Programming in the Life Sciences #20: extracting data from JSON

Egon Willighagen

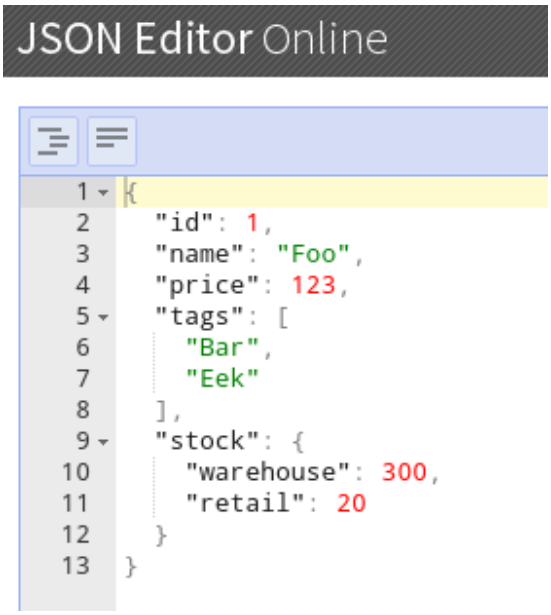
Published November 16, 2014

Citation

Willighagen, E. (2014, November 16). Programming in the Life Sciences #20: extracting data from JSON. *Chem-bla-ics*. https://doi.org/10.59350/szpy6-v8c44

Keywords

Pra3006



I previously wrote about the JavaScript Object Notation (JSON) which has become a de facto standard for sharing data by web services. I personally still prefer something using the Resource Description Framework (RDF) because of its clear link to ontologies, but perhaps JSON-LD combines the best of both worlds.

The Open PHACTS API support various formats and this JSON is the default format used by the ops.js library. However, the amount of information returned by the Open PHACTS cache is complex, and generally includes more than you want to use in the next step. Therefore, it is needed to extract data from the JSON document, which was not covered in the post #10 or #11.

Let's start with the example JSON given in that post, and let's consider this is the value of a variable with the name jsonData:

```
{
    "id": 1,
    "name": "Foo",
    "price": 123,
    "tags": [ "Bar", "Eek" ],
    "stock": {
        "warehouse": 300,
        "retail": 20
    }
}
```

We can see that this JSON value starts with a map-like structure. We can also see that there is a list embedded, and another map. I guess that one of the reasons why JSON has taken such a flight is how well it integrates with the JavaScript language: selecting content can be done in terms of core language features, different from, for example, XPath statements needed for XML or SPARQL for RDF content. This is because the notation just follows core data types of JavaScript and data is stored as native data types and objects.

For example, to get the price value from the above JSON code, we use:

```
var price = jsonData.price;
```

Or, if we want to get the first value in the Bar-Eek list, we use:

```
var tag = jsonData.tags[0];
```

Or, if we want to inspect the warehouse stock:

```
var inStock = jsonData.stock.warehouse;
```

Now, the JSON returned by the Open PHACTS API has a lot more information. This is why the online, interactive documentation is so helpful: it shows the JSON. In fact, given that JSON is so much used, there are many tools online that help you, such as jsoneditoronline.org (yes, it will show error messages if the syntax is wrong):

chem-bla-ics

BTW, I also recommend installing a JSON viewer extension for Chrome or for Firefox. Once you have installed this plugin, you can not just read the JSON on Open PHACTS' interactive documentation page, but also open the Request URL into a separate browser window. Just copy/paste the URL from this output:

And with a JSON viewing extension, opening this *https://beta.openphacts.org/1.3/pathways/...* URL in your browser window will look something like:

And because these extensions typically use syntax highlighting, it is easier to understand how to access information from within your JavaScript code. For example, if we want the number of pathways in which the compound testosterone (the link is the ConceptWiki URL in the above example) is found, we can use this code:

```
var pathwayCount = jsonData.result.primaryTopic.pathway_count;
```