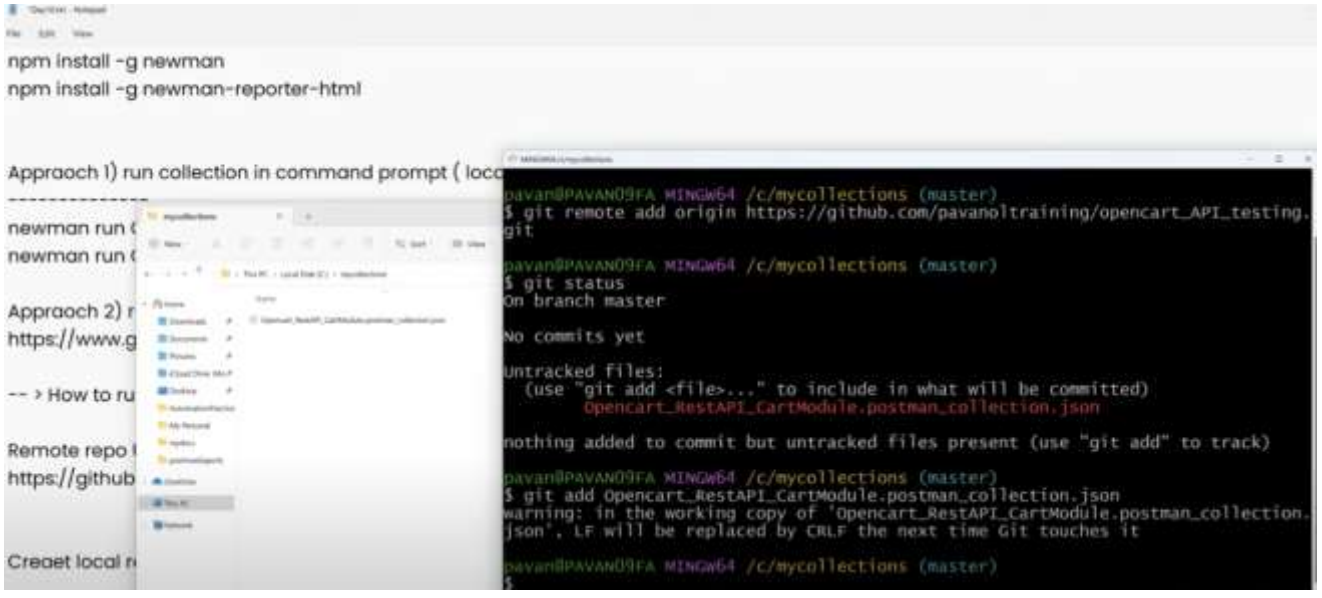

localhost:8080/job/OpenCart_Postman_Collection/configure
Dashboard
OpenCart_Postman_Collection
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
Add timestamps to the Console Output
Inspect build log for published Gradle build scans
Terminate a build if it's stuck
With Ant
Build
Execute Windows batch command
Command
See the list of available environment variables
newman run https://www.getpostman.com/collections/02078d0786a69a8fd7ec
Advanced...
Add build step
Post-build Actions
Add post-build action
Save
Apply

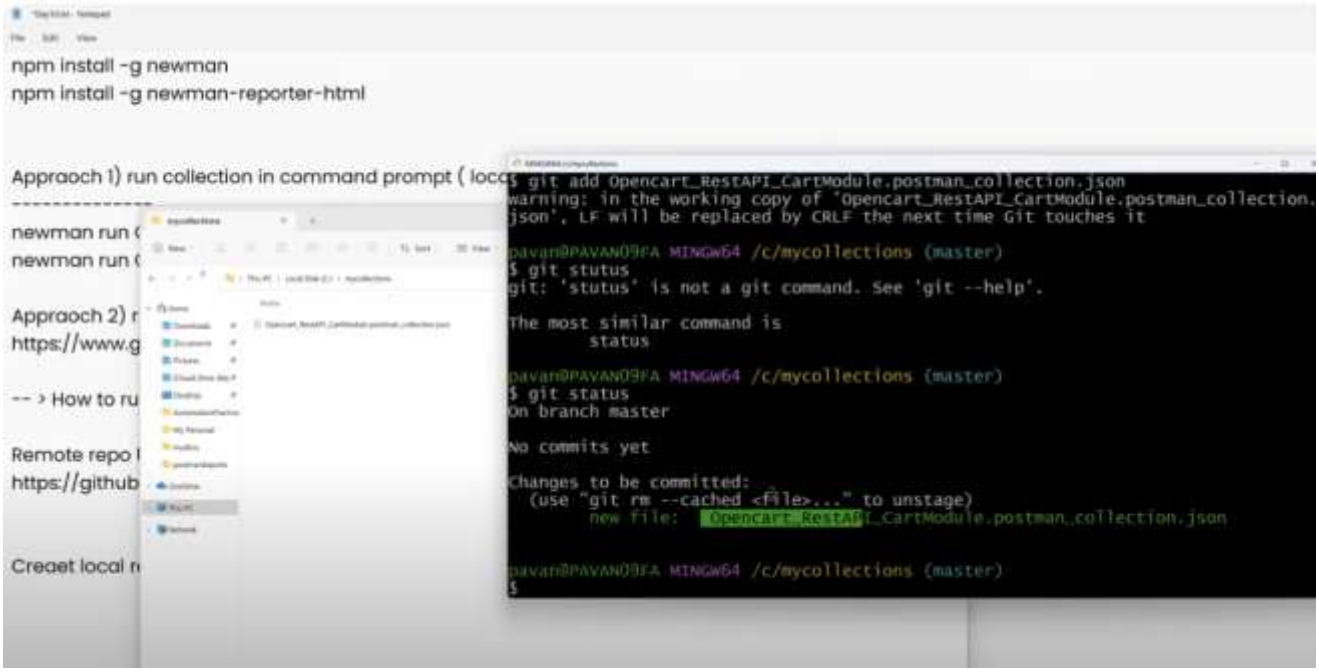

Jenkins
Search
Dashboard > OpenCart_Postman_Collection > #2
Back to Project
Status
Changes
Console Output
View as plain text
Edit Build Information
Delete build '#2'
Previous Build
Console Output
Started by user admin
Running as SYSTEM
Building in workspace C:\Users\pavan\.jenkins\workspace\OpenCart_Postman_Collection
[OpenCart_Postman_Collection] $ cmd /c call C:\Users\pavan\AppData\Local\Temp\jenkins1168509186817476072?.bat
C:\Users\pavan\.jenkins\workspace\OpenCart_Postman_Collection>newman run https://www.getpostman.com/collections/02078d0786a69a8fd7ec
newman
Opencart_RestAPI_CartModule
Create Session/Token
POST http://192.168.0.107/opencart/upload/index.php?route=api/login [200 OK, 627B, 138ms]
Status code is 200
checking success msg in response
Add product to cart
POST http://192.168.0.107/opencart/upload/index.php?route=api/cart/add&api_token=aa367fbe012082932cb938f27c [200 OK, 30GB, 75ms]
Status code is 200
checking success msg in response
Cart content
GET http://192.168.0.107/opencart/upload/index.php?route=api/cart/products&api_token=aa367fbe012082932cb938f27c [200 OK, 615B, 76ms]
Status code is 200
Edit product quantity in cart
POST http://192.168.0.107/opencart/upload/index.php?route=api/cart/edit&api_token=aa367fbe012082932cb938f27c [200 OK, 30GB, 58ms]
Status code is 200
checking success msg in response
Remove product from cart
POST http://192.168.0.107/opencart/upload/index.php?route=api/cart/remove&api_token=aa367fbe012082932cb938f27c [200 OK, 30GB, 56ms]
checking success msg in response
1:43:15 / 2:03:17
Bilingual

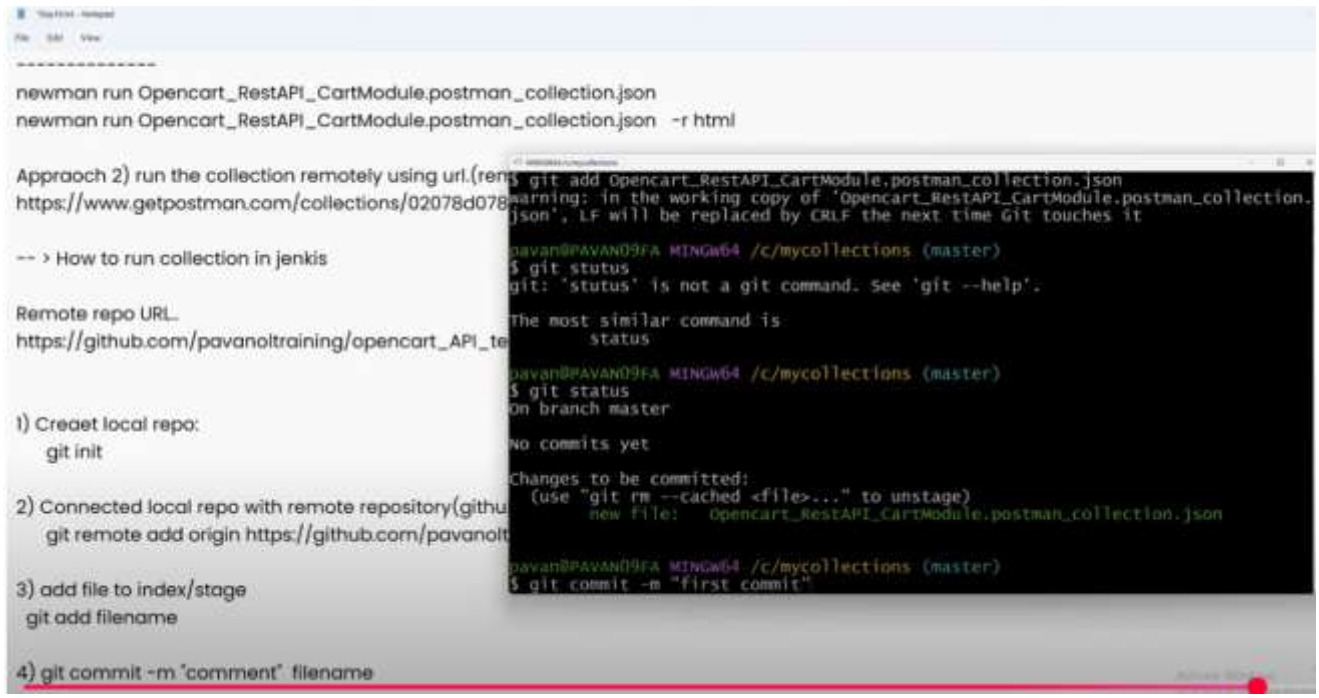

github.com
Search or jump to...
Pull requests
Issues
Marketplace
Explore
Recent Repositories
New
Find a repository...
pavanoltraining/OpencartV123
pavanoltraining/Opencart_V121
pavanoltraining/MySampleProject
pavanoltraining/TestDemo2
pavanoltraining/testdemo
pavanoltraining/InetbankingV1
pavanoltraining/nopCommerceV1
Show more
Recent activity
When you take actions across GitHub, we'll provide links to that activity here.
Following
For you
Beta
chandinishetty started following you 14 hours ago
chandinishetty
1 repository
Follow
nikhil-k1200 forked nikhil-k1200/JenkinsPipelineDemoProject from pavanoltraining/JenkinsPipelineDemoProject 15 hours ago
pavanoltraining/JenkinsPipelineDemoProject
HTML
Updated Jun 13
Star
ChangeMak started following you 18 hours ago
Viresh Borkar ChangeMak
3 repositories
Follow
abhimanyu-sun started following you yesterday
abhimanyu kumar abhimanyu-sun
software engineer
6 repositories
Follow


github.com/new
Search or jump to...
Pull requests
Issues
Marketplace
Explore
Create a new repository
A repository contains all project files, including the revision history. Already have a project repository elsewhere?
Import a repository.
Owner *
Repository name *
pavanoltraining
opencart_API_testing
Great repository names are short
opencart_API_testing is available.
tion? How about miniature-disco?
Description (optional)
Public
Anyone on the internet can see this repository. You choose who can commit.
Private
You choose who can see and commit to this repository.
Initialize this repository with:
Skip this step if you're importing an existing repository.
Add a README file
This is where you can write a long description for your project. Learn more.
Add .gitignore
Choose which files not to track from a list of templates. Learn more.
.gitignore template: None
Choose a license
A license tells others what they can and can't do with your code. Learn more.
License: None


npm install -g newman
npm install -g newman-reporter-html
Appraoch 1) run collection in command prompt (loca
newman run
newman run
Appraoch 2) r
https://www.g
-- > How to ru
Remote repo
https://github
Creaet local r
pavan@PAVAN09FA MINGW64 /c/mycollections
$ git init
Initialized empty Git repository in C:/mycollections/.git/
pavan@PAVAN09FA MINGW64 /c/mycollections (master)
$ git remote add origin https://github.com/pavanoltraining/opencart_API_testing.
git
pavan@PAVAN09FA MINGW64 /c/mycollections (master)
$ git status
On branch master
No commits yet
Untracked files:
(use "git add <file>..." to include in what will be committed)
Opencart_RestAPI_CartModule.postman_collection.json
nothing added to commit but untracked files present (use "git add" to track)
pavan@PAVAN09FA MINGW64 /c/mycollections (master)
$ git add Opencart_RestAPI_CartModule.postman_collection.json


npm install -g newman
npm install -g newman-reporter-html
Appraoch 1) run collection in command prompt (loca
newman run
newman run
Appraoch 2) r
https://www.g
-- > How to ru
Remote repo
https://github
Creaet local r
pavan@PAVAN09FA MINGW64 /c/mycollections (master)
$ git remote add origin https://github.com/pavanoltraining/opencart_API_testing.
git
pavan@PAVAN09FA MINGW64 /c/mycollections (master)
$ git status
On branch master
No commits yet
Untracked files:
(use "git add <file>..." to include in what will be committed)
Opencart_RestAPI_CartModule.postman_collection.json
nothing added to commit but untracked files present (use "git add" to track)
pavan@PAVAN09FA MINGW64 /c/mycollections (master)
$ git add Opencart_RestAPI_CartModule.postman_collection.json
warning: in the working copy of 'Opencart_RestAPI_CartModule.postman_collection.
json', LF will be replaced by CRLF the next time Git touches it
pavan@PAVAN09FA MINGW64 /c/mycollections (master)





newman run Opencart_RestAPI_CartModule.postman_collection.json -r html

Appraoch 2) run the collection remotely using url.(remote)
https://www.getpostman.com/collections/02078d0786a69a8fd7ec

-- > How to run collection in jenkis

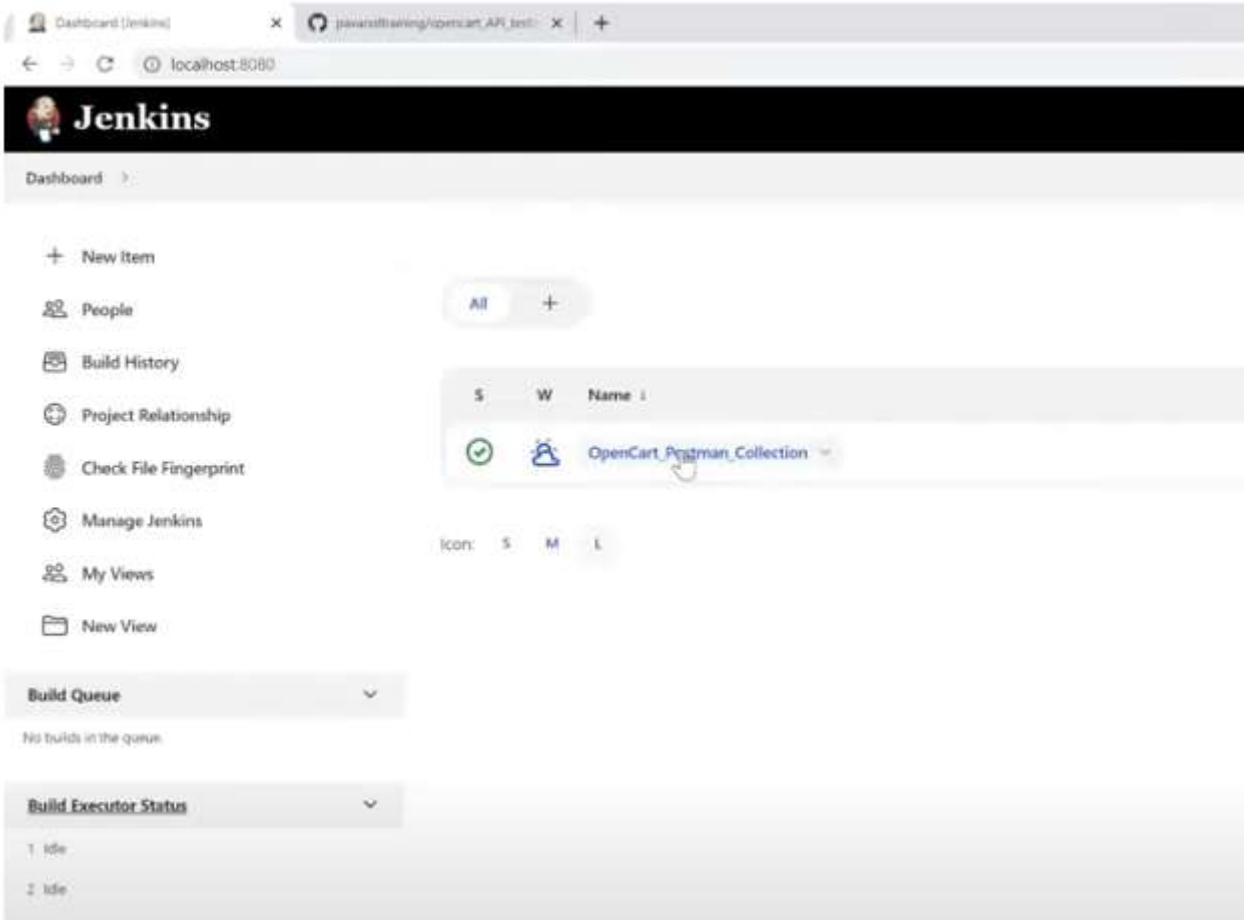
Remote repo URL..
https://github.com/pavanoltraining/opencart_API_testing.git

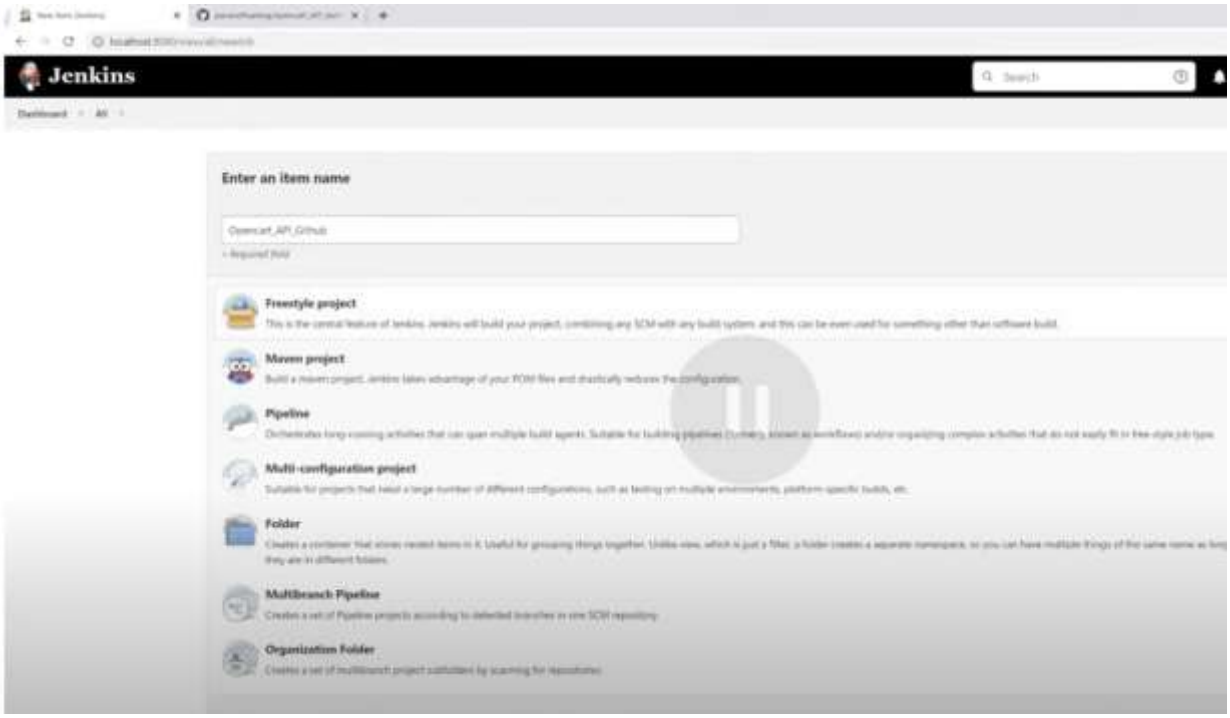
1) Creaet local repo:
git init

2) Connected local repo with remote repository(github)
git remote add origin https://github.com/pavanoltraining/opencart_API_testing.git

3) add file to index/stage
git add filename

4) git commit -m "comment"





Remote repo URL..
https://github.com/pavanoltraining/opencart_API_testing.git

1) Creaet local repo:
git init

2) Connected local repo with remote repository(github)
git remote add origin https://github.com/pavanoltraining/opencart_API_testing.git

3) add file to index/stage
git add filename

4) git commit -m "comment"

5) git push origin master

```
  (use "git rm --cached <file>..." to unstage)
        new file:   Opencart_RestAPI_CartModule.postman_collection.json

pavan@PAVAN09FA MINGW64 /c/mycollections (master)
$ git commit -m "first commit"
[master (root-commit) 6646ba5] first commit
 1 file changed, 409 insertions(+)
 create mode 100644 Opencart_RestAPI_CartModule.postman_collection.json

pavan@PAVAN09FA MINGW64 /c/mycollections (master)
$ git status
On branch master
nothing to commit, working tree clean

pavan@PAVAN09FA MINGW64 /c/mycollections (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.72 KiB | 587.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
```

commands to install above components..

npm install -g newman
npm install -g newman-reporter-html

Appraoch 1) run collection in command prompt (local)
---
newman run Opencart_RestAPI_CartModule.postman_collection.json
newman run Opencart_RestAPI_CartModule.postman_collection.json -r html

Appraoch 2) run the collection remotely using url.(remote)
Shared URL: https://www.getpostman.com/collections/02078d0786a69a8fd7ec

newman run https://www.getpostman.com/collections/02078d0786a69a8fd7ec -r html

-- > How to run collection in jenkis

Remote repo URL..
https://github.com/pavanoltraining/opencart_API_testing.git

newman run Opencart_RestAPI_CartModule.postman_collection.json
newman run Opencart_RestAPI_CartModule.postman_collection.json -r html

Appraoch 2) run the collection remotely using url.(remote)
Shared URL: https://www.getpostman.com/collections/02078d0786a69a8fd7ec

newman run https://www.getpostman.com/collections/02078d0786a69a8fd7ec -r html

Appraoch 2) Run the collection in jenkins(locally)

Appraoch 3) Run the colelction in jenkins from Github
Remote repo URL:
https://github.com/pavanoltraining/opencart_API_testing.git

1) Creaet local repo:
git init

2) Connected local repo with remote repository(github)
git remote add origin https://github.com/pavanoltraining/opencart_API_testing.git

3) add file to index/stage
git add filename