[API](https://www.mulesoft.com/platform/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.

An API defines functionalities that are independent of their respective implementations, which allows those implementations and definitions to vary without compromising each other. Therefore, [a good API](https://blogs.mulesoft.com/biz/anypoint-platform/a-good-api-strategy-can-help-turn-your-data-into-revenue/) makes it easier to develop a program by providing the building blocks.

When developers create code, they don’t often start from scratch. [APIs enable developers](https://www.mulesoft.com/resources/api/development-best-practices) can make repetitive yet complex processes highly reusable with a little bit of code. The speed that APIs enable developers to build out apps is crucial to the current pace of application development.

Developers are now much more productive than they were before when they had to write a lot of code from scratch. With an API they don’t have to reinvent the wheel every time they write a new program. Instead, they can focus on the unique proposition of their applications while outsourcing all of the commodity functionality to APIs.

**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 the [API is designed correctly](https://www.mulesoft.com/lp/whitepaper/api/design-apis), hopefully, your order won’t crash!

API endpoint

**Featured snippet from the web**

In simple terms, an **API endpoint** is the point of entry in a communication channel when two systems are interacting. It refers to touchpoints of the communication between an **API** and a server.

Paths and Parameters

Narrow down on a specific piece of data. Ex. Joke API can be configured to receive the kind of Joke we want. Using Joke API, we can have a specific path for a category of joke.

For ex.  **paths** are endpoints (resources), such as /users or /reports/summary/, that your API exposes, and **operations** are the HTTP methods used to manipulate these paths, such as GET, POST or DELETE.

Paths

API paths and operations are defined in the global paths section of the API specification.

1. paths:
2. /ping:
3. ...
4. /users:
5. ...
6. /users/{id}:
7. ...

All paths are relative to the [API server URL](https://swagger.io/docs/specification/api-host-and-base-path/). The full request URL is constructed as <server-url>/path. Global servers can also be overridden on the path level or operation level (more on that [below](https://swagger.io/docs/specification/paths-and-operations/#overriding-servers)). Paths may have an optional short summary and a longer description for documentation purposes. This information is supposed to be relevant to all operations in this path. description can be [multi-line](http://stackoverflow.com/questions/3790454/in-yaml-how-do-i-break-a-string-over-multiple-lines) and supports [Markdown](http://commonmark.org/help/) (CommonMark) for rich text representation.

1. paths:
2. /users/{id}:
3. summary: Represents a user
4. description: >
5. This resource represents an individual user in the system.
6. Each user is identified by a numeric `id`.
7. get:
8. ...
9. patch:
10. ...
11. delete:
12. ...

Parameters

Help to narrow down the criteria

Go after the URL of endpoint or path and then after it, follows a question mark.

Ex. [www.abc.com/apiv/xyz\_api\_name/Path\_name?contains=”keyword’”\](http://www.abc.com/apiv/xyz_api_name/Path_name?contains=”keyword’”\)

Here contains is a parameter

Contains=”query” is called a key value relationship

Parameters can be more than one and each one of it is separated by an ampersand symbol &.

API Authentication

It can be used for a variety of purposes. You can use an API Authentication formonitoring the usage of API such as the number of requests made, when are they made, through which platform. Allt his can be measured using an authentication developer account and then measure the statistics. This helps in variety of situations. For ex. The OpenWeather API

# The OpenWeather API

We can use this API for telling weather updates.

# Mailchimp API

This API can be used for collecting user email addresses using sign up forms.

Both the projects and their info is visible in the source code.