

# API Testing

**By Zain**

# Agenda of API Testing



## API TESTING

API and API Testing



## API Methods

Different API Methods



## TOOL SELECTION

Right tool for API Testing



## ADVANTAGE

S

Advantages of API Testing



## WORKFLOW

How API works ?



## DEMO

Live Demo

# What is API?

---

API stands for  
Application  
Programming Interface.

API is an software to software  
interface, not a user interface.

With APIs, applications talk  
to each other without any  
user knowledge or  
intervention.

Implemented by writing function  
calls in the program.



# What is API Testing?

---

API testing uses software to send calls to the API and get the output.

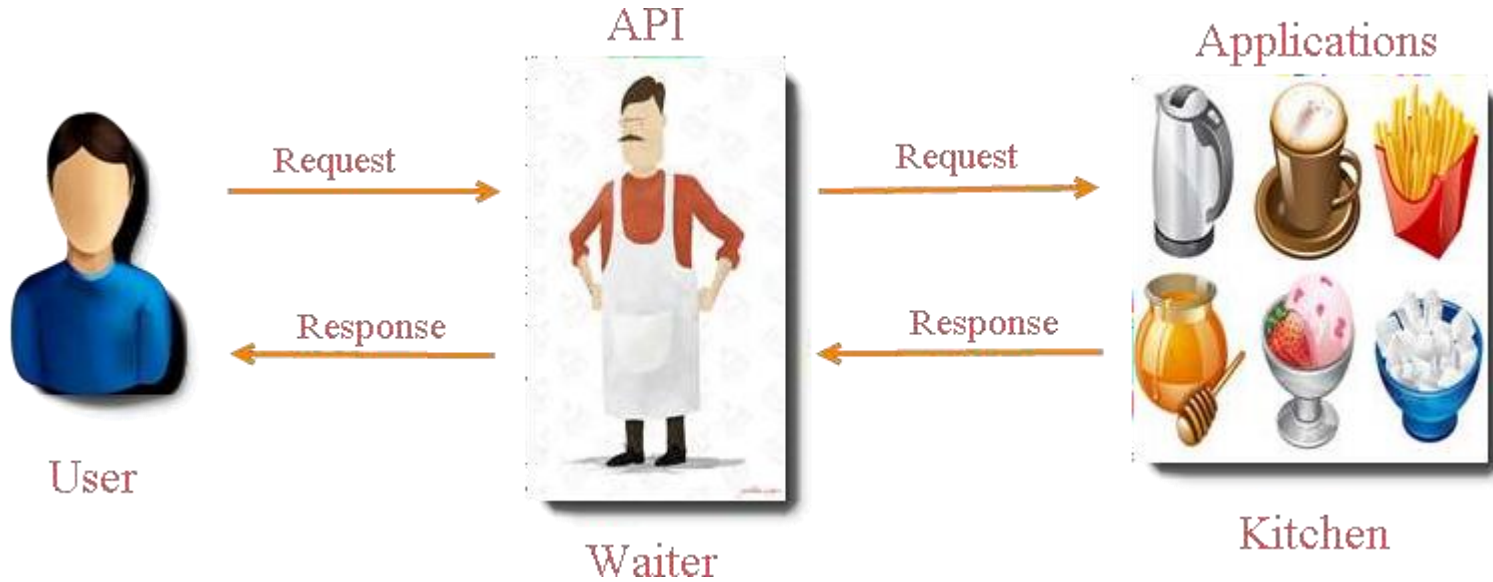
API testing treats the component under test as a black box.

The goal of API testing is to verify correct performance and error handling of the component prior to its integration into an application.



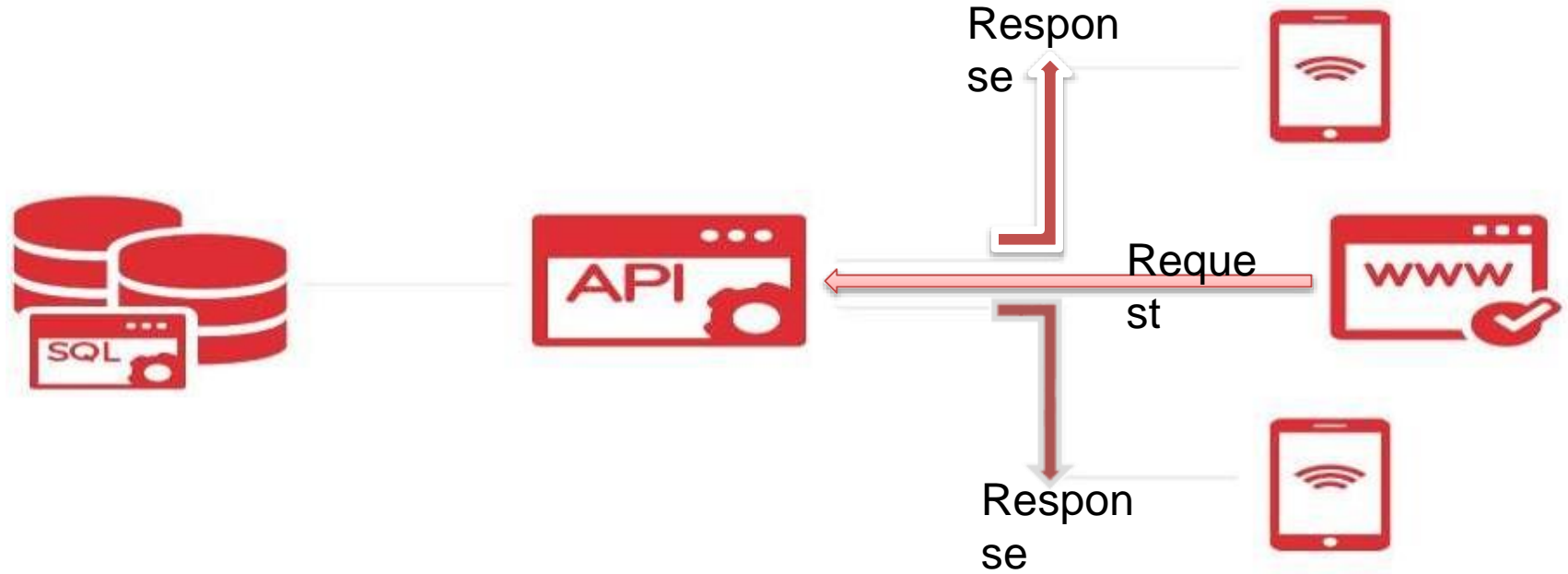
# API Workflow

---



# API Workflow

---



# REST API

---



**REST** – stands for Representational State Transfer.

- It is a set of functions to which the developers perform requests and receive responses. In REST API interaction is made via HTTP protocol.
- REST also allows computers to talk to each other over a network.
- It involves using HTTP methods to send and receive messages, and does not require a strict message definition, unlike Web services.
- REST messages often take the form of XML, or JavaScript Object Notation (JSON).

# HTTP Methods

---

## 4 Commonly Used Methods:-

**GET** : - Provides a read only access to a resource.

**POST** :- Used to update an existing resource or create a new resource.

**PUT** :- Used to create a new resource.

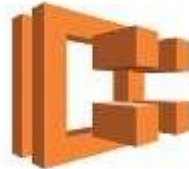
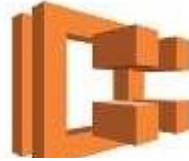
**DELETE** :- Used to Remove a resource.

RESOUCE	GET	PUT	POST	DELETE
URL	Retrieve the member	Create / Replace a new resource	Create new entry	Used to remove a resource



# What is API Testing ?

---



# REST API Inputs

---



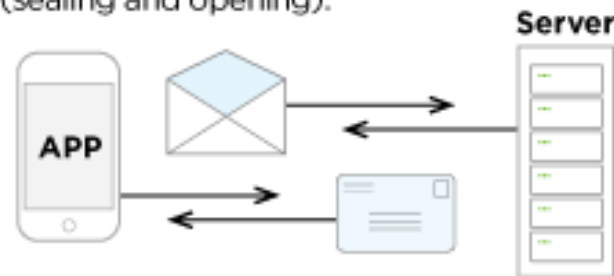
GET	POST	PUT	DELETE
Method - GET	Method - POST	Method - PUT	Method - DELETE
URL	URL	URL	URL
Custom-Header	Custom-Header	Custom-Header	Custom-Header
	Input JSON		

# SOAP METHOD

## SOAP vs. REST APIs

SOAP is like using an envelope

Extra overhead, more bandwidth required, more work on both ends (sealing and opening).



REST is like a postcard

Lighterweight, can be cached, easier to update.

upwork

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <personnel>
3   <person id="Big.Boss">
4     <name>
5       <family>
6         (The end tag is indented at the same level
7         as the start tag.)
8       </family>
9     </name>
10    <address>
11      <city>
12        <zip>
13          <code>(The children of an element increase
14          the indentation level.)</code>
15        </zip>
16      </city>
17    </address>
```

# GET Methods

---

## Request

GET

## Response

Status : 200 OK

```
{
  message: "Beam Detail fetched successfully"
  data:
    {
      name          : "name goes here"
      eventStatus   : "COMPLETED"
      startTime     : "10 Oct 2015
14:25:00"
      viewCount     : "2354"
      likeCount     : 88
      disLikeCount  : 99
      commentCount  : 5
      duration      : 1490
      performerId   : 10244
    }
  status: 1
}
```

# POST Methods

---

## Request

### POST

```
"email"      : "zainsch707@gmail.com",  
"password"   : "admin",  
"medium"     : "MANUAL",  
"apiVersion" : "1.0",  
"appKey"     : "myAppKey",  
"actionName" : "login"  
}
```

## Response

Status : 200 OK

```
{  
  "status": 1,  
  
  "message": "Login successfully",  
  "data": {  
    "access_token": "rkd676up5kit3akd93a",  
  
    "email": "admn.fameliive@gmail.com",  
    "roles": "[ROLE_USER]",  
  
    "userId": "1",  
    "isAccountVerified": true,  
    "timeZoneName": "GMT +05:30 - New York",  
    "timeZoneld": 8,  
    "fameName": "admin.fameliive",  
    "code": 10001  
  },  
  "imageName": "zsdasdaw",  
  "userChannel": "a468784e6e2a9874619e7f",  
}
```

# PUT Methods

---

## Request

GET

## Response

Status : 200 OK

```
{  
  message: "Beam Detail fetched successfully"  
  data:  
    {  
      name: "name goes here"  
      eventStatus: "COMPLETED"  
      startTime: "10 Oct 2015 14:25:00"  
      viewCount: "2354"  
      likeCount: 88  
      disLikeCount: 99  
      commentCount: 5  
      duration: 1490  
      performerId: 10244  
    }  
} status: 1
```

# DELETE Methods

---

## Request

GET

## Response

Status : 200 OK

```
{  
  message: "Beam Detail fetched successfully"  
  data:  
    {  
      name: "name goes here"  
      eventStatus: "COMPLETED"  
      startTime: "10 Oct 2015 14:25:00"  
      viewCount: "2354"  
      likeCount: 88  
      disLikeCount: 99  
      commentCount: 5  
      duration: 1490  
      performerId: 10244  
    }  
} status: 1
```

# HTTP Response Codes

---

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

**200 OK:** - Code indicates that the request was successful.

**201 Created:-** Code indicates that request was successful and a resource was created. It is used to confirm success of a PUT or POST request.

**400 Bad Request :-** It happens especially with POST and PUT requests, when the data does not pass validation, or is in the wrong format.

**404 Not Found:-** response indicates that the required resource could not be found.

**401 Unauthorized:-** error indicates that you need to perform authentication before accessing the resource.

**405 Method Not Allowed:-** HTTP method used is not supported for this resource.

**409 Conflict:-** Conflict request to create the same resource twice.

**500 Internal Server Error:-** Occurs due to some error on Server side.



# Advantages of API Testing

---

## Advantages of API Testing:-

- Time Effective
- Language Independent
- Test Core Functionality
- Reduce Testing cost
- Reduced Risks

# Common Types of Tests in API Testing

---

## Common API Bugs:-

- Verify if API doesn't return any response.
- Based on input request, return request should be checked.
- Verification of the API whether it triggers some other event or calls another API
- Verification of the API whether it is updating any data structure
- Delayed in API Response time
- Response Data is not structured
- Difficulty in connecting and getting response from API

# Basic Inputs

URL : `http://qa-api.livfame.com/api/v11_1`

URL Parameter Key	Value
Content-Type	application/json
Header	Value

form-data x-www-form-urlencoded raw JSON

TO

POST

GET

POST

PUT

PATCH

DELETE

COPY

HEAD

OPTIONS

LINK

UNLINK

PURGE

## Header

form-data x-www-form-urlencoded raw JSON

```
{
  "apiVersion": "1.0",
  "actionName": "login",
  "email": "myname1santhony",
  "password": "123456",
  "medium": "MANUAL",
  "appKey": "myAppKey"
}
```

Sand Preview Add to collection

## Input JSON

## Method

STATUS 200 OK TIME 389 ms

Preview

JSON XML

```
1 {
2   "status": 1,
3   "message": "Login successfully"
}
```

## Output Response