Node.js Workshop 3: Processing JSON and using API's

After completing this workshop the student is knows how to:

- Process JSON files from web
- Parse JSON and show the output on browser
- Deploy the code in Heroku

Create a new folder called WS3 for these assignments. Place all your code there.

Read and process JSON files locally

Use lecture notes as a guide. There is plenty of tutorials available in the Internet.

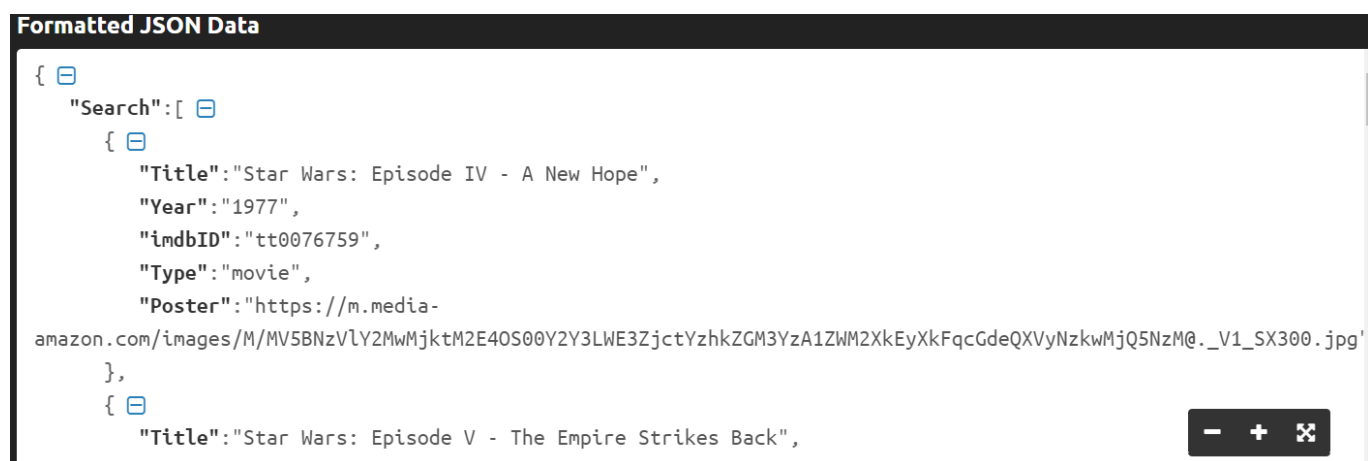## Read and process JSON files online through API's

Exercise 1: movies.js

1. Create a new program "movies.js", which will call JSON API from the web, parse it and output on the console and to the browser. This time we will use the Open Movie Database API, see the docs here: http://www.omdbapi.com/.

   You can see a sample set of results here:
   http://www.omdbapi.com/?s=star+wars&apikey=cbbc6750

   Often times Online JSON formatters make reading API responses much easier. Try pasting the response here: https://jsonformatter.curiousconcept.com/

   Formatted JSON Data

   ```
   { ⊟
      "Search":[ ⊟
         { ⊟
            "Title":"Star Wars: Episode IV - A New Hope",
            "Year":"1977",
            "imdbID":"tt0076759",
            "Type":"movie",
            "Poster":"https://m.media-
   amazon.com/images/M/MV5BNzVlY2MwMjktM2E4OS00Y2Y3LWE3ZjctYzhkZGM3YzA1ZWM2XkEyXkFqcGdeQXVyNzkwMjQ5NzM@._V1_SX300.jpg'
         },
         { ⊟
            "Title":"Star Wars: Episode V - The Empire Strikes Back",
   ```

2. Pasting the JSON response in JS console as variable ("var json") helps to understand it:

```
> json.Search

<· ▼(10) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}] ⓘ
       ▶ 0: {Title: "Star Wars: Episode IV - A New Hope", Year: "1977", imdbID: "tt0076759"…
       ▶ 1: {Title: "Star Wars: Episode V - The Empire Strikes Back", Year: "1980", imdbID:…
       ▶ 2: {Title: "Star Wars: Episode VI - Return of the Jedi", Year: "1983", imdbID: "tt…
       ▶ 3: {Title: "Star Wars: The Force Awakens", Year: "2015", imdbID: "tt2488496", Type…
       ▶ 4: {Title: "Star Wars: Episode I - The Phantom Menace", Year: "1999", imdbID: "tt0…
       ▶ 5: {Title: "Star Wars: Episode III - Revenge of the Sith", Year: "2005", imdbID: "…
       ▶ 6: {Title: "Star Wars: Episode II - Attack of the Clones", Year: "2002", imdbID: "…
       ▶ 7: {Title: "Rogue One: A Star Wars Story", Year: "2016", imdbID: "tt3748528", Type…
       ▶ 8: {Title: "Star Wars: The Last Jedi", Year: "2017", imdbID: "tt2527336", Type: "m…
       ▶ 9: {Title: "Solo: A Star Wars Story", Year: "2018", imdbID: "tt3778644", Type: "mo…
         length: 10
       ▶ __proto__: Array(0)

> json.Search[1].Title
<· "Star Wars: Episode V - The Empire Strikes Back"

> json.Search[1].Type
<· "movie"

> json.Search[1].Year
<· "1980"

> json.Search[1].Poster
<· "https://m.media-amazon.com/images/M/MV5BYmU1NDRjNDgtMzhiMi00NjZmLTg5NGItZDNiZjU5NTU4O
   TE0XkEyXkFqcGdeQXVyNzkwMjQ5NzM@._V1_SX300.jpg"

> |
```

3. Then we can loop through it:

```
> for (var i=0; i < 10; i++){
  console.log(json.Search[i].Title);
  }
  Star Wars: Episode IV - A New Hope                          VM591:2
  Star Wars: Episode V - The Empire Strikes Back              VM591:2
  Star Wars: Episode VI - Return of the Jedi                  VM591:2
  Star Wars: The Force Awakens                                VM591:2
  Star Wars: Episode I - The Phantom Menace                   VM591:2
  Star Wars: Episode III - Revenge of the Sith                VM591:2
  Star Wars: Episode II - Attack of the Clones                VM591:2
  Rogue One: A Star Wars Story                                VM591:2
  Star Wars: The Last Jedi                                    VM591:2
  Solo: A Star Wars Story                                     VM591:2
<· undefined
```

4. Register here in order to get your own API key: http://www.omdbapi.com/apikey.aspx

   ```
   You can use the API by adding parameters after the URL.
   http://www.omdbapi.com/?apikey=[yourkey]&
   ```

5. Create a web server and query for your favourite movies/series and parse the output as HTML formatted table to the browser.
6. Notice that the resultset has an image link, make sure your HTML code renders it nicely. You can see examples on howto use the API here: http://www.omdbapi.com/#usage

Exercise 3: Deploy your code to Heroku web service.

Push (upload) all the exercises completed here to you GitHub-account. Then deply the site to Heroku.

Heroku is a web service for deploying web applications online. Heroku can get your code from GitHub, so you need to authenticate Heroku to read your repository.

Some considerations though.

1) You need to use variable called PORT instead of a fixed port in your code
2) You need to have a file called package.json in your app directory, with the following structure

```
index.js:

const PORT = process.env.PORT || 5000;
var http = require("http");

//create a server object:
http
  .createServer(function(request, response) {
    response.writeHead(200, { "Content-Type": "text/plain" });
    response.write("Hello World!\n"); //write a response to the client
    response.end("This is the end"); //end the response
  })
  .listen(PORT); //the server object listens on port 8080


package.json:

{
  "name": "node workshop 3",
  "version": "0.1.0",
  "description": "A sample Node.js app in Heroku",
  "engines": {
    "node": "12.x"
```

```
    },
    "main": "index.js",
    "scripts": {
      "start": "node index.js",
      "test": "node test.js"
    }

"dependencies": {
    "axios": "^0.21.1",
    "request": "^2.88.2"
  }
}
```

Laurea-ammattikorkeakoulu
Ratatie 22, 01300 Vantaa

Puhelin (09) 8868 7150
Faksi (09) 8868 7200

etunimi.sukunimi@laurea.fi
www.laurea.fi

Y-tunnus   1046216-1
Kotipaikka   Vantaa