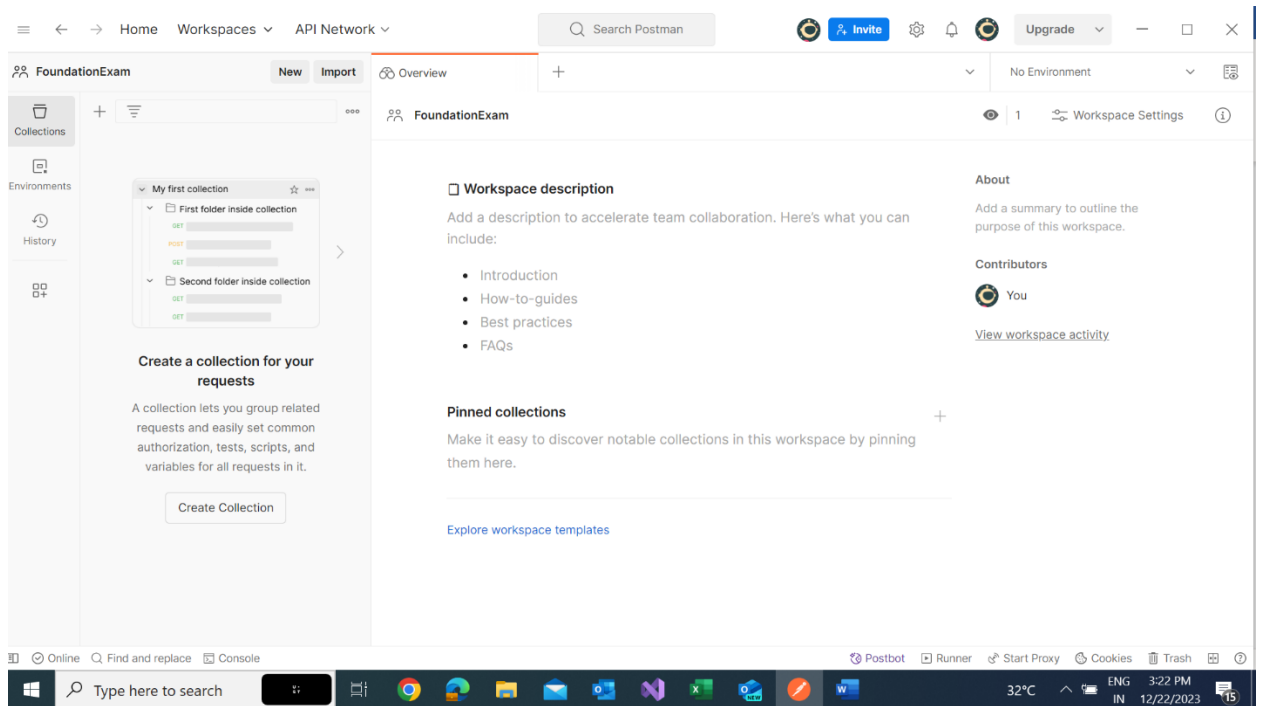
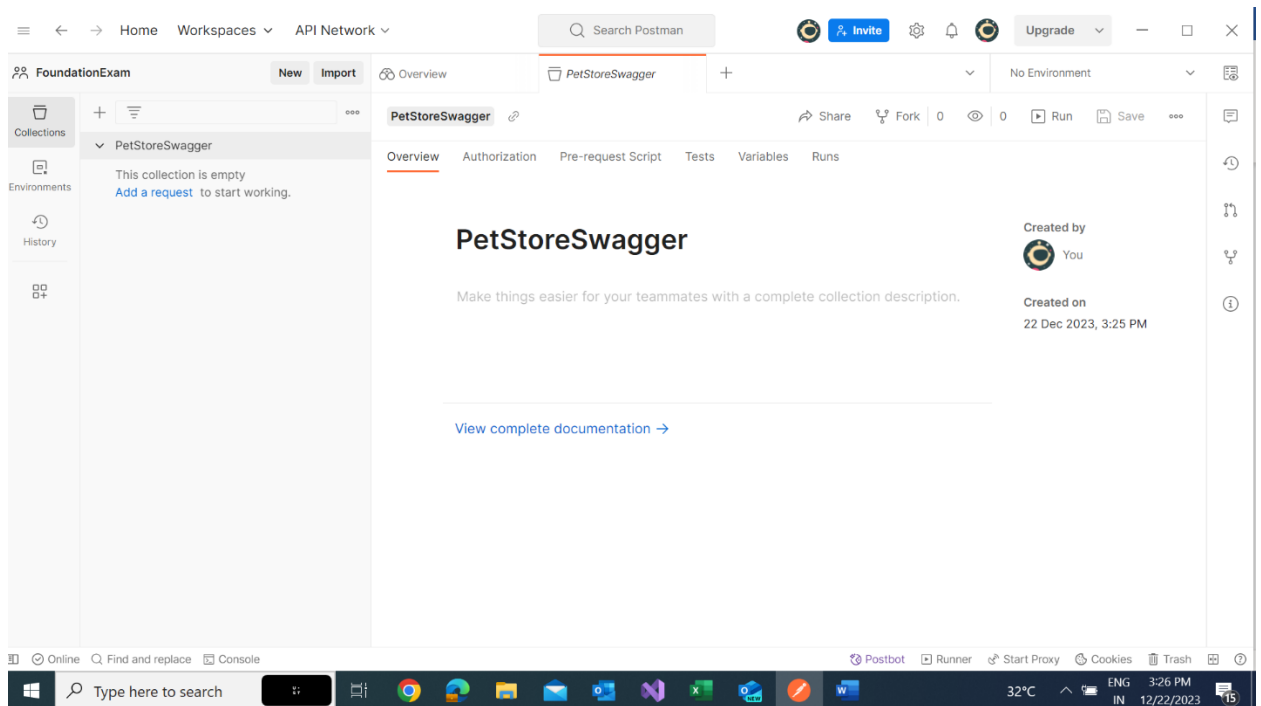


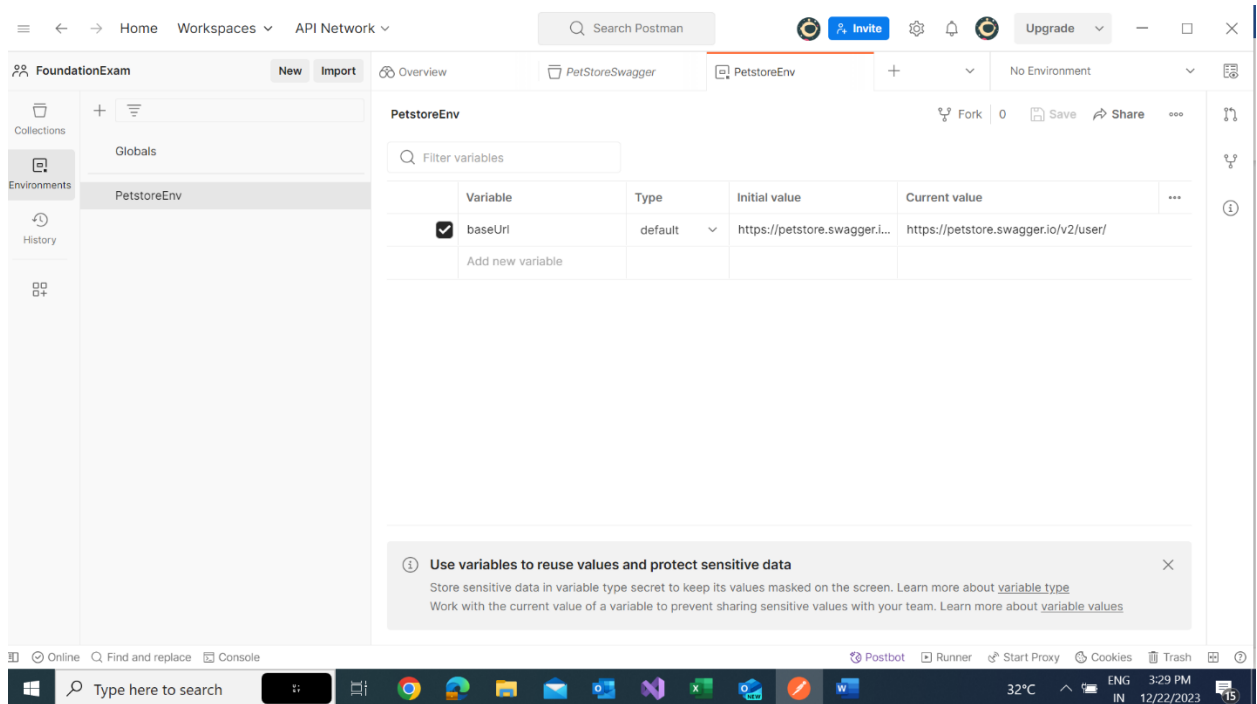
- <https://petstore.swagger.io/#/user>
- Creating Workspace



- Creating Collection

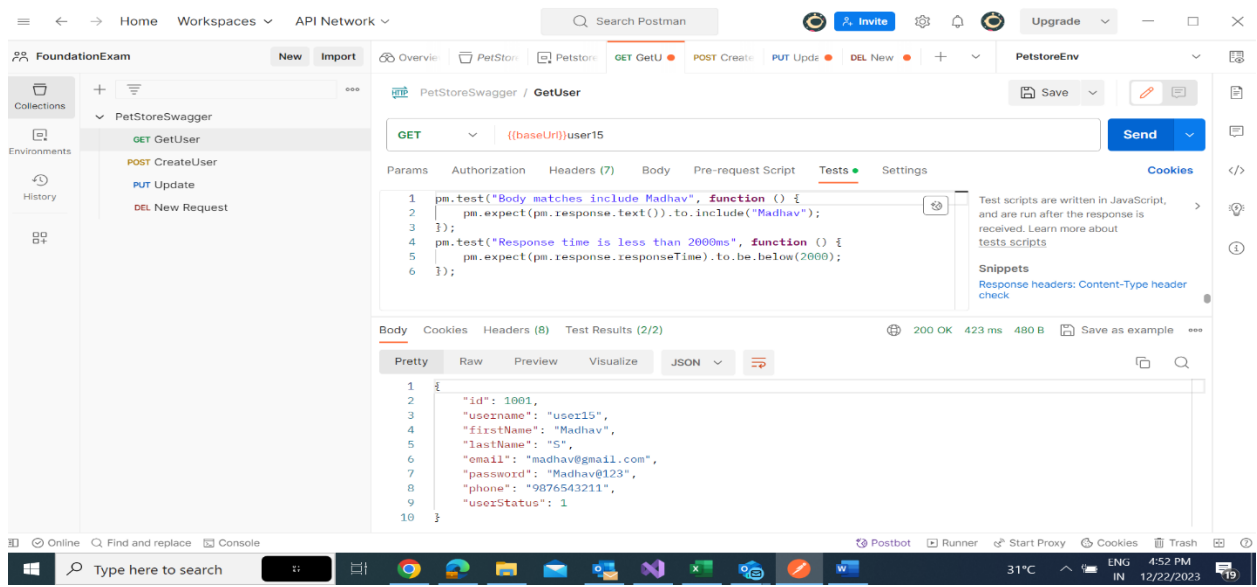


- Creating Environment



Get

- Get request with response.



- Response time less than 2000 test and Body include string test and results

The screenshot shows the Postman interface with a GET request to `((baseUrl))user15`. The request is configured with the following test scripts:

```
1 pm.test("Body matches include Madhav", function () {
2   pm.expect(pm.response.text()).to.include("Madhav");
3 });
4 pm.test("Response time is less than 2000ms", function () {
5   pm.expect(pm.response.responseTime).to.be.below(2000);
6 });
```

The test results show two passed assertions:

- PASS Body matches include Madhav
- PASS Response time is less than 2000ms

The status bar indicates a 200 OK response with 1069 ms response time and 480 B body size.

POST

- Post request and response

The screenshot shows the Postman interface with a POST request to `((baseUrl))createUser`. The request body is a JSON object:

```
1 {
2   "username": "aravind",
3   "firstName": "Aravind",
4   "lastName": "vijayan",
5   "email": "a@gmail.com",
6   "password": "qwerty",
7   "phone": "9876543210",
8   "userStatus": 1
9 }
```

The test results show a successful response with the following JSON body:

```
1 {
2   "code": 200,
3   "type": "unknown",
4   "message": "9223372836854775807"
5 }
```

The status bar indicates a 200 OK response with 1322 ms response time and 387 B body size.

- Post successful Test and results

The screenshot shows the Postman application interface. On the left, the 'Collections' pane shows 'PetStoreSwagger' with 'POST CreateUser' selected. The main panel displays the 'POST' method for 'CreateUser' with the URL '({baseUrl})'. The 'Tests' tab is active, showing a JavaScript test script:

```
1 pm.test("Successful POST request", function () {
2   pm.expect(pm.response.code).to.be.oneOf([200, 202]);
3 });
4
```

 The 'Test Results' section at the bottom shows a 'PASS' status for the 'Successful POST request'. The status bar at the bottom indicates '200 OK', '421 ms', and '387 B'.

UPDATE

- Update Request and response

The screenshot shows the Postman application interface for a PUT request. The 'PUT' method is selected for the 'Update' endpoint with the URL '({baseUrl})user1'. The 'Body' tab is active, showing a JSON payload:

```
1 {
2   "username": "amaya",
3   "firstName": "Amaya",
4   "lastName": "vijayan",
5   "email": "a@gmail.com",
6   "password": "qwerty",
7   "phone": "9876543210",
8   "userStatus": 1
9 }
```

 The 'Test Results' section at the bottom shows a '200 OK' status with a response body:

```
1 {
2   "code": 200,
3   "type": "unknown",
4   "message": "9223372036854775887"
5 }
```

 The status bar at the bottom indicates '200 OK', '1091 ms', and '387 B'.

- Type Check test and result

The screenshot shows the Postman interface with a workspace named 'FoundationExam'. The left sidebar shows a collection 'PetStoreSwagger' with endpoints: GET GetUser, POST CreateUser, and PUT Update. The main panel shows a PUT request to '((baseUrl))user1'. The 'Tests' tab is active, displaying a JavaScript test script:

```
1 pm.test("Type Check Test", function () {
2   var jsonData = pm.response.json();
3   pm.expect(jsonData.type).to.eql("unknown");
4 });
```

The 'Test Results' tab shows a single test 'Type Check Test' with a status of 'PASS'. The response status is 200 OK, with a response time of 1432 ms and a body size of 387 B. The response body is not visible in this view.

DELETE

- Delete Request with response

The screenshot shows the Postman interface with a workspace named 'FoundationExam'. The left sidebar shows a collection 'PetStoreSwagger' with endpoints: GET GetUser, POST CreateUser, PUT Update, and DEL New Request. The main panel shows a DELETE request to '((baseUrl))user1'. The 'Body' tab is active, displaying the response body in JSON format:

```
1 {
2   "code": 200,
3   "type": "unknown",
4   "message": "user1"
5 }
```

The response status is 200 OK, with a response time of 1088 ms and a body size of 373 B. The response body is displayed in the 'Body' tab, which also includes tabs for Pretty, Raw, Preview, and Visualize. The 'JSON' tab is selected.

- Responsebody message check and result.

The screenshot displays the Postman application interface. On the left sidebar, the 'Collections' tab is active, showing a collection named 'PetStoreSwagger' with several endpoints: 'GET GetUser', 'POST CreateUser', 'PUT Update', and 'DEL New Request'. The 'DEL New Request' endpoint is selected. The main panel shows the details of this request, including the method 'DELETE', the URL '({baseUrl})string', and a 'Send' button. Below the URL bar, tabs for 'Params', 'Authorization', 'Headers (7)', 'Body', 'Pre-request Script', 'Tests', and 'Settings' are visible. The 'Tests' tab is active, showing a JavaScript test script:

```
1 pm.test("Message Test", function () {
2   var jsonData = pm.response.json();
3   pm.expect(jsonData.message).to.eql("string");
4 });
```

To the right of the script, there is a sidebar with 'Test scripts' and 'Snippets' sections. The 'Test Results (1/1)' tab is active, showing a single test result: 'PASS Message Test'. The status bar at the bottom indicates '200 OK', '1023 ms', and '374 B'. The Windows taskbar at the very bottom shows the system clock as 4:09 PM on 12/22/2023.