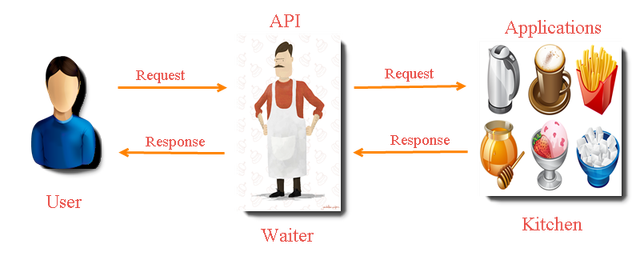
**What is API?**

API stands for Application Programming Interface. It is a software to software interface, not a user interface. With APIs, applications converse with each other with no client learning or intervention. It can be executed by composing capacity brings in the system.

**API Workflow Example**



**What is API Testing?**

API testing utilizes programming to send calls to the API and get the yield. It testing regards the segment under test as a black box. The objective of API testing is to confirm right execution and blunder treatment of the part preceding its coordination into an application.

**REST API**

**REST: Representational State Transfer.**

* It’s an arrangement of functions on which the testers performs requests and receive responses. In REST API interactions are made via HTTP protocol.
* REST also permits communication between computers with each other over a network.
* For sending and receiving messages, it involves using HTTP methods, and it does not require a strict message definition, unlike Web services.
* REST messages often accepts the form either in form of XML, or JavaScript Object Notation (JSON).

**4 Commonly Used API Methods:-**

1. **GET: –** It provides read only access to a resource.
2. **POST: –** It is used to create or update a new resource.
3. **PUT: –**It is used to update or replace an existing resource or create a new resource.
4. **DELETE: –** It is used to remove a resource.

**REST API Inputs:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **GET** | **POST** | **PUT** | **DELETE** |
| **Method – GET** | **Method – POST** | **Method – PUT** | **Method – DELETE** |
| **URL** | **URL** | **URL** | **URL** |
| **Custom-Header** | **Custom-Header** | **Custom-Header** | **Custom-Header** |
|  | **Input JSON** | **Input JSON** | **Input JSON** |

**Browser Specific Rest Clients List to Test API Manually:-**

|  |  |  |
| --- | --- | --- |
| **Browser** | **Tool Name** | **Web Site** |
| **Firefox** | **RESTClient** | [**https://addons.mozilla.org/en-US/firefox/addon/restclient/**](https://addons.mozilla.org/en-US/firefox/addon/restclient/) |
| **Firefox** | **REST Easy** | [**https://addons.mozilla.org/en-us/firefox/addon/rest-easy/**](https://addons.mozilla.org/en-us/firefox/addon/rest-easy/) |
| **Chrome** | **Advanced REST Client** | [**http://chromerestclient.appspot.com/**](http://chromerestclient.appspot.com/) |
| **Chrome** | **Postman** | [**https://www.getpostman.com/**](https://www.getpostman.com/) |
| **Chrome** | **DHC – REST/HTTP API Client** | [**http://restlet.com/products/dhc/**](http://restlet.com/products/dhc/) |

**Steps to Test API Manually**:-

To use API manually, we can use browser based REST API plugins.

**a)**Install POSTMAN(Chrome) / REST(Firefox) plugin

**b)**Enter the API URL

**c)**Select the REST method

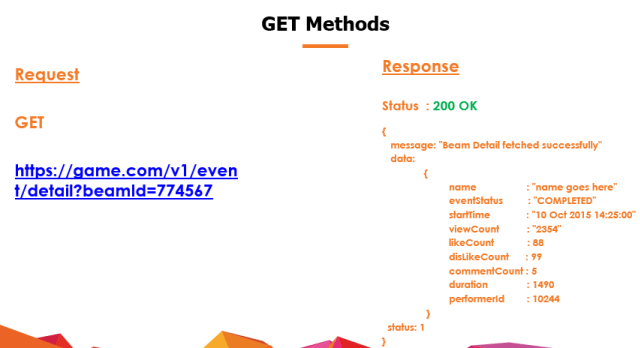
**d)**Select content-Header

**e)**Enter Request JSON (POST)

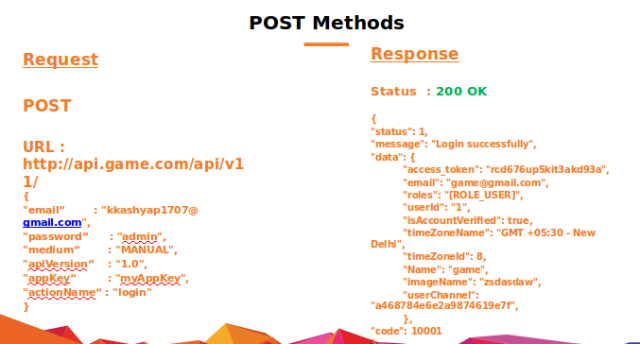
**f)**Click on send

**g)**It will return output response

**Example:-**



**Fig. :: GET method Example**



**Fig. :: POST method Example**

**HTTP Response Codes**

**Some HTTP response codes, which are often used with REST:-**

**200 OK: –** This code indicates that the request was made successful.

**201 Created: –** This code indicates that request was successful and a resource was created and it validated the success of a PUT or POST request.

**400 Bad Request: –** This error code happens with POST and PUT requests, when the data does not pass validation or the passed data is in the wrong format.

**404 Not Found: –** This code indicates that the required resource could not be found or not available.

**401 Unauthorized: –** This error code indicates that you need to provide authentication before accessing the resource.

**405 Method Not Allowed: –** Method Not Allowed error indicates that the requested HTTP method is not supported for this resource.

**409 Conflict: –** This code indicates that the user create the same resource twice by which conflict occurs.

**500 Internal Server Error: –** Due to server side issue this error code occurs.

**Common Types of Tests and Bugs in API Testing**

* Delayed in API Response time
* To verify when API doesn’t return any response data.
* Response Data is not structured
* Difficulty in connecting and getting response from API
* Output response should be checked on the basis of provided input request.
* Verification of the API whether it triggers some other event or request another API
* Verification of the API whether it is updating any data structure

**Advantages of API Testing**

* It is time effective.
* It is language independent.
* It is quite helpful in testing Core Functionality.
* It helps us to reduce the testing cost.
* It helps us to reduce the risks.

**Challenges of API Testing**

* Parameter Combination, Parameter Selection, and Call Sequencing are three major challenges in API testing.
* Due to absence of GUI it is quite difficult to provide input values.
* Parameters selection and categorization required must be known to the tester.

**Why API Automation?**

Below are few points which invokes us to start API Automation:-

* Testing of manual tests takes far longer than just executing them via automation.
* The UI is one of the least stable interfaces of rapid changing Web/App Ecosystem, so we can start automating API’s to test the backend functionality.
* Maintenance of the UI tests takes a significant amount of time.
* Execution of UI test is slow, and sometimes cumbersome.
* Tests become flaky.
* Tests break for the wrong reasons at UI end.

**Prerequisites to start API Automation**

Setup below mentioned tools on your development machine:-

* Your favorite IDE (Eclipse Recommended)
* Java, Maven should be installed
* Java, Maven, TestNG knowledge
* HTTP Library: A Java based library for HTTP communications
* GSON to handle or manipulate JSON response.

**What is JSON?**

* JSON stands for Java Script Object Notation.
* It provides readable format for structuring data into a syntax for storing and exchanging.
* There are two primary parts of JSON format i.e.: keys and values. Key is always a string while value can be String, Array, Boolean Expression or an object.

{  
“status”: 1,  
“message”: “Login successfully”,  
“data”: {  
                                “access\_token”: “rcd676up552kit3oqfhkajjf18akd93a”,  
                                “email”: “kkashyap1707@[gmail.com](http://gmail.com/)“,  
                                “userId”: “1”,  
                                “timeZoneName”: “GMT +05:30 – New Delhi”,  
                                “timeZoneId”: 8,  
                                “gameName”: “ikeshav”,  
                },  
“code”: 10001  
}

**JSON Representation**

**Difference Between JSON and XML :-**

|  |  |
| --- | --- |
| **JSON** | **XML** |
| Java Script Object Notation | Extensible Markup Language |
| Developed by Douglas Crockford | Developed by W3C |
| Faster to write | Difficult to write |
| Light Weighted | Not so lighter as compared to JSON as it takes more space to represent same data. |
| Doesn’t supports comments | Supports Comments |
| Supports data types including Integer , string and Arrays. | No direct support of Array |
| Data oriented and can be mapped easily | Document oriented & need more efforts for mapping. |