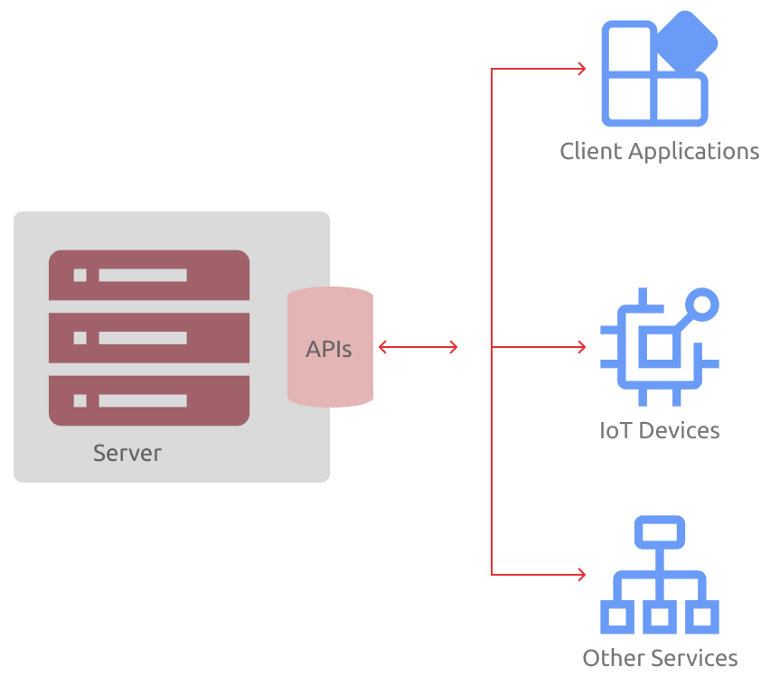
**What is an API? -** [Sahil Fruitwala](https://sahil-fruitwala.medium.com/?source=post_page-----b53f95aa79a7--------------------------------) Apr 22, 2022

Application Programming Interface or API, the term every developer has heard and used. From sending a WhatsApp message to playing a YouTube video, every application uses API(s). You might hardly find any application that uses the internet and doesn’t use API(s). So, you might be wondering, “**what are these APIs?”**

**What is an API?**

APIs are just pieces of code that help two server applications (backend) or one client application(frontend) and one server application to communicate. This communication is essential for any modern application to work. Therefore, APIs are considered the backbone of the latest software and/or services.

If I have to say in simple words what APIs do, I will say, APIs are just services that return some kind of data on demand. This data can be anything such as an image file, sound file, raw text, etc.



The basic working of an API

Here, I have illustrated the basic architecture overview of how an API works. As I mentioned earlier and have shown in the picture, any system, service and software can use APIs.

**Explanation of how API works for 7-year olds**

Let’s assume the server that serves us APIs is a pizza parlour. Now, what do we do when we want a specific kind of pizza?

*Right! We call a waiter and either we will ask for a menu or will order another pizza.*

The same goes for APIs as well. We call an API endpoint[\*](https://www.notion.so/What-is-an-API-c36a9b2038f84d6cbbaed6d16d1db3d9) and request it to get some data or validate some data for us.

So, after one request is served, what now?

*Correct! Either we ask for a check or more pizzas. It can be the same or different kinds of pizza.*

Now, let’s look at this in the context of an API. After one request is served, either we can do nothing or request API to do another task (we can do nothing means there is no need for another request).

*At most of the places, instead of the fixed pizzas, we can order customized pizzas.*

We can do the same for APIs as well. If API has the capabilities, we can ask for specific data (customized data) or operate on particular data. For example, instead of asking for all user data, we can ask for the user data with a specific ID. Or we can request API to delete the data of a user with a particular ID. Some APIs provide certain functionalities, whereas some don’t. It all depends on how APIs are designed.

*This is just a top-level explanation.* In real-life API, many things are going on underneath that we can discuss in another blog.

**Types of API. The Past, Present and the Future!**

Throughout the history of APIs, there have been mainly three kinds.

1. RPC (Remote Procedure Call)
2. SOAP (Simple Object Access Protocol)
3. REST (Representational State Transfer)

**RPC** or **Remote Procedure Call** APIs seem to be the most **simple and ancient** APIs. In general, RPC requires developers or clients to run a block of code on a server. RPC-style makes it possible for developers to easily create and invoke multiple programs and services on a server. Nowadays, you won’t see much use of RPC APIs. There are two sub-type of RPC APIs:

* **XML-RPC:** A XML-RPC is basically an RPC API coded in return XML formatted data.
* **JSON-RPC:** A JSON-RPC is basically an RPC API coded to return JSON formatted data.

**SOAP** or **Simple Object Access Protocol** uses XML format to transfer data over HyperText Transfer Protocol (HTTP). SOAP uses Web Services Definition Language(WSDL) to publish the definition of its interface. Before the REST APIs came into the picture, SOAP had dominance on how APIs were created back then. SOAP was the first API protocol that can be created and invoked on any kind of operating system and, two decades ago, it was a huge deal.

**REST** or **Representational State Transfer** is a modern web services API. REST follows an architectural style instead of a generic protocol style. The main benefit of this style is it detaches consumers and providers. It means, because of its architectural style, consumers can use it without worrying about its underlying protocols. REST APIs support both JSON and XML formats. But most developers prefer JSON format as it is easy to use and read. Currently, REST is the most popular type of API. If you research, you will find that Netflix, Amazon, Google or any other giant tech company uses REST APIs in their products.

**Accessibility of APIs**

Not every API is a public API. There are many private and partner APIs as well. **Public APIs** are the APIs that can be accessed by any person in the world. It can be free, paid or on a subscription base. **Private APIs** can be developed by the organization for internal uses. Private APIs can be used in products that are available to the public. **Partner APIs** are the kind of APIs whose access is restricted to some user, or these APIs can be used with some Applications or services only. For example, Twitter API is a public API. But, there might be some APIs that Twitter is using for some internal tasks. Now, if some organization has created some API that can only be used with certain services of other organizations, then it will be called partner API.

**General Use of APIs and Basic Functionalities of APIs**

As a developer, we use and develop APIs for many purposes and for many use-cases. Some of those use-cases are:

1. To fetch data from the server (Database)
2. To add data to the Database
3. To update record (data) in the Database
4. To delete record in the Database
5. To validate a user or for authentication
6. To request other services (s) to perform some operations

***Note:*** According to [RapidAPI](https://rapidapi.com/blog/api-glossary/endpoint/" \t "_blank), the API endpoint is the point of entry in a communication channel when two systems are interacting. It means that the API endpoint is a URI through which you can access API or communicate with the server.

**Conclusion**

Now, there is a lot about APIs that can only be understood by writing the code. There can be many other types of APIs, and there can be many functionalities that I might have missed. Mention everything in the comment section that I might have missed. So others can get to know and understand.

That was it. Thank you for reading. I hope now you know **What is API?**

If you like this blog, let me know in the comment section. Let me know if you need any help or want to discuss something.