What is API

Benefit of API

How to Test

https://smartbear.com/learn/api-testing/what-is-api-testing/

>>

**1) What is an API?**

API is a contract. A promise to perform described services when asked in specific ways.

**2) How is it used?**

According to the rules specified in the contract. The whole point of an API is to define how it's used.

**3) When and where is it used?**

It's used when 2 or more separate systems need to work together to achieve something they can't do alone.

API stands for Application Programming Interface, i.e. API is the way for an application to interact with certain system/application/library/etc.

For example, there are API's for OS (WinAPI), API's for other applications (like databases) and for specific libraries (for example, image processing), etc.

APIs are usually developed in a form consumable by a client application. For C/C++ applications, it a set header files and dynamic/static libraries. For Java - set of jars. And so on.

In layman's terms, I've always said an API is like a translator between two people who speak different languages. In software, data can be consumed or distributed using an API (or translator) so that two different kinds of software can communicate. Good software has a strong translator (API) that follows rules and protocols for security and data cleanliness.

**Where it is used**

An example, You are buying an item in online through your credit card. You will provide credit card details and press continue button. It will tell you whether your information is correct or not. To provide these results, there are lot of things in the background.

The application will send your credit card details to a remote application which will validate your information and send the result back your application. API is used in this scenario.

I think hope it helps for the beginners who doesn't understand really what API is.

With an API, the exact structure of request and response is documented upfront by[weather.com](http://weather.com/), and is likely to remain constant, regardless of whether the website changes its look and feel for human visitors.

**REST**

REST means REpresentational State Transfer; it is an architecture that generally runs over HTTP. The REST style emphasizes the interactions between clients and services, which are enhanced by having a limited number of operations. REST is an alternative to SOAP (Simple Object Access Protocol) and instead of using XML for request REST uses simple URL in some cases. Unlike SOAP, RESTFUL applications uses HTTP build in headers to carry meta-information.

There are various code that REST use to determine whether user has access to API or not like code 200 or 201 indicates successful interaction with response body while 400 indicates a bad request or the request URI does not match the APIs in the system. All API request parameters and method parameters can be sent via either**POST** or **GET** variables.

Rest API supports both XML and JSON format. It is usually preferred for [mobile](http://www.guru99.com/mobile-testing.html) and web apps as it makes app work faster and smoother

Postman:

deepakapi / Deepak

[my.request.1@yopmail.com](mailto:my.request.1@yopmail.com) / 123456

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9..RHK4FAAMk3kIWtZpuKPcROIihQm1KgqVOlFOKSH71Jo

POST data in table

Authorization

Bearer {eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9..RHK4FAAMk3kIWtZpuKPcROIihQm1KgqVOlFOKSH71Jo}

## Methods of Sending Information to Server

A web browser communicates with the server typically using one of the two HTTP (Hypertext Transfer Protocol) methods — GET and POST. Both methods pass the information differently and have different advantages and disadvantages, as described below.

**GET and POST:**

http://java67.blogspot.in/2014/08/difference-between-post-and-get-request.html

## The GET Method

In GET method the data is sent as URL parameters that are usually strings of name and value pairs separated by ampersands (&). In general, a URL with GET data will look like this:

[http://www.example.com/action.php?**name**=*john*&**age**=*24*](http://www.example.com/action.php?name=john&age=24)

## Advantages and Disadvantages of Using the GET Method

* Since the data sent by the GET method are displayed in the URL, it is possible to bookmark the page with specific query string values.
* The GET method is not suitable for passing sensitive information such as the username and password, because these are fully visible in the URL query string as well as potentially stored in the client browser's memory as a visited page.
* Because the GET method assigns data to a server environment variable, the length of the URL is limited. So, there is a limitation for the total data to be sent.

## The POST Method

In POST method the data is sent to the server as a package in a separate communication with the processing script. Data sent through POST method will not visible in the URL.

## Advantages and Disadvantages of Using the POST Method

* It is more secure than GET because user-entered information is never visible in the URL query string or in the server logs.
* There is a much larger limit on the amount of data that can be passed and one can send text data as well as binary data (uploading a file) using POST.
* Since the data sent by the POST method is not visible in the URL, so it is not possible to bookmark the page with specific query.

But, even if you are not following RESTful principles, it can be useful to think in terms of using GET for retrieving / viewing information and POST for creating / editing information.

You should never use GET for an operation which alters data. If a search engine crawls a link to your evil op, or the client bookmarks it could spell big trouble.

Use GET if you don't mind the request being repeated (That is it doesn't change state).

Use POST if the operation does change the system's state.

GET: Usually used for submitted search requests, or any request where you want the user to be able to pull up the exact page again.

Advantages of GET:

* URLs can be bookmarked safely.
* Pages can be reloaded safely.

Advantages of POST:

* Name-value pairs are not displayed in url. (Security += 1)
* Unlimited number of name-value pairs can be passed via POST.

Disadvantages of POST:

* Page that used POST data cannot be bookmark. (If you so desired.)

The first important thing is the *meaning* of GET versus POST :

* GET should be used to... get... some information **from** the server,
* while POST should be used to send some information **to** the server.

After that, a couple of things that can be noted :

* Using GET, your users can use the "back" button in their browser, and they can bookrmark pages
* There is a limit in the size of the parameters you can pass as GET *(2KB for some versions of Internet Explorer, if I'm not mistaken)* ; the limit is much more for POST, and generally depends on the server's configuration.

**aim trying to get Yahoo weather Api with Celsius temperature i did added &u=c but its still in Fahrenheit**

**this is the url i'm using**

<http://stackoverflow.com/questions/21092164/yahoo-weather-api-celsius>

<http://query.yahooapis.com/v1/public/yql?q=select%20item%20from%20weather.forecast%20where%20location=%22LEXX0003%22&format=json&u=c>

#### PLAYER 1: JSON [LINK](https://www.smashingmagazine.com/2012/02/beginners-guide-jquery-based-json-api-clients/#player-1-json)

[JSON](http://www.json.org/) (or JavaScript Object Notation) is a lightweight, easy and popular way to exchange data. jQuery is not the only tool for manipulating and interfacing with JSON; it’s just my and many others’ preferred method.

A lot of the services we use everyday have JSON-based APIs: Twitter, Facebook and Flickr all send back data in JSON format.

# Fetching JSON data from REST APIs

<https://cran.r-project.org/web/packages/jsonlite/vignettes/json-apis.html>

This section lists some examples of public HTTP APIs that publish data in JSON format. These are great to get a sense of the complex structures that are encountered in real world JSON data. All services are free, but some require registration/authentication. Each example returns lots of data, therefore not all output is printed in this document.

## Github

Github is an online code repository and has APIs to get live data on almost all activity. Below some examples from a well known R package and author:

hadley\_orgs <- fromJSON("https://api.github.com/users/hadley/orgs")

hadley\_repos <- fromJSON("https://api.github.com/users/hadley/repos")

gg\_commits <- fromJSON("https://api.github.com/repos/hadley/ggplot2/commits")

gg\_issues <- fromJSON("https://api.github.com/repos/hadley/ggplot2/issues")

Paste the above url in the url and test with Rest Console:

### JSON Webservices

GeoNames offers most webservices in XML and JSON format. JSON has the advantage that is can be used to directly access the geonames webservice from javascript code, whereas using XML the browser will throw a security exception if an xml call to another webserver is made.

All JSON services accept an optional parameter 'callback' for a javascript function call and a parameter 'formatted=true' to format the output with linefeeds and indentation. The latter is useful to view the JSON result in a browser but should not be used in production usage (waste of bandwith).

JSON Examples :  
[Placename autocomplete](http://www.geonames.org/export/ajax-postalcode-autocomplete.html)   
[full text search on google maps](http://www.geonames.org/maps/json-googlemaps-example.html)

#### Places

##### Cities and Placenames

Webservice Type : REST   
Url : api.geonames.org/citiesJSON?  
Parameters :   
north,south,east,west : coordinates of bounding box   
callback : name of javascript function (optional parameter)   
lang : language of placenames and wikipedia urls (default = en)  
maxRows : maximal number of rows returned (default = 10)  
  
Result : returns a list of cities and placenames in the bounding box, ordered by relevancy (capital/population). Placenames close together are filterered out and only the larger name is included in the resulting list.  
  
Example : <http://api.geonames.org/citiesJSON?north=44.1&south=-9.9&east=-22.4&west=55.2&lang=de&username=demo>   
  
This service is also available in XML output :   
Example : <http://api.geonames.org/cities?north=44.1&south=-9.9&east=-22.4&west=55.2&username=demo>

##### Wikipedia Fulltext Search

Webservice Type : XML or JSON   
Url : api.geonames.org/wikipediaSearch?  
api.geonames.org/wikipediaSearchJSON?  
Parameters : q : place name ([urlencoded utf8](http://forum.geonames.org/gforum/posts/list/8.page))  
title : search in the wikipedia title (optional)  
lang : language code, supported languages are de,en,es,fr,it,nl,pl,pt,ru,zh (default = en)  
maxRows : maximal number of rows returned (default = 10)  
Result : returns the wikipedia entries found for the searchterm as xml document   
Example <http://api.geonames.org/wikipediaSearch?q=london&maxRows=10&username=demo> 

**Example:**

<http://www.geonames.org/export/ws-overview.html>

**JSON**, or JavaScript Object Notation, is a minimal, readable format for structuring data. It is used primarily to transmit data between a server and web application, as an alternative to XML.

**Benefit:**

JSON stands for JavaScript Object Notation, and it's a text format that makes it easy to share data between devices like clients and servers.

Because it is smaller and easier to convert into a data structure, it's a great alternative to other formats like XML.

 One of the advantages to using JSON is how easy it is to read. JSON uses minimal formatting--really just a few special characters in addition to the data. Another advantage to JSON is that it's super easy to parse.

 This means that JSON data will take less space and load faster into your web applications. Plus, parsing an XML object can be complicated and time consuming, whereas JSON is easily mapped into a JavaScript object and so it takes less time to process.

**REST:**

<http://www.javatpoint.com/soap-vs-rest-web-services>

REST is an **architectural style**.

REST stands for **REpresentational State Transfer**.

REST **can use SOAP** web services because it is a concept and can use any protocol like HTTP, SOAP.

REST **uses URI to expose business logic**.

**JAX-RS** is the java API for RESTful web services.

REST does not define too much standards like SOAP.

REST **requires less bandwidth** and resource than SOAP.

RESTful web services **inherits security measures** from the underlying transport.

REST **permits different** data format such as Plain text, HTML, XML, JSON etc.

REST **more preferred** than SOAP.

**REST** describes a set of architectural principles by which data can be transmitted over a standardized interface (such as HTTP). REST does not contain an additional messaging layer and focuses on design rules for creating stateless services. A client can access the resource using the unique [URI](http://searchsoa.techtarget.com/definition/URI) and a representation of the resource is returned. With each new resource representation, the client is said to transfer state. While accessing RESTful resources with HTTP protocol, the URL of the resource serves as the resource identifier and GET, PUT, DELETE, POST and HEAD are the standard HTTP operations to be performed on that resource.

**What is API:**

API stands for Application Programming Interface, i.e. **API is the way for an application to interact with certain system/application/library/etc.**

In layman's terms, I've always said an **API is like a translator between two people who speak different languages. In software,** data can be consumed or distributed using an API (or translator) so that two different kinds of software can communicate. Good software has a strong translator (API) that follows rules and protocols for security and data cleanliness.

**Where it is used**

An example, You are buying an item in online through your credit card. You will provide credit card details and press continue button. It will tell you whether your information is correct or not. To provide these results, there are lot of things in the background.

The application will send your credit card details to a remote application which will validate your information and send the result back your application. API is used in this scenario.

**API Vs Web Services:**

Web services are a specific subset of APIs.   
  
APIs can have many forms, whether they communicate over the Web, over other Internet protocols, or even inter-process on the same machine, or inside the same machine and process.

A Web service uses only three styles of use: SOAP, REST and XML-RPC for communication whereas API may use any style for communication.

**API:**An application programming interface (API) is a set of routines, data structures, object classes and/or protocols provided by libraries and/or operating system services in order to support the building of applications.  
 **Webservice:**  
A Web Service is defined by the W3C as "a software system designed to support interoperable machine-to-machine interaction over a network"  
  
Clearly, both are means of communications. The diference is that Web Service almost always involves communication over network and HTTP is the most commonly used protocol. Web service also uses SOAP, REST, and XML-RPC as a means of communication. While an API can use any means of communication e.g. DLL files in C/C++, Jar files/ RMI in java, Interrupts in Linux kernel API etc.   
So, you can say that-

Web Service. A web service is a collection of APIs working together to perform a particular task.

**Web service** can be accessed using a transport protocol. HTTP is a far more popular transport protocol to send a request and get a response to and forth from a web service. Using a web service does requires us to be online in the first place.

1. Web Service is an API wrapped in HTTP.  
   2. All Web Services are API but APIs are not Web Services.  
   3. Web Service might not perform all the operations that an API would perform.  
   4. A Web Service needs a network while an API doesn't need a network for its operation.

**Benefit of API Testing :**

**Test for Core Functionality**  
The first major advantage of API testing is access to the application without a user interface.

**Time Effective**

**Reduced Testing Costs**

### Reduced Technical Debt

### Earlier Remediation

## Reduce Risks

### Web Services Example:

**For example: MAP**

<http://maps.googleapis.com/maps/api/staticmap?center=Sydney,NSW&zoom=14&size=400x400&sensor=false>

### 

**Most often-used types of web service:**

* SOAP (Simple Object Access Protocol)
* XML-RPC
* JSON-RPC
* **REST**

### What is REST:

**REST means REpresentational State Transfer; it is an architecture that generally runs over HTTP.** The REST style emphasizes the interactions between clients and services, which are enhanced by having a limited number of operations. **REST is an alternative to SOAP (Simple Object Access Protocol) and instead of using XML for request REST uses simple URL in some cases.** Unlike SOAP, RESTFUL applications uses HTTP build in headers to carry meta-information.

**REST provides a lighter weight alternative. Instead of using XML to make a request**

REST can use four different HTTP 1.1 verbs (GET, POST, PUT, and DELETE) to perform tasks

You can find REST-based Web services that output the data in Command Separated Value (CSV), JavaScript Object Notation (JSON) and Really Simple Syndication (RSS).

Rest API supports both XML and JSON format. It is usually preferred for [mobile](http://www.guru99.com/mobile-testing.html) and web apps as it makes app work faster and smoother.

[JSON](http://www.json.org/) (or JavaScript Object Notation) is a lightweight, easy and popular way to exchange data. jQuery is not the only tool for manipulating and interfacing with JSON; it’s just my and many others’ preferred method.

A lot of the services we use everyday have JSON-based APIs: Twitter, Facebook and Flickr all send back data in JSON format.

### 

### Free Web Services:

### <http://www.geonames.org/export/ws-overview.html>

### API Testing :

### Example : <http://api.geonames.org/citiesJSON?north=44.1&south=-9.9&east=-22.4&west=55.2&lang=de&username=demo>  This service is also available in XML output :  Example : <http://api.geonames.org/cities?north=44.1&south=-9.9&east=-22.4&west=55.2&username=demo>

### Sample API:

### http://reqres.in/

### [my.request.1@yopmail.com](mailto:my.request.1@yopmail.com) / 123456