

Phase-3 Practice Project: Assisted Practice

2. Demonstrate how to create the first API request.

The screenshot shows the Postman application interface. On the left, the sidebar lists collections, environments, and history. A collection named "simplilearn_aug_11" is selected, showing various requests. In the main workspace, a request to "openweathermap website / req 3" is being edited. The method is set to "GET" and the URL is "https://api.openweathermap.org/data/2.5/weather?q=karnataka&appid=2f15b0744ac5f3aa170bf81ec4803102". The "Params" tab is active, showing "q" with the value "karnataka" and "appid" with the value "2f15b0744ac5f3aa170bf81ec4803102". Below the request editor, the response pane displays the JSON data returned from the API. The status is 200 OK, and the response body is:

```
33 ],
34 "dt": 1691995201,
35 "sys": {
36   "country": "IN",
37   "sunrise": 1691973786,
38   "sunset": 1692619122
39 },
40 "timezone": 19800,
41 "id": 1267701,
42 "name": "Karnataka",
43 "cod": 200
44 }
```

4. Explain how to work on GET requests, with an example.

The screenshot shows the Postman application interface. On the left, the sidebar lists collections, environments, and history. A collection named "simplilearn_aug_11" is selected, showing various requests. In the main workspace, a request to "gorest website / Request 24" is being edited. The method is set to "GET" and the URL is "https://gorest.co.in/public/v2/users/4432068". The "Body" tab is active, showing the raw JSON payload:

```
1 {
2   "name": "meghana G P",
3   "email": "meghanapatelgp020@gmail.com",
4   "gender": "Female",
5   "status": "active"
6 }
7 }
```

Below the request editor, the response pane displays the JSON data returned from the API. The status is 200 OK, and the response body is:

```
1 {
2   "id": 4432068,
3   "name": "meghana G P",
4   "email": "meghanapatelgp020@gmail.com",
5   "gender": "female",
6   "status": "active"
7 }
```

5. Demonstrate how to work with POST requests.

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (selected), 'Environments', 'Monitors', and 'History'. The main workspace shows a list of requests under 'POST Request 3' and 'POST Request 24'. The 'POST Request 24' card is expanded, showing a POST request to 'https://gorest.co.in/public/v2/users'. The 'Body' tab is selected, displaying JSON data:

```
1 "name": "meghana G P",
2 "email": "meghanapatelgp020@gmail.com",
3 "gender": "Female",
4 "status": "active"
```

Below the body, the response status is shown as 'Status: 201 Created Time: 419 ms Size: 1.19 KB'. The bottom of the screen shows the Windows taskbar with various pinned icons like File Explorer, Task View, and Microsoft Edge.

7. Demonstrate how to create a Collection.

The screenshot shows the Postman interface with the 'Collections' sidebar selected. A context menu is open over a POST request, with the 'Save request' option highlighted. A 'SAVE REQUEST' dialog box is displayed in the center. It contains fields for 'Request name' (set to 'https://gorest.co.in/public/v2/users/4432068'), 'Add description', and 'Save to' (set to 'Reqres'). At the bottom, there are 'Save' and 'Cancel' buttons.

Postman interface showing the 'SAVE REQUEST' dialog box. The 'Request name' field contains 'https://gorest.co.in/public/v2/users/4432068'. The 'Params' section shows 'non'. The 'Body' section is set to 'Pretty'. The 'Save to' dropdown shows 'Search for collection or folder' with options like 'new collection', 'gorest website', 'openweathermap website', and 'Reqres'. At the bottom right are 'Save' and 'Cancel' buttons.

Postman interface showing the 'Request 6' screen. The request method is 'GET' and the URL is '({url})/https://gorest.co.in/public/v2/users?page=2'. The 'Params' tab is selected, showing a table with one row: 'page' with value '2'. The 'Response' section is visible at the bottom.

8. Demonstrate how to run a Collection using Collection Runner.

The screenshot shows the Postman application interface. At the top, there's a navigation bar with Home, Workspaces, and API Network options. Below the navigation is a header for the workspace "simplilearn_aug_11".

The main area displays a "Collections" sidebar on the left, which is currently expanded to show a collection named "gorest website". A context menu is open over this collection, with a blue arrow pointing to the "Run collection" option. Other visible options in the menu include Share, Move, Generate tests (BETA), Edit, Add request, Add folder, Monitor collection, Mock collection, Create a fork (Ctrl+Alt+F), Create pull request, Merge changes, Pull changes, View changelog, View documentation, Rename (Ctrl+E), Duplicate (Ctrl+D), Export, and Manage roles.

Below the collections sidebar, a list of requests is shown under "gorest website", numbered from Request 2 to Request 25. The requests are color-coded by method: GET (green), POST (orange), PUT (blue), and DELETE (red).

To the right of the requests, the "Runner" tab is selected in the navigation bar. The "Run order" section lists all 25 requests with checkboxes next to them, all of which are checked. Below this, there are sections for "Functional" and "Performance" runs, and a "Choose how to run your collection" section where "Run manually" is selected. The "Run configuration" section includes fields for Iterations (set to 1) and Delay (set to 0 ms). There are also checkboxes for Persist responses for a session and Advanced settings, and a prominent red "Run gorest website" button.

At the bottom of the interface, there's a toolbar with various icons and a status bar showing the date and time (14-08-2023, 12:47), along with weather information (27°C, Partly sunny).

Postman interface showing a collection named "simplilearn_aug_11". The left sidebar shows environments: "gorest website" (selected), "dev", "prod", "qa", and "staging". The main panel displays "gorest website - Run results" from a run today at 12:47:47. It shows 1 iteration, 13s 974ms duration, 0 failed tests, and an average response time of 464 ms. Below this, a list of requests is shown:

- GET Request 21: https://gorest.co.in/public/v2/users (200 OK, 431 ms, 1.162 KB)
- POST Request 22: https://gorest.co.in/public/v2/users (422 Unprocessable Entity, 418 ms, 1.039 KB)
- POST Request 23: https://gorest.co.in/public/v2/users (422 Unprocessable Entity, 449 ms, 1.039 KB)
- POST Request 24: https://gorest.co.in/public/v2/users (422 Unprocessable Entity, 397 ms, 1.047 KB)
- GET Request 25: https://gorest.co.in/public/v2/users (200 OK, 405 ms, 1.16 KB)

The bottom status bar shows: 27°C Partly sunny, Search, Runner, Capture requests, Cookies, Trash, 12:48, ENG IN, 14-08-2023.

11. Demonstrate how environments are used in Postman.

Postman interface showing a collection named "simplilearn_aug_11". The left sidebar shows environments: "Globals" (selected), "dev", "prod", "qa", and "staging". The main panel displays "gorest website / Request 6" with a GET method and URL {{url}}/https://gorest.co.in/public/v2/users?page=2. The "Params" tab is selected, showing a query parameter "page" with value "2". The "Response" section shows a placeholder message: "Click Send to get a response". A "Well done! You got the basics." message is displayed on the left. The bottom status bar shows: 23°C Mostly clear, Search, Runner, Capture requests, Cookies, Trash, 20:17, ENG IN, 15-08-2023.

Postman screenshot showing the 'dev' environment selected. A global variable 'url' is defined with the value 'https://dev.reqres.in'. A tooltip provides information on using variables.

Well done! You got the basics.

Postman also enables you to automate API testing, monitor API performance, mock APIs and more.

Dismiss Quickstart

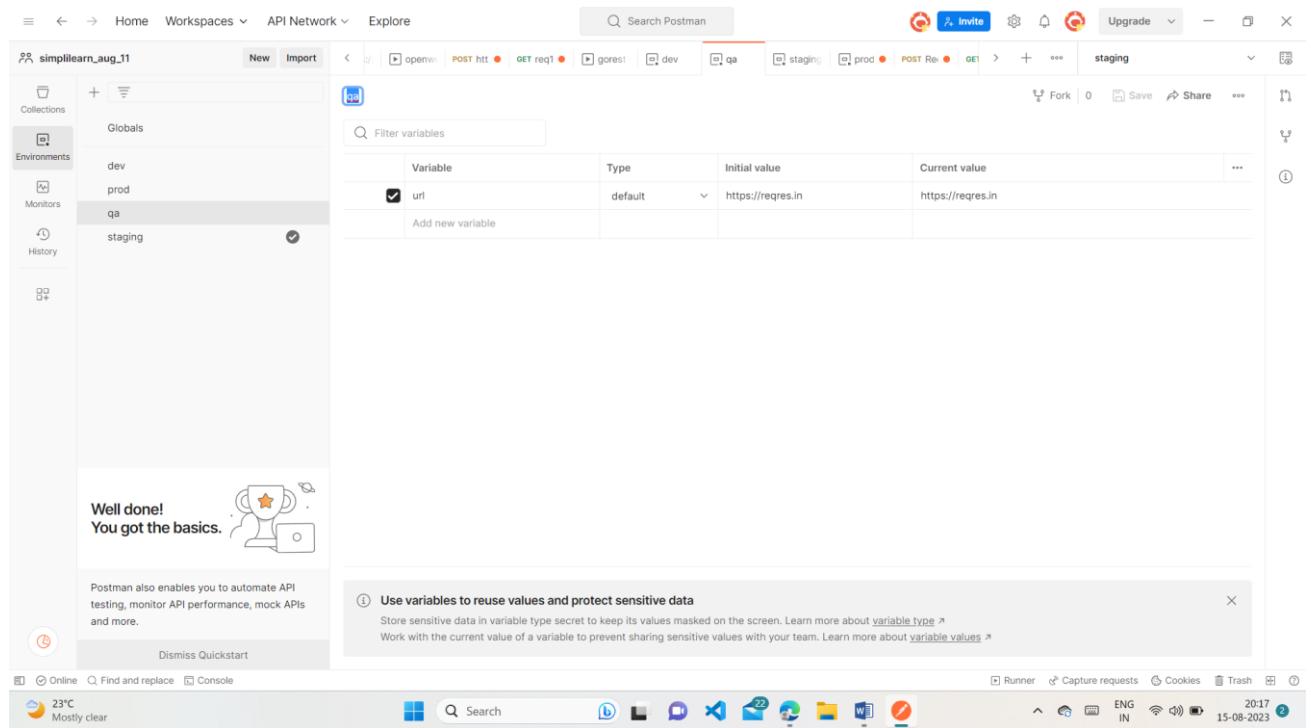
Postman screenshot showing the 'prod' environment selected. A global variable 'url' is defined with the value 'https://reqres.in'. A tooltip provides information on using variables.

Well done! You got the basics.

Postman also enables you to automate API testing, monitor API performance, mock APIs and more.

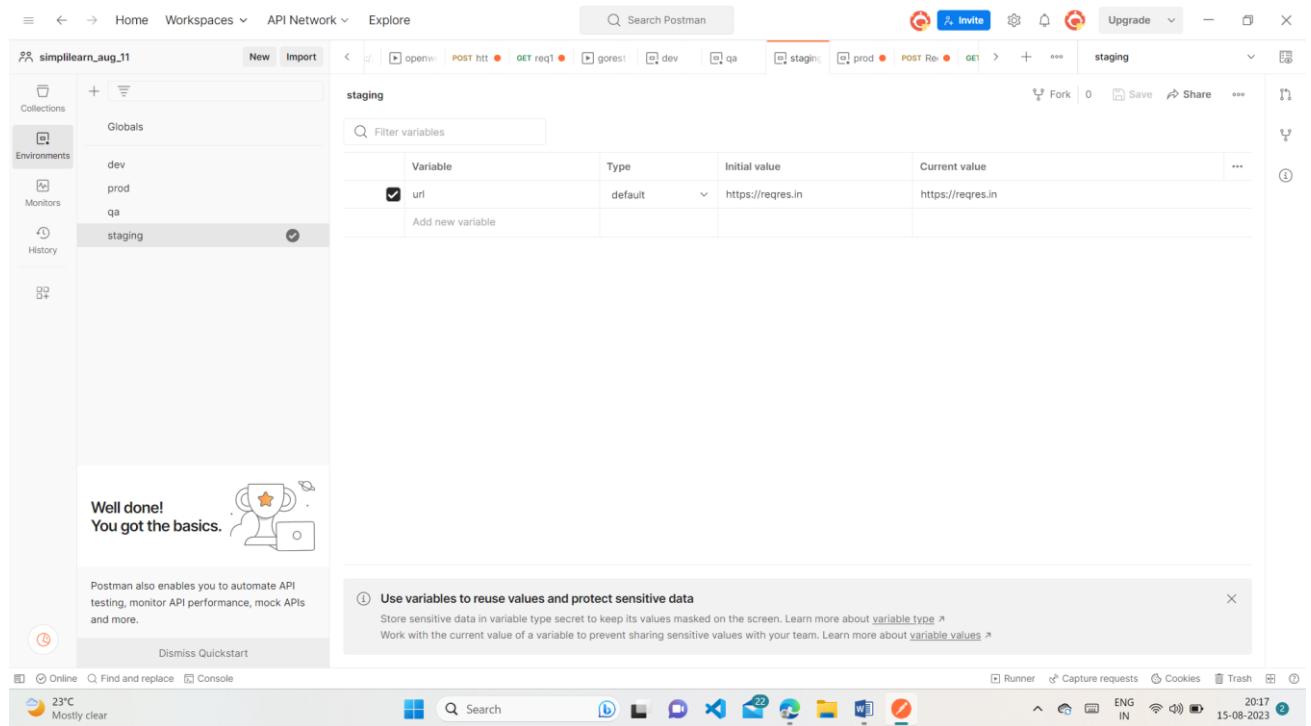
Dismiss Quickstart

Postman screenshot showing the 'Variables' section for the 'staging' environment. A variable 'url' is defined with a default value of 'https://reqres.in'. The interface includes a sidebar with collections, environments (dev, prod, qa, staging), monitors, and history. A 'Well done! You got the basics.' message is displayed. The bottom shows a Windows taskbar with various pinned icons.



	Variable	Type	Initial value	Current value
<input checked="" type="checkbox"/>	url	default	https://reqres.in	https://reqres.in
	Add new variable			

Postman screenshot showing the 'Variables' section for the 'staging' environment. A variable 'url' is defined with a default value of 'https://reqres.in'. The interface includes a sidebar with collections, environments (dev, prod, qa, staging), monitors, and history. A 'Well done! You got the basics.' message is displayed. The bottom shows a Windows taskbar with various pinned icons.



	Variable	Type	Initial value	Current value
<input checked="" type="checkbox"/>	url	default	https://reqres.in	https://reqres.in
	Add new variable			

12. Demonstrate how test scripts are written in Postman.

The screenshot shows the Postman interface with the following details:

- Header Bar:** Home, Workspaces, API Network, Explore, Search Postman, Invite, Upgrade.
- Sidebar:** Collections (empty), Environments (empty), Monitors (empty), History (empty).
- Request List:** simplilearn_aug_11 workspace, New, Import, qa, staging, prod, POST Req 1, GET Req 1, Runner, GET req 3, GET Req 4, GE1, +, No Environment.
- Current Request:** gorest website / Request 3 (POST) to https://gorest.co.in/public/v2/users.
- Test Script (Tests tab):**

```
1 var response = pm.response.json();
2 pm.globals.set("id", response.id);
3
4
5 pm.test("Response status code is 201", function () {
6   pm.response.to.have.status(201);
7 });
8
9
10 pm.test("Response has the required fields", function () {
11   const responseData = pm.response.json();
```
- Test Results:** Status: 201 Created, Time: 1217 ms, Size: 1.19 KB, Save as Example.
- Body:** Response status code is 201 (PASS), Response has the required fields (PASS), id is a non-negative integer (PASS), Email is in a valid email format (PASS), Gender is either 'male' or 'female' (PASS).

The screenshot shows the Postman interface with the following details:

- Header Bar:** Home, Workspaces, API Network, Explore, Search Postman, Invite, Upgrade.
- Sidebar:** Collections, Environments, Monitors, History, and a status indicator (27°C, partly sunny).
- Left Panel:** A tree view of collections and environments. The 'gorest website' collection is expanded, showing 14 requests (GET Request 2 is selected) and 15 others.
- Request Details:** GET Request 2 is selected. URL: https://gorest.co.in/public/v2/users/{id}.
 - Tests Tab:** Contains a JavaScript script:

```
1 pm.test("Response status code is 200", function () {  
2     pm.response.to.have.status(200);  
3 });  
4  
5 pm.test("Response has the required fields - id, name, email, gender, and status", function () {  
6     const responseData = pm.response.json();  
7  
8     pm.expect(responseData).to.be.an("object");  
9     pm.expect(responseData.id).to.exist;  
10    pm.expect(responseData.name).to.exist;  
11    pm.expect(responseData.email).to.exist;  
12    pm.expect(responseData.gender).to.exist;  
13  
14});
```
 - Test Results:** Status: 200 OK, Time: 1163 ms, Size: 1.15 KB. 5/5 tests passed.
- Right Sidebar:** Includes a 'Script with Postbot' feature (BETA), environment variables (ENG, IN, DE), and a 'Get a variable' section.

13. Demonstrate how Postman is used with Jenkins.

The screenshot shows the AWS EC2 Management Console. On the left, a sidebar lists services like EC2 Dashboard, EC2 Global View, Events, Instances (with Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store. The main pane displays a table of instances. A search bar at the top says "Find instance by attribute or tag (case-sensitive)" and has "[Alt+S]". The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. One row is selected for an instance named "postman" with Instance ID "i-095ad53135af2bcb9", which is "Running" on an "t2.micro" instance type, currently "Initializing". Its Public IPv4 DNS is "ec2-34-204-98-18.compute-1.amazonaws.com" and its Public IPv4 address is "34.204.98.18". A tooltip over the Public IPv4 address shows "Public IPv4 address copied". To the right, there are sections for Private IPv4 addresses ("172.31.32.131") and Public IPv4 DNS ("ec2-34-204-98-18.compute-1.amazonaws.com | open address").

The screenshot shows the Jenkins interface. At the top, it says "Not secure | ec2-34-204-98-18.compute-1.amazonaws.com:8080/job/poztman_demo/2/console". The main area is titled "Console Output". On the left, there are links for "Status", "Changes", "Console Output" (which is selected), "View as plain text", "Edit Build Information", "Delete build '#2'", "Git Build Data", and "Previous Build". The "Console Output" section contains the following log text:

```
Started by user meghanapatel
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/poztman_demo
The recommended git tool is: NONE
using credential 17e3fed-3610-4098-86c5-2995a69d102a
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/poztman_demo/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/meghanapatelgp/postman_demo.git # timeout=10
Fetching upstream changes from https://github.com/meghanapatelgp/postman_demo.git
> git -version # timeout=10
> git -version # 'git version 2.40.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/meghanapatelgp/postman_demo.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 7004b358bf1c492984215791cdc68ea933d4367 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 7004b358bf1c492984215791cdc68ea933d4367 # timeout=10
Commit message: "postman collection"
First time build. Skipping changelog.
Finished: SUCCESS
```

At the bottom, there is a Windows taskbar with icons for File Explorer, Task View, Mail, Edge, and others. The system tray shows the date and time as "16-08-2023 12:40 IN".

The screenshot shows the Jenkins configuration interface for a job named 'pozman_demo'. Under the 'Build Steps' section, there is a step titled 'Execute shell' with the command '/var/lib/jenkins/workspace/pozman_demo newman run openweathermap.postman_collection.json'. Below this, there is an 'Advanced' dropdown and a 'Post-build Actions' section.

Build Steps

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

Execute shell

Command

```
/var/lib/jenkins/workspace/pozman_demo
newman run openweathermap.postman_collection.json
```

Post-build Actions

Save Apply

Jenkins Build #4

Status: Build #4 (Aug 16, 2023, 6:41:57 AM)

Started 16 sec ago Took 3.2 sec

Changes: No changes.

Console Output: Started by user meghanapatel

Git Build Data: Revision: 7004b93581f1c492984215791cdc68ea933d4367 Repository: https://github.com/meghanapatelgp/postman_demo.git refs/remotes/origin/main

REST API Jenkins 2.401.3

Running Postman scripts on Jenkins

- Export the collection somewhere in your local machine and try running it using newman to check if everything is working fine.
- Push this collection to GitHub.
- Set up cloud machine by following the below steps –

1) Setting up the cloud machine

- Login into AWS Management console.
- Search for EC2 service. (Services > Compute > EC2)
- Click on Launch Instance Drop down > Select Launch Instance

- Provide a name for the new instance.
- Download key in .ppk (windows) or .pem(mac) format
- Click ‘Launch Instance’.
- Click ‘View All Instances’

2) Connecting to the cloud machine

For Window users –

- Launch putty in your machine.
- Enter public ip address of cloud machine in putty.
- Expand SSH > Expand Auth > Click Credentials > Browse the .ppk file downloaded while creating the cloud machine.
- Click Open button.
- Accept the Security Alert, a black screen will appear.
- Type ‘ec2-user’ on the black screen and hit enter.

[Run following commands using Putty on Cloud Machine](#)

1. sudo yum update -y
2. sudo wget -O /etc/yum.repos.d/jenkins.repo \https://pkg.jenkins.io/redhat-stable/jenkins.repo
3. sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
4. sudo yum install jenkins java -y
5. sudo chkconfig --add jenkins
6. sudo systemctl start jenkins
7. sudo systemctl status jenkins

Change Cloud machine’s security settings so that it can be accessed from any outer IP

1. Click on ‘Security’ tab.
2. Click on alphanumeric string under ‘Security groups’.
3. On next page, click on ‘Edit Inbound rules’.
4. Click on ‘Add’ button on next page and select following settings –
5. Click ‘Save Rules’ button.
6. Go to ‘Details’ tab again > Copy Public IPv4 DNS.
7. Open any browser, paste the DNS from last step and type ‘:8080’ at the end and hit enter.

After executing above 7 steps, we have to configure Jenkins, for that we have to read a temp password stored at a location which is shown here –

To read the password from this location, write the commands –

- On the next screen, click ‘Install Suggested Plugins’.

Configure Jenkins

1. Go to Manage Jenkins > Manage Plugins > Available tab > Search for GitHub Integration and GitHub Authentication.

Now install git in AWS instance so that we can pull the code into it. Run the following commands on Putty –

a) sudo yum install git -y

Get the path of installation of git in AWS machine by writing in Putty

a) which git

2. Go to Manage Jenkins > Tools > Update the path as shown –

Add cloud computer as a Trusted Source in GitHub

1. Write in putty –

ssh-keygen

You will see following screen –

2. The key generated in the previous step is to be read from the location specified. For that write –
sudo cat /home/ec2-user/.ssh/id_rsa.pub
3. A unique string will be printed on screen, copy that.
4. Go to GitHub and login with your credentials
5. Go to Settings > SSH and GPG Keys > New SSH Key > Paste the key copied in step 3 above > Add SSH Key
6. Go to Manage Jenkins > Tools > Under ‘JDK’ option, for ‘JDK installations’, provide any value to ‘Name’ field, check ‘install automatically’.

Postman test cases can be executed through Newman which is equivalent of running test cases through Command prompt. Here we will not be required to open Postman UI.

Installing Newman on Jenkins

1. sudo yum install -y gcc-c++ make
2. curl -sL https://rpm.nodesource.com/setup_12.x | sudo -E bash -
3. sudo yum install -y nodejs
4. sudo npm install -g newman

Executing test case on Cloud Machine

- Login into Jenkins > Go to Dashboard > New Item > Name the project > Choose FreeStyle Project > Ok

- On the next page Select GitHub project > Enter project URL
- Scroll Down to ‘Source Code Management’ > Git > Enter Repository URL
- Enter your GitHub username and Password Access Token
- Go to ‘Branches to Build’ > Update it to ‘*/main’
- Click Save and go to the Job page.
- Click ‘Build Now’, Jenkins will pull the file from Github and the job will get executed.
- Click on Console Output of the executed job > Get the location of directory in which the JSON file from Github has been pulled (/var/lib/jenkins/workspace/Postman_Demo).
- Go to Configure page of Job > ‘Build Steps’ > Choose ‘Execute Shell’ from drop-down > Write

≡ Execute shell ?

Command

See [the list of available environment variables](#)

```
cd /var/lib/jenkins/workspace/Postman_Demo
newman run ReqRes.postman_collection.json
```

- Click Save
- Click Build Now and execute the job.

14. Illustrate workspaces in Postman using an example.

The screenshot shows the Postman application interface. The left sidebar is titled 'Workspaces' and lists several workspaces: 'simplilearn_aug_11' (selected), 'gorest', 'openweathermap', and 'Reqres'. Below this, there are sections for 'Recently visited' (including 'simplilearn_aug_11' and 'simplilearn_aug_10') and 'More workspaces' (including 'API demo' and 'Team Workspace'). The main workspace area displays a request for 'https://api.openweathermap.org/data/2.5/weather?q=Bangalore&appid=a0128e93087e260d0e44fc4dd3790cc7'. The request details are as follows:

Header	Value	Description	... Bulk Edit
Value	Bangalore		
Value	a0128e93087e260d0e44fc4dd3790cc7		

Below the request details, there is a cartoon character pointing towards the 'Send' button, with the text 'Click Send to get a response'.

15. Demonstrate how monitors are used.

☰ ← → Home Workspaces v API Network v Explore Search Postman

New Import < Runner GET req 3 GET Request GET Req ● GET Req ● GET Req ● GET Req ● gorest / gorest / gorest / GET Request > + *** No Environment v

Collections Environments Monitors History

gorest website

- GET | Share
- POST | Move
- GET | Run collection
- POST | Generate tests BETA
- GET | Edit
- POST | Add request
- POST | Add folder
- DEL | Monitor collection
- DEL | Mock collection
- GET | Create a fork Ctrl+Alt+F
- POST | Create pull request
- POST | Merge changes
- GET | Pull changes
- POST | View changelog
- POST | View documentation
- GET | Rename Ctrl+E
- POST | Duplicate Ctrl+D
- POST | Export
- POST | Manage roles
- GET Request 25

HTTP gorest website / Request 12

GET https://gorest.co.in/public/v2/users/4365724

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

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description	...	

Response

Click Send to get a response

Runner Capture requests Cookies Trash

27°C Partly sunny

Windows Search

12:53 14-08-2023

Home Workspaces API Network Explore

New Import < Runner GET req 3 GET Req GET Req GET Req GET Req GET Req gorest gorest Create > + *** No Environment

Collections Environments Monitors History

gorest website

- GET Request 2
- POST Request 3
- GET Request 5
- GET Request 6
- POST Request 7
- GET Request 8
- POST Request 9
- POST Request 10
- DEL Request 11
- GET Request 12**
- DEL Request 13
- GET Request 14
- POST Request 15
- POST Request 16
- GET Request 17
- POST Request 18
- POST Request 19
- GET Request 20
- GET Request 21
- POST Request 22
- POST Request 23
- POST Request 24
- GET Request 25

Check your usage limits

Week timer Every day 12:00 PM

Regions

You can select one or more regions to monitor your requests from. Learn more

Automatically select region
 Manually select region

Receive email notifications for run failures and errors meghanapate@gmail.com

Add another recipient email

Stop notifications after 3 consecutive failures

Retry if run fails (This might affect your billing.)
 Set request timeout
 Set delay between requests
 Follow redirects
 Enable SSL validation

Cancel Create Monitor

Online Find and replace Console

27°C Partly sunny

Search

Runner Capture requests Cookies Trash

12:54 14-08-2023

Home Workspaces API Network Explore

New Import < Runner GET req 3 GET Req GET Req GET Req GET Req GET Req gorest gorest Create > + *** No Environment

Collections Environments Monitors History

gorest website

This monitor hasn't run yet

When it does, the results will appear here. You can either wait for it to run as per its schedule or trigger a run manually.

Run

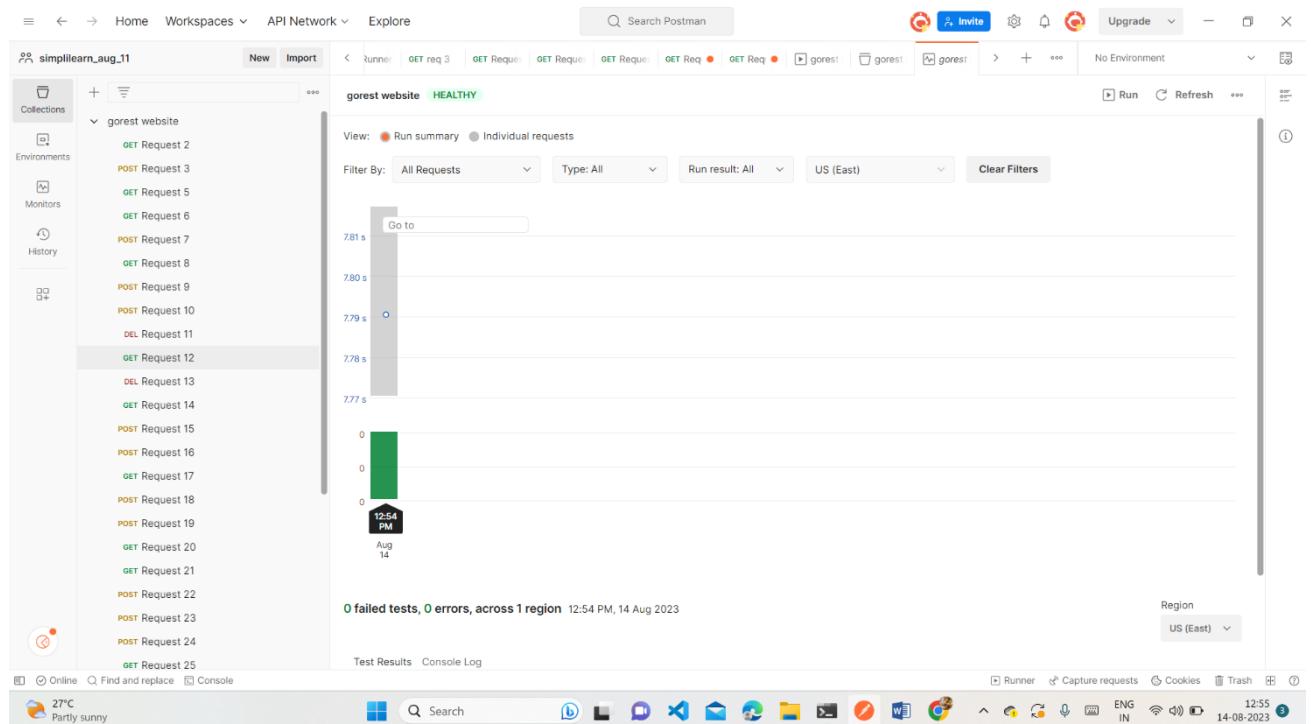
Online Find and replace Console

27°C Partly sunny

Search

Runner Capture requests Cookies Trash

12:54 14-08-2023



18. Demonstrate how to run a collection remotely with URL.

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections', 'Environments', 'Monitors', and 'History'. The main area displays a collection named 'gorest website' containing 25 requests. A modal window titled 'Share gorest website' is open, with the 'Via API' tab selected. It contains a warning about generating a key and a generated URL: https://api.postman.com/collections/29056143-a9ecb583-ec9b-42db-ac00-106fddd5680e?access_key=.... Below this, another section shows a JSON response for a collection status.

The screenshot shows a terminal window titled 'Terminal - meghanap2mphas@ip-172-31-18-231: ~/Desktop'. The user runs the command `newman run https://api.postman.com/collections/29056143-a9ecb583-ec9b-42db-ac00-106fddd5680e?access_key=PMA-T-01H7SDSPR5SX557WW9QNKCX0YW`. The terminal then lists the results for each request:

```

gorest website
- Request 2
GET https://gorest.co.in/public/v2/users/4365724 [404 Not Found, 1.03kB, 365ms]
- Request 3
POST https://gorest.co.in/public/v2/users [422 Unprocessable Entity, 1.04kB, 316ms]
- Request 5
GET https://gorest.co.in/public/v2/users [200 OK, 1.16kB, 326ms]
- Request 6
GET https://gorest.co.in/public/v2/users [200 OK, 1.16kB, 318ms]
- Request 7
POST https://gorest.co.in/public/v2/users [422 Unprocessable Entity, 1.04kB, 325ms]
- Request 8

```

```
Postman API Platform x | ChatGPT x | Practice Labs x - DEFAULT x + 
meghanap2mphasil0006.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guacadmin
Applications Terminal - meghanap2...
File Edit View Terminal Tabs Help
GET https://gorest.co.in/public/v2/users [200 OK, 1.16kB, 320ms]
- Request 9
POST https://gorest.co.in/public/v2/users [422 Unprocessable Entity, 1.04kB, 340ms]
- Request 10
POST https://gorest.co.in/public/v2/users [422 Unprocessable Entity, 1.04kB, 313ms]
- Request 11
DELETE https://gorest.co.in/public/v2/users/4365724 [404 Not Found, 1.03kB, 310ms]
- Request 12
GET https://gorest.co.in/public/v2/users/4365724 [404 Not Found, 1.03kB, 304ms]
- Request 13
DELETE https://gorest.co.in/public/v2/users/4365724 [404 Not Found, 1.03kB, 330ms]
- Request 14
GET https://gorest.co.in/public/v2/users/4365724 [404 Not Found, 1.03kB, 326ms]
- Request 15
POST https://gorest.co.in/public/v2/users [422 Unprocessable Entity, 1.05kB, 320ms]
- Request 16
POST https://gorest.co.in/public/v2/users/4365698 [404 Not Found, 4.92kB, 290ms]
- Request 17
GET https://gorest.co.in/public/v2/users/4365698 [200 OK, 1.16kB, 341ms]
- Request 18
POST https://gorest.co.in/public/v2/users [422 Unprocessable Entity, 1.04kB, 312ms]
```

```
Postman API Platform x | ChatGPT x | Practice Labs x - DEFAULT x + 
meghanap2mphasil0006.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guacadmin
Applications Terminal - meghanap2mphasil0006.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guacadmin
File Edit View Terminal Tabs Help
POST https://gorest.co.in/public/v2/users/4365698 [404 Not Found, 4.92kB, 290ms]
- Request 17
GET https://gorest.co.in/public/v2/users/4365698 [200 OK, 1.16kB, 341ms]
- Request 18
POST https://gorest.co.in/public/v2/users [422 Unprocessable Entity, 1.04kB, 312ms]
- Request 19
POST https://gorest.co.in/public/v2/users [422 Unprocessable Entity, 1.04kB, 321ms]
- Request 20
GET https://gorest.co.in/public/v2/users [200 OK, 1.16kB, 318ms]
- Request 21
GET https://gorest.co.in/public/v2/users [200 OK, 1.16kB, 327ms]
- Request 22
POST https://gorest.co.in/public/v2/users [422 Unprocessable Entity, 1.05kB, 337ms]
- Request 23
POST https://gorest.co.in/public/v2/users [422 Unprocessable Entity, 1.04kB, 333ms]
- Request 24
POST https://gorest.co.in/public/v2/users [422 Unprocessable Entity, 1.04kB, 315ms]
- Request 25
GET https://gorest.co.in/public/v2/users [200 OK, 1.16kB, 330ms]
```

```

Request 22
POST https://gorest.co.in/public/v2/users [422 Unprocessable Entity, 1.05kB, 337ms]

Request 23
POST https://gorest.co.in/public/v2/users [422 Unprocessable Entity, 1.04kB, 333ms]

Request 24
POST https://gorest.co.in/public/v2/users [422 Unprocessable Entity, 1.04kB, 315ms]

Request 25
GET https://gorest.co.in/public/v2/users [200 OK, 1.16kB, 330ms]



|                    | executed | failed |
|--------------------|----------|--------|
| iterations         | 1        | 0      |
| requests           | 23       | 0      |
| test-scripts       | 1        | 0      |
| prerequest-scripts | 0        | 0      |
| assertions         | 0        | 0      |


total run duration: 7.6s
total data received: 4.59kB (approx)
average response time: 323ms [min: 290ms, max: 365ms, s.d.: 14ms]
meghanap2mphas@ip-172-31-18-231:~/Desktop$ 

```

19. Demonstrate how to run SOAP requests in Postman.

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Shows a collection named "simplilearn_aug_11" containing various requests (POST Request 9, POST Request 10, etc.) and environments.
- Central Area:**
 - Request URL:** http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?WSDL
 - Method:** POST
 - Body Type:** XML
 - Body Content:**

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<?> <soap:Body>
<?> <CapitalCity xmlns="http://www.oorsprong.org/websamples.countryinfo">
<?> <sCountryISOCode>IN</sCountryISOCode>
<?> </CapitalCity>
<?> </soap:Body>
<?> </soap:Envelope>

```
 - Response Status:** 200 OK
 - Response Body:**

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
<?> <soap:Body>
<?> <m><CapitalCityResponse xmlns:m="http://www.oorsprong.org/websamples.countryinfo">
<?> <m>New Delhi</m></CapitalCityResponse>
<?> </m></CapitalCityResponse>
<?> </soap:Body>
<?> </soap:Envelope>

```
- Bottom Navigation:** Shows system status (26°C Partly sunny), search bar, and system icons.

Postman screenshot showing a POST request to http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?WSDL. The Headers tab is selected, showing the following configuration:

Header	Description
Content-Length	<calculated when request is sent>
Host	<calculated when request is sent>
User-Agent	PostmanRuntime/7.32.3
Accept	*/*
Accept-Encoding	gzip, deflate, br
Connection	keep-alive
Content-Type	application/soap+xml

The Body tab shows the XML response received from the API:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
3   <soap:Body>
4     <m:CapitalCityResponse xmlns:m="http://www.oorsprong.org/websamples.countryinfo">
5       <m:CapitalCityResult>New Delhi</m:CapitalCityResult>
6     </m:CapitalCityResponse>
7   </soap:Body>
8 </soap:Envelope>
```

At the bottom of the interface, there is a large blue arrow pointing to the right.