Public note

Bruno Pedro

Friday, October 7, 2022

With a major version bump and an acceptance of breaking changes from OpenAPI 3.x, I'd like to propose moving from JSON object structures to JSON array structures.

Found on Consideration for moving from objects to arrays. · Discussion #32 · OAImoonwalk on 2022-10-07 12:08:38.

Tags:
#ope-
napi
#ope-
napiv4
#v4
#dis-
cus-
sion
#spec-
ifi-
ca-
tion
#api-
specification

I don't necessarily agree with this proposal. The following is the comment I left on the discussion mentioned above.

Comment

I believe this proposal is interesting, however it creates a tradeoff between the importance of running query filters on user-provided keys and accessing individual elements.

Let's start by analyzing the motivation behind this proposal. According to the author, the goal of the proposed change is to **make it easier for API practitioners to work with the data represented by OpenAPI definitions**.

> Specifically, running query filters on names in JSON name-value pairs is difficult for lots of tooling (…)

The examples provided to obtain all the paths that contain the word "pets" make sense. You can clearly see that using JSON object structures makes it harder to obtain all the elements where their keys contain a given string.

On the other hand, accessing **individual** elements **becomes more difficult** after the proposed changes. Let's take the `/pets` path as an example and see how accessing it directly looks like with JSON object structures versus JSON array structures, using different methods.

Method	JSON object structures	JSON array structures
M1: `jq`, using `api.json` as the API definition document	`jq '.paths | .["/pets"]' < api.json`	`jq '.paths[] | select(.name=="/pets")'< api.json`
M2: JavaScript, using `api` as the object holding the API definition	`api.paths['/pets']`	`api.paths.find(path => path.name == '/pets')`
M3: Python, using `api` as the	`api['paths']['/pets']`	`[path for path in api['paths'] if path['name']=='/pets'][0]`

While **M1** doesn't convey the big difference in effort when using `jq`, if you look at the other methods, you will immediately see that there is a degree of difficulty introduced when looking up elements on array structures. Object structures can be easily referenced by their keys in most programming languages, while array structures can't be referenced by the value of arbitrary keys.

The value of this proposal is, in my opinion, affected by the importance of how you want to read information from an OpenAPI definition. If you feel it's more important to easily filter name-value pairs, then JSON array structures seem to be the appropriate solution. If, on the other hand, you want to access individual elements as easily as possible, then JSON object structures are the preferred solution.