What is an Application Programmer Interface (API)?

Killian Anderson

October 2024

1 Introduction

An Application Programmer Interface (API) will share datasets across the web, systems, and applications through a request and response methodology. Typically, a request will be sent to an API to retrieve data. The vast majority of this is summarized from the Hyper Text Transfer Protocol (HTTP) Working Group and can be found here at `https://httpwg.org/specs/`.

1.1 Principles

1. A client is responsible for starting the request and response handshake by initiating the process. A process may be a request or anything that would trigger an external event for the API to act upon.
2. A server where the API is located interacts with the request and response handshakes. The server will handle authentication, validation, retrieving, deletion, update, or even manipulation of datasets.
3. A request will is dependent upon the architecture of the API you decide to call.
    - An **endpoint** is a stand-alone user reference link (URL) to provide specific access to a resource.
    - A **method** indicates the operation called by the client to the API to perform an action for a given resource. Typically, retrieving, creating, deleting, and updating data.
    - A **parameter** is a variable passed to the API **endpoint** for instruction to start a process for the API to act upon. They are typically in the URL query somewhere.
    - A **request header** is a value representing details about a request process, i.e. formatting of datasets or credentials for access.

- A **body** is center piece of the request or call of the data for retrieving, create, update, or deletion of a resource. Essentially, it contains the data.

4. A response by the server housing the API will include some messages for the client.
    - A **status code** are three digits.
        - **Code:** 100 means progress on response
        - **Code:** 200 means success
        - **Code:** 300 means redirection somewhere.
        - **Code:** 400 means there is a client side error.
        - **Code:** 500 means there is a server side error with valid client request.
    - A **response header** is much like a **request header**, but for information about server response.
    - A **response body** includes datasets or content items the client requested as part of the request and response handshake. In other words, datasets, error messages, and additional information such as a time-stamp.

2 What is the purpose?

Why would we need an API? There are many good reasons and we shall note them here.

1. **Automation** of repetitive tasks for datasets or processes.
2. **Innovation** by making datasets or programs publicly accessible.
3. **Security** by forcing authorization to gain access to datasets or processes in programs.
4. **Efficiency** gains from using another source of datasets or programs without having to build anything from scratch.

3 What types of interfaces are out there?

The type of APIs that you may initiate a request and response handshake can be categorized in two main ways.

1. **Public** APIs provide access to an organizations data, processes, or services with which third party applications may rely upon to integrate their own software. Services may be free or require a payment.

2. **Private** APIs are usually internal to the users or not available to third party users. They typically have a login to set communication of users.

The most common APIs that you may encounter or use for a third party application will have the following architecture.

1. **REST** is the most popular architecture for an API. Resources are accessible via endpoint and operations performed via HTTP methods such as GET, POST, PUT, DELETE typically using the curl package on UNIX like family operating systems.

2. **Webhooks** is another popular architecture for event driven requests. These requests are automated for event triggers. Typically, this is through an HTTP request for a pre-configured URL relevant to the request and a response will be given through that URL. Many communications system might utilize this architecture, such as Discord.

4 Representative Diagram

We have included a diagram to better understand how we interact with various services, databases, or resources. In the traditional sense, an API is a program that accesses information on a server at a clients request and sends a response to that same client. Below, we have drawn a wonderful diagram representing this process.

