API Testing Using Postman

KTCTC

move with confidence

Day 1 :

➢Software Application

1. Front End/UI----Front End Developer
2. Middle Layer/APIs-----Back End Developers   API-Application Programming Interface
3. Back End/DB

➢ Why API Testing ??/Importance of API Testing

Day 2 :

- **Go through from POSTMAN UI**

3 Sections on POSTMAN UI

1. Header Section
2. Sidebar Section
3. Builder Section

- **Starting With Use of POSTMAN for API Testing**

CREATE API Request → Hit API Request → Response Analysis

- **Collection :** Group of API Requests saved & arranged in logical manner
- **Folders inside collection**

Day 3:

- **Collection Runner :**

Use/Importance of Collection Runner

- **Variables in POSTMAN**

- **Variable : Elements which can store different values**
- **Use :**
  a. **To reuse same values at various places**
  b. **To avoid repetitive data**
  c. **To avoid rework [in case of changes in values/To work with different data]**

- **Variable Declaration & use in POSTMAN :**

1. Collection Level 2. Global Level 3. Environment Level

KTCTC
move with confidence

Day 4-5:

- **Environment : Set of Key-Value Pairs**

  I. How to Create? II. Use of Environment

- **Scripts in Postman**

  - Snippets : Script Templates

- **Pre-Request & Post-Request Scripts/Test**

  I. Collection Level II. Folder Level III. Request Level

- **Set & Get Variables using script**

  I. Collection Level II. Global Level III. Environment Level

Day 6:

- **Debug in POSTMAN-**Use of POSTMAN Console
- **WORKSPACES :** Area where Collections of APIs are grouped, Organized & Managed(Available in version above 6.0)
    - **Create & Manage Workspace**
- **Request Chaining in POSTMAN**

Day 7:

- **Data Driven API Testing**

I. Using CSV File

```
Key1, Key2
Value1, value2
Value1, value2
```

II. Using JSON file

```
[
 {
  "Email": “Email Value1",
  "Password": “Password value 1"
 },
{
  "Email": “Email Value2",
  "Password": “Password value 2"
 },
]
```

Day 8:

- **Run Collection from Command Line(Newman-Command line collection runner)**

1. Install node.js[node –v , npm -v]
2. Install newman [npm install –g newman & newman –v]
3. Export collection & Run from command line [newman run collection.json]

- **Execution of Collection Remotely :**

Steps :

1. Get Collection url from POSTMAN
2. Install node.js & Newman command runner
3. Run collection with command **newman run collection-url**

Day 9:

➢ **HTML Reports through Newman**

npm -v
newman –v
npm install -g newman-reporter-html
npm install -g newman-reporter-htmlextra

Using Collection URL :

1. **To Run Collection using Collection Url & generate html reports:**

newman run CollectionUrl -r html /newman run CollectionUrl -r htmlextra

2. **To Run Collection using Collection Url + Environment json file & generate html reports:**

newman run CollectionUrl -e Env.json -r html /newman run CollectionUrl -e Env.json -r htmlextra

3. **To Run Collection using Collection Url + Environment json file along with number of iterations & generate html reports:**

newman run CollectionUrl -e Env.json –n count -r html /newman run CollectionUrl -e Env.json –n count-r htmlextra

Day 9:

➢ **Using Collection Json File :**

1. **To Run Collection using Collection json file & generate html reports:**

newman run collection.json -r html /newman run collection.json -r htmlextra

2. **To Run Collection using Collection Json file + Environment json file & generate html reports:**

newman run collection.json -e Env.json -r html /newman run collection.json -e Env.json -r htmlextra

3. **To Run Collection using Collection Json File + Environment json file along with number of iterations & generate html reports:**

newman run collection.json -e Env.json –n count -r html /newman run collection.json -e Env.json –n count-r htmlextra

newman run collection.json -e Env.json –d datafile.csv/json –n count -r html /newman run collection.json -e Env.json –n count-r htmlextra

Day 9:

- **With Title to HTML Report :**

newman run collection.json -e Env.json –n count -r htmlextra --reporter-htmlextra-title “ ReqRes APIs"

Day 10:

- **Authentication v/s Authorization**

- **Authentication :** Process of Verifying who a user is

- **Authorization :** Process of verifying what user have access to

E.g. Airport System, college system, Industry Areas

- **Status Code Categories:**

1. **1XX : Informational**
2. **2XX : Success**
3. **3XX : Redirection**
4. **4XX : Client Error**
5. **5XX : Server Error**

KTCTC
move with confidence

Day 10:

➢ Most Common Status Codes :

- 200(OK) : API Request hit by client is successful
- 201(Created) : Whenever new resource data is created on Server immediately
- 202(Accepted) : Whenever creation of new data on server is accepted & require more time to complete request.
- 204(No Content) : If server declines to send back any information in response to request raised by client
- 301(Moved Permanently) : If API's resource model is redesigned & new permanent Url has been assigned to it…Subsequent requests are not allowed
- 400(Bad Request) : If Request is not as per syntax
- 401(Unauthorized) : If client tries to access protected resources without providing proper authorization
- 403(Forbidden) : If request is correctly formed but client is not having necessary permissions for it.
- 404(Not Found) : If resource is not found…..subsequent requests are allowed
- 405(Method not allowed) : If client tries to use different Request method, which resource does not allow
- 500(Internal Server Error) : Returned in case of exception handled in code/Server side issue…Client can try with subsequent request & expect different status code