Understanding the Basics of APIs

Introducing APIs

An application programming interface (API) is how two or more computer programs communicate with each other. APIs are very important because they increase the speed, quality, and functionality behind every digital touchpoint that occurs. An API specification is a document that describes how to establish and use a connection or interface.

How Do APIs work

An API is essentially protocols, a set of rules, which dictate how computer programs communicate with one another. This means APIs connect machines, or computer programs together; it is not intended to be used to communicate with an end-user. An API often is made up of various parts, which act like tools or services that are used and available to a programmer writing the API. It is also common to have more than one API working together to achieve a larger goal.

Most web APIs are positioned between the application and the web server. The web user initiates an API call that tells the application to execute an action. Then, the application will use an API to request the web server to do something. The API, in this context, is the middleman that sits between the application and the web server. The API call is the request.

Example of an API

First, let us explore a simple API-like protocol. You want to know what the weather is like outside but have no intention of physically going outside. You call your friend on the phone to inquire about the weather outside. Your friend asks who is calling to make sure you are not a random stranger; this is the verification or authorization. You ask him the question, and he answers the question. The exchange is over, and you both hang up the phone.

Now, let us explore the same weather through an app and API. You, as the end-user, want to know the weather outside. You pick up the phone and open the weather app on your phone. The weather bureau's software system has daily weather data stored in it. The weather app on your phone communicates with the weather bureau's software to access their data through various API requests. Essentially, your phone app is asking different servers permission to access the information the end user is looking for.

Types of APIs

The four APIs commonly used in web services are public, partner, private, and composite. Public APIs are open and available for use by an outside developer business. Public APIs typically involve moderate authentication and authorization. Partner APIs require specially selected and authorized outside developers. This is intended to facilitate business-to-business activities. Because partner APIs have clear rights and licensing requirements, they generally incorporate strong authentication, authorization, and security measures. Internal APIs or private APIs are intended to be used within an enterprise to connect systems and data within a business. Internal APIs generally are intended to be used in an internal business setting, meaning they employ weak security and authentication. Composite APIs combine two or more APIs to craft a sequence of related or interdependent operations. These APIs address the complexity or tightly related API behaviors and might improve speed and performance over using individual APIs.

The decision of which type of API format should be used can have a profound and long-lasting impact on the success and adoption of the API. A programmer and their organization must select the API appropriate to the complexity of the information that is being exchanged, the level of security required, and the speed with which the information must be communicated. This will also affect the performance of the web application.

If a simpler format is chosen, it might be easier to code and maintain, but might not necessarily offer the level of security required by the enterprise. More complex formats provide the security element but pose higher learning curves for adopters. Tradeoffs are rarely simple, and all requirements must be considered for longevity of the enterprise.