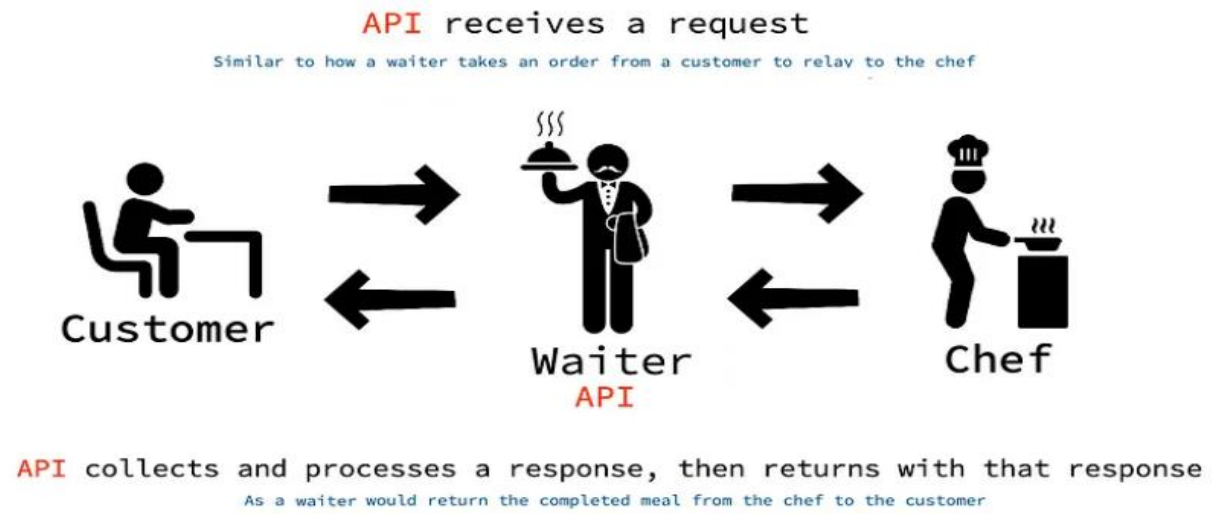
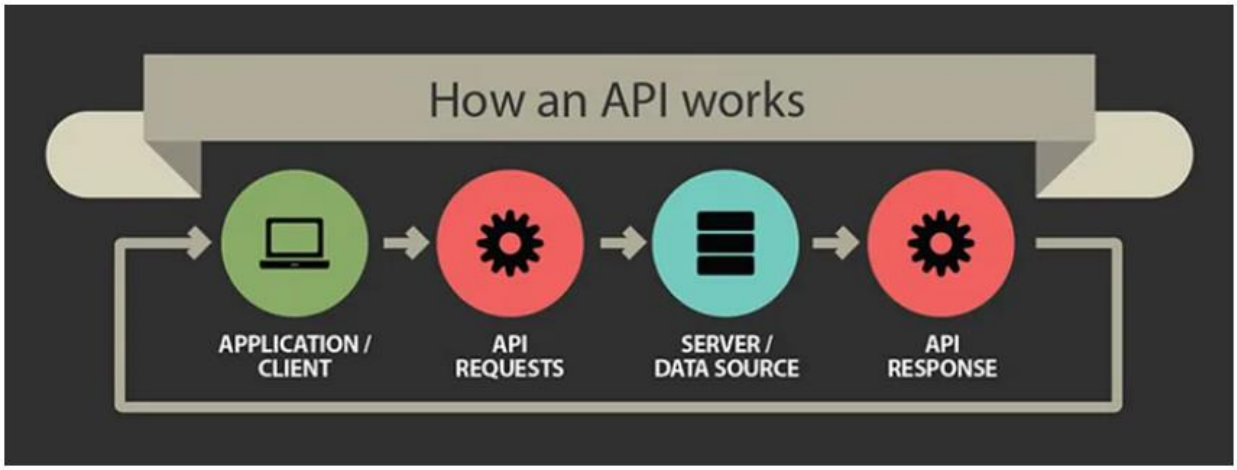
What exactly is an API?

API is an acronym for Application Programming Interface. It is defined as a system of tools and resources in an operating system which enable developers to create software applications. Typically, APIs use JSON (Javascript Object Notation) as the main language which provides methods designed to extract and manipulate data stored in HTML documents.





Purposes of API

1. **APIs makes life easier for developers —** Developers can easily add functionality.
2. **APIs control access to resources.**
3. **APIs are used for communication between services —** We can use APIs to access many online services from our application.

Types of APIs

Based on the release policies APIs can be divided into 3 as :

1. Private - designed for improving solutions and services within an organization.
2. Public - These are external APIs. They are available for third-party developers.
3. Partner - used by business partners who have reached an agreement with the publisher.

Use cases of APIs

1.Database APIs - allow communication between an application and a database management system.

2.Operating System APIs - his collection of APIs specifies how programs interact with operating system tools and services. Every operating system has its own series of APIs, such as the Windows API or the Linux API

3.Remote APIs -Remote APIs allow developers to communicate with remote services using protocols.

4. Web APIs — This is the most common use case of APIs. Developers use these Web APIs to increase the functionality of their applications. These APIs represent Client-Server architecture. These APIs predominantly use Hypertext Transfer Protocol (http) to deliver requests from web applications and responses from servers.

API Specifications

The number of programming languages and frameworks increase day by day. It is not possible for a developer to know all these languages. Therefore in order to handle the API requests and response, there should be some specifications or protocols. These API specifications provide the ability for diverse systems to seamlessly communicate with each other.

There are 4 main protocols. They are :

1. Remote Procedure Calls (RPC)
2. Service Object Access Protocol (SOAP)
3. Representational State Transfer Protocol (REST)
4. GraphQL

1. **Remote Procedure Calls (RPC) -** It has a straight forward interaction between a client and a server. The client remotely calls a method in the server and the server executes the method.
2. **Service Object Access Protocol (SOAP)-** OAP is a simple protocol for sharing structured data in a distributed, decentralized environment. It enables XML messaging between systems through HTTP. SOAP is most widely used in corporate web-based applications to guarantee the security of data. Payment gateways, identity protection, and CRM solutions, as well as financial and telecommunication services, all prefer SOAP APIs.

3. **Representational State Transfer (REST)-** REST is a software architecture style that has six constraints for creating HTTP-based programs, such as web services.

   REST is seen as a more user-friendly alternative to SOAP , REST makes data available as resources. Each resource has a unique URI and one can use this resource by specifying the URL. These resources can be modified by using the HTTP verbs such as GET to retrieve information, PATCH to update information, POST to insert data and DELETE to delete the resource.

   RESTful systems can send messages in a range of formats, including plain text, Javascript, YAML, XML, and JSON, while SOAP can only send messages in XML.

   REST is schemaless and the server decides how the data is returned, therefore we can't retrieve only certain information from the resources.

4. GraphQL - It has a schema and the client decides how the data is returned which means that the client can retrieve only the required data. GraphQL is initially created by Facebook for its internal use and it has a more efficient data loading due to increased mobile adoption. GraphQL is an API query language. It enables the client to choose the exact data it requires and simplifies data aggregation from various sources, allowing the developer to make only one API call to obtain all of the information required.