Reading API Documentation

How do you know what is possible with an API? You need to read the docs! Reading technical documentation is an important skill for developers, and that includes anyone working with APIs. Recall that we've been working with RESTful Web APIs, but here, API can also refer to any way we use code access functionality from a third party software package or web service (or even Python itself!).

Writing documentation is time consuming. So is keeping documentation updated - it's another step above and beyond writing code to implement new features. But APIs demand reliability and good documentation because they are how applications access data and functionality. Making a change to the Google Maps API, for instance, would immediately affect thousands of API developers, millions of websites, and billions of pageviews. This actually happened in June 2016, when Google Maps V3 removed keyless API access to their API, breaking thousands of sites that were making requests without including an API key. Clearly, documentation changes over time as an API evolves - so make sure you're reading the right documentation for the API version you're using. When in doubt, use the latest version.

Documentation Features

According to Pamela Fox's *Developer Support Handbook*, complete documentation should include these features:

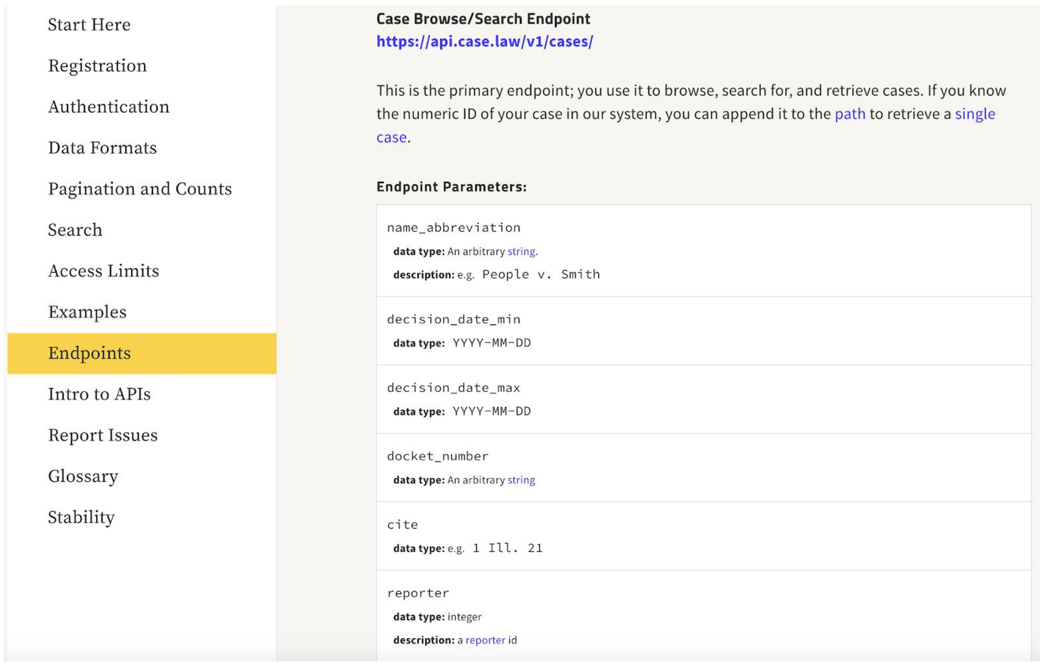
- Class Reference: A comprehensive listing of API functionality.
- Changelog: A reference of what changes in each API version.
- Code Samples: A set of examples showing typical API usage.
- Code Playground: An interactive explorer for trying the API live in the browser.
- Developers Guide: A conversational written guide to using the API.
- Articles: Tutorials and screencasts discussing different ways of using the API.

These common documentation sections provide different ways for developers to understand an API. You'll probably be searching for documentation for a variety of reasons, so any of them could potentially be useful. When you're learning a new API, the developer's guide and code samples will likely kickstart your process by showing what is possible. A code sandbox lets you test sample queries and play with live data quickly. Perhaps you're already familiar with the API and just need to know which parameters to use with a specific method outlined in the class reference.

Common Documentation Sections

Some other common sections include:

- Terms of service: some APIs restrict certain kinds of uses. Make sure to follow the terms of service or you may get banned!
- Registration: if you need to sign up to use the API, this will tell you how to obtain credentials
- Authentication: how you need to sign your requests to access the API (API key, OAuth, Basic Authentication, etc)
- Rate limits: how many requests you can make to the API in a specified window
- Cost: many APIs are paid. This may take the form of tiers of access or a cost per request.
- Data Formats: some APIs allow you to choose between JSON, XML, and other formats.
- Endpoints: these are where you send your requests
    - They typically take the form of a noun, like `/object` or `/painting`, which describes the type of resource you are interacting with.
    - The documentation should describe the type of HTTP methods the endpoint will accept (`GET, POST, PUT, DELETE,` etc) and the available of URL and query parameters you can use with the endpoint. The documentation should describe which parameters are required and which are optional, and what type of values you need to provide for each.
    - It should also define the type of response you will get back, typically through a listing of keys and their corresponding value types.
    - You may also find examples of API requests and responses here, which can be very helpful when developing your first queries.



Case Browse/Search Endpoint
https://api.case.law/v1/cases/

This is the primary endpoint; you use it to browse, search for, and retrieve cases. If you know the numeric ID of your case in our system, you can append it to the path to retrieve a single case.

Endpoint Parameters:

`name_abbreviation`
data type: An arbitrary string.
description: e.g. `People v. Smith`

`decision_date_min`
data type: `YYYY-MM-DD`

`decision_date_max`
data type: `YYYY-MM-DD`

`docket_number`
data type: An arbitrary string

`cite`
data type: e.g. `1 Ill. 21`

`reporter`
data type: integer
description: a reporter id

- Pagination: many APIs only allow you to access a limited number of responses in one request, and you must use pagination to get additional data
- Version: APIs evolve - make sure you're using the right version. Default to the latest if you're unsure.

Harvard Law School Caselaw Access Project documentation for the `cases` *endpoint.*

API Documentation Examples

Here are a few different Web API documentation guides. See if you can find some of the features outlined above in these examples:

- Twitter
- Harvard Law School Caselaw Access Project
- New York Times
- Genius
- Rotten Tomatoes
- Dark Sky (Weather)

Finding APIs

Finding the right API for your project can be difficult. Programmable Web is a good resource for this. You can search for APIs and filter across a number of different categories, which helps you to quickly understand whether or not an API is a good fit for what you want to accomplish.