



## Assignment 7 (HF, major subject)

---

*Due: Wed 09.12.2015; 12:00h*

### Goals

After doing these exercises,

- You know how to utilize the express-generator
- You can modify the basic set-up of an express app, in particular serving static directories.
- You have understood the principle of middleware.

### Task 1: Set up your app

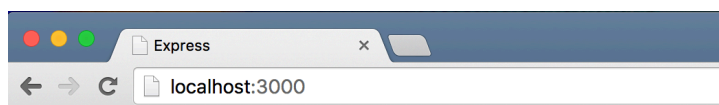
**Difficulty: Easy**

We want to use the [website](#) from last week's break-out and tutorial and serve it through NodeJS. To achieve this, we will have to prepare an Express web application. The modifications are done in Task 2 and 3 – here you only set up the basic web app and see if it starts.

Here's how you can do this:

1. Use the [express-generator](#) to create a standard express web-app that we are going to modify a little bit. Launch this command in a terminal from an empty directory:  
**express MusicApp**
2. Navigate to the newly created directory MusicApp (`cd MusicApp`) and install the required dependencies:  
**npm install**  
You now should have a directory called `node_modules` containing all necessary modules that the express app requires.
3. Launch the app, and verify that it starts up correctly, e.g.:  
**node ./bin/www**  
Open a browser and go to <http://localhost:3000/>. You should see a default page from Express.

Result:



## Express

Welcome to Express



## Task 2: Create a Spotify module

Difficulty: Intermediate

Now, we want to create a module that serves a static directory containing the HTML and CSS for the Spotify search.

Follow these steps:

1. Create a directory "spotifysearch" inside MusicApp and copy the breakout.html and style.css from GitHub in there.
2. Rename `breakout.html` to `index.html` to mark it as entry point for requests to this directory.
3. Create a new file "spotify.js" inside the routes directory. This file acts as our Spotify module that responds to the `/spotify` route.
4. Take a peek at `routes/index.js`. Its code is very similar to what you want to do in the Spotify module. However, the middleware function on the root path is not necessary. Instead, use the express.static middleware to serve the `spotifysearch` directory. Hint: `app.js` does a similar thing with the `public` directory, so you can find out how it's done.
5. Require the path module which is part of the node core. It allows you to easily resolve the relative path of the `spotifysearch` directory by calling the `.join()` method, like so:  

```
path.join(__dirname, '../spotifysearch')
```

## Task 3: Use the Spotify module

Difficulty: Easy

So far, the Spotify module is not used anywhere, so it is not possible to use the `/spotify` route just yet. This is what we need to change now.

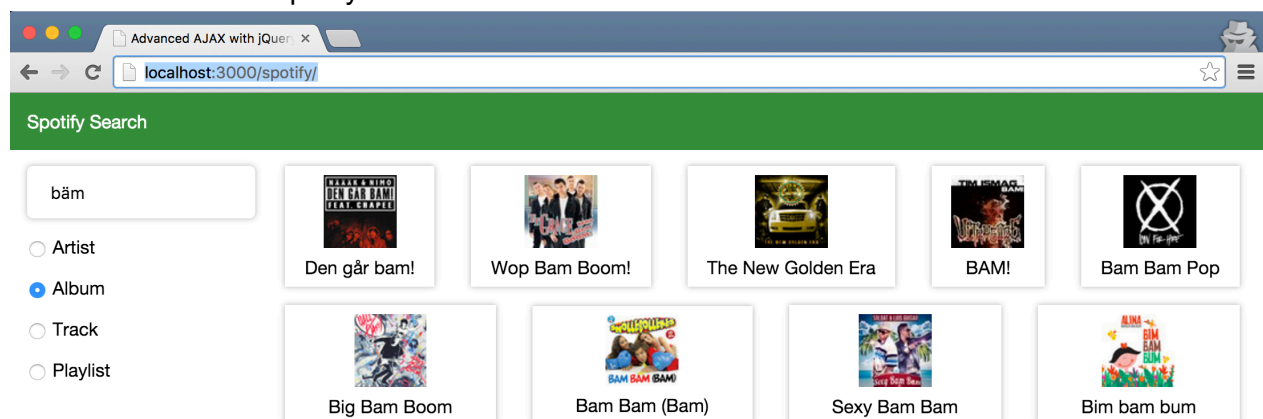
Follow these steps:

1. Require the Spotify module in `routes/index.js` and assign it to a variable `spotify`.
2. Tell the router object to use the Spotify module on the `/spotify` route.

That's it. Since the `index.js` file acts as a central module for routing all requests, you can now re-launch the app and open this URL in a browser:

<http://localhost:3000/spotify>

You should see the Spotify search then.





## Task 4: Visit Counting Module (Optional)

**Difficulty: Advanced**

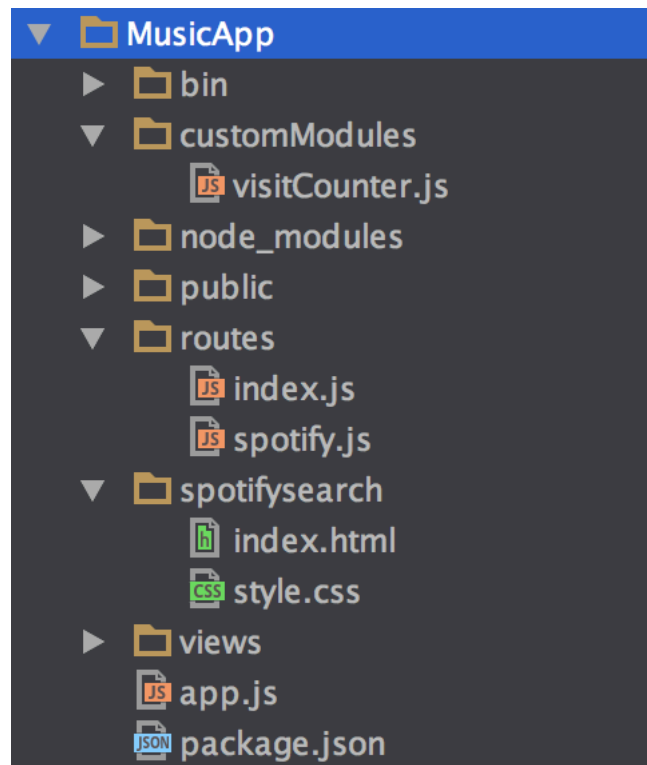
We provided a module named “visitCounter” in the [resource](#) folder for this assignment on GitHub. It already has some functionality that you don’t need to modify. It allows you to count requests to certain routes and paths.

Your task is to include it in your app to count all incoming requests. That means you have to use the middleware provided by the visitCounter in `app.js`.

Possible steps:

1. Require the module in `app.js`
2. Instantiate a VisitCounter object.
3. Use its middleware function `countMiddleware` very early in the middleware chain. In particular, include it before `express.static` and the `index.js` modules are used.
4. Create a route `/showVisits` that displays the request statistics. Use the `getCount()` method to retrieve an Object containing these statistics.
5. Send this object as JSON response.

In summary, you should have a structure like this:



## Submission

Put the solution to Task1 – Task4 into a ZIP folder. However, to keep the submission small, please do not include the `node_modules` directory. The correctors will do an `npm install` and do not need the `node_modules`.



Please turn in your solution via UniWorX. You can form groups of up to three people.  
After the submission deadline, push your solution to our [GitHub repository](#).

## Common Problems when first working with NodeJS

### Permissions

**Problem:** Permission denied when installing the express-generator.

**Potential reason:** The permissions for the directory /usr/local have been changed, so that your user cannot write anything in there.

**Solution** (for UNIX-based systems: change the owner of the directory with chown.

```
sudo chown -R {YOUR_USER_NAME} /usr/local
```

Confirm that the directory is now owned by your user with `ls -la /usr/`

```
spengler:~ Tobi$ ls -la /usr/
total 8
drwxr-xr-x@ 11 root wheel 374 28 Nov 13:04 .
drwxr-xr-x 34 root wheel 1224 3 Dez 10:20 ..
lrwxr-xr-x 1 root wheel 8 10 Sep 09:45 X11 -> /opt/X11
drwxr-xr-x 3 root wheel 102 20 Sep 02:11 adic
drwxr-xr-x 1054 root wheel 35836 28 Nov 12:58 bin
drwxr-xr-x 262 root wheel 8908 28 Nov 13:05 lib
drwxr-xr-x 183 root wheel 6222 28 Nov 12:58 libexec
drwxr-xr-x 22 Tobi admin 748 28 Nov 17:19 local
drwxr-xr-x 245 root wheel 8262 28 Nov 12:58 sbin
drwxr-xr-x 45 root wheel 1530 28 Nov 12:58 share
drwxr-xr-x 4 root wheel 136 18 Okt 01:36 standalone
```

**Problem:** Permission denied when running `node app.js`.

Potential reason: the executable flag is not set for the app file.

Solution: make the file executable, e.g.

```
chmod 775 app.js
```

### NodeJS binary does not exist

Problem: the node app does not start because there is “no such command” (or similar).

Potential reasons:

1. NodeJS was not installed correctly.
2. The directory, in which the node program is located has not been added to the PATH environment variable.
3. You are at the CIP pool and need to use `nodejs` instead of `node`

**If any other problems occur, please write us an email so we can extend this list.**