

Postman

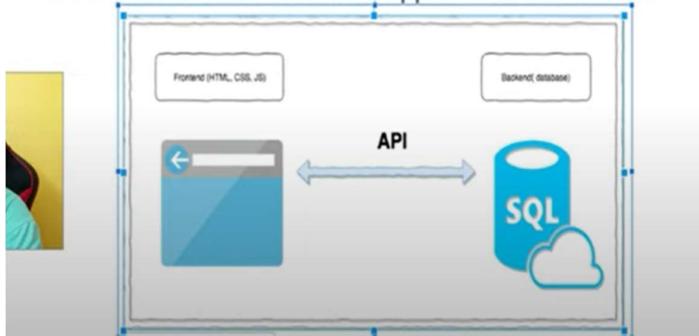
What is API ?

API stands for the Application Programming Interface,

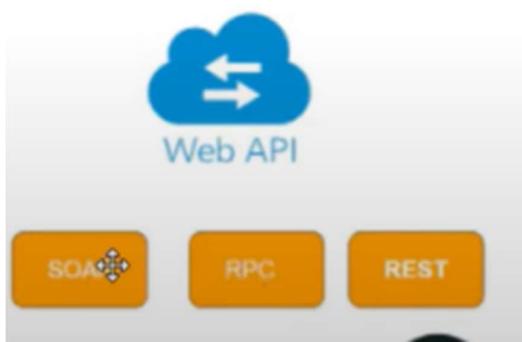


What is API ?

- Basically a collection of functions and procedures which allows us to communicate two applications or libraries.



➔ There are three types of web api



What is REST API?

REST API creates an object and thereafter sends the values of the objects in response to the client request

What is REST API?

- REST is an acronym for REpresentational State Transfer.
- The REST architectural style describes six constraints.
- These constraints, put on the architecture, were initially communicated by Roy Fielding in his doctoral dissertation and defines the basis of RESTful-style.

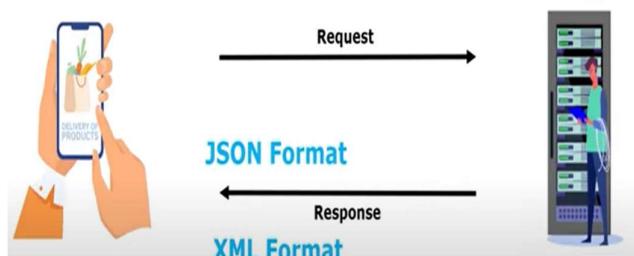
REST = Representational state transfer.

What is REST API?

- In Simple Terms REST API is nothing but a Design Pattern for Web API

Why do we need REST API?

We want the data to be in a structured format rather than a complete webpage



→ We received data in json format and xml format

JSON Format

```
{"city":{  
    "restauranname":{  
        "fooditem":"sandwich"  
    }  
}}
```

Object - value

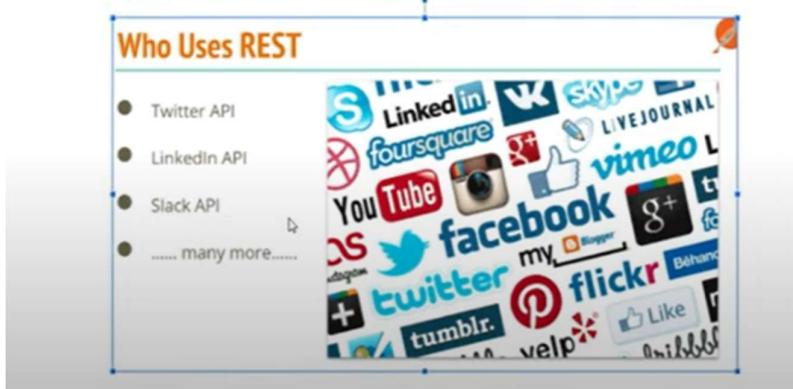
XML Format

```
<city>  
    <restauranname>  
        <fooditem>sandwich</fooditem>  
    </restauranname>  
</city>
```

Hierarchical Data Structure

REST Used by

Most of WebApps, Mobiles are relying on REST API



→ If any api can follow these 6 rules it means this is rest api

REST Constraints

- Uniform Interface
- Stateless
- Cacheable
- Client-Server

Layered System

Code on Demand

<https://scrolltest.com/rest/explained>

Types of testing in API testing

Functional Testing.

Integration Testing

Regression Testing

Security Testing

Load Testing

Penetration Testing

Fuzz testing - discover coding errors



francesc #BlackLivesMatter ... · 18h ✓

HTTP status codes as emoji .. this might be a good idea?

- 200
- 201
- 301
- 400
- 401
- 402
- 403
- 404
- 408
- 410
- 418
- 500

httpstatuses.com

HTTP Status Codes

httpstatuses.com is an easy to reference database of HTTP Status Codes with their definitions and helpful code references all in one place. Visit an individual status code via <http://httpstatuses.com/code> or browse the list below.

[@ Share on Twitter](#) [⊕ Add to Pinboard](#)

1xx Informational

100 Continue
101 Switching Protocols
102 Processing

2xx Success

200 OK
201 Created
202 Accepted
203 Non-authoritative Information
204 No Content
205 Reset Content
206 Partial Content
207 Multi-Status
208 Already Reported
226 IM Used

3xx Redirection

300 Multiple Choices
301 Moved Permanently
302 Found
303 See Other
304 Not Modified
305 Use Proxy
307 Temporary Redirect
308 Permanent Redirect

4xx Client Error

400 Bad Request
401 Unauthorized
402 Payment Required
403 Forbidden
404 Not Found
405 Method Not Allowed
406 Not Acceptable
407 Proxy Authentication Required
408 Request Timeout
409 Conflict
410 Gone
411 Length Required
412 Precondition Failed
413 Payload Too Large
414 Request-URI Too Long
415 Unsupported Media Type
416 Requested Range Not Satisfiable
417 Expectation Failed
418 I'm a teapot
421 Misdirected Request
422 Unprocessable Entity
423 Locked
424 Failed Dependency
426 Upgrade Required
428 Precondition Required
429 Too Many Requests
431 Request Header Fields Too Large
444 Connection Closed Without Response
451Unavailable For Legal Reasons
499 Client Closed Request

5xx Server Error

500 Internal Server Error
501 Not Implemented
502 Bad Gateway
503 Service Unavailable
504 Gateway Timeout
505 HTTP Version Not Supported
506 Variant Also Negotiates
507 Insufficient Storage
508 Loop Detected
510 Not Extended
511 Network Authentication Required
599 Network Connect Timeout Error

POSTMAN Basics

- OverView
- Import External Request.
- History
- WorkSpace in Postman
- Postman Tabs - Param, Auth, Headers, Body, Pre-req, Post Req(Test), Settings
- Postman Console.

The screenshot shows the Postman interface with a GET request to `https://reqres.in/api/users/2`. The 'Params' tab is active, displaying a table with one row:

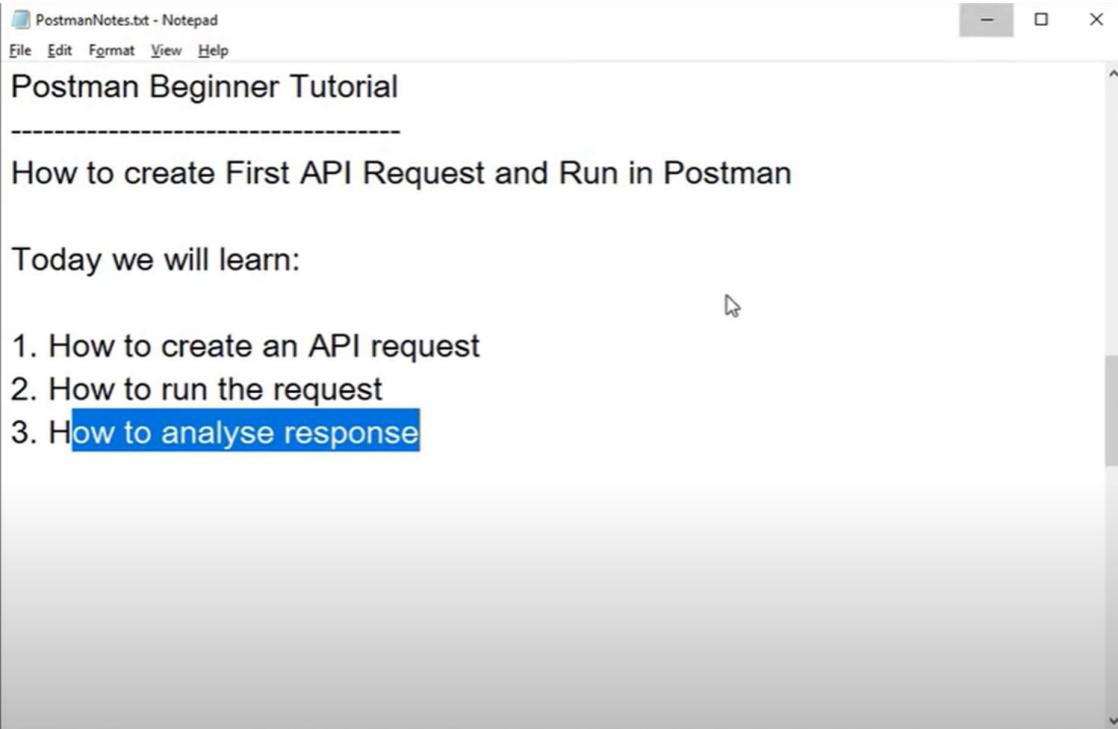
KEY	VALUE	DESCRIPTION
name	Janet	Description

Handwritten annotations include 'PATH PARAMS' written above the URL and 'Query Params' with an arrow pointing to the 'Params' tab.

Open test link = <https://reqres.in/>

→ <https://reqres.in/api/users/1> (data change)

→ <https://reqres.in/api/users/2> (data change)



→ If we want to have a sample api we use this website to get the sample api. = <https://reqres.in/>

→ Select any **GET** request and paste in new tab url you get the same data.

→ Copy url & also copy get request id & then paste whole url in new tab .

- Create a new workspace postman.
- Select get request & enter this URL <https://reqres.in/api/users/2> you get the same data.
- we get same data in postman also

Postman screenshot showing a successful GET request to <https://reqres.in/api/users/2>. The response status is 200 OK, time is 249 ms, and size is 1.2 KB. The response body is a JSON object:

```

1
2   "data": {
3     "id": 2,
4     "email": "janet.weaver@reqres.in",
5     "first_name": "Janet",
6     "last_name": "Weaver",
7     "avatar": "https://reqres.in/img/faces/2-image.jpg"
8   },
9   "support": {
10    "url": "https://reqres.in/#support-heading",
11    "text": "To keep ReoRes free. contributions towards server costs are appreciated!"
}

```

→ Types of data format

Postman screenshot showing a successful GET request to <https://reqres.in/api/users/2>. The response status is 200 OK, time is 57 ms, and size is 119 KB. The response body is a JSON object. A yellow annotation points to the JSON dropdown menu with the text "many format data".

```

1
2   "data": {
3     "id": 2,
4     "email": "janet.weaver@reqres.in",
5     "first_name": "Janet",
6     "last_name": "Weaver",
7     "avatar": "https://reqres.in/img/faces/2-image.jpg"
8   },
9   "support": {
10    "url": "https://reqres.in/#support-heading",
11    "text": "To keep ReoRes free. contributions towards server costs are appreciated!"
}

```

→ Status code ,response size, time taken

Postman screenshot showing a successful GET request to <https://reqres.in/api/users/2>. The response status is 200 OK, time is 57 ms, and size is 119 KB. The response body is a JSON object. A yellow annotation highlights the status bar showing the response size, request size, and total size.

Status: 200 OK Time: 57 ms Size: 119 KB

Response Size: 119 KB
Request Size: 232 B
Total Size: 232 B

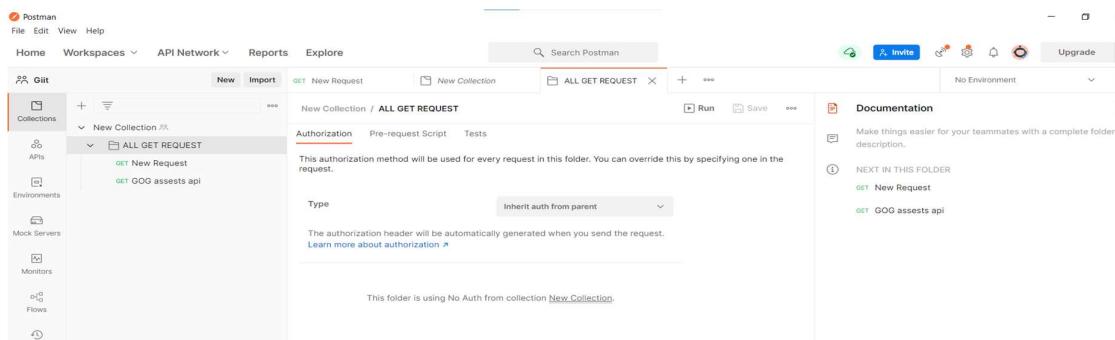
STEP 2:-

1. What is COLLECTION
2. How to create Collection
3. How to create folders inside collection
4. How to arrange requests inside collection

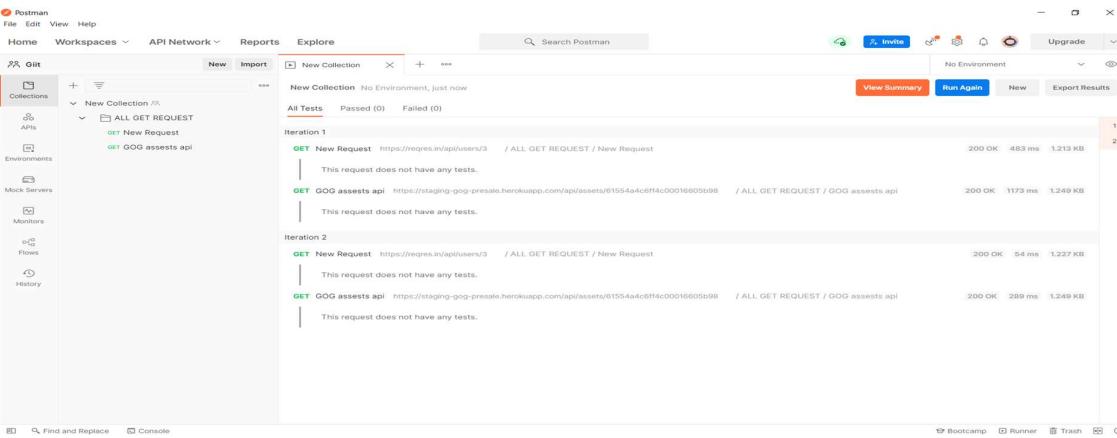
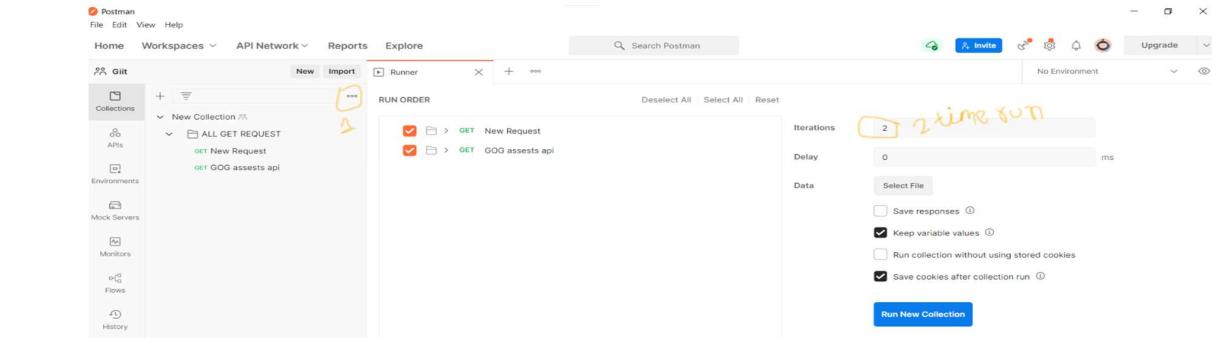
- Collection is a group of api request that can be stored and saved in logical arrangement.
- When you create a collection you have option to use security function like authorization, in authorization you can select **auth** to create username and password that means, firstly when user run your api Enter the name and password, then the API will provide you the data..
- If you don't want to use authentication you don't select it.
- We have also an option to share our collection with our colleague and anyone whom we want to share.
- We can also create an folder.

How to run COLLECTION
Today we will learn:
1. How to run a collection
2. How to analyse the result

We create a folder in the collection and in the folder we put all our **GET** request file, so when we run our folder all API are run continuously.



- Run multiple api in same time



VARIABLES

What | Why | When | How

Today we will learn:

1. What are variables in POSTMAN
2. Why to use variables
2. How to create variables
3. How to refer variables

WHAT

variables - elements(data store) that can take different values

WHY

to reuse values at multiple places

avoid repetition

to avoid re-work when value changes

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', 'Flows', and 'History'. Under 'Collections', 'GOOG' is expanded, and 'get GOOG assets api' is selected. A context menu is open over 'New Collection', listing options like 'Share', 'Move', 'Run collection', 'Edit', 'Add request', 'Add folder', 'Monitor collection', 'Mock collection', 'Create a fork', 'Create Put Request', 'Merge changes', 'View documentation', 'Rename', 'Duplicate', 'Export', and 'Manage roles'. The main workspace shows a 'New Collection' card with 'Authorization' set to 'No Auth'. The right panel displays 'Collection details' for the new collection, including its ID (16093185-5601e2b1-ee67-49b3-8720-55ae5a31818a), creation by 'You' on '07 Oct 2021, 3:42 PM', and no mock servers or monitors created. There are sections for 'Integrations' and 'Backup your collection'.

→ If we want to call a URL again and again, then we store that URL in a variable and keep it.

This screenshot shows the 'Variables' dialog box in Postman. It lists a single variable named 'url' with an initial value of 'https://regres.in' and a current value of 'http://regres.in'. There are buttons for 'Persist All', 'Reset All', and 'Set as variable'. A note at the bottom says: 'Use variables to reuse values in different places. Work with the current value of a variable to prevent sharing sensitive values with your team. Learn more about variable values'.

→ In the search URL, first we put the variable name in brackets where you store the common link.& then give the url - api details & you see your data is visible.

This screenshot shows a 'New Request' dialog box in Postman. The method is 'GET' and the URL is '(url)/api/users/2'. The 'Body' tab is selected, showing a JSON response with fields like 'data': { 'id': 2, 'email': 'emma.wong@reqres.in', 'name': 'Emma Wong', 'last_name': 'Wong', 'avatar': 'https://reqres.in/img/faces/3-image.jpg', 'support': { 'url': 'https://reqres.in/support-heading' } }. The status bar at the bottom indicates 'Status: 200 OK'.

→ If you want to see your data in postman console you have a option to see this

This screenshot shows the 'View' menu context menu in Postman. The menu items include 'Toggle Full Screen', 'Zoom In', 'Zoom Out', 'Reset Zoom', 'Toggle Sidebar', 'Toggle Two-Pane View', 'Next Tab', 'Previous Tab', 'Show Postman Console', and 'Developer'. To the right of the menu, there's a 'Reports' section with an 'Import' button.

Postman Console

File Edit View Help

Search messages

All Logs

	Response Status	Latency
GET https://staging-gog-presale.herokuapp.com/api/assets/61554a4c6ff4c00016605b98	200	1375 ms
GET https://staging-gog-presale.herokuapp.com/api/owner/61554a4c6ff4c00016605b98	404	348 ms
GET https://staging-gog-presale.herokuapp.com/api/assets/61554a4c6ff4c00016605b98	404	284 ms
GET https://staging-gog-presale.herokuapp.com/api/assets/61554a4c6ff4c00016605b98	200	310 ms
GET https://reqres.in/api/users/3	404	453 ms
GET https://reqres.in/api/users/2	404	328 ms
GET https://reqres.in/api/users/2	404	48 ms
GET https://reqres.in/api/users/1	404	345 ms
GET https://reqres.in/api/users/2	200	55 ms
GET https://reqres.in/api/users/2	200	66 ms
GET https://reqres.in/api/users/2	200	46 ms
GET https://reqres.in/api/users/2	200	58 ms
GET https://reqres.in/api/users/2	200	52 ms
GET https://reqres.in/api/users/3	200	48 ms

Show timestamps Hide network

→ How can we set & get variable through a script.

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Search Postman

New Import GET GOG assets api GET New Request GET New Request

New Collection / ALL GET REQUEST / New Request

GET {{url}}/api/users/3

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Write Script

Test scripts are written in JavaScript, and are run after the response is received. Learn more about tests scripts

SNIPPETS Get an environment variable Get a global variable Get a variable Get a collection variable Set an environment variable

Send

Body Cookies Headers (19) Test Results

Status: 200 OK Time: 219 ms Size: 1.19 KB Save Response

There are no tests for this request

Write a test script to automate debugging

→ Script =>

Postman

File Edit View Help

Workspaces API Network Reports Explore

Search Postman

New Import GET GOG assets api GET New Request GET New Request

New Collection / ALL GET REQUEST / New Request

GET {{url}}/api/users/3

Params Auth Headers (7) Body Pre-req. Tests Settings

1 console.log("Hello world");

Test scripts are written in JavaScript, and are run after the response is received. Learn more about tests scripts

SNIPPETS Get an environment variable Get a global variable Get a variable Get a collection variable Set an environment variable

Send

Body Cookies Headers (19) Test Results

200 OK 219 ms 1.19 KB Save Response

There are no tests for this request

Write a test script to automate debugging

No logs yet

Send a request to view its details in the console.

→ Result in console :-

Postman interface showing a test script in the Tests tab:

```
1 console.log("Hello world");
```

Test Results: 200 OK 183 ms 1.19 KB

Postman Console output:

```
15:57:25.574 > GET https://reqres.in/api/users?page=2
15:57:25.635 "Hello world"
15:57:25.650 "Value for url variable is : https://reqres.in/"
```

→ If we want to see our log files we see in console

Postman interface showing a test script in the Tests tab:

```
1 console.log("Hello world");
```

Test Results: 200 OK 183 ms 1.19 KB

Postman Console output:

```
15:57:25.574 > GET https://reqres.in/api/users?page=2
15:57:25.635 "Hello world"
15:57:25.650 "Value for url variable is : https://reqres.in/"
```

→ Using test case to print

Postman interface showing a collection named 'learn postman' with a 'Variables through scripting using tests' section:

- GET collection api
- Github
- Variable collection, variable name = URL
 - API Request collection
 - GET API 1
 - GET API 2
- Mock Servers
 - Variables through scripting using tests
 - GET using test print (" hello satyam..."...)
 - GET set variable name in tests 4 th li...
 - GET Fetch global environment value ...

Postman Console output:

```
15:57:25.574 > GET https://reqres.in/api/users?page=2
15:57:25.635 "Hello world"
15:57:25.650 "Value for url variable is : https://reqres.in/"
```

Request Body (Pretty) output:

```
1 {
2   "page": 2,
3   "per_page": 6,
4   "total": 12,
5   "total_pages": 2,
6   "data": [
7     {
8       "id": 7,
9       "email": "michael.lawson@reqres.in",
10      "first_name": "Michael",
```

→ Using test casr to set a variable

The screenshot shows the Postman interface with a collection named "learn postman". A test script is being run on a GET request to "https://reqres.in/api/users?page=2". The script sets a variable "urlVar" to the value of the "url" parameter from the URL. It then logs the value of "urlVar" and uses it to set a global environment variable "gift". Finally, it logs the value of "name" from the response body.

```

1 console.log("hello satyam....")
2 let urlVar = pm.variables.get("url");
3 console.log("Value for url variable is : "+urlVar);
4 pm.variables.set("name","gift (we set gift as a variable)");
5 console.log(pm.variables.get("name"))
    
```

The Postman Console output shows the logs from the test script running successfully.

→ Using test case to fetch global environment

The screenshot shows the Postman interface with a collection named "learn postman". A test script is being run on a GET request to "https://reqres.in/api/users?page=2". The script logs the value of "urlVar" and then logs the value of the "name" field from the response body. It also prints the value of the global environment variable "AUTOMATION".

```

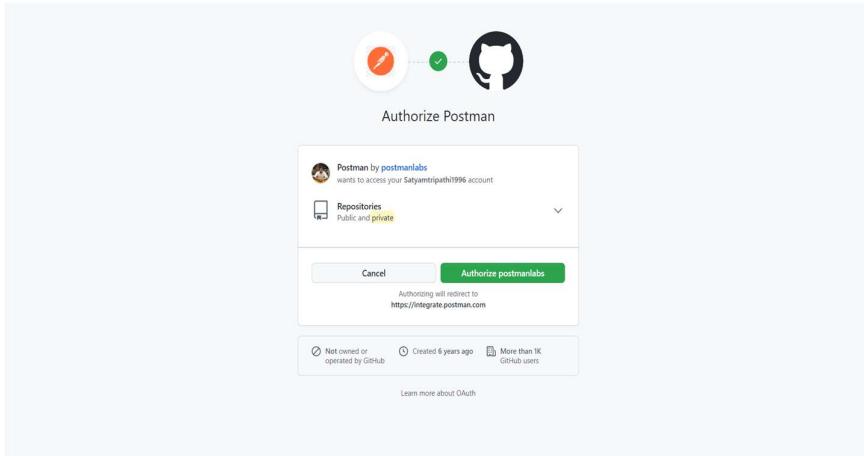
1 console.log("hello satyam....")
2 let urlVar = pm.variables.get("url");
3 console.log("Value for url variable is : "+urlVar);
4 pm.variables.set("name","gift (we set gift as a variable)");
5 console.log(pm.variables.get("name"))
6 let globalVar = pm.globals.get("Environment");
7 console.log(globalVar,"AUTOMATION is the value of Environment that is declare in global Environment.
Please see in postman console");
    
```

The Postman Console output shows the logs from the test script running successfully.

→ Connect postman to github

→ Api /import/ code repository /github

The screenshot shows the Postman interface with a collection named "learn postman". A modal dialog titled "Import" is open, showing options to import from a file, folder, link, raw text, or code repository. The "Code repository" tab is selected, with "GitHub" chosen as the provider. The "Import from code repository" section is visible, prompting the user to select a Git service to scan for files that can be imported.



→ Connected github in postman, choose your repo

The screenshot shows the Postman application interface. On the left, there's a sidebar with options like Home, Workspaces, API Network, Reports, and Explore. The main area shows a collection named 'learn postman' with a status of 'No APIs yet'. A central window is open for 'Import' from GitHub, with the following details:

- Authenticated on GitHub as: Satyamtripathi1996
- Select your GitHub organization: Satyamtripathi1996
- Select repository: box-designer-website
- Select branch: master

At the bottom of the import window, there's a preview of a JSON response with the following code:

```
6   data : 1
7   [
8     "id": 7,
```

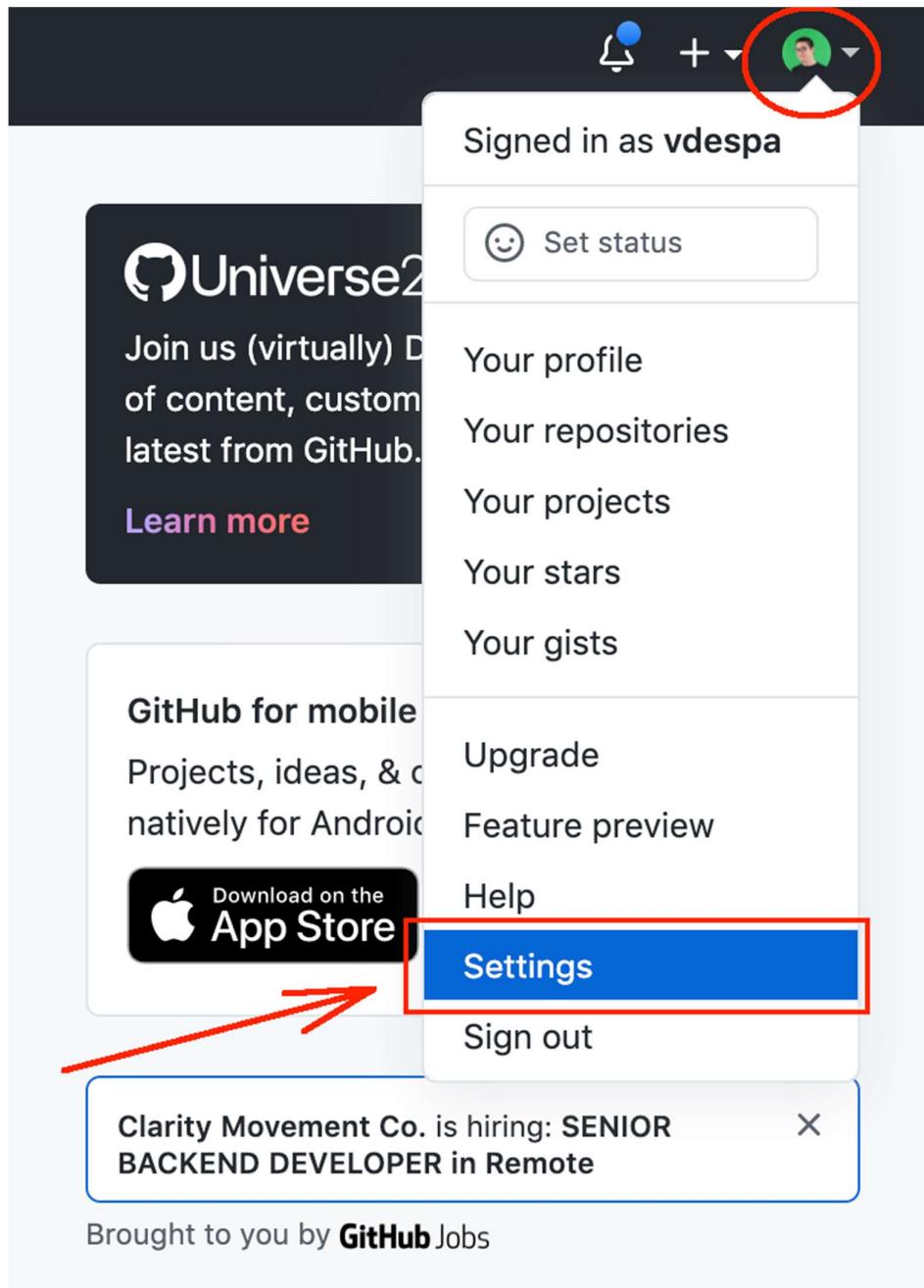
Github has changed the API starting November 13, 2020 and authentication using a username and a password is NO LONGER PERMITTED.

You need to generate a **personal access token**.

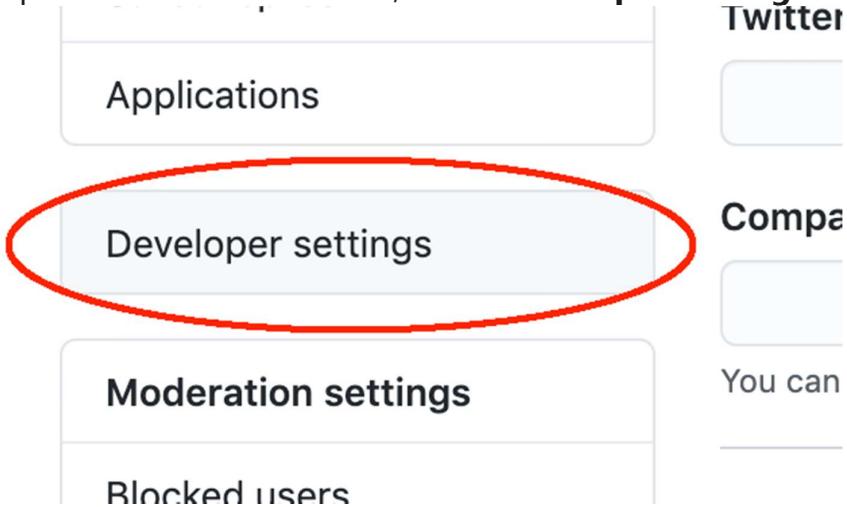
Only some of the content in this section will work BUT not all. I am sorry about this.

The main concepts remain the same.

Step 1. Create a Github.com account if you don't have one already and go to Settings.



Step 2. From the left sidebar, select **Developer Settings**.



Step 3. From the left sidebar, select **Personal access tokens**.



Step 3. Click on **Generate new token**.

Step 4. Name your token Postman or as you prefer. Select the permissions as in the image below.

Note

Postman

What's this token for?

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> write:packages	Upload packages to github package registry
<input type="checkbox"/> read:packages	Download packages from github package registry
<input type="checkbox"/> delete:packages	Delete packages from github package registry
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects
<input type="checkbox"/> admin:public_key	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys
<input type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input type="checkbox"/> write:repo_hook	Write repository hooks
<input type="checkbox"/> read:repo_hook	Read repository hooks
<input type="checkbox"/> admin:org_hook	Full control of organization hooks
<input type="checkbox"/> gist	Create gists
<input type="checkbox"/> notifications	Access notifications
<input type="checkbox"/> user	Update all user data
<input type="checkbox"/> read:user	Read all user profile data
<input type="checkbox"/> user:email	Access user email addresses (read-only)
<input type="checkbox"/> user:follow	Follow and unfollow users
<input checked="" type="checkbox"/> delete_repo	Delete repositories
<input type="checkbox"/> write:discussion	Read and write team discussions
<input type="checkbox"/> read:discussion	Read team discussions
<input type="checkbox"/> admin:enterprise	Full control of enterprises
<input type="checkbox"/> manage_billing:enterprise	Read and write enterprise billing data
<input type="checkbox"/> read:enterprise	Read enterprise profile data
<input type="checkbox"/> workflow	Update github action workflows
<input type="checkbox"/> admin:gpg_key	Full control of public user gpg keys (Developer Preview)
<input type="checkbox"/> write:gpg_key	Write public user gpg keys
<input type="checkbox"/> read:gpg_key	Read public user gpg keys

Generate token

Cancel

Step 5. Click on **Generate token**.

Step 6. Keep this page open otherwise, you will not see the token.

Personal access tokens

[Generate new token](#)

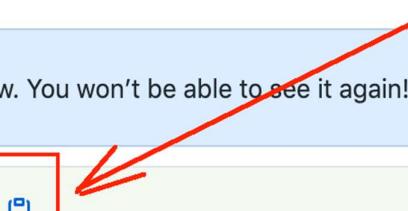
[Revoke all](#)

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your new personal access token now. You won't be able to see it again!

✓ 6e664f19461... Ocba3bafa [Copy](#)

[Delete](#)



Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Step 7. In Postman, instead of using Basic Auth, select Bearer Token.

GET https://api.github.com/user/repos

Params Authorization Headers (8) Body Pre-request Script Tests Settings

TYPE
Bearer Token

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Token
6e664f19461... Ocba3bafa

→ Aws Signature use

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

Github / New Request

GET https://api.github.com/user/repos

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Type AWS Signature

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Add authorization data to Request Header

AWS Region e.g. us-east-1

Service Name e.g. s3

Session Token Session Token

Status: 404 Not Found Time: 244 ms Size: 1.26 KB Save Response

Step 1

Add a new request to the collection that **creates a new Github repository** with the following properties:

- name: Test repository (*random string*)
- description: This is a test repository created by Postman

Add one **test** that checks the status code (expected 201).

Note: *by visiting your profile, you can notice that the repository is now visible.*

Hint: *in order to create the repository, you need to send JSON in your request body.*

Step 2

Add a new request to the collection that **retrieves (with GET) the newly created Github repository**.

Add one **test** that checks the status code (expected 200).

This is an additional test step that ensures that the repository was indeed created.

Step 3

Add a new request to the collection that **creates a few issue** inside the repository with the following content:

- title: Found a bug
- body: This issue has been automatically created by Postman.

Add one **test** that checks the status code (expected 201).

Step 4

Add a new request to the collection that **retrieves (with GET) the newly created Github repository issue**.

Add one **test** that checks the status code (expected 200).

Add one **test** that checks the issue title (expected "Found a bug").

This is an additional test step that ensures that the issues was created.

Step 5

Time to clean-up, otherwise your account will be filled with repositories which you don't need.

Make sure you **delete the repository** that you have created.

Add one **test** that checks the status code (expected 204).

Step 6

Try again to fetch (with GET) the repository that you have just deleted (similar to Step 2).

Add one **test** that checks the status code (expected 404 Not found).

Clear any variables that you have used in this collection.

Questions for this assignment

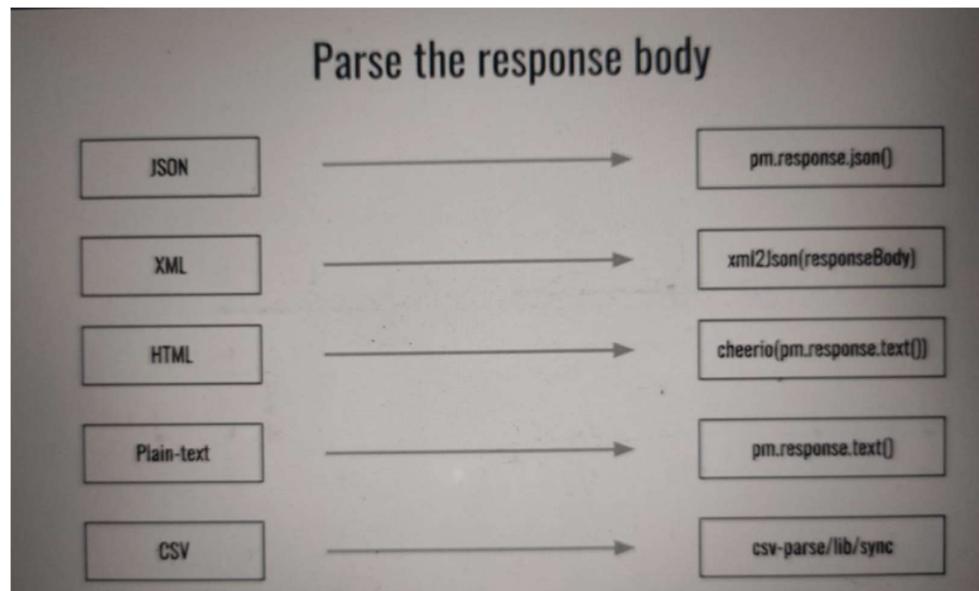
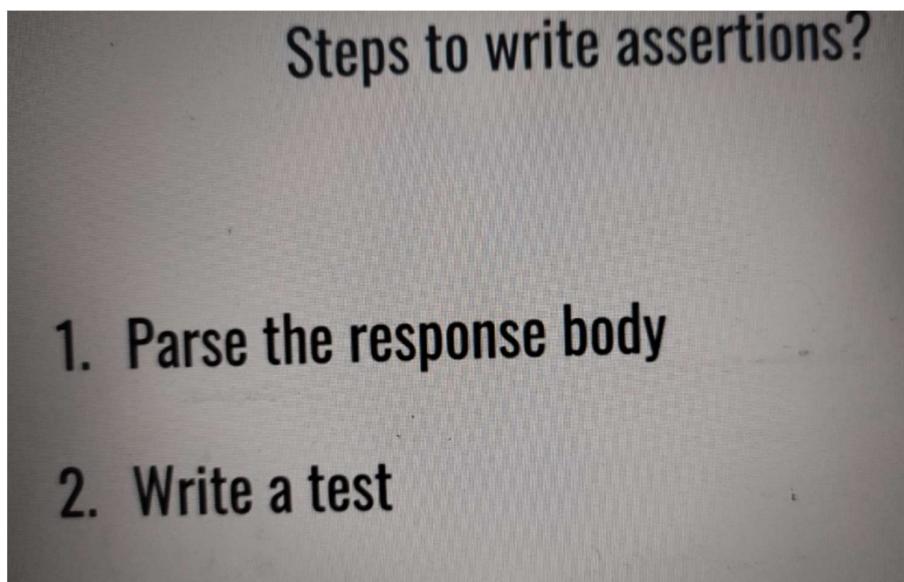
How does your collection look like?

Add here screenshots or better generate a link with your collection and use it here.

Important! Make sure you don't include any usernames / tokens / passwords.

How was this assignment?

Thank you for looking into this assignment. Please let me know how did you like it, if you feel you have learned something new. Any feedback on how to improve is welcome!



Assertion

→ Full response details print in postman console

The screenshot shows the Postman application interface. In the center, there's a 'Tests' tab with the following JavaScript code:

```
1 const response = pm.response.json();
2
3 console.log(response);
```

To the right, the 'Postman Console' window displays the response body in JSON format:

```
12:08:49.720 ✘ GET https://run.mocky.io/v3/25e2d802-a744-470d-8c60-6acb338407e9
12:08:49.746 ✘ {name: "Jane", age: 29, other: {email: "jane@example.com", social-media: ["Facebook", "Instagram"]}}
```

The 'Body' tab of the main interface shows the raw JSON response:

```
1
2   "name": "Jane",
3   "age": 29,
4   "other": {
5     "email": "jane@example.com",
6     "social-media": [
7       "Facebook",
8       "Instagram"
9     ]
10    }
```

→ In response only name is print in postman console

The screenshot shows the Postman application interface. In the center, there's a 'Tests' tab with the following JavaScript code:

```
1 const response = pm.response.json();
2
3 console.log(response.name);
```

To the right, the 'Postman Console' window displays the value 'Jane'.

The 'Body' tab of the main interface shows the raw JSON response:

```
1
2   "name": "Jane",
3   "age": 29,
4   "other": {
5     "email": "jane@example.com",
6     "social-media": [
7       "Facebook",
8       "Instagram"
9     ]
10    }
```

→ If jane is Visible then print it

The screenshot shows the Postman application interface. In the center, there's a 'Tests' tab with the following JavaScript code:

```
1 const response = pm.response.json();
2
3 console.log(response.name);
4
5 pm.test("person is Jane", () => {
6   pm.expect(response.name).to.eql("Jane");
7 });
```

To the right, the 'Postman Console' window displays the message 'PASS person is Jane'.

The 'Test Results' section at the bottom shows one result:

All	Passed	Skipped	Failed
PASS	1	0	0

The 'Body' tab of the main interface shows the raw JSON response:

```
12:32:31.350 ✘ GET https://run.mocky.io/v3/25e2d802-a744-470d-8c60-6acb338407e9
12:32:31.380 ✘ "Jane"
```

→ If Jake is visible then print it but jake is not available so that's why error will come.

The screenshot shows the Postman interface with a collection named "learn postman". A test script is being run:

```

const response = pm.response.json();
console.log(response.name);
pm.test("person is Jane", () => {
  pm.expect(response.name).to.eql("Jane");
});
  
```

The "Test Results" tab shows a single failed test case:

- All
- Passed
- Skipped
- Failed**

FAIL: person is Jane | AssertionError: expected 'Jane' to deeply equal 'Jake'

The Postman Console shows the log output:

```

12:43:22.153 ✖ GET https://run.mocky.io/v3/25e2d802-a744-470d-8c60-6acb338407e9
12:43:22.184 "Jane"
  
```

→ Monitor Collection

The screenshot shows the Postman interface with a collection named "learn postman". The left sidebar has a context menu open over the "Monitors" section, with the "Monitor collection" option highlighted.

The "Create a monitor" dialog is open, showing the configuration for the new monitor:

- Monitor name: 1 = Variable collection, variable name = URL
- Collection: 1 = Variable collection, variable name = URL
- Collection tag: CURRENT
- Environment: None
- Run this monitor: Week Timer, Every Day at 1:00 PM
- Regions: Automatically select region
- Receive email notifications for run failures and errors: satyamchamptripathi@gmail.com

The main interface shows the "Monitors" section with the newly created monitor collection listed.

The "Monitors" section displays a summary of the monitor's status:

1 = Variable collection, variable name = URL **HEALTHY**

View: Run summary Individual requests

Filter By: All Requests Type: All Run result: All US (East) Clear Filters

Region: Region

0 failed tests, 0 errors, across 1 region 01:40 PM, 13 Jan 2022

At the bottom, there is a search bar and a toolbar with various icons.

Newman

What is Newman

Newman is a command line Collection Runner for Postman.

It allows you to run and test a Postman Collection directly from the command line.

```
npm install -g newman
```

```
newman run yourcollection.json
```

NPM =Node packet manager

Newman install command = **npm install -g newman**

➔ Newman is a CLI tool which allows you to run a Postman collection directly from the command line.

NPM is the Node Package Manager. This is like a repository of projects and has knowledge of what requirements each project has. For example, you want to install Newman, but it depends on other libraries which themselves depend on other libraries. To figure out all the dependencies each project needs, you just need to have NPM installed and NPM will do this work for you.

In the next video, I have defined Newman as a dependency for my project. This means that the project I have created, cannot run without Newman. When NPM checks the packages.json file, it will figure out what it needs to install, in order for my project to run.

If you are familiar with **Docker**, there is a Docker image that has Newman and all the dependencies already installed. Just let me know so that I can point you in the right direction.

Install nodejs,NPM, NEWMAN in our Ubuntu machine

- Sudo apt-get update
- apt install nodejs
- node -version
- apt install npm
- npm install -g newman

```
root@ip-172-31-42-7:/home/ubuntu# npm install -g newman
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
/usr/local/bin/newman -> /usr/local/lib/node_modules/newman/bin/newman.js
npm WARN notsup Unsupported engine for commander@8.3.0: wanted {"node":">= 12"} (current: {"node":"10.19.0","npm":"6.14.4"})
npm WARN notsup Not compatible with your version of node/npm: commander@8.3.0

+ newman@5.3.1
added 111 packages from 167 contributors in 11.632s
root@ip-172-31-42-7:/home/ubuntu#
```

Troubleshooting Node.js / npm / Newman Problems (for Windows)

Windows - Troubleshooting Node.js / npm / Newman Problems

1. Node.js cannot be called from the command prompt

Once you have successfully installed Node and restarted your computer, when you open a command prompt terminal and type in `node -v`, you get an error and not a version like `v6.9.5`

Possible solution

In Windows, you need to set node.js folder path into system variables.

1) open Control Panel -> System and Security -> System -> Advanced System Settings -> Environment Variables

2) in "System variables" find variable `PATH` and add node.js folder path as value. Usually it is `C:\Program Files\nodejs;`. If variable doesn't exists, create it.

3) Restart your computer.

2. Newman cannot be called from the command prompt

Once you have installed newman and restarted your computer, when you open a command prompt terminal and type in `newman --version`, you get an error and not a version like `v4.1.0`

Possible solution

In Windows, you need to set the folder where node.js is storing global dependencies into system variables.

1) open Control Panel -> System and Security -> System -> Advanced System Settings -> Environment Variables

2) in "System variables" find variable `PATH` and add the folder path as value. Usually it is something like: `C:\Users\YOURUSERNAME\AppData\Roaming\npm`

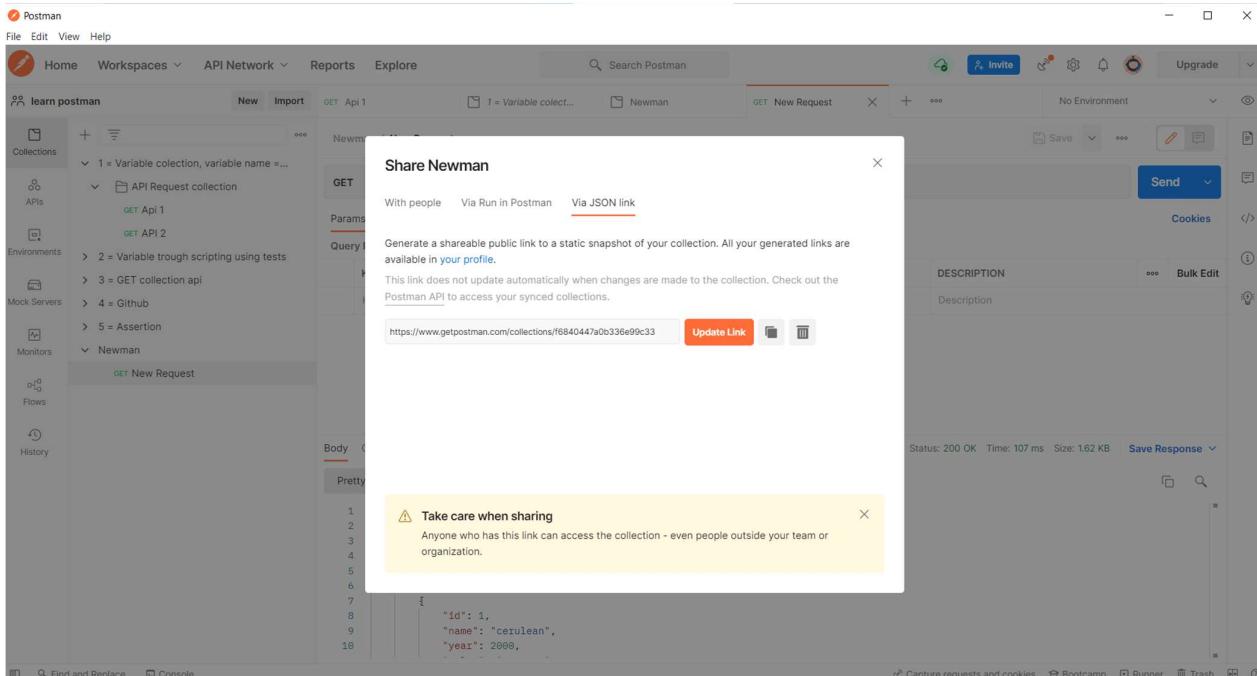
Inside the folder you should see a file called newman.

3) Restart your computer.

→ Running a collection with Newman

→ Copy link

→ Then paste in cmd before link write **newman run link**



→ Now our collection is run .

```
root@ip-172-31-42-7:/home/ubuntu# newman run https://www.getpostman.com/collections/f6840447a0b336e99c33
newman

Newman

→ New Request
GET https://reqres.in/api/unknown [200 OK, 1.65kB, 180ms]



|                    | executed | failed |
|--------------------|----------|--------|
| iterations         | 1        | 0      |
| requests           | 1        | 0      |
| test-scripts       | 0        | 0      |
| prerequest-scripts | 0        | 0      |
| assertions         | 0        | 0      |



total run duration: 247ms
total data received: 705B (approx)
average response time: 180ms [min: 180ms, max: 180ms, s.d.: 0µs]
```

```
root@ip-172-31-42-7:/home/ubuntu#
```

Running newman on Jenkins

Install Jenkins in Ubuntu virtual machine

Install java before Installing Jenkins :-

- sudo apt-get update
- sudo apt-cache search openjdk
- sudo apt-get install openjdk-8-jdk
- java -version

Install Jenkins :-

Follow this link :- <https://pkg.jenkins.io/debian-stable/>

- wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
- sudo vi /etc/apt/sources.list

inside source.list file add this line in last line .

- deb https://pkg.jenkins.io/debian-stable binary/

```
## Select root@ip-172-31-45-11:/home/ubuntu
## This repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe will NOT receive any
## security updates from the Ubuntu security team.
## deb http://archive.ubuntu.com/ubuntu/ focal universe
## deb-src http://archive.ubuntu.com/ubuntu/ focal universe
## deb http://archive.ubuntu.com/ubuntu/ focal-updates universe
## deb-src http://archive.ubuntu.com/ubuntu/ focal-updates universe
## N.B.: software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## universe WILL NOT receive any review or updates from the Ubuntu
## security team.
## deb http://archive.ubuntu.com/ubuntu/ focal multiverse
## deb-src http://archive.ubuntu.com/ubuntu/ focal multiverse
## deb http://archive.ubuntu.com/ubuntu/ focal-updates multiverse
## deb-src http://archive.ubuntu.com/ubuntu/ focal-updates multiverse
## deb http://archive.ubuntu.com/ubuntu/ focal-backports main restricted universe multiverse
## deb-src http://archive.ubuntu.com/ubuntu/ focal-backports main restricted universe multiverse
## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
## This software is not part of Ubuntu, but is offered by Canonical and the
## Canonical partners, as a service to Ubuntu users.
## deb http://archive.canonical.com/ubuntu focal partner
## deb-src http://archive.canonical.com/ubuntu focal partner
deb http://security.ubuntu.com/ubuntu focal-security main restricted
deb-src http://security.ubuntu.com/ubuntu focal-security main restricted
deb http://security.ubuntu.com/ubuntu focal-security universe
deb-src http://security.ubuntu.com/ubuntu focal-security universe
deb http://security.ubuntu.com/ubuntu focal-security multiverse
deb-src http://security.ubuntu.com/ubuntu focal-security multiverse
deb https://pkg.jenkins.io/debian-stable binary/ -- INSERT --
```

- sudo apt -get update
- sudo apt-cache search Jenkins

we use ths command to check the Jenkins version .

- sudo apt-cache madison Jenkins

```
root@ip-172-31-45-11:/home/ubuntu# sudo apt-cache madison Jenkins
jenkins | 2.319.2 | https://pkg.jenkins.io/debian-stable binary/ Packages
jenkins | 2.319.1 | https://pkg.jenkins.io/debian-stable binary/ Packages
jenkins | 2.303.3 | https://pkg.jenkins.io/debian-stable binary/ Packages
jenkins | 2.303.2 | https://pkg.jenkins.io/debian-stable binary/ Packages
jenkins | 2.303.1 | https://pkg.jenkins.io/debian-stable binary/ Packages
jenkins | 2.289.3 | https://pkg.jenkins.io/debian-stable binary/ Packages
jenkins | 2.289.2 | https://pkg.jenkins.io/debian-stable binary/ Packages
jenkins | 2.289.1 | https://pkg.jenkins.io/debian-stable binary/ Packages
jenkins | 2.277.4 | https://pkg.jenkins.io/debian-stable binary/ Packages
jenkins | 2.277.3 | https://pkg.jenkins.io/debian-stable binary/ Packages
jenkins | 2.277.2 | https://pkg.jenkins.io/debian-stable binary/ Packages
jenkins | 2.277.1 | https://pkg.jenkins.io/debian-stable binary/ Packages
jenkins | 2.263.4 | https://pkg.jenkins.io/debian-stable binary/ Packages
jenkins | 2.263.3 | https://pkg.jenkins.io/debian-stable binary/ Packages
jenkins | 2.263.2 | https://pkg.jenkins.io/debian-stable binary/ Packages
jenkins | 2.263.1 | https://pkg.jenkins.io/debian-stable binary/ Packages
```

If we want install jenkins in latest version we use this command.

→ sudo apt-get install jenkins-y

```
c:\ Select root@ip-172-31-45-11:/home/ubuntu
root@ip-172-31-45-11:/home/ubuntu# sudo apt-get install jenkins -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
daemon net-tools
The following NEW packages will be installed:
daemon jenkins net-tools
0 upgraded, 3 newly installed, 0 to remove and 32 not upgraded.
Need to get 72.1 MB of archives.
After this operation, 73.4 MB of additional disk space will be used.
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 daemon amd64 0.6.4-1build2 [96.3 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 net-tools amd64 1.60+git20180626.aebd88e-1ubuntu1 [196 kB]
Get:1 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.319.2 [71.8 MB]
Fetched 72.1 MB in 2min 1s (596 kB/s)
Selecting previously unselected package daemon.
(Reading database ... 79518 files and directories currently installed.)
Preparing to unpack .../daemon_0.6.4-1build2_amd64.deb ...
Unpacking daemon (0.6.4-1build2) ...
Selecting previously unselected package net-tools.
Preparing to unpack .../net-tools_1.60+git20180626.aebd88e-1ubuntu1_amd64.deb ...
Unpacking net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.319.2_all.deb ...
Unpacking jenkins (2.319.2) ...
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Setting up daemon (0.6.4-1build2) ...
Setting up jenkins (2.319.2) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.13) ...
root@ip-172-31-45-11:/home/ubuntu#
```

→ sudo service Jenkins status

```
c:\ Select root@ip-172-31-45-11:/home/ubuntu
root@ip-172-31-45-11:/home/ubuntu# sudo service jenkins status
● jenkins.service - LSB: Start Jenkins at boot time
  Loaded: loaded (/etc/init.d/jenkins; generated)
  Active: active (exited) since Mon 2022-01-17 08:41:30 UTC; 1h 20min ago
    Docs: man:systemd-sysv-generator(8)
   Tasks: 0 (limit: 1147)
  Memory: 0B
  CGroup: /system.slice/jenkins.service

Jan 17 08:41:28 ip-172-31-45-11 systemd[1]: Starting LSB: Start Jenkins at boot time...
Jan 17 08:41:29 ip-172-31-45-11 jenkins[6070]: Correct java version found
Jan 17 08:41:29 ip-172-31-45-11 jenkins[6070]: * Starting Jenkins Automation Server jenkins
Jan 17 08:41:29 ip-172-31-45-11 su[6103]: (to jenkins) root on none
Jan 17 08:41:29 ip-172-31-45-11 su[6103]: pam_unix(su-l:session): session opened for user jenkins by (uid=0)
Jan 17 08:41:29 ip-172-31-45-11 su[6103]: pam_unix(su-l:session): session closed for user jenkins
Jan 17 08:41:30 ip-172-31-45-11 jenkins[6070]: ...done.
Jan 17 08:41:30 ip-172-31-45-11 systemd[1]: Started LSB: Start Jenkins at boot time.
root@ip-172-31-45-11:/home/ubuntu#
```

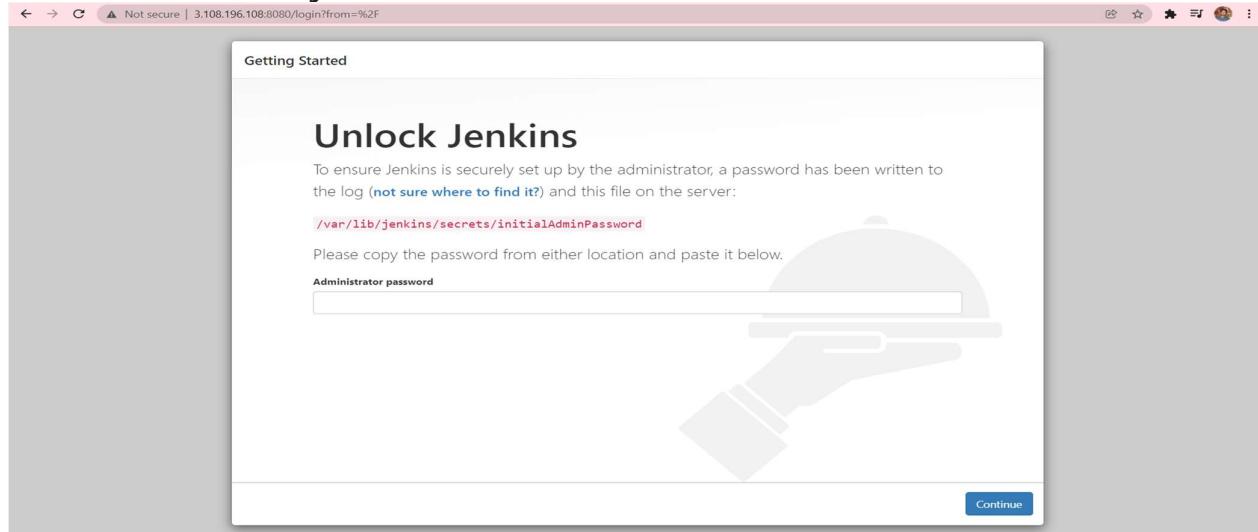
→ allow port 8080 in aws machine

□	Name	Security group rule...	IP version	Type	Protocol	Port range
□	-	sgr-050e6c4820a7eb0...	IPv6	HTTPS	TCP	443
□	-	sgr-0942f0fd4051a3dd	IPv4	Custom TCP	TCP	8080
□	-	sgr-0b07e4c28d7799c...	IPv4	HTTP	TCP	80
□	-	sgr-000eff3cd53298615	IPv4	SSH	TCP	22
□	-	sgr-0038ce7d6bcfa8572	IPv4	HTTPS	TCP	443

→ hit your public ip in browser and then put :8080

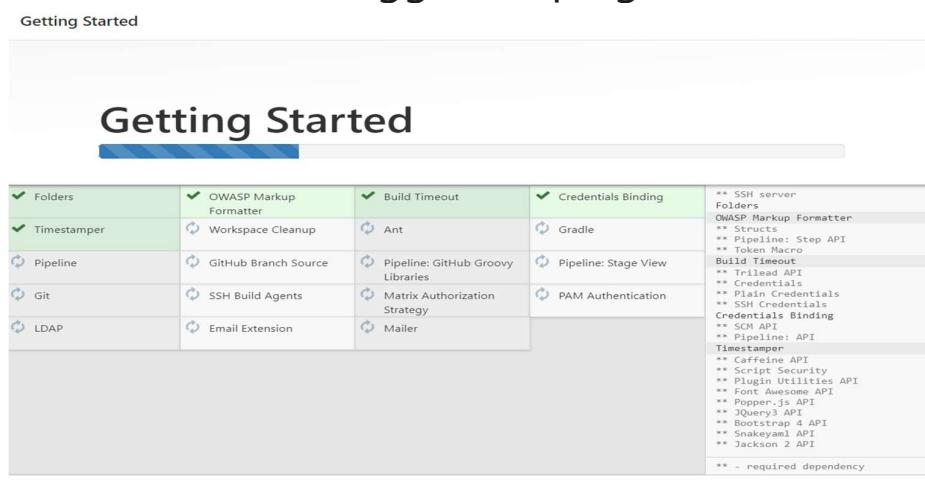
→ copy the given path and go to the terminal use this command to get your password, and copy password in the terminal then put a password here your Jenkins will ready.

→ sudo cat /var/lib/jenkins/secrets/initialAdminPassword



```
root@ip-172-31-45-11:/home/ubuntu# sudo cat /var/lib/jenkins/secrets/initialAdminPassword  
b8cd4a77a01d4b8e92dfe514e8518087  
root@ip-172-31-45-11:/home/ubuntu#
```

→ Click install suggested plugins



Create First Admin User

Username:	Satyamgii
Password:	*****
Confirm password:	*****
Full name:	Satyam Tripathi
E-mail address:	Satyamchamptripathi@gr

Jenkins 2.319.2

Skip and continue as admin

Save and Continue

- If I want to change my public ip to private ip we have a option to change it here but right now I don't want to give my private ip but you have a option to private it .

Instance Configuration

Jenkins URL:

http://3.108.196.108:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.319.2

Not now

Save and Finish

- Jenkins dashboard is ready in aws cloud

The screenshot shows the Jenkins dashboard with the following details:

- Header:** Jenkins URL: http://3.108.196.108:8080/ (Not secure)
- User Information:** Satyam Tripathi, log out
- Left Sidebar:**
 - New Item
 - People
 - Build History
 - Manage Jenkins
 - My Views
 - Lockable Resources
 - New View
- Build Queue:** No builds in the queue.
- Build Executor Status:**
 - 1 Idle
 - 2 Idle
- Welcome Message:** Welcome to Jenkins!
- 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

Before install newman install nodejs

→ Go this aws doc & install nodejs step by step :-

→ <https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/setting-up-node-on-ec2-instance.html>

Install Newman

→ npm install -g newman

→ npm -version

```
root@ip-172-31-45-11:/home/ubuntu# npm install -g newman
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated uid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.

added 111 packages, and audited 112 packages in 13s

5 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities

npm notice New patch version of npm available! 8.3.0 -> 8.3.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.3.1
npm notice Run npm install -g npm@8.3.1 to update!
npm notice
root@ip-172-31-45-11:/home/ubuntu# newman --version
5.3.1
```

→ Click new item

The screenshot shows the Jenkins dashboard at localhost:8080. The left sidebar includes links for New Item, People, Build History, Manage Jenkins, My Views, Lockable Resources, and New View. The main area displays a table of build jobs:

S	W	Name	Last Success	Last Failure	Last Duration
●	✖	Hello job	N/A	8 mo 22 days - #7	6.4 sec
○	✖	Hello-jobs	N/A	N/A	N/A
●	✖	satyam	8 mo 22 days - #4	N/A	2.2 sec
●	✖	Satyam Tripathi	8 mo 22 days - #1	N/A	20 sec

Below the table, there are sections for Build Queue (No builds in the queue) and Build Executor Status (1 Idle, 2 Idle). The bottom status bar shows the REST API and Jenkins version (2.27.3).

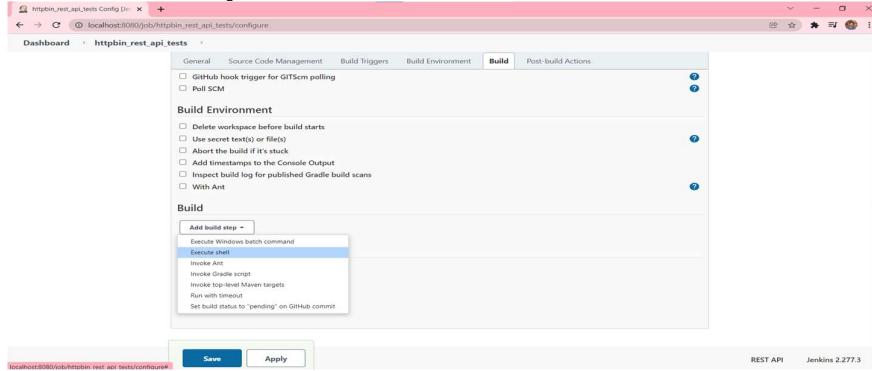
→ Create new names and click freestyle project then enter ok

The screenshot shows the Jenkins 'Enter an item name' dialog at localhost:8080/view/all/newJob. The input field contains 'HelloWorldProject'. Below the input field, there are four project types listed:

- 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.
- 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.

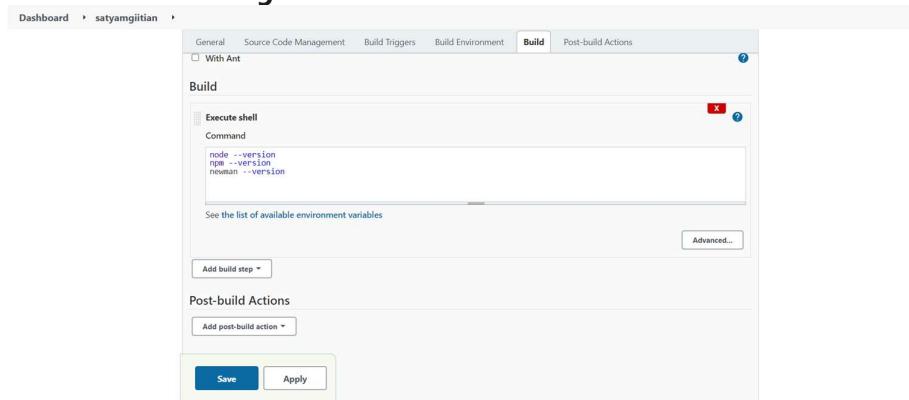
At the bottom of the dialog, there is an 'OK' button and a note about GitHub Organization.

→ Go to build option and click execute shell



→ Enter all the details to check the version.

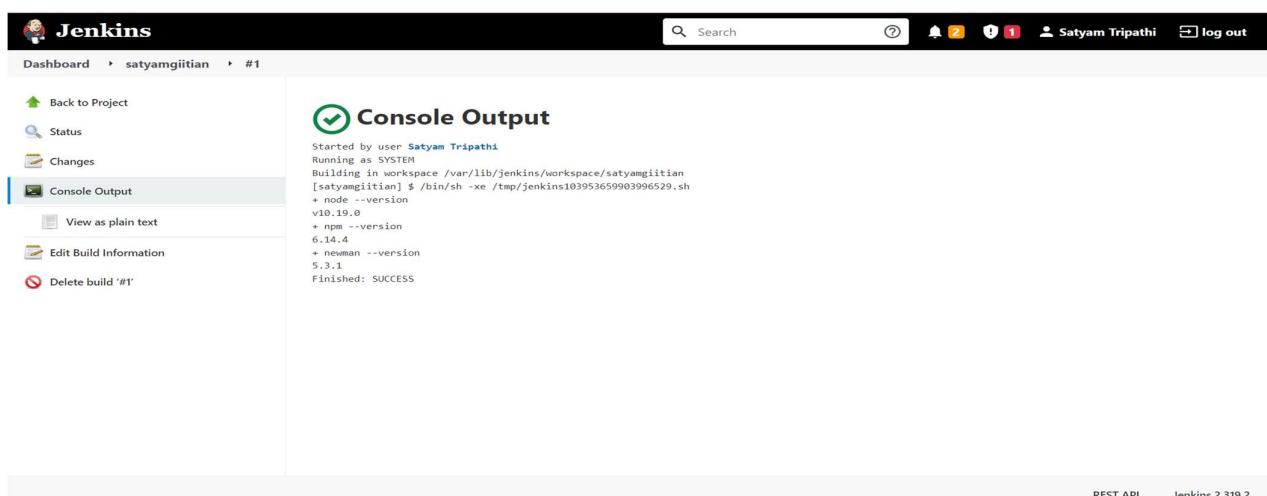
→ Click save the configuration



→ Our project has been created and now click build now, after build now click #1 to see the output .

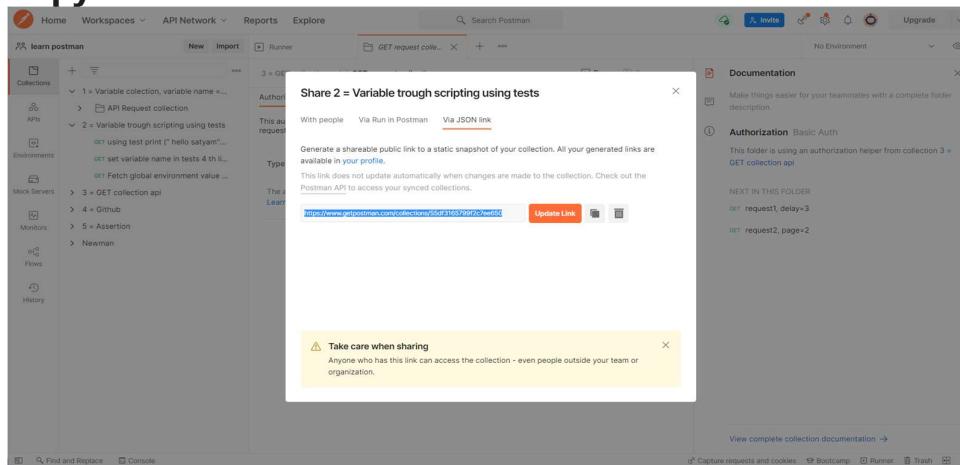


→ Open console output



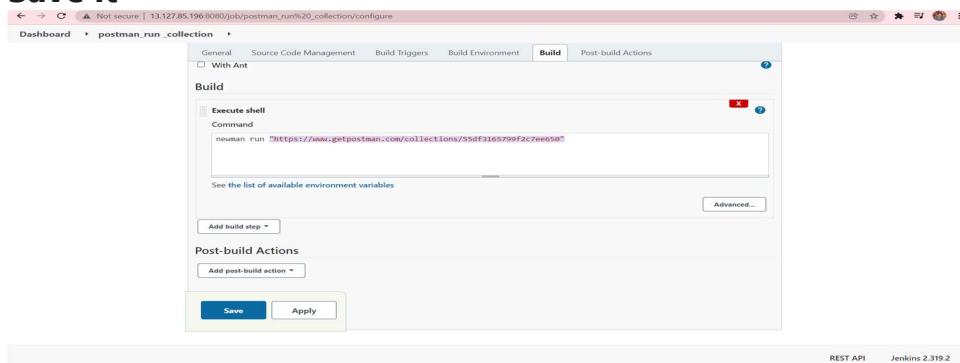
Run Postman collection using Jenkins:-

→ Copy collection link

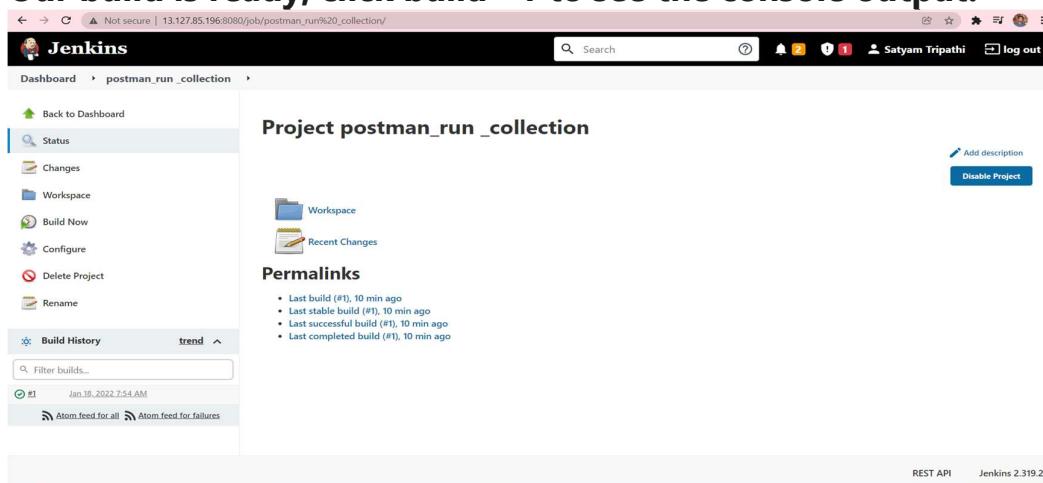


→ Create new project & paste the link before link use newman run to run the collection.

→ Save it



→ Our build is ready, click build #1 to see the console output.



← → ⚡ ▲ Not secure | 13.127.85.196:8080/job/postman_run_collection/1/console

Jenkins

Dashboard postman_run_collection #1

Back to Project Status Changes Console Output View as plain text Edit Build Information Delete build '#1'

Console Output

Started by user Satyam Tripathi
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/postman_run_collection
[postman_run_collection] \$ /bin/sh -xe /tmp/jenkins9057970715230223309.sh
+ newman run https://www.getpostman.com/collections/55df3165799f2c7ee650
newman

2 = Variable through scripting using tests

```
+ using test print (" hello satyam") show in postman console
GET https://reqres.in/api/users?page=2 [200 OK, 1.93kB, 140ms]
| 'hello satyam...'
| 'Value for url variable is : https://reqres.in/'
```

```
+ set variable name in tests 4 th line
GET https://reqres.in/api/users?page=2 [200 OK, 1.93kB, 17ms]
| 'hello satyam...'
| 'Value for url variable is : https://reqres.in/'
| 'gilt (we set gilt as a variable)'
```

```
+ Fetch global environment value in 7th line
GET https://reqres.in/api/users?page=2 [200 OK, 1.93kB, 21ms]
| 'hello satyam...'
```

```
+ Fetch global environment value in 7th line
GET https://reqres.in/api/users?page=2 [200 OK, 1.93kB, 21ms]
| 'hello satyam...'
```

```
+ Fetch global environment value in 7th line
GET https://reqres.in/api/users?page=2 [200 OK, 1.93kB, 21ms]
| 'hello satyam...'
| 'Value for url variable is : https://reqres.in/'
| 'gilt (we set gilt as a variable)'
| undefined, "AUTOMATION is the value of Environment that is declare in global Environment.Please see in postman console'
```

	executed	failed
iterations	1	0
requests	3	0
test-scripts	6	0
prerequest-scripts	3	0
assertions	0	0
total run duration:	427ms	
total data received:	3.09kB (approx)	
average response time:	59ms [min: 17ms, max: 140ms, s.d.: 57ms]	

Finished: SUCCESS

ENCL 12/24

Running a collection with Jenkins & Generating a report and html report

→ We see in add post-build option we can't see the htm

The screenshot shows the Jenkins configuration interface for a job named "Html_reports_in_jenkins". The "Build" tab is active. A context menu is open over the "Execute shell" section, listing various build actions. At the bottom left of this menu, there is a button labeled "Add post-build action".

General Source Code Management Build Triggers Build Environment **Build** Post-build Actions

With Ant ?

Build

Execute shell x ?

Command

Aggregate downstream test results
Archive the artifacts
Build other projects
Publish JUnit test result report
Record fingerprints of files to track usage
Git Publisher
E-mail Notification
Editable Email Notification
Set GitHub commit status (universal)
Set build status on GitHub commit [deprecated]
Delete workspace when build is done

Add post-build action ▾

Save **Apply**

→ npm install -g newman-reporter-htmlextra

```
ubuntu@ip-172-31-44-171:~$ sudo su
root@ip-172-31-44-171:/home/ubuntu# npm install -g newman-reporter-htmlextra
npm WARN deprecated highlight.js@0.18.5: Support has ended for 0.x series. Upgrade to @latest
npm WARN deprecated http-signature@0.12.0: REMOVED: event-stream from gulp-header
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated unix@0.1.0: Please see https://github.com/lydell/unix#deprecated
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
/usr/local/bin/newman-reporter-htmlextra -> /usr/local/lib/node_modules/newman-reporter-htmlextra/bin/htmlextra.js

> highlight.js@0.18.5 postinstall /usr/local/lib/node_modules/newman-reporter-htmlextra/node_modules/highlight.js
> node deprecated.js

Verion 9 of Highlight.js has reached EOL. It will no longer
be supported or receive security updates in the future.
Please upgrade to version 10 or encourage your indirect
dependencies to do so.

For more info:
  https://github.com/highlightjs/highlight.js/issues/2877
  https://github.com/highlightjs/highlight.js/blob/master/VERSION_10_UPGRADE.md

+ newman-reporter-htmlextra@1.22.4
added 239 packages from 480 contributors in 26.66s
root@ip-172-31-44-171:/home/ubuntu#
```

→ npm install -S newman-reporter-htmlextra

```
root@ip-172-31-44-171:/home/ubuntu# npm install -S newman-reporter-htmlextra
npm WARN deprecated highlight.js@0.18.5: Support has ended for 0.x series. Upgrade to @latest
npm WARN deprecated gulp-header@0.12.0: REMOVED: event-stream from gulp-header
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated unix@0.1.0: Please see https://github.com/lydell/unix#deprecated
/usr/local/bin/newman-reporter-htmlextra -> /usr/local/lib/node_modules/newman-reporter-htmlextra/bin/htmlextra.js

> highlight.js@0.18.5 postinstall /home/ubuntu/node_modules/highlight.js
> node deprecated.js

Verion 9 of Highlight.js has reached EOL. It will no longer
be supported or receive security updates in the future.
Please upgrade to version 10 or encourage your indirect
dependencies to do so.

For more info:
  https://github.com/highlightjs/highlight.js/issues/2877
  https://github.com/highlightjs/highlight.js/blob/master/VERSION_10_UPGRADE.md

+ newman-reporter-htmlextra@1.22.4
added 239 packages from 480 contributors and audited 239 packages in 22.679s
3 packages are looking for funding
  run `npm fund` for details
Found 1 moderate severity vulnerability
  run `npm audit fix` to fix them, or `npm audit` for details

  New major version of npm available! 6.14.4 → 8.3.1
  Changelog: https://github.com/npm/cli/releases/tag/v8.3.1
  Run npm install -g npm to update!

root@ip-172-31-44-171:/home/ubuntu#
```

→ apt install docker.io

```
root@ip-172-31-44-171:/home/ubuntu# apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base libidn11 pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
```

→ docker pull dannydainton/htmlextra

```
ubuntu@ip-172-31-44-171:~$ sudo su
root@ip-172-31-44-171:/home/ubuntu# docker pull dannydainton/htmlextra
Using default tag: latest
latest: Pulling from dannydainton/htmlextra
c548a046273e: Pull complete
143451bd116c: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:40600ff0a8c50236e1742dcaceef0e2d2f03eb312fad4c1f122c0a5e2a1bfe172
Status: Downloaded newer image for dannydainton/htmlextra:latest
docker.io/dannydainton/htmlextra:latest
root@ip-172-31-44-171:/home/ubuntu#
```

Running Newman with gitlab

Docs to install gitlab =<https://docs.gitlab.com/runner/install/linux-repository.html>

→ curl -L "https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.deb.sh" | sudo bash

```
curl -L "https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.deb.sh" | sudo bash
root@ip-172-31-44-171:/home/ubuntu# curl -L "https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.deb.sh" | sudo bash
  % Total    % Received % Xferd  Average Speed   Time     Time  Current
          Dload  Upload   Total Spent    Left  Speed
100  5945  100  5945    0     0  18520      0 --:--:-- --:--:-- 18520
Detected operating system as Ubuntu/focal.
Checking for curl...
Detected curl...
Checking for gpg...
Detected gpg...
Running apt-get update... done.
Installing apt-transport-https... done.
Installing /etc/apt/sources.list.d/runner_gitlab-runner.list...done.
Importing packagecloud gpg key... done.
Running apt-get update... done.

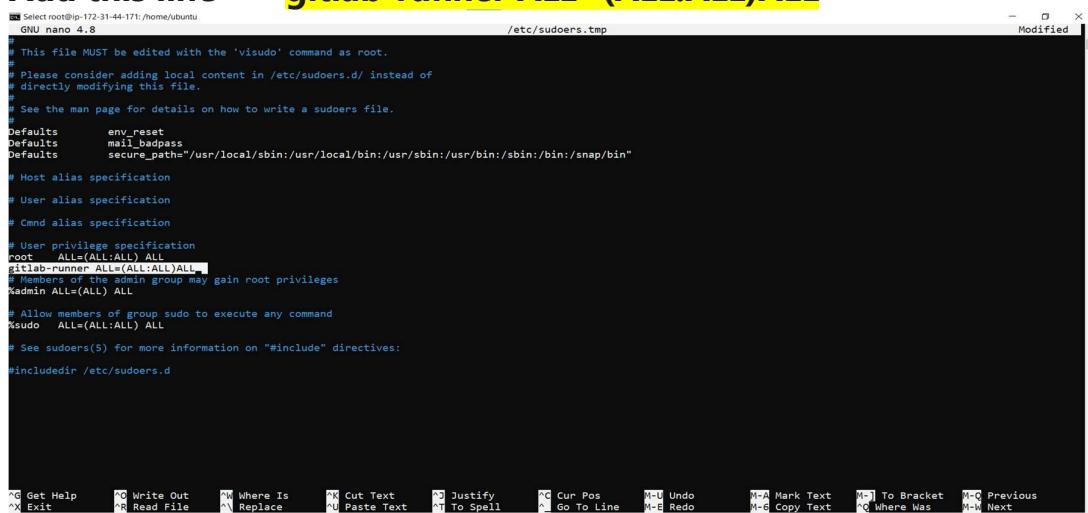
The repository is setup! You can now install packages.
root@ip-172-31-44-171:/home/ubuntu#
```

→ sudo apt-get install gitlab-runner

```
root@ip-172-31-44-171:/home/ubuntu# sudo apt-get install gitlab-runner
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  docker-engine
The following NEW packages will be installed:
  gitlab-runner
0 upgraded, 1 newly installed, 0 to remove and 15 not upgraded.
Need to get 439 MB of archives.
After this operation, 479 MB of additional disk space will be used.
Get:1 https://packages.gitlab.com/runner/gitlab-runner/ubuntu focal/main amd64 gitlab-runner amd64 14.6.0 [439 MB]
Fetched 439 MB in 10s (45.8 MB/s)
```

→ Sudo visudo

Add this line => gitlab-runner ALL=(ALL:ALL)ALL



```
root@ip-172-31-44-171:/home/ubuntu
GNU nano 4.8
/etc/sudoers.tmp
#
# This file MUST be edited with the 'visudo' command as root.
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL)ALL
gitlab-runner ALL=(ALL:ALL)ALL
# Members of the admin group may gain root privileges
admin    ALL=(ALL) ALL
# Allow members of group sudo to execute any command
sudo    ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "#include" directives:
#includein /etc/sudoers.d
```

→ sudo gitlab-runner --version

```
root@ip-172-31-44-171:/home/ubuntu# sudo gitlab-runner --version
Version:      14.6.0
Git revision: 5316d4ac
Git branch:   14-6-stable
Go version:   go1.13.8
Built:        2021-12-17T17:36:04+0000
OS/Arch:      linux/amd64
root@ip-172-31-44-171:/home/ubuntu#
```

→ sudo gitlab-runner status

```
root@ip-172-31-44-171:/home/ubuntu# sudo gitlab-runner status
Runtime platform                                arch=amd64 os=linux pid=44052 revision=5316d4ac version=14.6.0
gitlab-runner: Service is running
root@ip-172-31-44-171:/home/ubuntu#
```

→ Sudo gitlab-runner start

→ Sudo gitlab-runner stop

→ Sudo gitlab-runner restart

→ Open this

→ Copy, & enter this command to cli = sudo gitlab-runner register

The screenshot shows the 'Register a runner' section of the GitLab Docs. It includes the command `sudo gitlab-runner register` and its proxy export equivalents:

```
export HTTP_PROXY=http://yourproxyurl:3128
export HTTPS_PROXY=https://yourproxyurl:3128
sudo -E gitlab-runner register
```

→ Open this path & copy this url & paste it console .

The screenshot shows the 'Runners' settings page in the GitLab UI. It displays two sections: 'Specific runners' and 'Shared runners'. The 'Available specific runners' table shows one entry: 'my-ubuntu-runner'.



```
ubuntu@ip-172-31-44-171:~$ sudo su
root@ip-172-31-44-171:/home/ubuntu# sudo gitlab-runner register
Runtime platform                                arch=amd64 os=linux pid=44199 revision=5316d4ac version=14.6.0
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com/):
https://gitlab.com/
Enter the registration token:
bJnkm_yF-S8CLtWylkX3
Enter a description for the runner:
[ip-172-31-44-171]: my-ubuntu-runner
Enter tags for the runner (comma-separated):
ssh
Registering runner... succeeded                         runner=bJnkm_yF
Enter an executor: custom, docker, parallels, virtualbox, kubernetes, docker-ssh, shell, ssh, docker+machine, docker-ssh+machine:
shell
Runner registered successfully. Feel free to start it, but if it's running already the config should be automatically reloaded!
root@ip-172-31-44-171:/home/ubuntu#
```

→ Now , again refresh a page we see our new available specific runner details .

The screenshot shows the 'Runners' settings page in the GitLab UI after registration. It displays two sections: 'Specific runners' and 'Shared runners'. The 'Available specific runners' table shows two entries: 'my-ubuntu-runner' and 'green-shared-gitlab-org.runners-manager.gitlab.com'. The 'Available shared runners' table shows one entry: 'green-shared-gitlab-org.runners-manager.gitlab.com'.

RUN COLLECTION IN JENKINS BUT API ROUTE DIVERSION USING TEST CASE

(CREATE a basic workflow & scenario)

The screenshot shows the Postman interface with the 'Request 1(login)' tab selected. The request method is 'GET' and the URL is 'https://reqres.in/api/login'. In the 'Tests' tab, there is a single test script line: 'postman.setTextRequest("Request 1(single)").' The left sidebar shows a collection named '1 = Variable collection, variable name ...' containing 'API Request collection' and 'GET API 1'. The right sidebar contains a 'SNIPPETS' section with options like 'Get an environment variable', 'Get a global variable', etc.

1=

The screenshot shows the Postman interface with the 'Request 2(create)' tab selected. The request method is 'GET' and the URL is 'https://reqres.in/api/users'. In the 'Tests' tab, there is a single test script line: 'postman.setTextRequest("Request 2(single)").' The left sidebar shows the same collection structure as the first screenshot. The right sidebar contains a 'SNIPPETS' section with options like 'Get an environment variable', 'Get a global variable', etc.

2=

The screenshot shows the Postman interface with the 'Request 3(single)' tab selected. The request method is 'GET' and the URL is 'https://reqres.in/api/unknown/2'. In the 'Tests' tab, there is a single test script line: 'postman.setTextRequest("Request 2(create)").' The left sidebar shows the same collection structure. The right sidebar contains a 'SNIPPETS' section with options like 'Get an environment variable', 'Get a global variable', etc.

3=

The screenshot shows the Postman interface with the 'Request 4(List_Users)' tab selected. The request method is 'GET' and the URL is 'https://reqres.in/api/users?page=2'. In the 'Tests' tab, there is a single test script line: '1'. The left sidebar shows the same collection structure. The right sidebar contains a 'SNIPPETS' section with options like 'Get an environment variable', 'Get a global variable', etc.

S

Not secure | 13.127.85.196:8080/job/gitian/1/console

Dashboard > gitian > #1

BACK TO PROJECT

Console Output

Started by user Satyam Tripathi
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/gitian
[gitian] \$ /bin/sh -xe /tmp/jenkins65205538c291213638.sh
+ Newman run https://www.getpostman.com/collections/59e31cf04bbacfb4dcfaef
newman

8 = change request to another request

→ Request 1(login)
GET https://reqres.in/api/login [200 OK, 1.6kB, 195ms]

Attempting to set next request to Request 3(single)

→ Request 3(single)
GET https://reqres.in/api/unknown/2 [200 OK, 1.13kB, 16ms]

Attempting to set next request to Request 2(create)

→ Request 2(create)
GET https://reqres.in/api/users [200 OK, 1.89kB, 13ms]

	executed	failed
iterations	1	0
requests	3	0
test-scripts	3	0
prerequest-scripts	0	0

Not secure | 13.127.85.196:8080/job/gitian/1/console

Dashboard > gitian > #1

Delete build #1

Attempting to set next request to Request 3(single)

→ Request 3(single)
GET https://reqres.in/api/unknown/2 [200 OK, 1.13kB, 16ms]

Attempting to set next request to Request 2(create)

→ Request 2(create)
GET https://reqres.in/api/users [200 OK, 1.89kB, 13ms]

	executed	failed
iterations	1	0
requests	3	0
test-scripts	3	0
prerequest-scripts	0	0
assertions	0	0

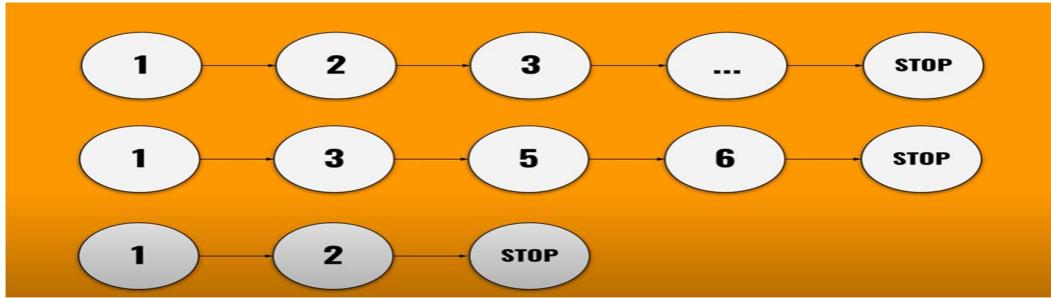
total run duration: 370ms

total data received: 1.93kB (approx)

average response time: 74ms [min: 13ms, max: 195ms, s.d.: 85ms]

Finished: SUCCESS

Advanced workflows (request chaining) with Postman

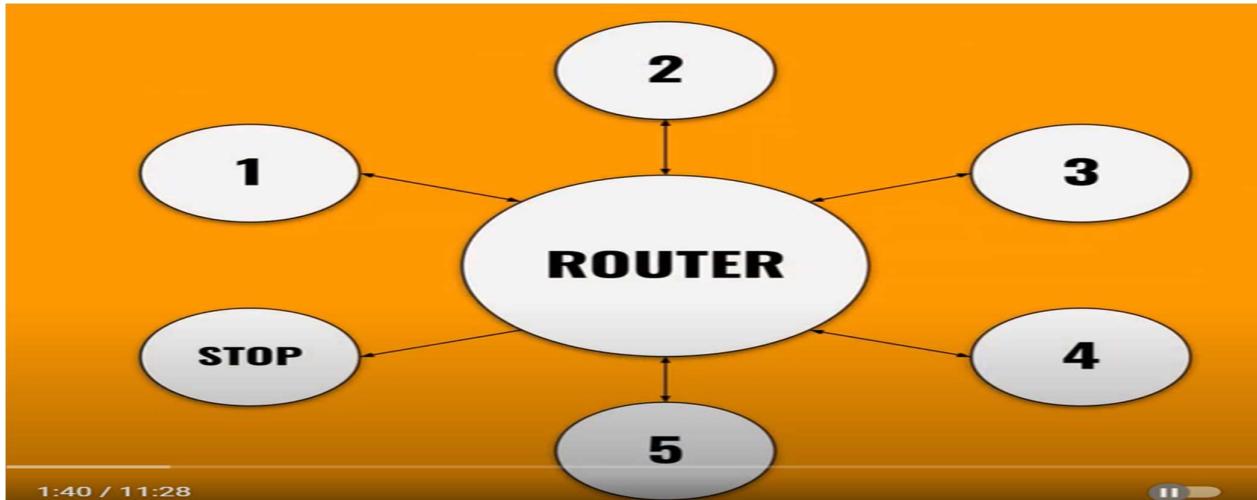


- We write this line to change the request
- For example :- I want always to go from the first request to the second one or from the second one to the fourth one from the fourth one to the third one .

postman.setNextRequest("the name of the request")

one scenario I could have used postman
set next request in

Router = Router is usually used in web application in order to contain the information that is necessary in order to navigate somewhere , for example :- to navigate from a page to another page



Convert postman request to code format

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

learn postman New Import Overview 10 = data -driven POST New Request + ***

POST https://reqres.in/api/users

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Body (Text)

```
1 {  
2   "name": "Satyam",  
3   "job": "devops"  
4 }
```

Send Cookies (

Status: 201 Created Time: 912 ms Size: 867 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {  
2   "id": "924",  
3   "createdAt": "2022-01-20T10:51:40.206Z"  
4 }
```

Find and Replace Console Capture requests and cookies Bootcamp Runner Trash

Postman

File Edit View Help

Home Workspaces API Network Reports Explore

learn postman New Import Overview 10 = data -driven POST New Request + ***

POST https://reqres.in/api/users

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies

Body (Text)

```
1 {  
2   "name": "Satyam",  
3   "job": "devops"  
4 }
```

Code snippet Python - Requests

```
1 import requests  
2 url = "https://reqres.in/api/users"  
3 payload = {"name": "Satyam",  
4            "job": "devops"}  
5 headers = {  
6            "Content-Type": 'text/plain'  
7          }  
8  
9 response = requests.request("POST", url,  
10                            headers=headers, data=payload)  
11  
12 print(response.text)
```

Status: 201 Created 912 ms 867 B Save Response

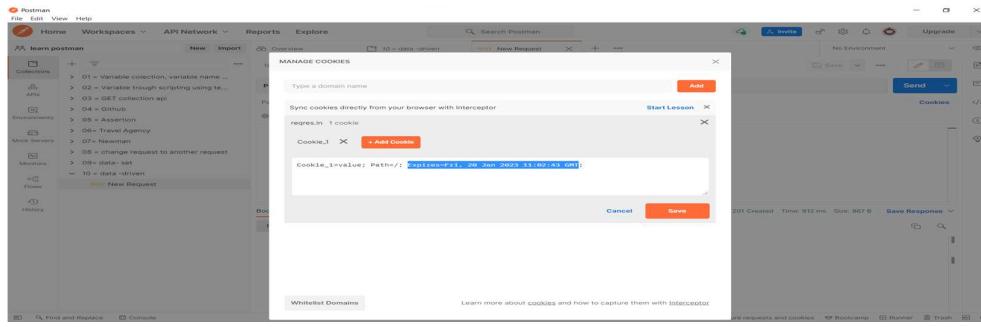
Pretty Raw Preview Visualize JSON

```
1 {  
2   "id": "924",  
3   "createdAt": "2022-01-20T10:51:40.206Z"  
4 }
```

Find and Replace Console Capture requests and cookies Bootcamp Runner Trash

Cookies

- We have an option to add cookies.
- Cookies is like a key pair value we define cookies .
- when you create a cookie after 1 cookie will expire you have an option to extend it.

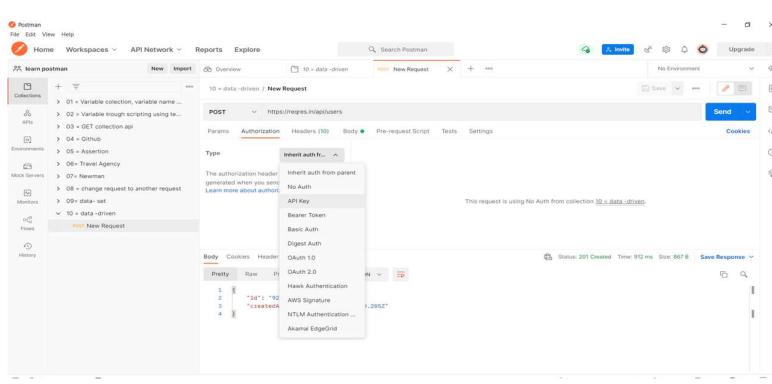


Authentication

- we mostly use bearer token & API Key

Authentication

- Basic Auth - Base64 (username and password)
- Digest authentication - User, Pass with Encryption
- Bearer Tokens - They are predominant type of access token used with OAuth 2.0. Opaque string, not intended to have any meaning to clients using it.
- JWT - Structured tokens



Postman Objects

Postman Objects

- pm.info.requestName:String
- pm.sendRequest
- pm.variables
- pm.environment
- pm.collectionVariables
- pm.globals
- pm.request
- pm.response
- pm.iterationData
- pm.cookies
- pm.cookies.jar
- pm.test
- pm.expect
- pm.response.to.be.*

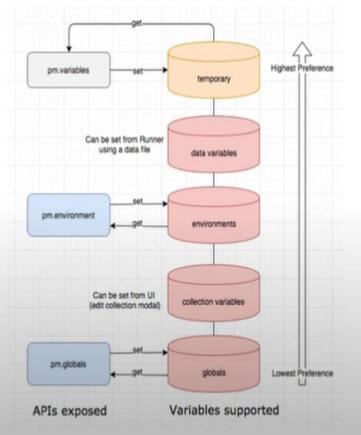
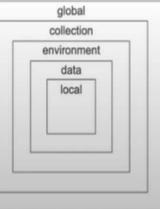
Postman Variables

POSTMAN Variables

Variable scopes

Postman supports the following variable scopes:

- Global
- Collection
- Environment
- Data
- Local



Crud = create read update delete

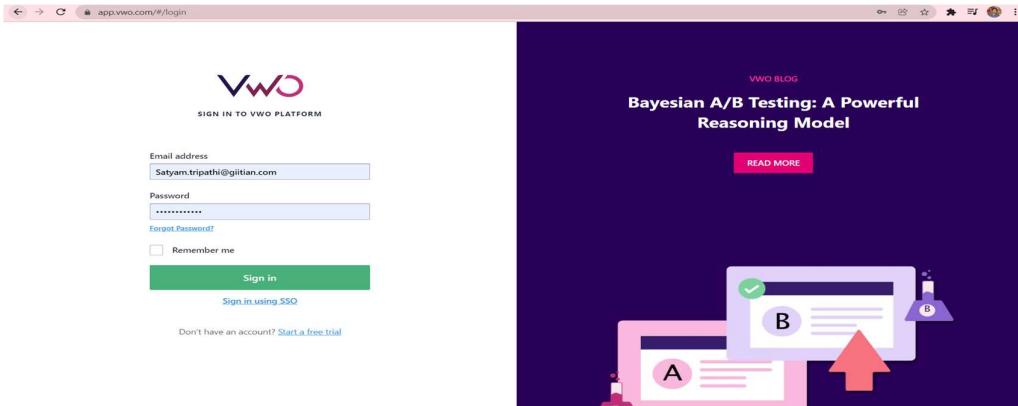
CRUD with Local Dummy Server.

- Install NodeJS.org
- Install JSON Server
 - `npm i -g json-server@v0.12.2`
- Create db.json
- Run Collection

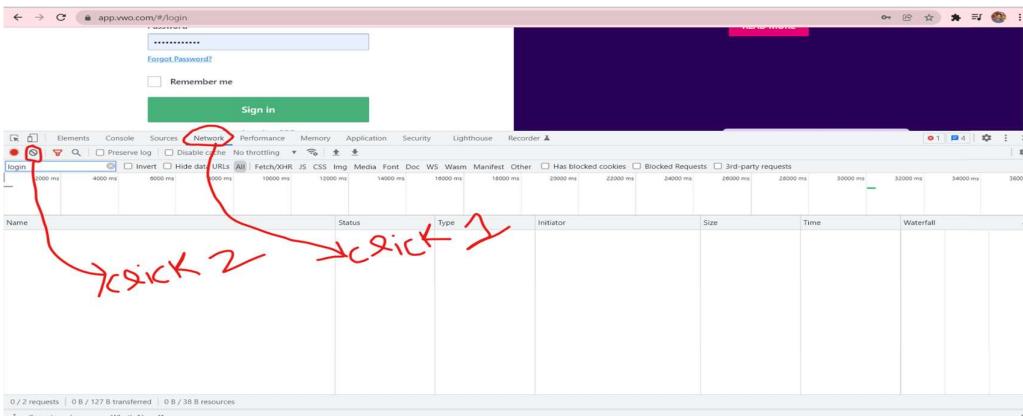
```
{  
  "person": [  
    {  
      "firstName": "John",  
      "lastName": "doe",  
      "age": 26,  
      "id": 1,  
      "address": "Chicago,USA",  
      "phoneNumbers": "123-456-78910"  
    }  
  ]  
}
```

Data-Driven

- open this :-
- <https://app.vwo.com/#/login>
- Create a account for testing



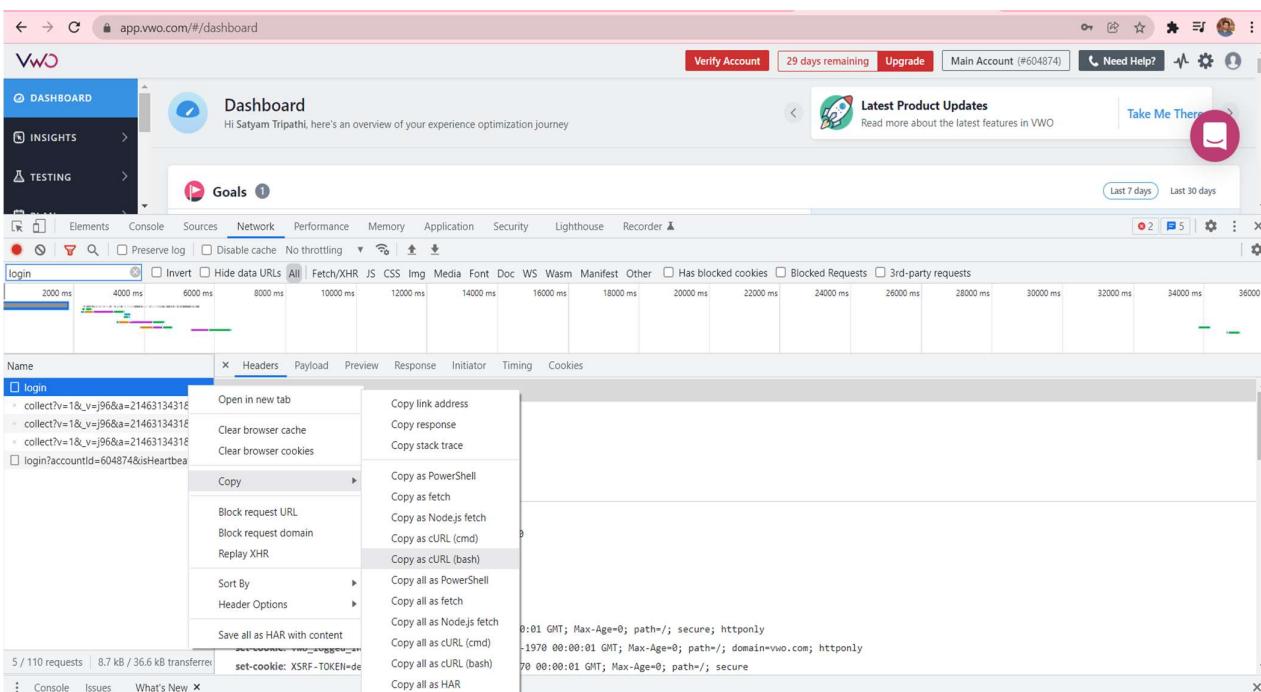
- Open inspect and go to network and clear all logs after clearing logs then sign it account.



0 / 2 requests | 0 B / 127 B transferred | 0 B / 38 B resources

Console Issues What's New

- After sign it filter login & after filtering click login left click and go to copy and copy , choose copy as cURL (bash)

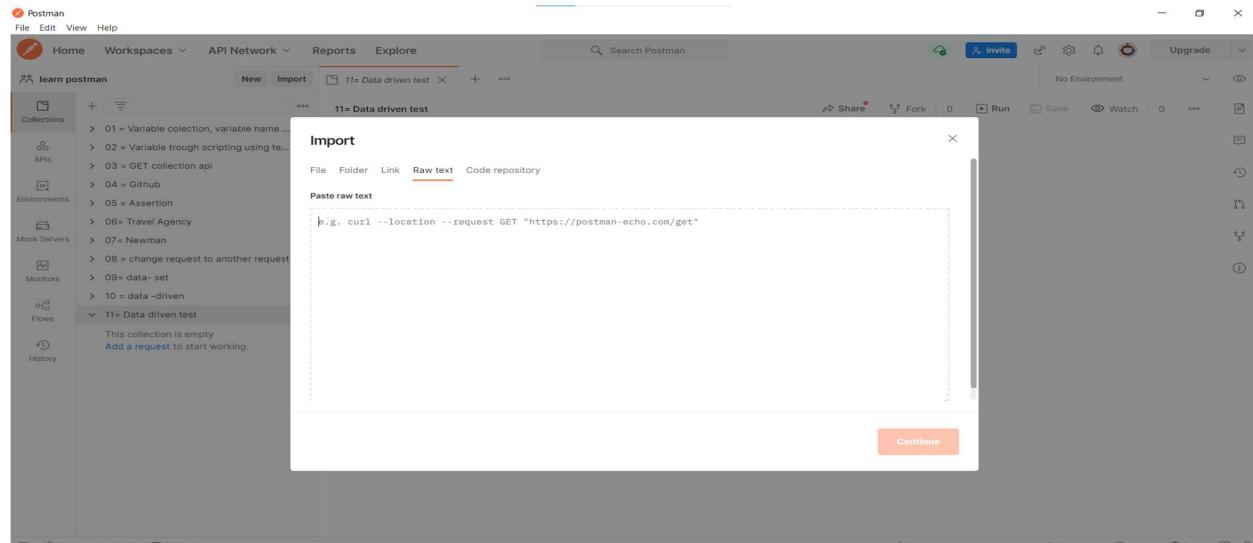


Verify Account | 29 days remaining | Upgrade | Main Account (#604874) | Need Help? | Take Me There

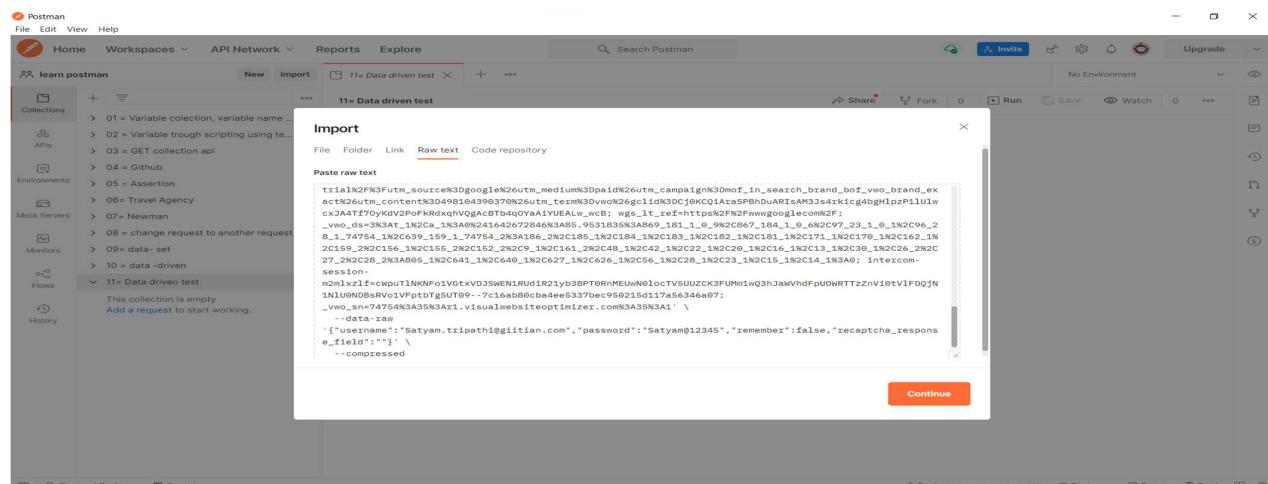
Last 7 days | Last 30 days

0:01 GHT; Max-Age=0; path=/; secure; httponly
-1970 00:00:01 GMT; Max-Age=0; path=/; domain=vwo.com; httponly
70 00:00:01 GMT; Max-Age=0; path=/; secure

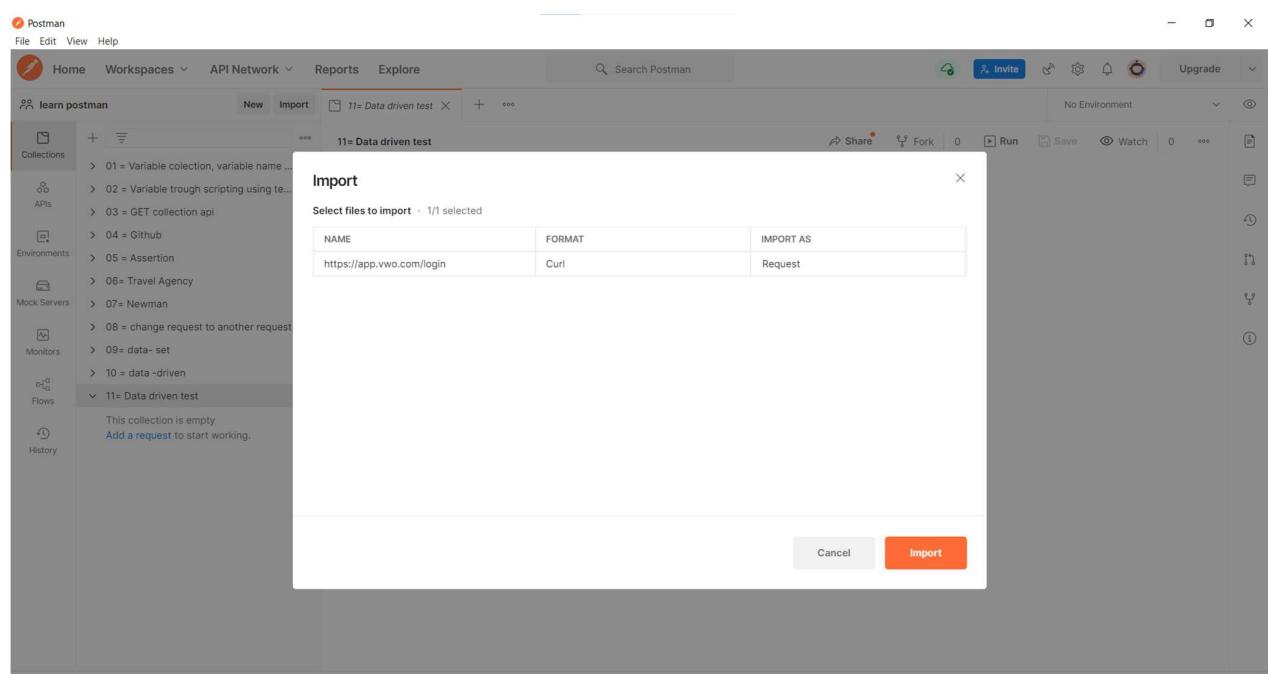
- Open postman and create a collection and select collection and then choose import option, when import option is open select raw text .



-> paste in raw text, copy as cURL (bash), then click continue



→ Click import



- Open body and then send a request you see your login credentials are visible in body
- Status code is 200 come

The screenshot shows the Postman interface with a collection named 'learn postman'. A POST request is made to <https://app.vwo.com/login>. The Body tab contains the following JSON payload:

```
{
  "username": "Satyam.tripathi@giltian.com",
  "password": "Satyam@12345",
  "remember": false,
  "recaptcha_response_field": ""
}
```

The response status is 200 OK, Time: 1994 ms, Size: 16.69 KB.

Cross checking :-

- First clear all cookies in postman tool ,because everytime when you run your request, cookies can store your data, so before running your new request clear your cookies.

The screenshot shows the 'MANAGE COOKIES' dialog in Postman. It displays a list of domains with their cookie counts: app.vwo.com (0 cookies), vwo.com (0 cookies). There is a red button labeled 'Add Cookie'.

- After clearing cookies, I change my password and send request I got invalid user.

The screenshot shows the Postman interface with the same collection and request setup as before. However, the response status is now 401 Unauthorized, Time: 1153 ms, Size: 1.11 KB. The response body is:

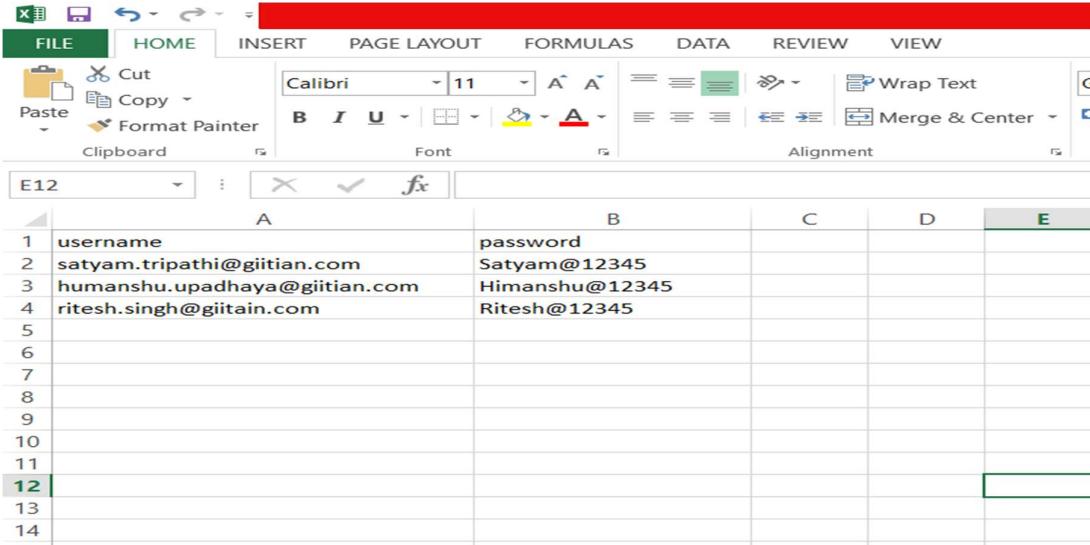
```
{
  "message": "Invalid User"
}
```

Swagger

- Swagger allows you to describe the structure of your APIs so that machines can read them
- Swagger includes automated documentation, code generation, and test-case generation.
- **Why is it so great?** Well, by reading your API's structure, we can automatically build beautiful and interactive API documentation.

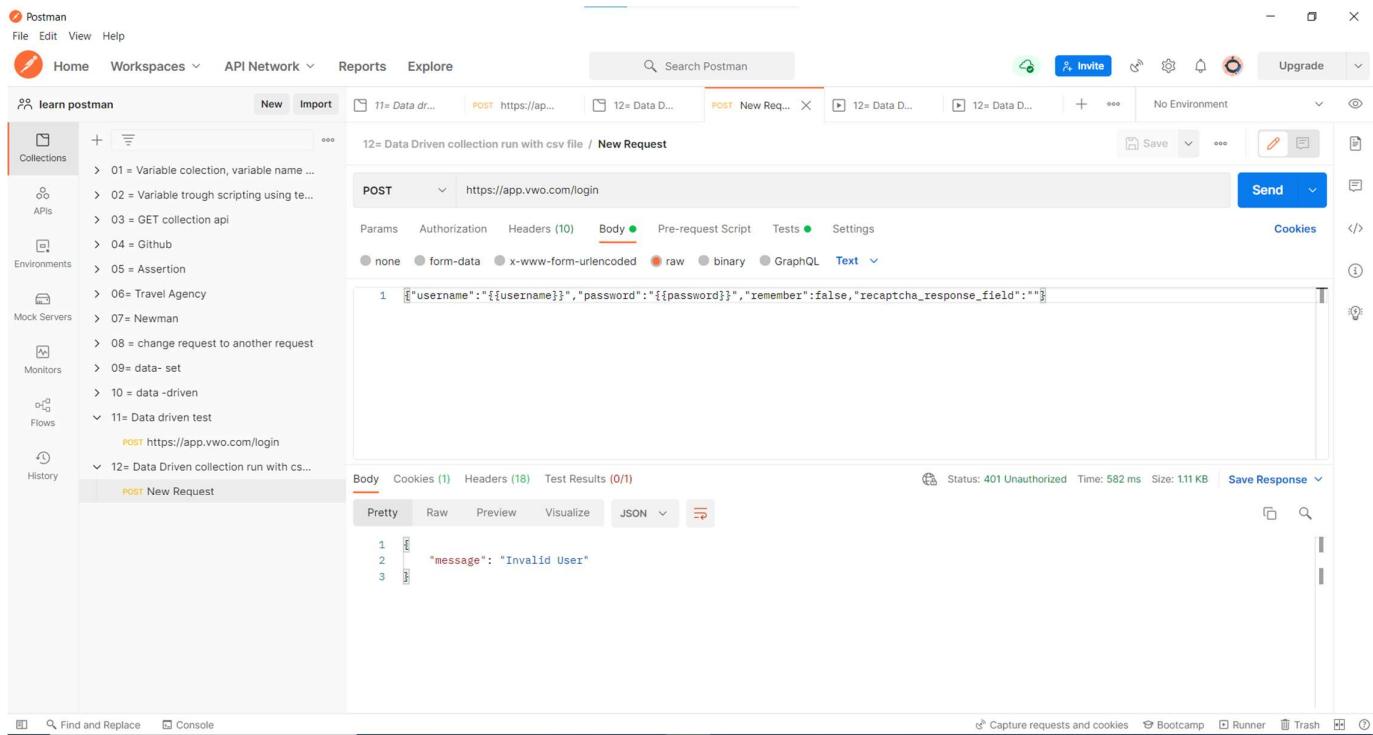
DATA Driven Collection Run using CSV file

→ Create csv file with login details.



	A	B	C	D	E
1	username	password			
2	satyam.tripathi@gitian.com	Satyam@12345			
3	humanshu.upadhaya@gitian.com	Himanshu@12345			
4	ritesh.singh@gitain.com	Ritesh@12345			
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					

→ In body we define key value pair , key is username and value is also username, it means value username is fetch in csv file and same procedure in password also.



The screenshot shows the Postman application interface. On the left, there's a sidebar with various sections like Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main area shows a 'New Request' dialog for a POST method to 'https://app.vwo.com/login'. The 'Body' tab is selected, showing a JSON template: { "username": "{{username}}", "password": "{{password}}", "remember": false, "recaptcha_response_field": "" }. Below the dialog, the 'Body' tab of the response panel shows the raw JSON response: { "message": "Invalid User" }. The status bar at the bottom indicates a 401 Unauthorized status.

→ In test case we write status code is 200

The screenshot shows the Postman interface. On the left, the sidebar lists collections, APIs, environments, mock servers, monitors, flows, and history. The main area shows a collection named "learn postman". A POST request is selected with the URL "https://app.vwo.com/login". The "Tests" tab is active, containing the following JavaScript code:

```
1 pm.test("status code is 200", function () {  
2     pm.response.to.have.status(200);  
3});
```

Below the tests, the "Body" tab is selected, showing a JSON response with a single key "message": "Invalid User". The status bar at the bottom indicates a 401 Unauthorized status.

→ Run collection and select csv file

The screenshot shows the Postman interface with the "Runner" tab selected. The "Data" field is set to "Select File" and "data1.csv". The "Data File Type" dropdown is set to "text/csv". There are several checkboxes with descriptions: "Save responses" (unchecked), "Keep variable values" (checked), "Run collection without using stored cookies" (unchecked), and "Save cookies after collection run" (checked). A blue button at the bottom right says "Run 12= Data Driven collection r...".

→ See csv file preview

PREVIEW DATA

X

Iteration	username	password
1	"satyam.tripathi@giitian.com"	"Satyam@12345"
2	"humanshu.upadhaya@giitian.com"	"Himanshu@12345"
3	"ritesh.singh@giitain.com"	"Ritesh@12345"

→ Run collection & we see the status .

The screenshot shows the Postman application interface. On the left is a sidebar with navigation links: Home, Workspaces, API Network, Reports, Explore, and a search bar. Below these are sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main area displays a collection named "12= Data Driven collection ...". It shows three iterations of a POST request to "https://app.vwo.com/login". Each iteration is labeled with a green "Pass" status and a status code of 200. The first iteration took 1737 ms and 17.094 KB. The second iteration took 468 ms and 16.295 KB. The third iteration took 445 ms and 16.295 KB.

Iteration	Request Type	URL	Status	Time	Size
1	POST	https://app.vwo.com/login	Pass	1737 ms	17.094 KB
2	POST	https://app.vwo.com/login	Pass	468 ms	16.295 KB
3	POST	https://app.vwo.com/login	Pass	445 ms	16.295 KB