API Testing Using JMeter

Prepared By: Bipin Gupta

Profile: QA Engineer

Table Of Contents

❑What is an API ?

- API stands for **Application Programming Interface**. An API is a software intermediary that allows two applications to talk to each other. In other words, an API is the messenger that delivers your request to the provider that you're requesting it from and then delivers the response back to you.

- An API defines functionalities that are independent of their respective implementations, which allows those implementations and definitions to vary without compromising each other. Therefore, a good API makes it easier to develop a program by providing the building blocks. APIs are beneficial because they allow developers to add specific functionality to an application, without having to write all of the code themselves.

- Here is a diagram of where API testing typically comes into play:
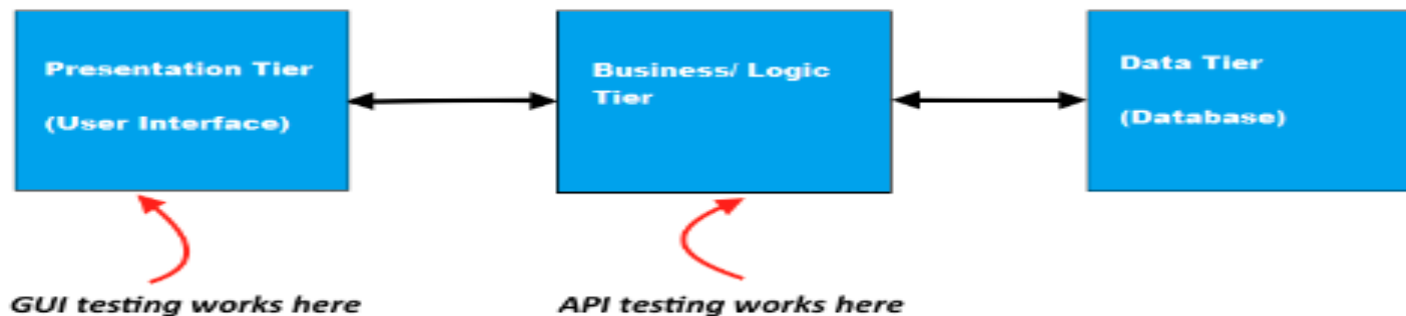
❑How do APIs work?

- Imagine a waiter in a restaurant. You, the customer, are sitting at the table with a menu of choices to order from, and the kitchen is the provider who will fulfill your order. You need a link to communicate your order to the kitchen and then to deliver your food back to your table.

- It can't be the chef because she's cooking in the kitchen. You need something to connect the customer who's ordering food and the chef who prepares it. That's where the waiter — or the API — enters the picture.

- The waiter takes your order, delivers it to the kitchen, telling the kitchen what to do. It then delivers the response, in this case, the food, back to you. Moreover, if the API is designed correctly, hopefully, your order won't crash!

- Here is a diagram of how API works:
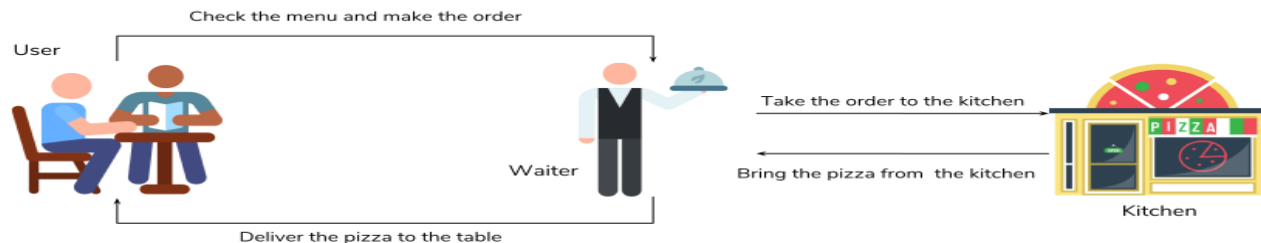
❑Main types of Web APIs:

- There are four main types of APIs:

1. **Open APIs:** Also known as Public API, there are no restrictions to access these types of APIs because they are publicly available.

2. **Partner APIs:** A developer needs specific rights or licenses in order to access this type of API because they are not available to the public.

3. **Internal APIs:** Also known as Private APIs, only internal systems expose this type of API. These are usually designed for internal use within a company. The company uses this type of API among the different internal teams to be able to improve its products and services.

4. **Composite APIs:** This type of API combines different data and service APIs. It is a sequence of tasks that run synchronously as a result of the execution, and not at the request of a task. Its main uses are to speed up the process of execution and improve the performance of the listeners in the web interfaces.

❑Types of Web service APIs:

- A web service is a system or software that uses an address, i.e., URL on the World Wide Web, to provide access to its services.

The following are the most common types of web service APIs:

1. **SOAP (Simple Object Access Protocol):** This is a protocol that uses XML as a format to transfer data. Its main function is to define the structure of the messages and methods of communication. It also uses WSDL, or Web Services Definition Language, in a machine-readable document to publish a definition of its interface.

2. **XML-RPC:** This is a protocol that uses a specific XML format to transfer data compared to SOAP that uses a proprietary XML format. It is also older than SOAP. XML-RPC uses minimum bandwidth and is much simpler than SOAP.

3. **JSON-RPC:** This protocol is similar to XML-RPC but instead of using XML format to transfer data it uses JSON.

4. **REST (Representational State Transfer):** REST is not a protocol like the other web services, instead, it is a set of architectural principles. The REST service needs to have certain characteristics, including simple interfaces, which are resources identified easily within the request and manipulation of resources using the interface.

❑What is HTTP & Types of Methods used for API testing ?

- The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers. HTTP works as a request-response protocol between a client and server. A web browser may be the client, and an application on a computer that hosts a website may be the server. **Example**: A client (browser) submits an HTTP request to the server; then the server returns a response to the client. The response contains status information about the request and may also contain the requested content.

- These are the most common methods involve in API Testing:

1. **GET:** The most common and widely used methods in APIs and websites. It is used to **retrieve data from a server at the specified resource**. For example, say you have an API with a /users endpoint. Making a GET request to that endpoint should return a list of all available users.

2. **POST:** In web services, POST requests are used to **send data to the API server** to create or update a resource. The data sent to the server is stored in the request body of the HTTP request. The simplest example is a contact form on a website. When you fill out the inputs in a form and hit *Send*, that data is put in the **response body** of the request and sent to the server. This may be JSON, XML, or query parameters (there's plenty of other formats, but these are the most common).

3. **PUT:** Similar to POST, PUT requests are used to send data to the API to **create or update a resource**. The difference is that **PUT requests are idempotent**. That is, calling the same PUT request multiple times **will always produce the same result**. In contrast, calling a POST request repeatedly make have side effects of creating the same resource multiple times.

4. **DELETE:** Removes all current representations of the target resource given by a URI. The simplest example is If a new user is created with a POST request to /users, and it can be retrieved with a GET request to /users/{{userid}}, then making a DELETE request to /users/{{userid}} will completely remove that user

5. **PATCH** : PATCH is used to update partial resources. For instance, when you only need to update one field of the resource, Putting a complete resource representation might be cumbersome and utilizes more bandwidth.

6. **HEAD** : HEAD is almost identical to GET, but without the response body. HEAD transfers the status line and the header section only.

7. **OPTIONS** : The OPTIONS method describes the communication options for the target resource.

❑The Benefits of API Testing:

- API testing is the only way to provide truly secure, reliable and scalable connections between platforms. Testing provides these benefits:

1. **Access to application without user interface:** The major core advantage of API testing is that it provides access to application without users actually having to interact with a potentially disparate system. This helps the tester to detect and recognize the errors early, instead of them becoming larger issues during GUI testing.

2. **Protection from malicious code and breakage:** API test requires extraordinary conditions and inputs, which protects the application from malicious code and breakage. Basically, API tests push software to their connective limits. API testing helps remove vulnerabilities.

3. **Time Efficiency vs. functional and validation testing:** API testing is far more less time consuming than functional and validation testing.

4. **Cost Effective / Reduces Testing Cost:** Testing the API level functionality of the application provides an early evaluation of its overall build strength before running GUI tests. Early detection of errors reduces the manual testing cost.

5. **Technology Independent:** In an API test, the data is interchanged using XML or JSON and compromised of HTTP requests and responses. These all are technology independent and used for development. Thus an API test allows you to select any core language when using automated API testing services for your application.

❑Top 10 open-source API Testing Tools:

1. Katalon Studio
2. SoapUI
3. Postman
4. JMeter
5. Rest-Assured
6. Postwoman
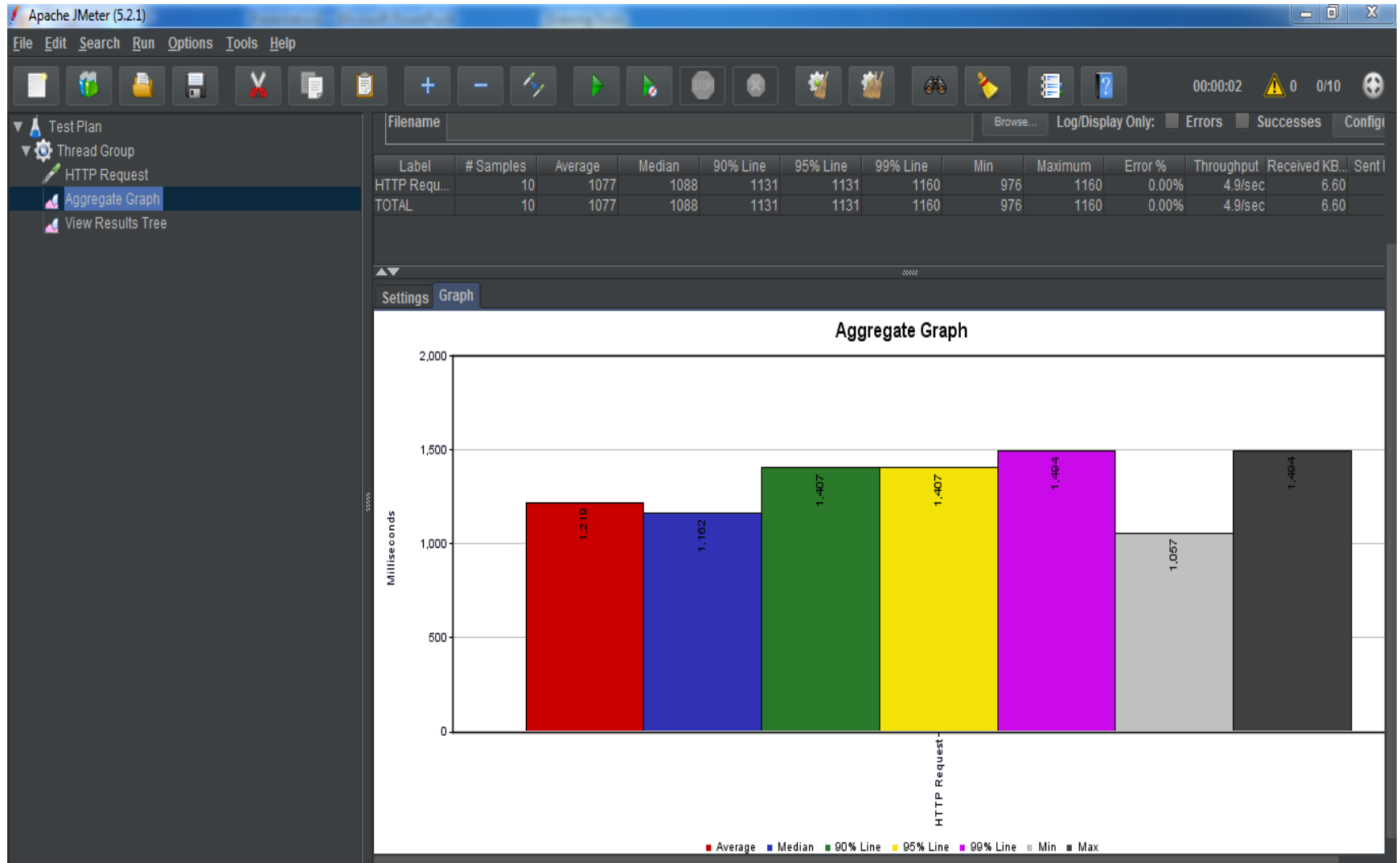7. Karate
8. Fiddler
9. Citrus Framework
10. Taurus

❏ How to test an API using Apache JMeter?

- Perform the below mentioned steps:

1. Add a new Thread Group under Test Plan.
2. Right click on **Thread Group** and **Select Add >> Sampler>>HTTP Request**
3. Add **API** details in **HTTP Request Sampler.** (Refer https://openweathermap.org/api for public APIs).
4. Now, Right click on Thread Group and **Select Add>> Listener>> View Results Tree or Aggregate Graph** for observing the output response.
5. After performing all above steps, your **HTTP Request Sampler** should be displayed as the image in next slide.
6. Run Test.

❑ Screenshot of HTTP Request Sampler Window:
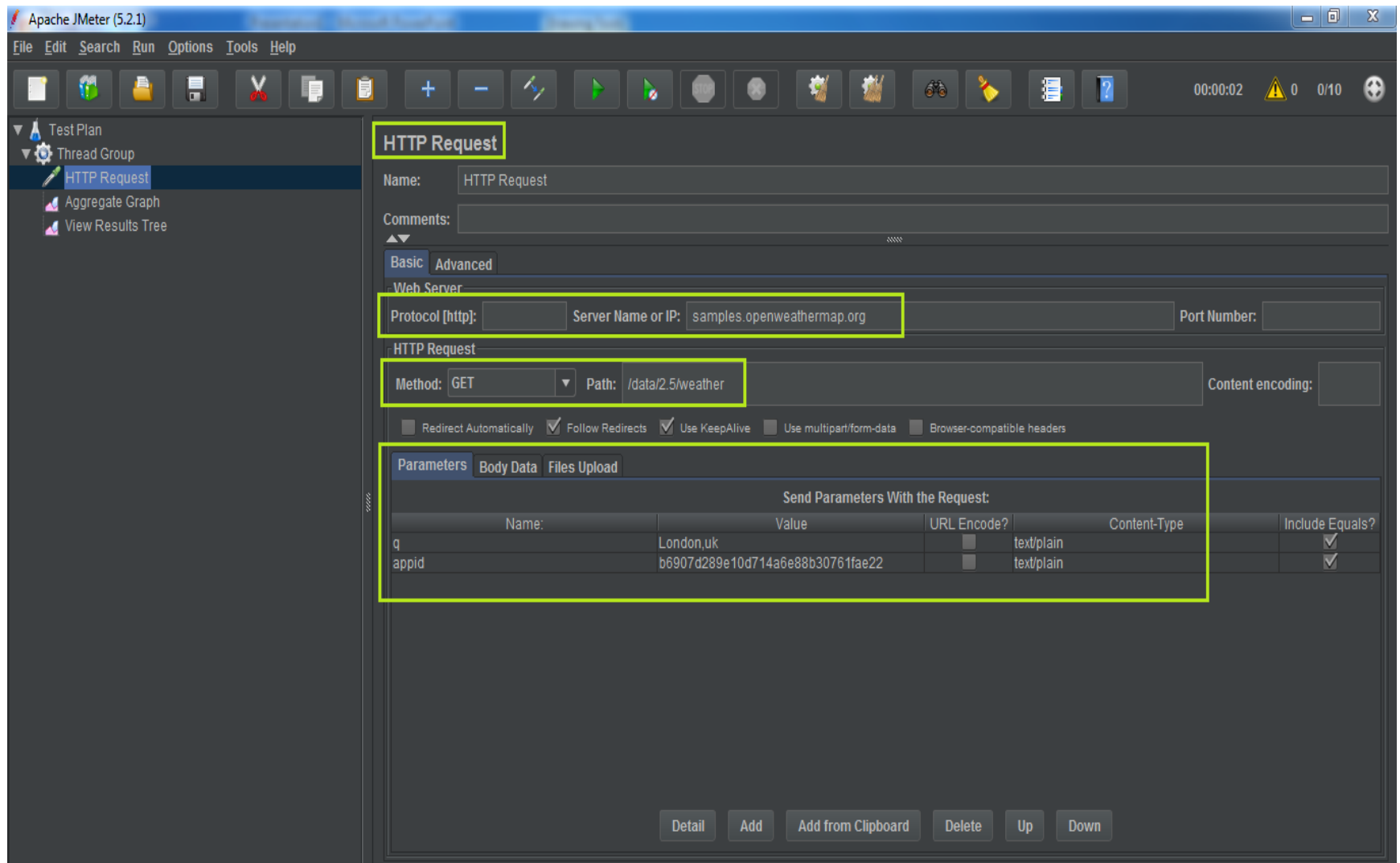
❑Screenshot of an output response:

Thank You!