

**Practical 8**  
**Software Testing and Quality Assurance**  
**2CSDE80**

**Mistry Unnat**  
**20BCE515**

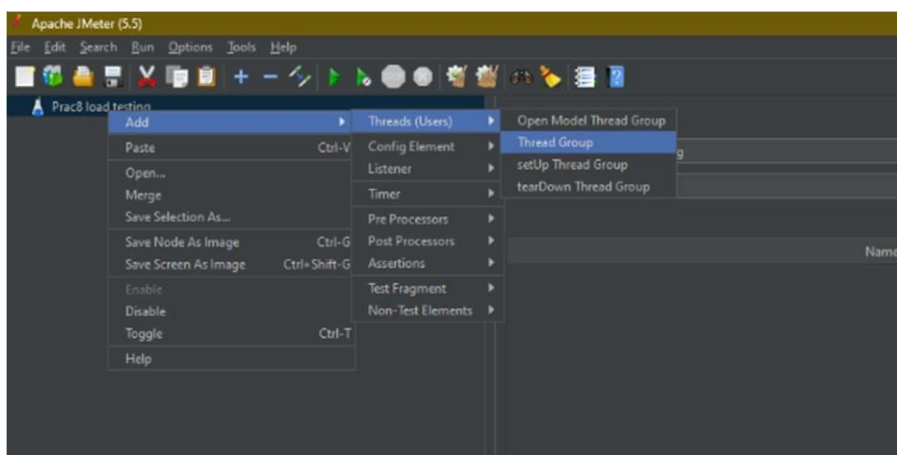
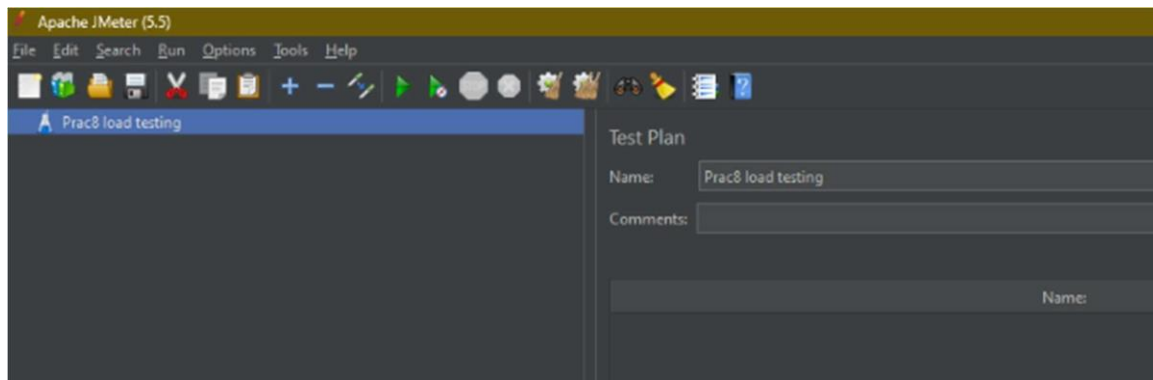


Department of Computer Science and Engineering  
Institute of Technology  
Nirma University  
Ahmedabad

**AIM :** To implement load testing method.

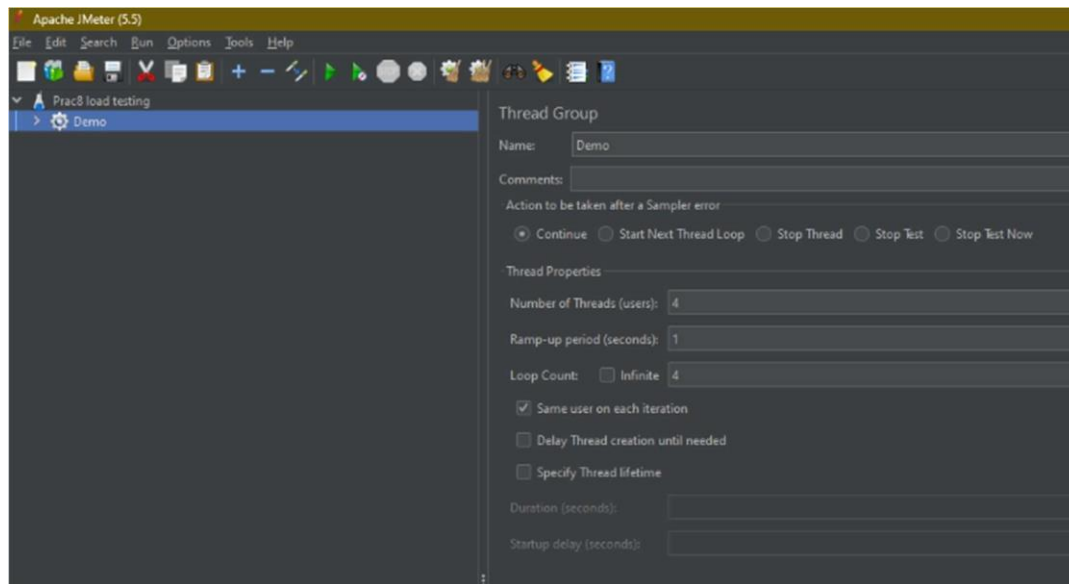
### Step-1:

Right click on Test Plan and go to Add -> Threads -> Thread Group.



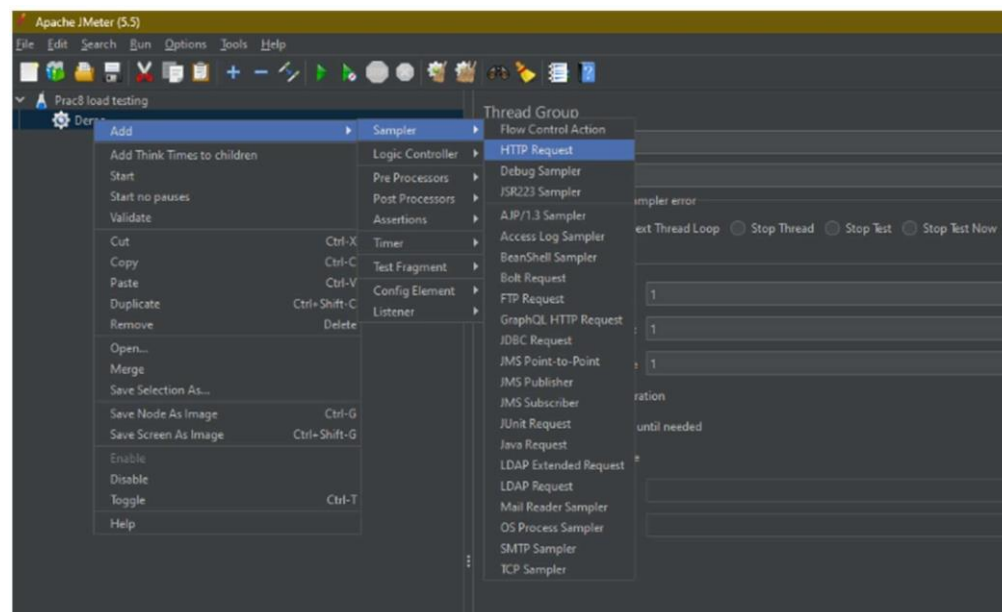
### Step-2:

Give the name of thread group and in Thread properties enter the number of users in the Number of Threads(Users) field.



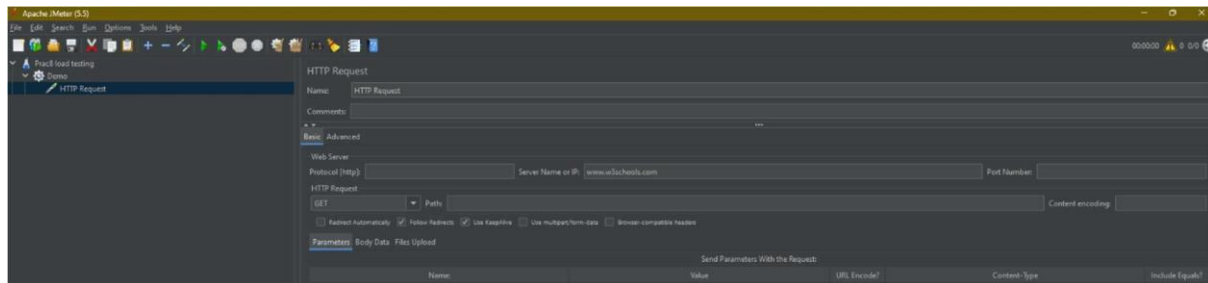
### Step-3:

Right click on the created Test Group and go to Add -> Sampler -> HTTP Request.

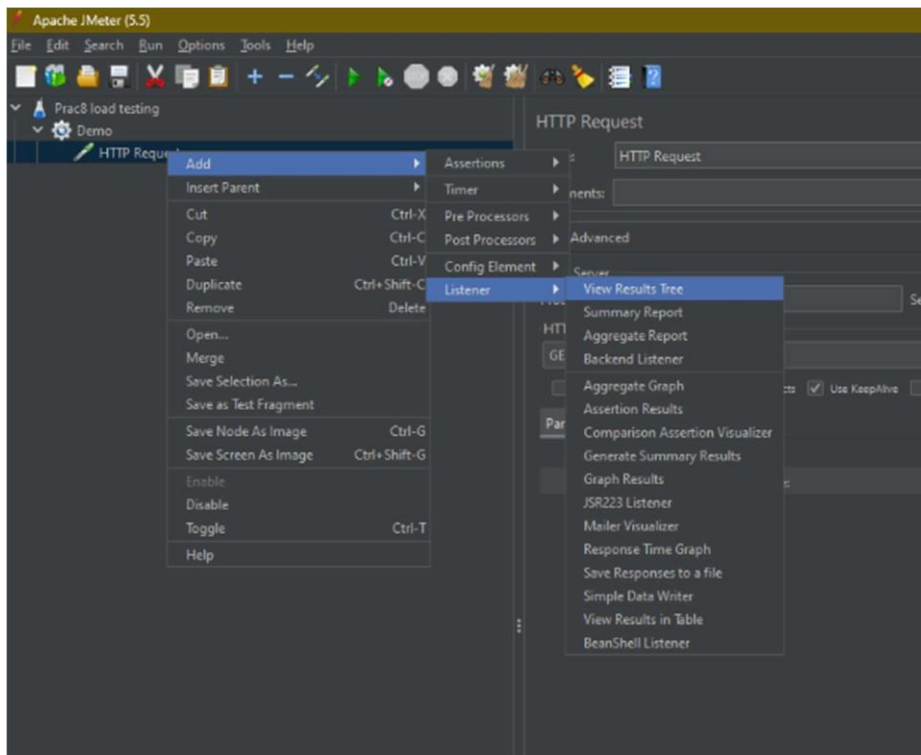


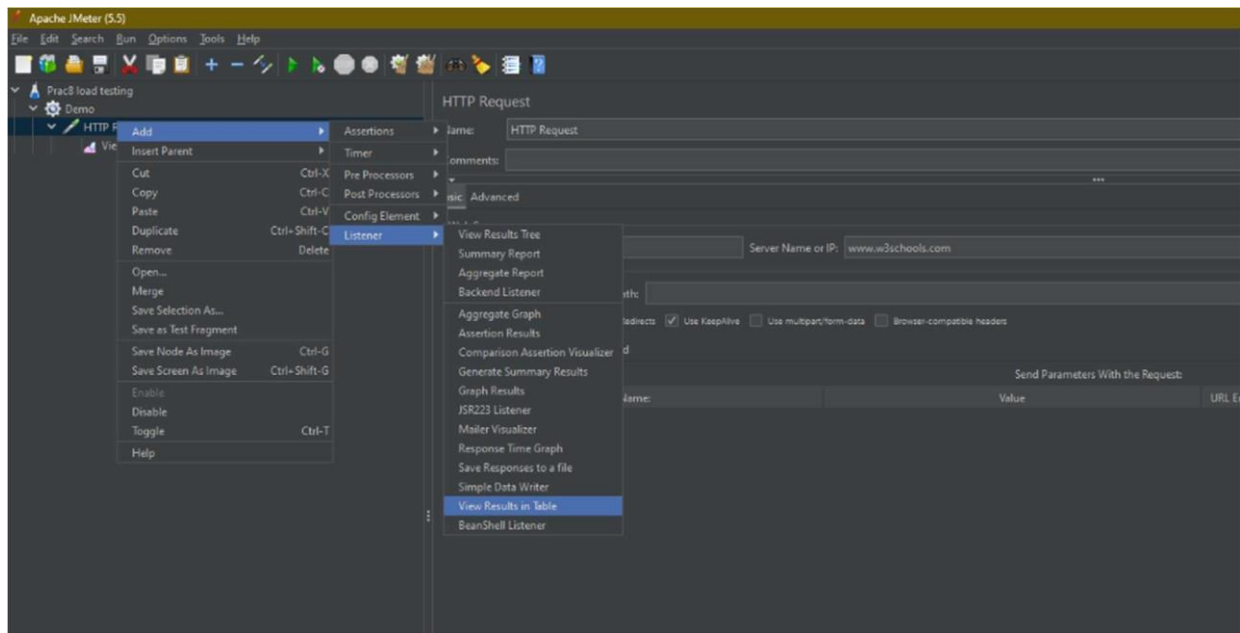
**Step-4:**

Give the url of the website on the performance test is to be done in Server Name or IP and click on Add at the Bottom Pane.

**Step-5:**

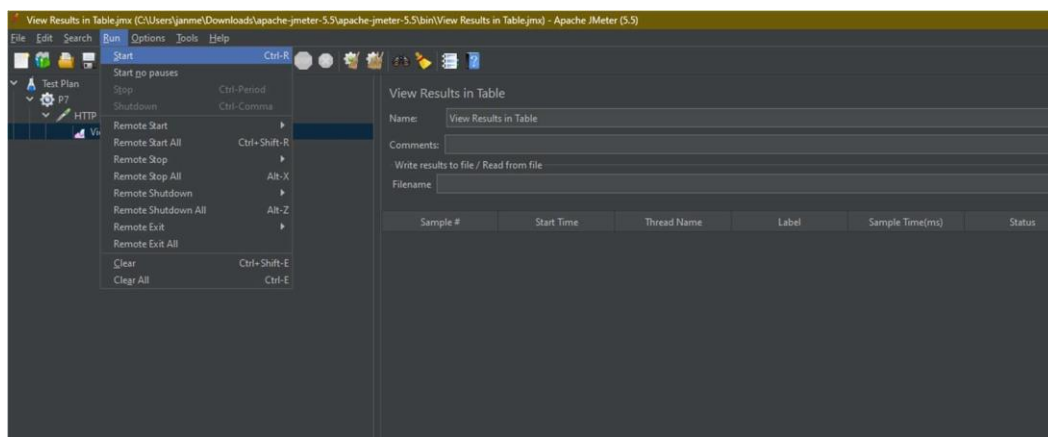
Right click on HTTP Request and go to Add->Listener->View Result Tree sameway add View Results in Table.





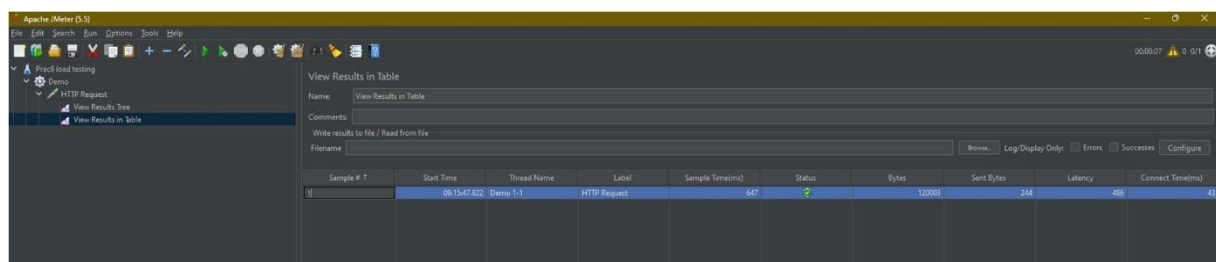
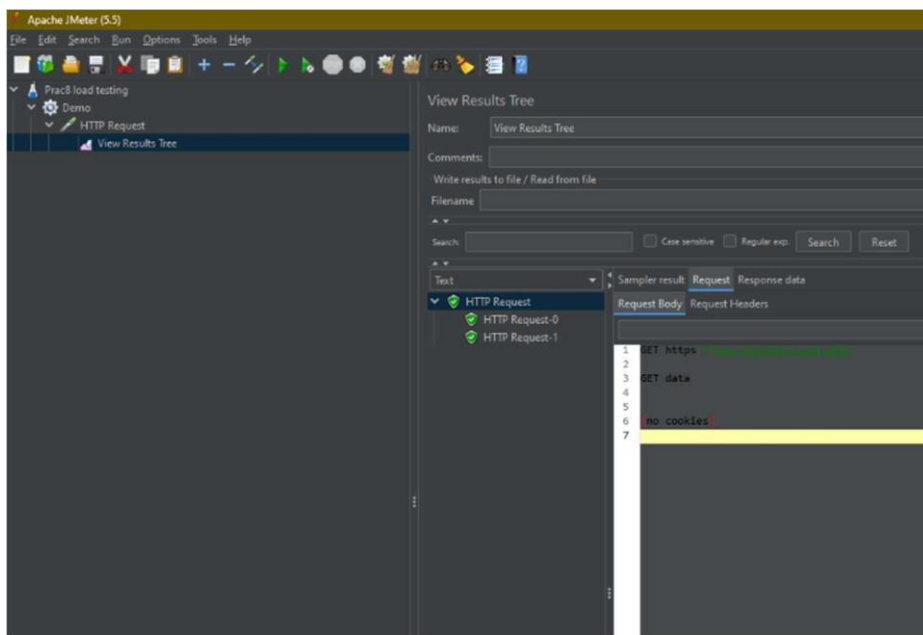
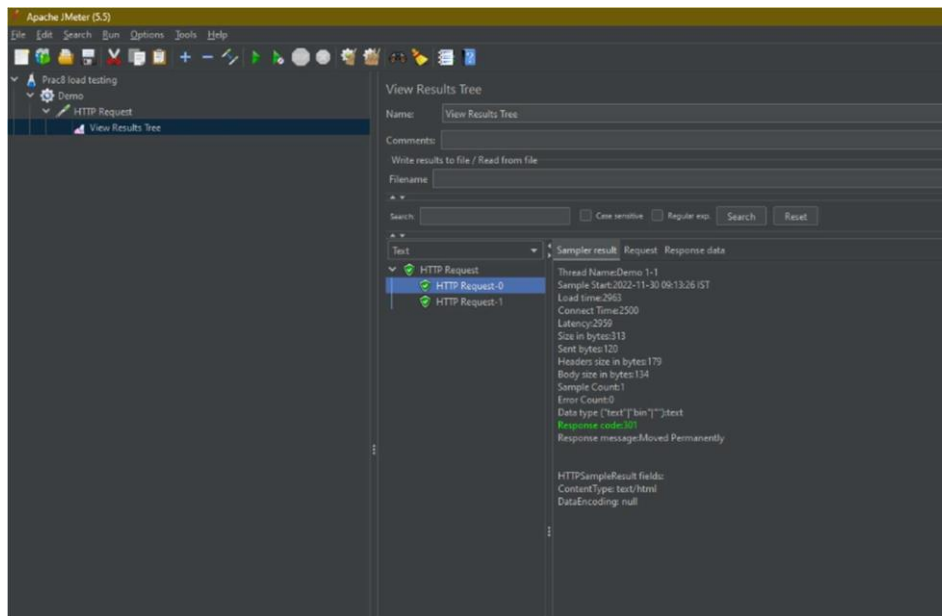
### Step-6:

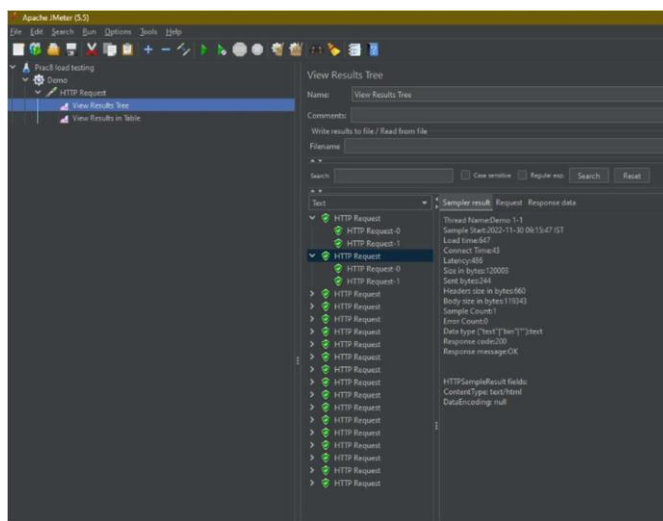
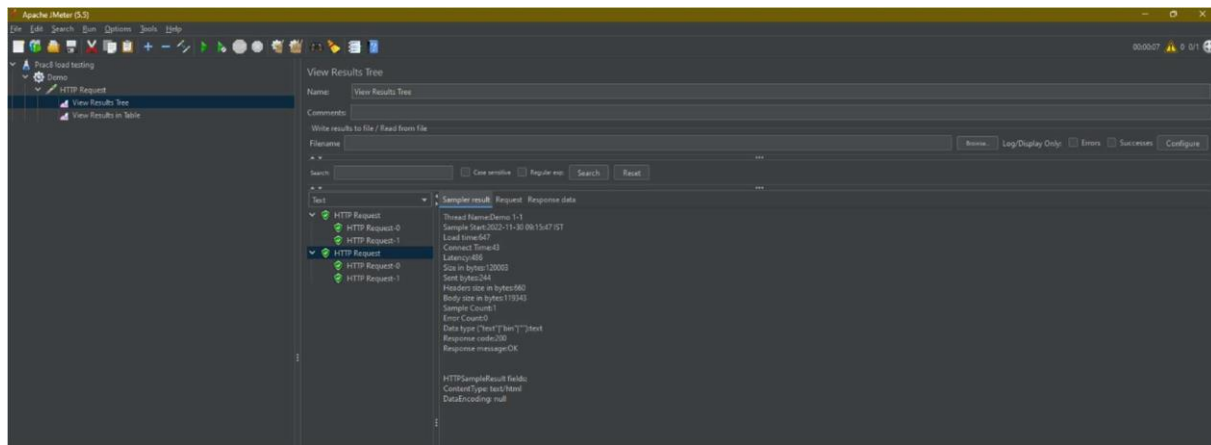
In the top pane click on Run and click on Start.



## Step-7:

You can see all the test cases recorded and their status.





View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename:

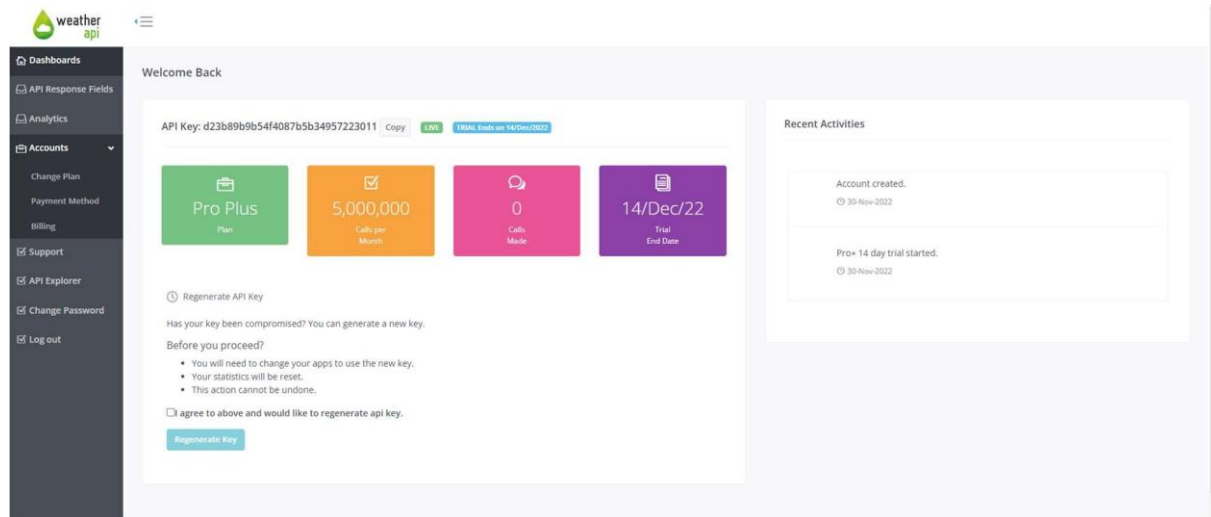
Search:  Case sensitive ☐ Regular exp. ☐ Search:  Reset:

Sample #	T	Start Time	Thread Name	Label	Sample Time (ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
1		09:15:47.022	Demo 1-1	HTTP Request	647	✓	12000	244	466	43
2		09:17:23.445	Demo 1-1	HTTP Request	575	✓	12000	244	400	36
3		09:17:23.708	Demo 1-2	HTTP Request	515	✓	12000	244	393	28
4		09:17:24.021	Demo 1-1	HTTP Request	376	✓	12000	244	347	0
5		09:17:23.951	Demo 1-3	HTTP Request	512	✓	12000	244	373	26
6		09:17:24.259	Demo 1-2	HTTP Request	463	✓	12000	244	414	0
7		09:17:24.197	Demo 1-1	HTTP Request	424	✓	12000	244	367	0
8		09:17:24.179	Demo 1-4	HTTP Request	750	✓	12000	244	604	26
9		09:17:24.463	Demo 1-3	HTTP Request	651	✓	12000	244	619	0
10		09:17:24.723	Demo 1-2	HTTP Request	414	✓	12000	244	363	0
11		09:17:24.821	Demo 1-1	HTTP Request	374	✓	12000	244	345	0
12		09:17:24.893	Demo 1-4	HTTP Request	403	✓	12000	244	366	0
13		09:17:25.118	Demo 1-2	HTTP Request	427	✓	12000	244	353	0
14		09:17:25.139	Demo 1-4	HTTP Request	394	✓	12000	244	364	0
15		09:17:25.114	Demo 1-3	HTTP Request	669	✓	12000	244	610	0
16		09:17:25.733	Demo 1-4	HTTP Request	404	✓	12000	244	372	0
17		09:17:25.763	Demo 1-3	HTTP Request	393	✓	12000	244	363	0

## Load Testing on WeatherApi

### Step-1:

Create an account in WeatherApi.com



### Step-2:

Copy the API key and Go to API Explorer.

## Interactive API Explorer

WeatherAPI.com interactive API explorer or IO Docs allows you to test our APIs and methods. It returns response headers, response code and response body.

For complete documentation please visit our [Weather API Documentation](#) section.

Your API Key

Protocol

HTTP

Format

JSON

Parameter	Value	Type	Location	Description
q	<input type="text" value="London"/>	string	query	Pass US Zipcode, UK Postcode, Canada Postcode, IP address, Latitude/Longitude (decimal degree) or city name. Visit <a href="#">request parameter</a> section to learn more.

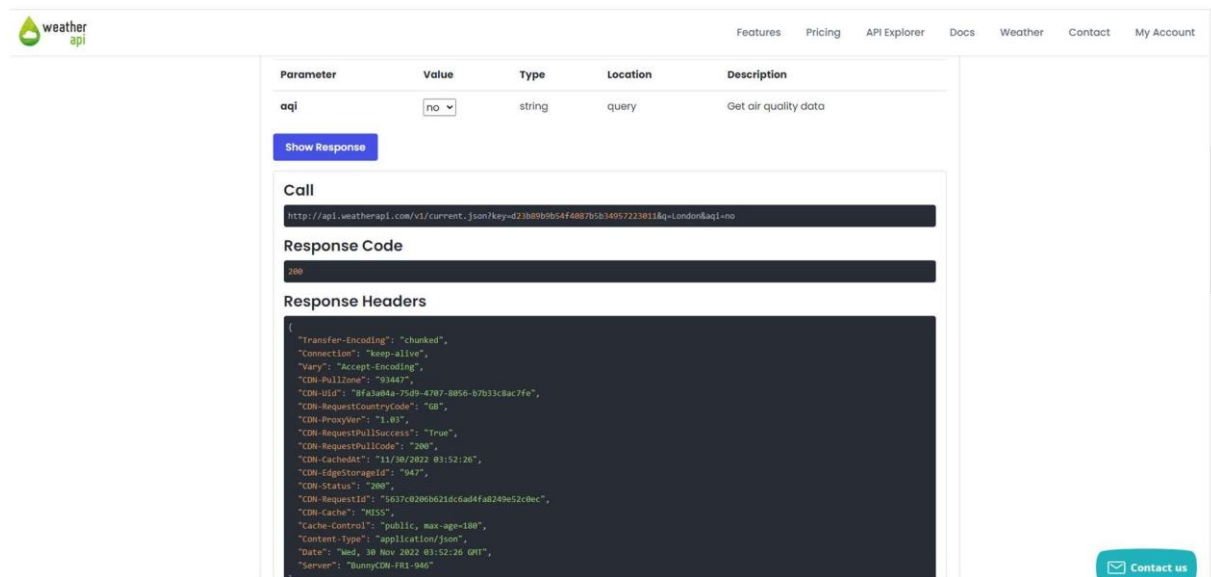
Current Forecast Search/Autocomplete History Future Astronomy Time Zone Sports

Parameter	Value	Type	Location	Description
-----------	-------	------	----------	-------------

Contact us



Copy and paste your API key and click show response to check if the key is valid or not.

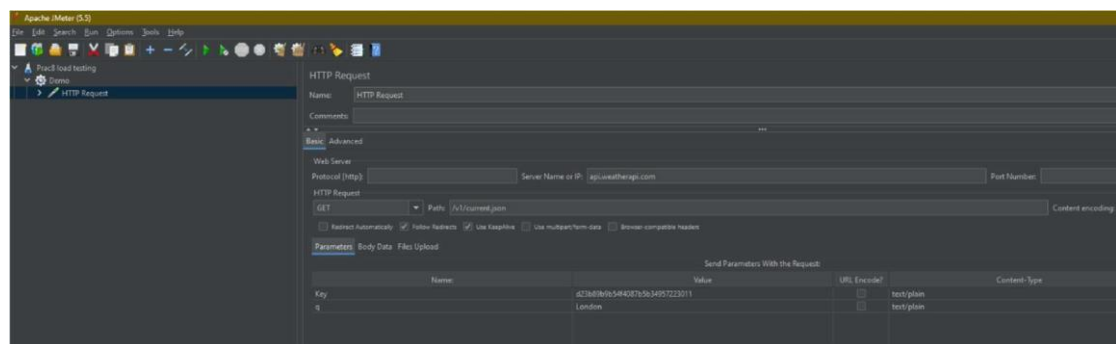


The screenshot shows the weatherapi.com website. At the top, there is a navigation bar with links: Features, Pricing, API Explorer, Docs, Weather, Contact, and My Account. Below the navigation bar, there is a table with columns: Parameter, Value, Type, Location, and Description. The table has one row with the parameter 'aqi', a value of 'no' in a dropdown menu, type 'string', location 'query', and description 'Get air quality data'. Below the table, there is a 'Show Response' button. To the right of the button, there is a 'Call' section with a text area containing the API call: `http://api.weatherapi.com/v1/current.json?key=d23b89b9b54f4887b5b349572238118q=London&q=no`. Below the 'Call' section, there is a 'Response Code' section with a text area containing '200'. Below the 'Response Code' section, there is a 'Response Headers' section with a text area containing the following headers: 

```
{
  "Transfer-Encoding": "chunked",
  "Connection": "keep-alive",
  "Vary": "Accept-Encoding",
  "CDN-Pullzone": "93447",
  "CDN-UID": "8fa3a84a-7549-4787-8856-b7b33cac7fe",
  "CDN-RequestCountryCode": "GB",
  "CDN-ProxyVer": "1.83",
  "CDN-RequestPullSuccess": "True",
  "CDN-RequestPullCode": "200",
  "CDN-CacheHit": "11/30/2022 03:52:26",
  "CDN-EdgeStorageId": "943",
  "CDN-Status": "200",
  "CDN-RequestId": "5637c8200b621dc6ad4fa249e52c8ec",
  "CDN-Cache": "MISS",
  "Cache-Control": "public, max-age=180",
  "Content-Type": "application/json",
  "Date": "Wed, 30 Nov 2022 03:52:26 GMT",
  "Server": "BunnyCDN-FR1-940"
}
```

### Step-3:

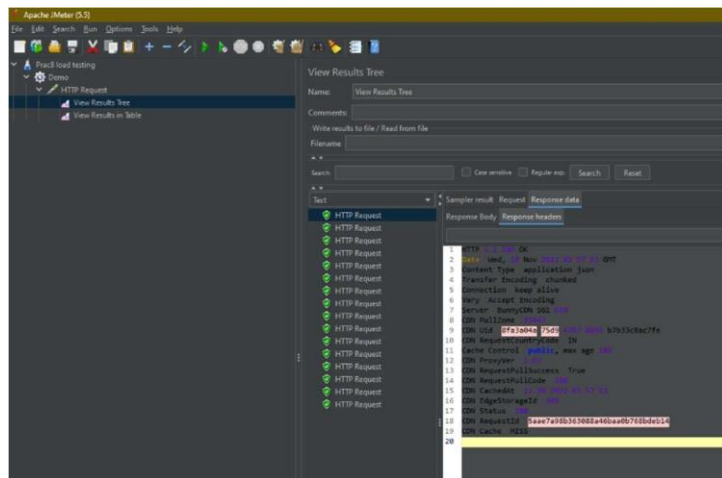
Configure the HTTP accordingly.

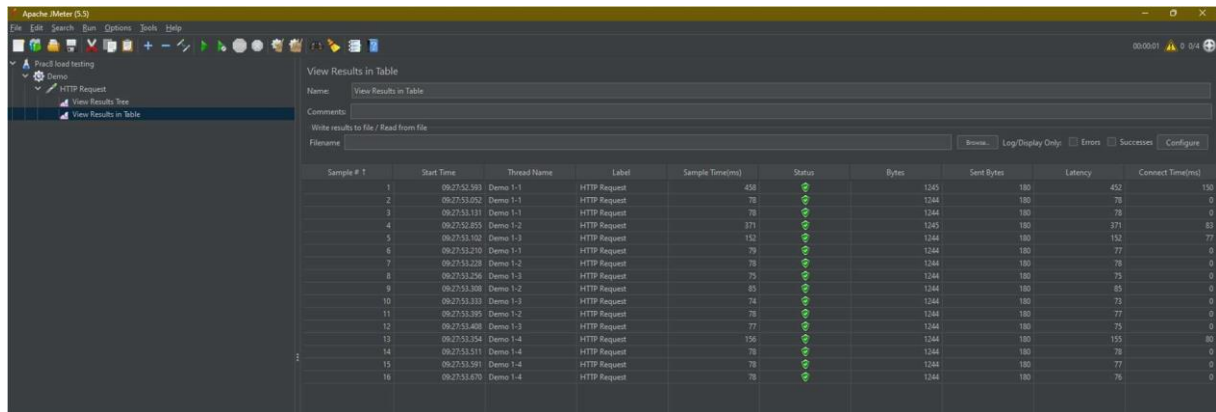


The screenshot shows the Apache JMeter (5.5) interface. The 'HTTP Request' tab is selected. The 'Name' field is set to 'HTTP Request'. The 'Protocol (http)' is set to 'http'. The 'Server Name or IP' is set to 'api.weatherapi.com'. The 'Port Number' is set to '80'. The 'Method' is set to 'GET'. The 'Path' is set to '/v1/current.json'. The 'Content encoding' is set to 'UTF-8'. The 'Parameters' tab is selected, showing a table with columns: Name, Value, URL Encode?, and Content-Type. The table has two rows: 'key' with value 'd23b89b9b54f4887b5b349572238118' and 'q' with value 'London'. Both rows have 'URL Encode?' checked and 'Content-Type' set to 'text/plain'.

request

In the top pane click on Run and click on Start.





Apache JMeter (5.5)

View Results in Table

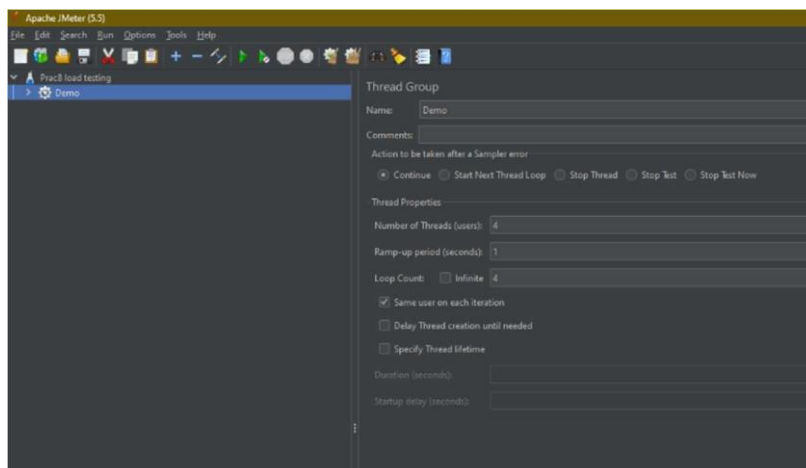
Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only Errors Successes Configure

Sample #	T	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1		09:27:52.993	Demo 1-1	HTTP Request	458	Success	1245	180	452	150
2		09:27:53.052	Demo 1-1	HTTP Request	78	Success	1244	180	78	0
3		09:27:53.131	Demo 1-1	HTTP Request	78	Success	1244	180	78	0
4		09:27:52.955	Demo 1-2	HTTP Request	371	Success	1245	180	371	83
5		09:27:53.102	Demo 1-3	HTTP Request	152	Success	1244	180	152	77
6		09:27:53.210	Demo 1-1	HTTP Request	79	Success	1244	180	77	0
7		09:27:53.228	Demo 1-2	HTTP Request	78	Success	1244	180	78	0
8		09:27:53.256	Demo 1-2	HTTP Request	75	Success	1244	180	75	0
9		09:27:53.308	Demo 1-2	HTTP Request	85	Success	1244	180	85	0
10		09:27:53.333	Demo 1-3	HTTP Request	74	Success	1244	180	73	0
11		09:27:53.395	Demo 1-2	HTTP Request	78	Success	1244	180	77	0
12		09:27:53.488	Demo 1-3	HTTP Request	77	Success	1244	180	75	0
13		09:27:53.354	Demo 1-4	HTTP Request	156	Success	1244	180	155	80
14		09:27:53.311	Demo 1-4	HTTP Request	78	Success	1244	180	78	0
15		09:27:53.391	Demo 1-4	HTTP Request	78	Success	1244	180	77	0
16		09:27:53.670	Demo 1-4	HTTP Request	78	Success	1244	180	76	0



Apache JMeter (5.5)

Thread Group

Name: Demo

Comments:

Action to be taken after a Sampler error

☒ Continue ☐ Start Next Thread Loop ☐ Stop Thread ☐ Stop Test ☐ Stop Test Now

Thread Properties

Number of Threads (users): 4

Ramp-up period (seconds): 1

Loop Count: ☐ infinite 4

☒ Same user on each iteration

☐ Delay Thread creation until needed

☐ Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

---

END

---