**Postman**

**(Java –jar Prathap.jar –server.port=8086)**

It is an API client for develop, test, Share,

Menu Bar: File, Edit, view, and help.

Header Section: Where we have New, Runner, Import and Inspector etc..

**Environment:**

To use the properties again and again, we can create environment by giving key, value pair.

It work as a local repository.

* Go to settings icon on right
* Add environment
* Give name and key, value

(Ex: localhost, uri = <http://localhost:8080>)

* And retrieve this property by like {{key}}

(Ex: {{url}})

* We can add global variables also (by clicking on global)

**Collection:**

**What is collection?**

It is group of API requests, where that requests can be stored and saved in logical arrangement.

It forms the basis for advanced operations in postman

You can create a new collection from the:

* Sidebar
* **New** button
* Launch screen

By using collections

* We can add folders like IDE project structure
* We can share collection with other workpsaces by using **“share”** option.
* We can run our collection periodically (ex:every 1hr, 2hr) by using **Monitor collections.**
* **We can select authorization**
* **Pre-requites scripts**
* **We can add Tests**: These tests will execute after every request in this collection

**How to run collection?**

* We can run all scripts in this collection at a time by using **“Run”** option
* While running this scripts **we can add environment**, where we have to run
* We can add **iterations**, how many times you can run scripts.
* We can get log response for fail or success results.
* After run we get **Run summery,** it’s a high-level summery of pass or fail scripts

**Variables:**

These are elements (data-store) that can take different values.

* To reuse values at multiple places
* To avoid repetition
* To avoid rework when values changes
* We can create variable in multiple locations
* For ex:
  + Collection level
  + Environment level

**Set and Get Variables through Script:**

* Goto Request
* Click on Tests

// GET methods

let URI = pm.variable.get("url");

console.log("Actual Url is:"+URI);

// SET methods

pm.variable.set("Name","Prathap");

console.log( pm.variable.set("Name"));

**Set and get variables at global level:**

let GlobalVar=pm.globals.get("Name");

console.log("Global value: "+GlobalVar);

//set

pm.globals.set("Age2","28");

console.log(pm.globals.set("Age"));

**PostmanConsol:**

* Go to view
* Click on postman console

By using this, we get logs, when we are running scripts

**Script Snippets:**

Snippets – Script templates

Scripts are two types

* Pre-requist script: These tests will execute befor every request in this collection
  + This script we can use when dynamically changing values (for ex: dynamically changing headers)
* Test Script : These tests will execute after every request in this collection
* We can create scripts in multiple levels
  + Collections
  + Folder
  + Request

**Creating Test:**

Postman tests are JavaScript code that is executed after receiving the response from server.

* By using snippets we can create test easily

**Debug:**

We have to debug the scripts by using Postman CONSOL (alt+ctrl+c)

App Debug Logs: View>> developer>>Developertools

**RunFrom Command Line(NewMan):**

How to run postman requests/tests from command line

1. Install node.js

2. Install Newman

3. Export collection and run from command line

**Newman is a command line collection runner for Postman**

**Step 1:** check if node.js is already installed cmd node -v npm -v

**Step 2:** Install node.js [https://nodejs.org/en/download/](https://www.youtube.com/redirect?q=https%3A%2F%2Fnodejs.org%2Fen%2Fdownload%2F&redir_token=OZGsnc5mOt6DLtT-i3SzXpn7tAN8MTUzMTcxNDc5MUAxNTMxNjI4Mzkx&v=f2yMmOGZU7M&event=video_description) Check if node and npm are installed

**Step 3:** Install Newman

* **npm install -g newman**

**Step 4:** Export collection as json file to your localsystem(ex: create afolder in desktop and export to there file name: Student1.postman\_collection.json)

**Step 5:** On cmd goto location of above file (Student1.postman\_collection.json)

**Step 6:** Run command

* **newman run filename.json (Student1.postman\_collection.json)**

**Step 7:** To get all options from newman

* **newman run –h**

**Postman With Jenkins:**

* Start Jenkins
* Select freestyle project
* Goto Build click on add build step
* Select Execute windows batch command
* Then give location of script (cd C:\Users\RITHWIK\Desktop\Postman)
* Give newman run command (newman run Student1.postman\_collection.json)

**Work Space:**

Work Space is an area where you can group, organize and manage your collections.

* It available after postman 6.0 versions
* You can create no. of workspaces
* If you press on all workspaces, you can see all details in browser of all your workspaces.
* We can add collections and environments from one workspace to another workspace by clicking “Add to workpsace”.
* We can delete work space

**Data Driven Testing:**

We can get data from variables as well as from external files.

Variables:

* We can create variable data either environment or collection level
* We can call those data by referring key like {{key}} in request.

External file:

* We can use data from CSV or JSON files.
* Create filrs or getfiles
* Make sure Files format is correct
* Goto Run at collection
* Select folder, which is going to run
* Select file and preview
* Then run

**Collections Run remotely:**

* Get Collection url
  + Go to Collection
  + Click on share Collection
  + Click on GetLink
  + Copy Link
  + Goto Commandprompt
  + Newman run above link(url) (Ex: newman run https://www.getpostman.com/collections/657ccbb4932d1dcfb4a3)
* We can run in remote machine also with same url

**OPTION Method:**

It will show you in headers section what methods the url will allows.

Ex: {{url}}/student --- **Allow →**POST

{{url}}/student/list --- **Allow →**PUT,DELETE,GET,HEAD,PATCH

**HEAD Method:**

It is same like get method but there is no response body. But it give Headers.

**How to Run SOAP Requests:**

* Create Collection
* Create Request
* Get WSDL
* Goto body in reqest
* Select body as “raw”
* Select text/xml
* Provide request data in body
* Goto Soapui or thirdparty plugins(ex:wizdler)
* Get Request Body
* Enter Data or Variables from global
* We can run in Cmd also like rest services.