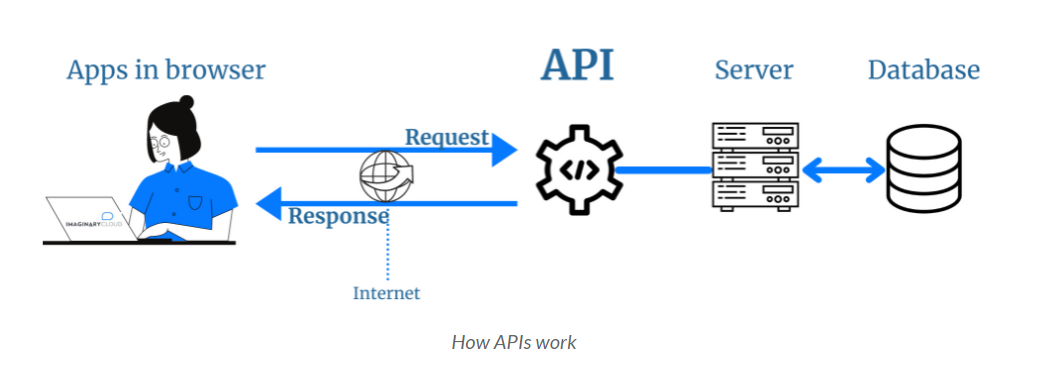
**Q. What is API?**

**API** (*Application Programming Interface*) helps in communication and data exchange between two software systems. API act as an interface between two applications and allows the two software systems communicate with one another. API is a collection of functions which can be executed by another software program.

API works as; it takes a request from the source, takes that request to the database, fetches the request data from the database and returns a response to the source. API takes the requests from the user and gives the response without exposing the internal details. API acts as Abstraction.

**Example:** Amazon API, Google Map API



Let's imagine we are booking a hotel. We go to the hotel booking page on our laptop, and that page - which is connected to the Internet - sends data (our request) to a server. In turn, the server retrieves the data, interprets it, and then, once the required actions are executed, it sends a response back to us with the information on our interface. This process happens thanks to APIs.

### 2) What is API testing?

**API testing** is a type of software testing that involves testing APIs directly. API is a part of integration testing to check whether the API meets expectations in terms of functionality, reliability, performance, and security of applications. Multiple API system can perform API testing. In API testing, our primary focus is on Business Logic Layer of the software architecture.

### 3) What are the types of API testing?

API testing involves the following types of testing:

* Unit Testing
* Functional Testing
* Load Testing
* Runtime/Error Detection
* Security Testing
* UI Testing
* Interoperability and WS compliance Testing
* Penetration Testing
* Fuzz Testing

**4) What are the protocols used in API Testing?**

Protocols used in API testing are:

* HTTP
* REST
* SOAP
* JMS
* UDDI

**5) What are the tools used for API Testing?**

* Apigee --- **cross-cloud API testing software** powered by [JavaScript](https://www.imaginarycloud.com/blog/a-javascript-ecosystem-overview/)
* SoapUI
* Katalon -- Katalon is a popular [test automation tool](https://www.imaginarycloud.com/blog/top-automation-testing-tools/) for APIs, as well as web applications and desktop and mobile apps. It supports **SOAP and REST requests** and provides multiple parameterization features and commands. Additionally, Katalon follows a data-driven approach, supports **CI/CD integrations**, and allows testers to execute automated and exploratory testing.
* PostMan
* Tricentis
* Assertible

### 6) What is API test environment?

For API the test environment is a quite complex method where the configuration of server and database is done as per the requirement of the software application. **API testing does not involve graphical user interface (GUI).**

API is checked for its proper functioning after installation.

### 7) What is API framework?

API framework is described by the config. File which consist of the list of all APIs that are required to be activated and are activated for any particular program run. This is essential as every test run does not require all APIs.

### 8) What are the limits of API usage?

Many APIs have certain limit set up by the provider. Hence, try to estimate our usage and understand how that will impact the overall cost of the offering.

**9) What are the advantages of API testing?**

Advantages of API testing are:

* **Test for core functionality:** API testing provides access to the application without the user interface. The core functionality of the application will be tested before the GUI tests. This will help to detect the minor issue which can become bigger during the GUI testing.
* **Time effective:** API testing is less time consuming than GUI testing. Particularly, API test requires less code so it can provide better and faster test coverage compare to GUI test automation. This will reduce the cost for the testing project.
* **Language Independent:** In API testing data is exchange using XML or JSON. These transfer mode are completely language-independent, which allows users to select any code language when adopting automation test service for the project.
* **Easy Integration with GUI:** API tests provide highly integrable tests which is useful to perform functional GUI tests after GUI tests. Simple integration would allow new user accounts to be created within the application before GUI started.

**10) What are the principles of an API test design?**

Here, are the seven principles of API test design.

1. **Exhaustive Testing:** Exhaustive testing is not possible. Instead we need optimal amount of testing which is based on the risk assessment of the application.
2. **Defect Clustering:** Defect Clustering states that a small number of modules contain the most of the defect detected. Approximately 80% of the defect found in 20% of the modules. By experience we can identify such risky modules. But this approach has its own problems. If the same tests are repeated over and over again, eventually the same test case will no longer find new bugs.
3. **Pesticide Paradox:** Testers cannot depend on existing technique. They must have to look continually to improve the existing method to make testing more effective. But even all these hard work in testing we can never claim our product is bug free. To overcome this, test cases need to be regularly reviewed and revised add new and different test cases to help find more defects.
4. **Testing shows presence of defects:** Testing principle states that- testing talks about the presence of defects not about the absence of defect. Software testing reduces the probability of undiscovered defects remaining in the software but even if no defects found, it is not a proof of correctness.

But if we work hard, taking all precautions and make our software products 99% bug free. The software does not meet the needs and requirements of the client.

1. **Absence of error -fallacy:** This can be possible the software which is 99% bug free is still unusable. The case can be if the system is tested for the wrong requirement. Software testing is not finding the defects but also to check that software addresses the business needs. The absence of error is fallacy i.e. finding and fixing defects does not help if the system build is unusable and doesn't fulfill the user's needs and requirements.
2. **Early Testing:** Testing should start as soon as possible in the software development lifecycle. So that defects in the requirement or design phase captured in the early stages. It is cheaper to fix defect in the early stages of testing. We should start finding the bug at the moment the requirements are defined.
3. **Testing is context dependent:** Testing is context dependent that we test an e-commerce site will be different from the way we test the commercial. All the developed software's are not identical. We will use different methodology; techniques and type of testing depend on the application type.

### 11) What is API framework?

A framework or software framework is a platform for developing software applications. API framework is a foundation on which software developer can build applications for a specific platform.

**Example:** A framework can include predefined classes and functions that can be used to process input, manage hardware devices and interact with system software.

Framework is similar to an Application Programming Interface, technically framework includes API. Framework serves foundation for programming while API provides access to the elements supported by the framework. Framework also includes code libraries, compiler and other programs used in the software development process.

API framework is defined by configuration file which consists the list of all APIs that is required to be activated and activated for a particular program run.

**12) What are the common tests that performed on API?**

Here, are the common tests that performed on API are as:

1. Response of the API should be verified based on the request. We will verify that the return value is based on request.
2. When API is updating any data structure we should verify the system is authenticating the outcome.
3. We will verify whether the API is trigger other event or request another API.
4. We will verify the behavior of the API when no value is return.

**13) What exactly needs to verify in API testing?**

In API testing, we send a request to API with the known data and then analysis the response.

1. We will verify the accuracy of the data.
2. Will see the HTTP status code.
3. We will see the response time.
4. Error codes in case API returns any errors.
5. Authorization would be check.
6. Non-Functional testing such as performance testing, security testing.

### 14) What are the differences between API and Web Services?

|  |  |
| --- | --- |
| **API** | **WEBSERVICES** |
| All APIs are not web service. | All web services are API |
| API may or may not need network for its operations. | Web Services always need network for its operation. |
| API can perform all the operations which web service can't perform. | Web service can't perform all the operations like API. |
| API can be communicated through SOAP, REST, XML-RPC and CURL calls as well. API can also be exposed in number of ways like JAR, DLL, XML over HTTP, JSON over HTTP etc. | Web service can be communicated through SOAP |

**15) What is API documentation?**

A good documentation is must for any foundation. API documentation serves as quick reference for accessing library or working within a program.

When we use any such documents, it must consists of proper plan, content source, proper layout, information related to each function etc.

There are various documentation tools like Doxygen and JavaDoc. Here, are the functions which are documented which revolve around the parameters like:

* Function description
* Type and syntax of error message that may occur
* Syntax, elements and sequence needed for each parameter
* Links regarding functions

**16) What is the most used template for API documentation?**

Here, are the various documentation template that make the whole process simple and easy. They are:

* Swagger
* Miredot
* Slate
* FlatDoc
* API blueprint
* RestDoc
* Web service API specification

**17) What are the types of bug that can be found during API testing?**

API testing helps us to find many types of bugs which are:

* Stress
* Security
* Duplicate or missing functionality
* Reliability
* Unused flags
* Performance
* Incompatible error handling
* Multi-threaded issue
* Improper errors

### 18) What are the difference between API testing and UI testing?

UI (User Interface) testing means the testing of the graphical user interface. The focus of UI testing is on the look and feel of the application. In user interface testing the main focus is on how users can interact with app elements such as images, fonts, layout etc. are checked.

API testing allows the communication between two software systems. API testing works on backend also known as backend testing.

### 19) What is SOAP?

SOAP (Simple Object Access Control) . It is an XML based protocol that helps in exchanging information among computers.

### 20) What is REST API?

**REST API** is a set of function helps the developer’s performing requests when the response is receiving. Through HTTP protocol interaction is made in REST API.

REST is defined as Representational state transfer. It is an effective standard for API creation.

### 21) What are the differences between SOAP and REST API?

|  |  |
| --- | --- |
| **SOAP** | **REST API** |
| **SOAP** stands as Simple Object Access Protocol. | **REST** stands as Representational State Transfer. |
| **SOAP** is a protocol. | **REST** is an architectural pattern. |
| **SOAP** can work with XML format. In SOAP all the data passed in XML format. | **REST** permit different data format such as Plain text, HTML, XML, JSON etc. But the most preferred format for transferring data is in JSON. |

**22) What are the major challenges faced during API testing?**

The major challenges faced during the API testing are:

* Parameter Selection
* Parameter Combination
* Call sequencing
* Output verification and validation
* A major challenge is providing input values which are very difficult because GUI is not available.

### 23) What are the difference between API Testing and Unit Testing?

|  |  |
| --- | --- |
| **API** | **UNIT TESTING** |
| API testing is a form of black box testing. | Unit testing is a form of white box testing. |
| API testing is performed after the project completion during the test. | Unit testing is performed when the project is created. |
| In API testing there is a wide scope of testing. | In Unit testing there is a limited scope of testing we can test only the basic functionality. |
| API testing is done by the testers. The whole purpose of API testing is end to end testing of the functionality. | Unit testing is done by the developer. In unit testing every functionality is separately tested. |

**24) What is a RESTFUL web services?**

There are two kinds of web services

1. SOAP Web Services
2. RESTFUL Web Services

**1. SOAP (Simple Object Access Protocol) -** SOAP is a XML based method which is used in Web Services.

**2. RESTFUL Web Services -** **RESTful** Web Services are basically REST Architecture based Web Services. In REST Architecture everything is a resource. RESTFUL Web Services defines URI (Uniform Resource Identifier), and also provides resource representation like JSON and a set of HTTP method. **Facebook, Twitter, and Google** expose their functionality in the form of Restful web services. The underlying protocol for REST is HTTP. REST stands for Representational State Transfer.

**25) What is Resource in REST?**

REST architecture treats any content as resource, which can be text files, HTML pages, images, videos or dynamic business information. REST server gives the functionality to access the resources and modifies them. We can identify each resources by URIs/ global IDs.

**26) What is the way to represent the resource in REST?**

REST uses different representation to define the resources like text, JSON and XML. The most popular representation of resources is JSON and XML.

**27) What protocol is used by the RESTFUL Web Services?**

RESTFUL Web Services uses the HTTP protocol. They use the HTTP protocol as a medium of communication between the client and the server.

**28) What are the characteristics of REST?**

Here, are the two characteristics of REST.

1. REST is stateless. With the use of the REST API the server has no status, we can restart the server between two calls, in spite of all the data is transferred to the server.
2. Web Services uses POST method to perform operations, while REST uses GET method to access the resources.

**29) What is messaging in RESTFUL Web Services?**

RESTFUL Web Services use the HTTP protocol as a communication tool between the client and the server. This is the technique when the client sends a message in the form of HTTP request the server send back the HTTP reply which is called Messaging. This message consists message data and Meta data i.e. information on the message itself.

**30) What are the components of an HTTP request?**

An HTTP request have five components. These are:

1. **Action showing HTTP method** like GET, PUT, POST, DELETE.
2. **Uniform Resource Identifier (URI):** URI is the identifier for the resource on the server.
3. **HTTP version:** Indicate the HTTP version like- HTTP V1.1.
4. **Request Header:** Request Header carries metadata for the HTTP request message. Metadata could be a client type, format supported by the client, format of a message body, cache setting etc.
5. **Request Body:** Resource body indicates message content or resource representation.

**31) What is the HTTP protocol supported by REST?**

**GET:** GET is used to request data from the specified resource.

GET request can be cached and bookmark. It remains in the browser history and has length restriction. When dealing with sensitive data GET requests should not be used.

**POST:** POST is used to send data to server for creation or updating the resources.

POST requests are never cached or bookmark.

**PUT:** PUT replaces the current representation of the target resource with the request payload.

**DELETE:** DELETE removes the specified resource.

**OPTIONS:** OPTION is used to describe the communication option for the target resources.

**HEAD:** HEAD asks for response which is identical to GET requests, but without the response body.

**32) Can we use GET request instead of PUT to create a resource?**

PUT or POST method is used create a resource. GET is only used to request the resources.

**33) What is URI? What is the purpose of web-based service and what is it's format?**

URI stands for Uniform Resource Identifier. It is a string of characters designed for unambiguous identification of resources and extensibility by the URI scheme. The purpose of URI is to locate the resource on the server hosting of the web service.

A URIs format is ***<protocol>://<service-name>/<Resource Type>/<ResourceID>***

**34) What are SOAP Web Services?**

**SOAP** (*Simple Object Access Protocol*) is defined as the XML based protocol. SOAP is also known for developing and designing web services and also enable the communication between the applications developed on different platform by using different programming languages on the internet. SOAP is platform and language independent.

### 35) When we can use SOAP API?

We can use SOAP API to perform the operation on records like create, retrieve, update or delete. We can use API to manage password, perform searches etc.