What is APIs.json? And What is Next for the API Discovery Format?

by Kin Lane · Mar. 21, 16 · Integration Zone · Analysis



As part of a renewed focus on the API discovery definition format APIs.json, I wanted to revisit the proposed machine readable API discovery specification and see what is going on. First, what is APIs.json? It is a machine-readable JSON specification, that anyone can use to define their API operations. APIs.json does not describe your APIs like OpenAPI Spec and API Blueprint do, it describes your surrounding API operations, with entries that can reference your Open API Spec, API Blueprint, or any other format that you desire.

APIs.json is an Index for API Operations

APIs.json provides a machine readable approach that API providers can put work in describing their API operations, similar to how website providers describe their websites using sitemap.xml. Here are the APIs, who are describing their APIs using APIs.json:

APIs.json Indexes can be Created by 3rd Parties

One important thing to add is that these APIs.json files can also be crafted, and published by external parties. An example of this is with the Trade.gov APIs. I originally created that APIs.json file and coordinated with them to eventually it get published under their own domain, making it an authoritative APIs.json file. Many APIs.json files will be born outside of the API operations they describe, something you can see in my API stack project:

- The API Stack - Provides almost 1000 APIs.json files, that describe the API operations of many leading public API platforms. There is also around 300 OpenAPI specifications, for some of the platforms described

APIs.json Can Be Used To Describe API Collections
Beyond describing a single API, within a single domain, APIs.json can also be used to describe entire collections of APIs, providing a machine readable way to organize, and share valuable collections of API resources. Here are a few examples of projects that are producing APIs.json driven collections.

APIs.json Can Be Used To Describe Collections of Collections
Then taking things up another rung up the chain, APIs.json can also provide a collection of collections, something I do with my own APIs. Each Github organization on my network has a master APIs.json, providing include links to all other APIs.json within the organization. In this scenario, I have over 30 other APIs.json indexed, which can all operate independently of each other, but can also be considered a collection of API collections.

- Master - A master collection of API collections I maintain as part of the API Evangelist network operations.

The First Open Source Tooling For APIs.json
Up until now, this post is all about APIs.json, where in reality the format is useless without their being any tooling built on top of the specification, bringing value to the table. This is why the 3Scale team got to work building an open source APIs.json driven search engine:

- APIs.io as an open source tool dedicated to APIs.json
- APIs.io as a public API search engine, with APIs.json as an index.
- APIs.io as a private API search engine, with APIs.json as an index.

APIs.json Driving Other Open Tooling
APIs.io is just the beginning. It won't be enough to convince all API providers that they should be producing APIs.json index of their site operations, just for the API discovery boost. We are going to need APIs.json driven tooling that will service every other stop along the life cycle, including:

- **HTTP Client/Hub / Workbenches**
- **Documentation**
- **Testing**
- **Monitoring**
- **Virtualization**
- **Visualization**

APIs.json Integrated Into Existing Platforms
What areas would you like to see served? Personally, I would like to have the ability to load / unload my APIs.json collections into any service that I use. Allowing me to organize my internal, public, and 3rd party APIs I depend on any platform out there that is servicing the API space. Here are a handful of those types of integrations that are already happening:

APIs.json Linking To The Human Aspects Of API Operations
APIs.json is just the scaffolding to hang links to essential aspects of your operations, it doesn't care what you link to. You can start by referencing essential links for your API operations like:

- **Signup** - How to signup for a service.
- **Support** - Where to get support.
- **Terms of Service** - Where are the terms of service.
- **Pricing** - Where to find the pricing for a service.

APIs.json Linking to Machine Readable Aspects of API Operations
These do not have to be machine-readable links, they can reference important things the humans will need first. However, ultimately the goal is to make as much of the APIs.json index as machine-readable as possible, using a variety of existing API definition formats, available for a variety of purposes.

Defining New, Machine Readable Property Elements For APIs.json
While the APIs.json spec will evolve, something I talk about below, its real strength lies in its ability to incentivize the development of entirely new, machine readable API definitions, bringing, even more, value to the API discovery process. Here are a few of the additional specs being crafted independent of, but inspired by APIs.json:

- API Plans, for pricing, plans & rate limits.
- API Monitoring, for monitoring & testing.
- API Changelog, for operational monitoring.
- API SDK, for SDK reference.
- API Conversations - for the stream around API operations

Roadmap for Version 0.16 of APIs.json
That is the 100K view of what is APIs.json now, and the short term plan for the future. Most of the change within the universe APIs.json is mapping will occur add the individual API, and within the machine readable specs that describe them like OpenAPI Spec, API Blueprint, and Postman. Secondarily, there will be additional, machine readable, API types being defined and added into the spec.

Even with this reality, we do have a handful of changes planned for the 0.16 version of APIs.json:

- **commons** - Establish a top-level collection of common property elements that apply to ALL APIs being referenced in an APIs.json
- **country** - Adding a top level country reference using ISO 3166.
- New Proper Elements - Suggesting a handful of new property elements to reference common API operation building blocks
    - Registration
    - Blog
    - Github
    - Twitter

I doubt we will see many new additions like commons and country. In the future most of the structural changes to APis.json will be derived from first class property elements (ie. adding documentation or Github), making this the proving ground for defining what are truly the most important aspects of API operations, and what should be machine readable vs human readable.

The Hard Work That Lies Ahead for APIs.json

That concludes defining what is APIs.json, and what is next for APIs.json. Now we really have to get to work, doing the heavy lifting around:

- Getting more API providers to describe their API operations using APIs.json, and publish in the root of the domain for their API ecosystem.
- Encourage more API evangelists, brokers & analysts using to describe their collections, using APIs.json, building more meaningful indexes and directories of high-value APIs.
- Encourage platforms to build APIs.json into their operations, as a storage and organization schema, but also as import / export format.
- Incentivize the development of more meaningful tooling that employs APIs.json, and uses it to better serve the API life cycle.
- Continue to add new API property elements, making sure as many of them as possible evolve to be machine-readable, as well as first class citizens in the APIs.json specification.

You can stay involved with what we are up to via the APIs.json website, and the APIs.json Github repository. You can also stay in tune with what is going on with APis.io via the website, and its Github repository. If you are doing something with APIs.json, ranging from using it as an index for your API operations, to platform integrations, please let me know. Also, if you envision some interesting tooling you'd like to see happen, make sure and submit a Github issue letting us know.

While we still have huge amounts of work to do, when it comes to delivering meaningful API discovery solutions that the industry can put to work, I am pretty stoked with what we have managed to do over the last two years of work on the APIs.json specification, and supporting tooling--momentum that I feel picking up in 2016.

Like This Arti