# APPLICATION PROGRAMMIN INTERFACES(API)

**Basic Definition:**

An API is a service provided by an application to other external applications to exchange sensitive data. Basically it is someone else's code but have legally given permission to use it in our application through an API.

**Purpose & Use:**

The stated purpose of an API was simply to create a standard that allows two servers to communicate and exchange data anywhere in the world. APIs are sets of requirements that govern how one application can talk to another. They make it possible to move information between programs. On the web, APIs make it possible for sites to let other apps and developers use their data for their own applications and purposes. They work by "exposing" some limited internal functions and features so that applications can share data without developers having direct access to the behind-the-scenes code.

**Basic Understanding:**

A good example for this would be a Weather App. We don't have enough resources to go all over the world and places sensors to get that kind of information but someone or some government organisation already has such resources and have installed it and also have written code to retrieve all the information from those sensors and are also willing to share that information to other developers through an API to build their own application either for free or with some cost involved. All we need is that information, using the API provided by them we can build our own weather app and display that to our users, we don't care about all that code written to retrieve weather data.

Some websites already know that we want their data and want to help us out. Twitter, for example, figures we might want to track some social metrics, like tweets, mentions, and hashtags. They help us out by providing developers with an API.

**Challenges faced building an API:**

Some sites just may not be able to develop their own APIs, or may not have the capacity to support or maintain them. Some other challenges that might prevent sites from developing their own APIs include:

- Security – APIs may provide sensitive data that shouldn't be accessible by everyone. Protecting that data requires upkeep and development know-how.
- Support – APIs are just like any other program and require maintenance and upkeep over time. Some sites may not have the manpower to support an API consistently over time.
- Mixed users – Some sites develop APIs for internal use, others for external. A mixed user base may need more robust APIs or several, which may cost time and money to develop.

**Limitations:**

Just because an API is available doesn't mean it always will be. Twitter, for example, limited third-party applications' use of its APIs. Companies have also shut down services and APIs in the past, whether because they go out of business, want to limit the data other companies can use, or simply fail to maintain their APIs. Google regularly shuts down their APIs if they find them to be unprofitable. Two examples of which include the late Google Health API and Google Reader API.

While APIs can be a great way to gather data quickly, they're just not reliable. The biggest issue is that sites have complete control over their APIs. They can decide what information to give, what data to withhold, and whether or not they want to share their API externally. This can leave plenty of people in the lurch when it comes to gathering necessary data to run their applications or inform their business.