

## 12.3.2

### Load a JSON file with D3.json()

**Roza** is now ready to load the belly button data into her script. First, she needs to download the data file to her computer. She'll then use two things to read the data file: the D3.js library and a local server.

To download the zip file containing the data file, use the following link.

**Download the data** [\\_ \(https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module\\_12/data-12-3-2-resources.zip\)](https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_12/data-12-3-2-resources.zip)

Recall the previous syntax used to place an API call:

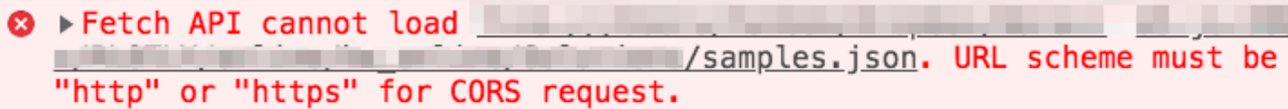
```
const url = "https://api.spacexdata.com/v2/launchpads";  
d3.json(url).then();
```

Here, the URL string is received by `d3.json()` as an argument. The `d3.json()` method then retrieves the data from the address specified by the URL. After the data is fully retrieved, the function inside the `then()` method is executed.

The syntax used to retrieve data from an external data file, instead of a web API, is the same:

```
d3.json("samples.json").then(function(data){  
  console.log("hello");  
});
```

When we open the browser, however, nothing is printed to the console. We get this error message:



Fetch API cannot load [redacted]/samples.json. URL scheme must be "http" or "https" for CORS request.

What gives? What is a CORS request?

The short explanation is that, for security reasons, a local server must be run when loading an external file into a JavaScript script file. If you don't understand these security issues right now, don't worry. We will come back to it later.

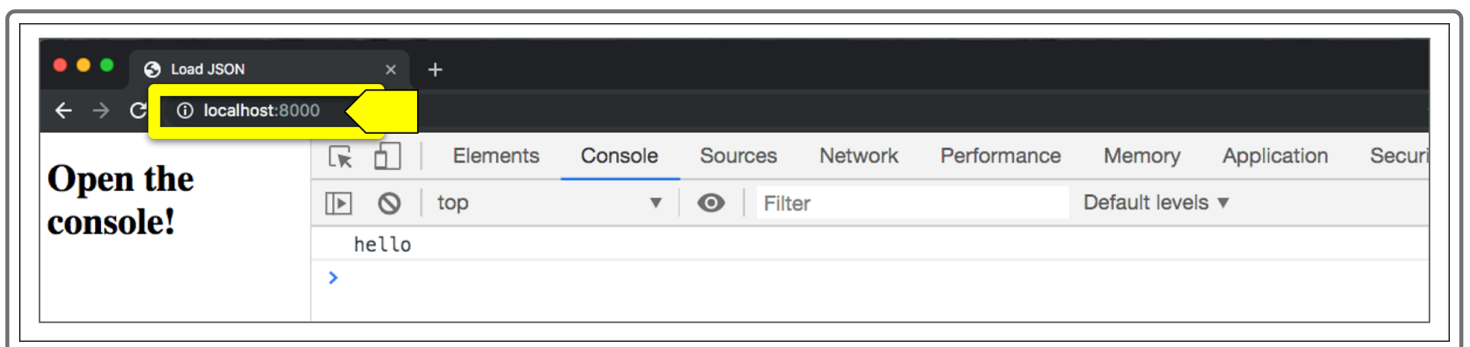
To load the page, navigate to the directory where `samples.json` and `index.html`, as well as the script file, `plots.js`, are located. Open the command line (Terminal or Git Bash) and type the following:

```
python -m http.server
```

You should see the following message in the command line:

```
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...  
127.0.0.1 - - [30/Oct/2019 13:23:53] "GET / HTTP/1.1" 304 -
```

Navigate to the given port number in your browser: `localhost:8000`. When you do this, you'll see the following screen:



### IMPORTANT

When reading an external data file such as a CSV or JSON file into a script, you must run a server. You cannot directly open `index.html` with your browser.



```
▼ {names: Array(153), metadata: Array (153), samples: Array (153)} ⓘ  
  ▼ metadata: Array (153)  
    ▼ [0 ... 99]  
      ▼ 0:  
        age: 24  
        bbtype: "I"  
        ethnicity: "Caucasian"  
        gender: "F"  
        id: 940  
        location: "Beaufort/NC"  
        wfreq: 2
```

The `metadata` array contains objects, each of which contains details of a volunteer, such as age, location, ethnicity, ID number, and weekly washing frequency of the belly button.

 [Retake](#)

 [Retake](#)

 [Retake](#)

Now let's examine the rest of the dataset. Looking at the image below, we can see that `names` is simply an array of the ID numbers of the volunteers. Even though this information is included in the `metadata` array, the `names` array may be useful in rapidly retrieving an ID number when creating a plot.

```
▼ {names: Array(153), metadata: Array(153), sample:  
  ► metadata: (153) [{...}, {...}, {...}, {...}, {...}, {...},  
  ▼ names: Array(153)  
    ▼ [0 ... 99]  
      0: "940"  
      1: "941"  
      2: "943"  
      3: "944"  
      4: "945"  
      5: "946"  
      6: "947"
```

Next, check out the `samples` array and inspect the first element.

```
▼ 0:  
  id: "940"  
  ► otu_ids: (80) [1167, 2859, 482, 2264, 41, 1189, 352, 189, 2318, 1977, 3450, 874, 1959,  
  ► otu_labels: (80) ["Bacteria;Bacteroidetes;Bacteroidia;Bacteroidales;Porphyromonadaceae;  
  ► sample_values: (80) [163, 126, 113, 78, 71, 51, 50, 47, 40, 40, 37, 36, 30, 28, 25, 23,
```

The array's first element is an object, with four key-value pairs. Note the following:

- The `id` key identifies the ID number.
- The `otu_ids` property is an array of the ID numbers of all the bacteria found in this person's navel. OTU stands for Operational Taxonomic Unit, and here it means species or bacterial type. In this instance, there were 80 bacterial types with distinct ID numbers.
- The `sample_values` array contains the corresponding species name for each bacterial ID number. Some bacterial species have different ID numbers, but are clumped together under the same `otu_label`.

In her final visualization of the belly button data, Roza would like to be able to select an individual from a dropdown menu. Once a person's ID number is selected, she would like to display the demographic information of that individual. Since each individual is represented by an object, she'll need to access both keys and values inside an object in order to do this.

**REWIND**

The `Object.entries()` method allows access to both an object's keys and values. It returns each key-value pair as an array.

 [Retake](#)

**REWIND**

The `forEach()` method allows access to each element of an array.

 [Retake](#)

Now test your skills in the following Skill Drill:

**SKILL DRILL**

Use `Object.entries()` and `forEach()` to print all the metadata of the first person in the `samples.json()` dataset (ID `940`).

Roza has made a definite step forward. With the following code, we can display the metadata of any individual from the dataset:

```
d3.json("samples.json").then(function(data){  
  firstPerson = data.metadata[0];  
  Object.entries(firstPerson).forEach(([key, value]) =>  
    {console.log(key + ': ' + value)});  
});
```

In this case, we are extracting the metadata of the first person in the dataset, as indicated by the zero index position in `metadata[0]`. We then use the `Object.entries()` method to return each key-value pair in an array, and the `forEach()` method to access each element of these pairs.

Open the browser console to see the results:

```
▶ {id: 940, ethnicity: "Caucasian", gender: "F", age: 24, location: "Beaufort/NC", ...}  
id: 940  
ethnicity: Caucasian  
gender: F  
age: 24  
location: Beaufort/NC  
bbtype: I  
wfreq: 2
```

Here, we manually specify the individual by the index position. Ultimately, we need to be able to choose an ID number from a dropdown menu and then display the metadata associated with that ID.

© 2020 - 2022 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.