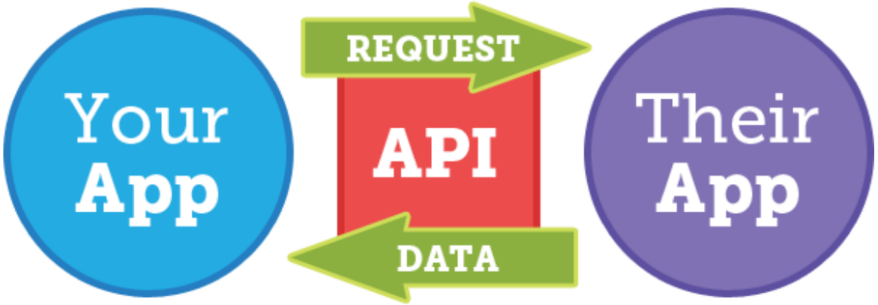
Application programming interface (API)

What is An API and How Does It Work?

Most people have never heard of API much less what it does. They probably think it is a cool new acronym that makes their texting easier like LOL or LMAO.

They don’t know that it is a powerful technology capable of achieving great things. It is being used today in applications we use every day. Companies like Yelp use it to locate nearby restaurants. People use APIs all the time and they don’t even know it. Whenever you want to share a blog post or article or recipe and click on the Facebook, Twitter, or LinkedIn little icon usually at the top or on the sidebar to share that piece of information then you are using an API.

So now that you know a little of how APIs are used all the time let’s find out what exactly an API is.



What is an API?

API is an acronym for Application Programming Interface that software uses to access data, server software, or other applications and has been around for quite some time.

In layman's terms, it is a software intermediary that allows two applications to talk to each other. Think of an API as a translator between two people who don't speak the same language but can communicate using a go-between

APIs are very versatile and can be used on web-based systems, operating systems, database systems, and computer hardware.

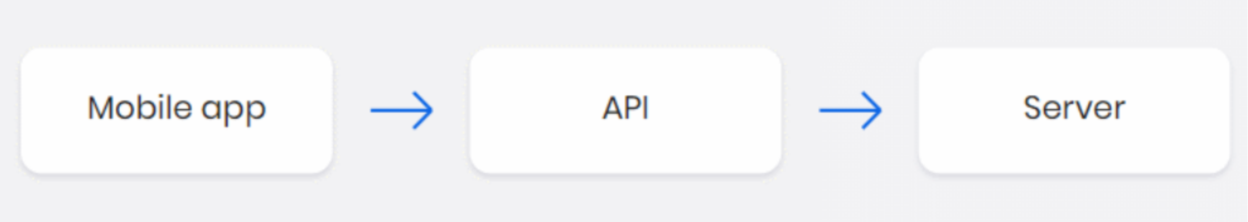
Developers use APIs to make their jobs more efficient by reusing code from before and only changing the part that is relevant to the process they want to improve. A good API makes it easier to create a program because the building blocks are in place. APIs use defined protocols to enable developers to build, connect and integrate applications quickly and at scale.

Now that you have a better understanding of what an API is let's look at how they work.

How Does An API Work?

APIs communicate through a set of rules that define how computers, applications, or machines can talk to each other. The API acts as a middleman between any two machines that want to connect with each other for a specified task.

A simplified example would be when you sign into Facebook from your phone you are telling the Facebook application that you would like to access your account. The mobile application makes a call to an API to retrieve your Facebook account and credentials. Facebook would then access this information from one of its servers and return the data to the mobile application.

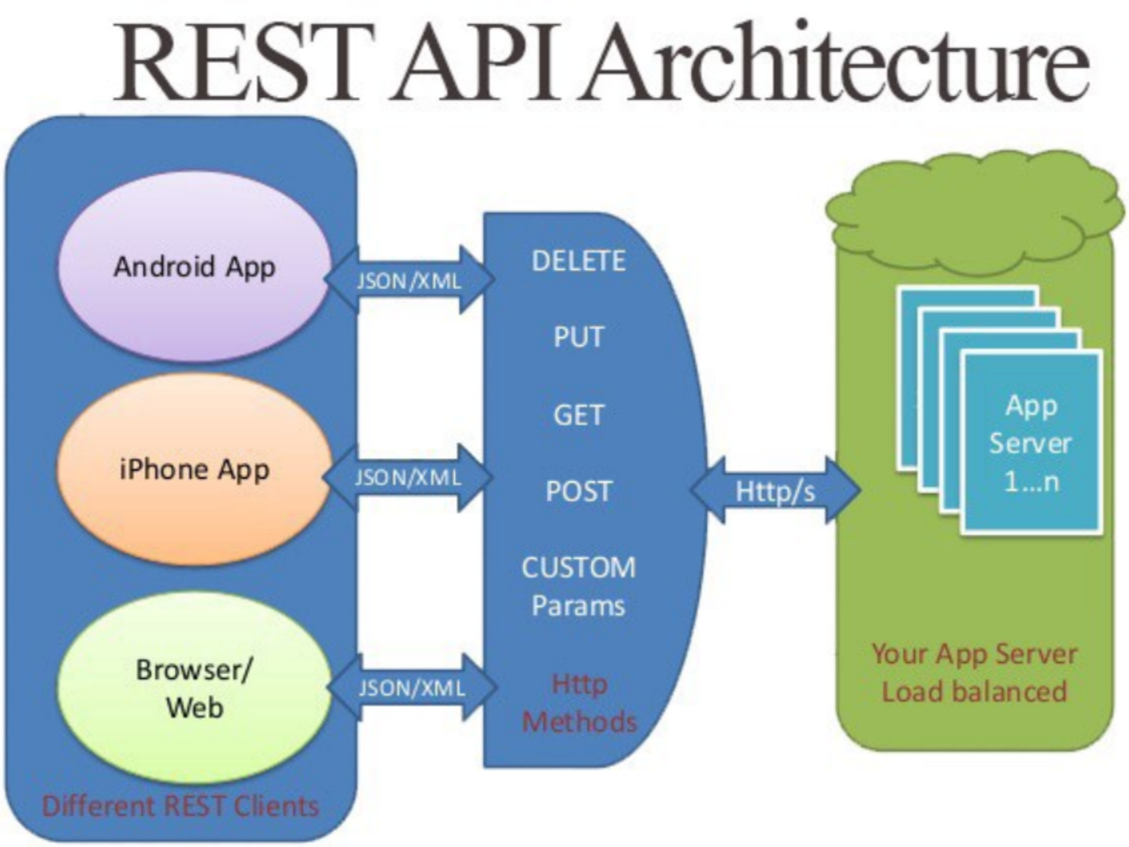


These type of APIs called web APIs are the most common but limited just to the web. There are APIs for virtually every machine or system that expects to interact with other machines or systems.
APIs have been around for a long time but only recently have gained in popularity. Companies use this technology to gain an edge over others by finding more efficient ways to retrieve information faster to serve the customer.
As I mentioned earlier web APIs are the not only ones that exist. We will dive into a few of them in the next section.

Types of APIs

Web APIs are what gets utilized the most but there are others that exist that you should know about. These are just as reliable and efficient as the web API but not well known.

REST (RESTful) API — stands for representational state transfer and delivers data using the lightweight JSON format. Most public APIs use this because of its fast performance, dependability and ability to scale by reusing modular components without affecting the system as a whole.



This API gives access to data by using a uniform and predefined set of operations. REST APIs are based on URLs and the HTTP protocol and are based on these 6 architectural constraints:

1. Client-server based —
The client handles the front end process while the server handles the backend and can be both replaced independently of each other.

2. Uniform interface — defines the interface between client and server and simplifies the architecture to enable each part to develop separately.

3. Stateless — each request from client to server must be independent and contain all of the necessary information so that the server can understand and process it accordingly.

4. Cacheable — maintains cached responses between client and server avoiding any additional processing

5. Layered system — layers are arranged hierarchically so that each one can only 'see' the corresponding layer they are interacting with.

6. Code-on-demand — allows client functionality to be extended by downloading and executing code in the form of applets and scripts. This simplifies clients by reducing the number of features required to be pre-implemented.
Once you follow these defined constraints the API you create is said the be RESTful.

SOAP — Simple Object Access Protocol is a little more complex then REST because it requires more upfront about how it sends its messages. This API has been around since the late 1990s and uses XML to transfer data. It requires strict rules and advanced security that requires more bandwidth.
This protocol does not have the ability to cache, has strict communication and needs every piece of information about an interaction before any calls are even thought of to be processed.

XML-RPC — known as extensible markup language — Remote Procedure Calls. This protocol uses a specific XML format to transfer data and is older and simpler than SOAP. A client performs a RPC by sending an HTTP request to a server that implements XML-RPC and receives the HTTP response.

JSON-RPC — is very similar to XML-RPC in that they work the same way except that this protocol uses JSON instead of XML format. The client is typically software that calls on a single method of a remote system.

Conclusion

This was a quick introduction to this foreign acronym that many people do not know about. They may have heard the term in passing and were not clear on what this technology exactly is. We discussed what they are, how they work and even a fee of the different types in use today.

APIs have been around for a long time and will continue to be as long as businesses utilize them as part of their operations. They are very versatile and quick to implement. We should not be intimidated by APIs now that we have a little better understanding of what they are and how they work.