**Summary of API Testing**

**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.

**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 it’s designed correctly, hopefully, your order won’t crash!

**A real example of an API**

How are APIs used in the real world? Here’s a very common scenario – **booking a flight.**

When you search for flights online, you have a menu of options to choose from. You choose a departure city and date, a return city and date, cabin class, and other variables like your meal, your seat, or baggage requests.

To book your flight, you need to interact with the airline’s website to access the airline’s database to see if any seats are available on those dates, and what the cost might be based on the date, flight time, route popularity, etc.

You need access to that information from the airline’s database, whether you’re interacting with it from the website or an online travel service that aggregates information from multiple airlines. Alternatively, you might be accessing the information from a mobile phone. In any case, you need to get the information, and so the application must interact with the airline’s API, giving it access to the airline’s data.

The API is the interface that, like your helpful waiter, runs and delivers the data from the application you’re using to the airline’s systems over the Internet. It also then takes the airline’s response to your request and delivers right back to the travel application you’re using. Moreover, through each step of the process, it facilitates the interaction between the application and the airline’s systems – from seat selection to payment and booking.

APIs do the same for all interactions between applications, data, and devices. They allow the transmission of data from system to system, creating connectivity. APIs provide a standard way of accessing any application data, or device, whether it’s accessing cloud applications like [Salesforce](http://mulesoft.com/integration-solutions/saas/salesforce), or shopping from your mobile phone.

**Types of APIs**

There are numerous types of APIs. For example, you may have heard of Java APIs, or interfaces within classes that let objects talk to each other in the Java programming language. Along with program-centric APIs, there are also Web APIs such as the Simple Object Access Protocol (SOAP), Remote Procedure Call (RPC), and perhaps the most popular—at least in name— Representational State Transfer (REST). There are 15,000 publicly available APIs, according to [Programmable Web](http://snip.ly/wwnim), and many thousands of more private APIs that companies use to expand their internal and external capabilities.

**API Testing**

An **API** or Application Programming Interface is a set of programming instructions for accessing a web-based software application. There are mainly 4 methods involve in **API Testing**like GET, POST, Delete, and PUT. We need to check response code, response message and response body inAPIi **Testing**.

**How is API testing done**

**API testing** is a type of software **testing** that involves **testing** application programming interfaces (**APIs**) directly and as part of integration **testing** to determine if they meet expectations for functionality, reliability, performance, and security. Since **APIs** lack a GUI, **API testing** is **performed** at the message layer.

**Tools used for API testing**

SoapUI is an automation **testing tool** for SOAP and REST **API**. You can perform functional **testing**, performance **testing**, security **testing**, and data-driven **testing**. It will provide you with the reports for **testing** and will allow you to export the data.

**API testing with postman**

Postman is an interactive and automatic tool for verifying the APIs of your project. Postman is a Google Chrome app for interacting with HTTP APIs. It presents you with a friendly GUI for constructing requests and reading responses. It works on the backend, and makes sure that each API is **working** as intended.

A **test** in **Postman** is fundamentally a JavaScript code, which run after a request is sent and a response has been received from the server. **POSTMAN** is very easy to use. It provides collection of **API** calls, and one has to follow that collection of **API**calls for **testing APIs** of application

**How to run API in postman**

**Enable authorization**

In Postman, select an API method.

Click the Authorization tab.

Choose OAuth 2.0 and add the following information from the table below.

Click Get access token.

Postman starts the authentication flow and prompts you to save the access token.

Select Add token to header.

**How to test an API**

**API Testing Best Practices**

Test for the typical or expected results first.

Add stress to the system through a series of API load tests.

Test for failure. ...

Group test cases by test category.

Prioritize API function calls so that it will be easy for testers to test quickly and easily.