Stoplight

Five Ways JSON Schema Can Help You Create Better APIs

In this post, we'll look at JSON's popularity and examine how the combination with JSON Schema makes it a great data format for most APIs.

Janet Wagner
Apr 23, 2021

In the earliest days of modern APIs, XML was the top data format. There is a standard for how XML is formatted and many other standards are built upon it. One strong aspect of XML is using schemas to define your XML data. However, the flexibility of the JSON data format increased its popularity. For years, JSON remained schemaless. Now, JSON Schema brings some formality to the data format, while maintaining what made it popular.

In this post, we'll look at JSON's popularity and examine how the combination with JSON Schema makes it a great data format for most APIs.

CONTENTS

Why Are JSON APIs Popular?

Recently ProgrammableWeb declared "JSON is Clearly the King of API Data Formats:"

> "Returning to the question of JSON versus XML, we see that in the last two years JSON has been used as a response data format five-fold as many times as XML. XML is by far the number two response format used by API providers, but it is clear that JSON has been the unquestioned format of choice."

If you do a web search for "XML vs. JSON," you'll find many different opinions as to why JSON has skyrocketed past XML in popularity. While JSON is less verbose, more lightweight, and easier to use than XML, these aren't the only reasons for JSON's increasing popularity. For starters, JSON support is included in every modern browser. Also, JSON is the native format for data in JavaScript applications—and JavaScript is the top language used by developers to build applications.

It makes sense that JavaScript rose in popularity and JSON rose along with it. You could try to use XML as the data interchange format for a JavaScript application, but it would take a lot more effort than if you use JSON. Similarly, most languages had data structures that closely mimic JSON, which makes it easy to translate.

JSON Schema Eliminates XML Holdouts

In the early days of JSON, schemas were a key difference (and to many, an advantage) of XML. XML Schema is a language that lets you describe the structure of an XML document. Proponents of this approach advocate this structured description of your data models. You can also use a schema to validate documents. JSON by default does not include a schema in the way that XML does. However, in 2010, the JSON Schema project was born, which allows anyone to define the structure of JSON.

JSON Schema is a vocabulary and standard for annotating and validating JSON documents. The main goal for JSON Schema is validation—you can use it to validate data from another source and then fit that data to your data model. Early on, some developers worried JSON Schema would become as complex as XML. But the goal of JSON Schema was always to help developers make sense of JSON data without the schema getting in the way.

One of the early use cases for JSON Schema was NoSQL databases that use JSON as data representation. These storage options, like CouchDB and MongoDB, also gained popularity in the early 2010s. While NoSQL databases don’t always require schemas, you still likely want one. A schema makes your data predictable and easier to query, especially if the data has a complicated structure and many levels.

More recently, JSON Schema is applied to other data tools, including APIs. In particular, its inclusion within the OpenAPI standard enables teams to create more consistent, predictable APIs.

JSON Schema Can Help You Build Better APIs

You can use JSON Schema to improve the design of your APIs along with third-party tools that support it. Many third-party tools have implemented JSON Schema, and you can find a list on the JSON Schema website for tools that support draft-06 or later.

With JSON Schema and third-party tools you can:

- Perform client-side validation.
- Simplify contract testing.
- Create mock servers.
- Create style guides for your structured data.
- Generate documentation for your API automatically.

Let's examine five ways JSON Schema can improve the design of your API in more detail.

Perform Client-Side Validation

API validation rules can live in two places: the server and the client. Developers often reproduce server-side validation in the client-side, which is not the best approach. Validation rules should not be stored only inside the backend code. Instead, you can use JSON Schema to create validation rules that can be seen by API clients. Client-side validation allows applications to notice new validation rules almost immediately and avoid breaking when there are minor validation changes. Find out more about this topic in this APIs You Won’t Hate post by Stoplight’s Phil Sturgeon.

Simplify Contract Testing

When you talk to developers about “contract testing,” they usually think you mean “producer contract testing.” You can simplify contract testing by using JSON Schema along with a schema matcher in your integration tests. The standard process usually starts with the API development team creating a test that records all parts of an interface. Then they run tests on pull requests to the API repository to make sure that there are no accidental changes to the code. However, if you use JSON Schema models along with OpenAPI, you can take the schemas and assert that the response matches it without having to write things like properties, data formats, and validations.

Create Mock Servers

With JSON Schema and an API definition standard like OpenAPI, you can generate a complete mock server. A mock server helps you accelerate the development of an API by allowing you to get feedback on the API design before you spend time on development. You can also ensure that your API fulfills consumer requirements and that any changes you make to your API won’t break the applications of your customers. Some mock API servers include dynamic mocking, where you can leverage fake data to respond dynamically to requests, thus simulating real server behavior.

Create Style Guides for Your Structured Data with a JSON Linter

If you want your developers to build consistent APIs, you need an API style guide. It's not a good idea for your developers to make guesses about API design components, like naming conventions and field formats. With a JSON Linter, you can create a style guide that keeps all your API developers on the same page and your API designs consistent across your organization.

Generate Documentation for Your API Automatically

If you're providing an API for developers to consume, you must also provide well designed, easily understandable API documentation. It's not enough to design your API well, you need to tell developers how to use it in a clear and concise manner. You can find many tools that allow you to generate nicely designed, human-readable API documentation from your JSON Schema files. Using JSON Schema, via an OpenAPI document, can also help you ensure that your API documentation is always up to date.

Design Top-Notch APIs with JSON Schema and Stoplight

We've highlighted five ways that you can use JSON Schema to improve the design of your APIs. Unsurprisingly, you can do all of them with Stoplight. You can create reusable models so the names and data objects remain consistent. Then, share these API design assets across a single API or your whole organization.

The best way to get started is to start a new workspace and start creating APIs with JSON schema for free today.

Related