

Lab 2

Node application working with Core and Third party modules

Due: Thursday March 29, 2018 @ 11:59pm

Overview:

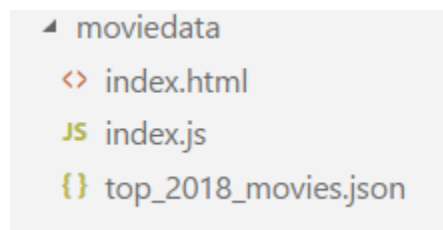
For this lab, you will walk through the following steps to creating a simple node application that displays specific movie data based on the requested route. This lab introduces working with core modules, third-party modules, functions, methods, JSON data, routing and http/https request and response objects in the Node environment.

Resources needed for this lab:

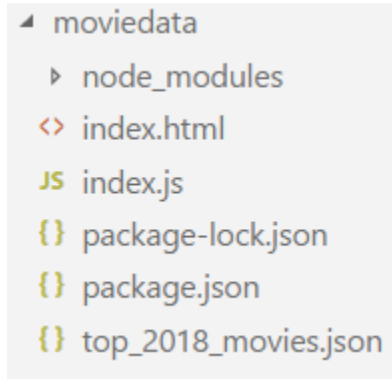
- Text Editor
- top_2018_movies.json file
- Github Repository and Local Repository
- Language – NodeJs
- NPM and Core modules
 - Nodemon
 - http
 - fs

Requirements:

- 1) Create a project folder called **"moviedata"**
- 2) Version control project and initialize Git by running the command **git init**
- 3) Open the project folder in your preferred editor
- 4) Create the project folder listed below in image:
- 5) Create and add the files listed below folder.
 - a. Create a **index.html** file
 - b. Create a **index.js** file
 - c. Include the **top_2018_movies.json** file
 - d. Your project folder should look similar to the folder structure listed below:



- 6) Run the **npm init** command to create package.json and the package-lock.json file
- 7) Install nodemon third-party module - **npm install nodemon --save**
Your project folder should look similar to the image below:



- 8) Open the index.html page and code out the following:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Top Grossing Movie Break-down</title>
  </head>
  <body>
    <h1>This site contains the total domestic box office for the top 100 movies released in 2018/2017</h1>
    <h2>Please click on the links below to display the movies based on criteria</h2>
    <p><a href="/all_movies">Click here to view all top grossing movies</a></p>
    <p><a href="/action">Click here to view all movies that grossed above 20 million and genre is action</a></p>
    <p><a href="/pg">Click here to view movies that are rated "PG-13" and number of tickets sold is between 1 and 5 million</a></p>
    <p><a href="/distributor">Click here to sort movies by distributor </a></p>
  </body>
</html>
```

- 9) Open the index.js file and include the three statements at the top of your file.

```
const http = require('http');
const fs = require('fs');
const data = require('./top_2018_movies.json');
```

- 10) After you include the three statements, type out the four single-line comments below and create functions that return the following data.

```
//Create a function that list all the movies
```

```
//Create a function that list the movies that grossed above 20 million and genre is action
```

```
//Create a function that list the movies that are rated "PG-13" and number of tickets sold is between 1 and 5 million
```

```
//Create a function that sorts the movies based on "distributor"
```

- 11) Create a server that listens on port 3000 using the `http.createServer()` and `listen()` methods.
- 12) Include logic that will allow you display the movie data based on the routes below:
 - a. `"/"`
 - b. `"/all_movies"`
 - c. `"/action"`
 - d. `"/pg"`
 - e. `"/distributor"`
- 13) If a route other than the ones listed is entered, output a 404 message and change the statusCode to 404 error.
- 14) Use the fs method `readFile()` to display the HTML contents of `index.html` for route `"/"`
- 15) Start your server by entering `nodemon index.js`
- 16) Navigate to <http://localhost:3000> and type in the routes from step 12 to render movie data.
- 17) Kill your server by entering **Control + C**
- 18) Run Git commands to commit files:
 - a. Check the status of files - `git status`
 - b. Add files to staging area - `git add .`
 - c. Commit files - `git commit -m "adding files and changes"`
- 19) Create Remote repository on Github - Name the repository **Lab_2_[your_dunwoody_username]**
- 20) Copy the commands that will allow you to push an already existing repository to remote repository.

Submission of Lab

The url to your public repository will be due on the date listed above.