Assignment 7 (HF, major subject)

Due: Wed 14.12.2016; 20:00h (1 Week)

Goals

This assignment trains you in...

- Setting up node.js apps with the express generator.
- Serving static directories with express.

Task 1: Web 2.0? Difficulty: Easy

a) Name three characteristics of the Web 2.0
b) Is the Web 2.0 still the *state of the art*? Discuss the question by writing down three arguments for and against the statement.

Write down the answers in a file "task01.txt" and include it in your submission.

Task 2: Create a Node App with the Express Generator Difficulty: Easy

Use the express generator as shown in the tutorial, or download the example from GitHub: https://github.com/MIMUC-MMN/tutorials-16-17/tree/master/tutorial07/examples/MMNExpressApp

a) Inside the app root directory, run "npm install" to download the necessary dependencies
b) Run "npm start" to launch the app.
c) Go to http://localhost:3000 to verify that the app works.
d) Contact us, if it doesn't work.

No submission necessary, task continues in Task 3.

Task 3: Serve a static directory with NodeJS + Express Difficulty: Medium

Now, we want to create a module that serves a static directory containing the HTML file containing everything necessary to search the Spotify API.

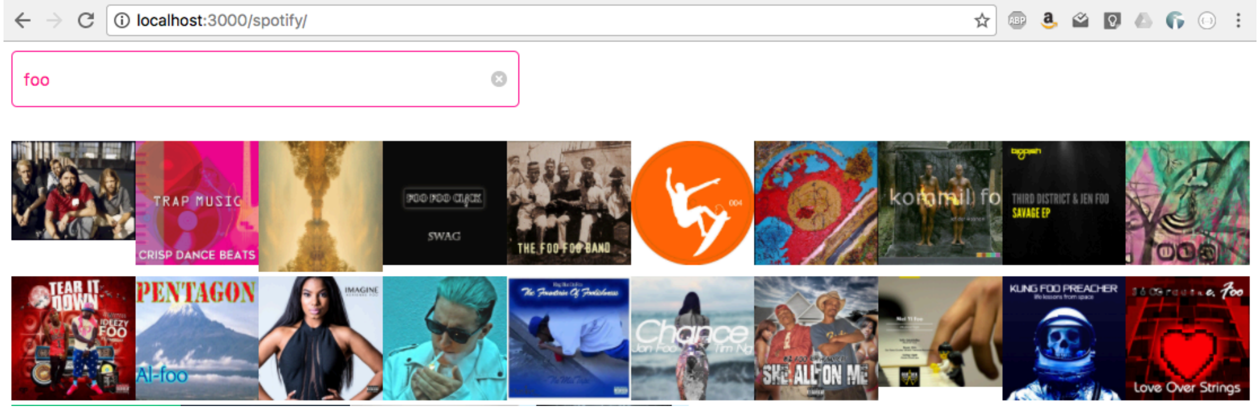
This task is more like a step-by-step tutorial than a difficult problem. To prepare yourself for the tutorials next week, it's important you know what's going on perform all necessary steps yourself.

Task continues on next page

Follow these steps:

1. Create a directory “`spotifysearch`” inside the app. In should be on the same level as the “public” directory.
2. Download the solution of the break out session in tutorial 05 here: https://raw.githubusercontent.com/MIMUC-MMN/tutorials-16-17/master/tutorial05/breakout-solution/spotify-breakout.html
   Put the file into the spotifysearch directory.
3. Rename the downloaded file to **`index.html`**. This will automatically mark it as entry point for the static directory.
4. Create a new file “`spotify.js`” inside the `routes` directory. This file acts as our Spotify module that responds to the `/spotify` route.
5. At the top of spotify.js, require the following modules
   a. **express**: provides the app functionality.
   b. **path**: is part of the node core. It allows you to easily resolve the relative path of the `spotifysearch` directory.
6. Instantiate the router Object for the spotify module:
   ```
   var router = express.Router();
   ```
7. Declare the middleware to serve the “spotifysearch” directory that you created earlier (and which contains the index.html) from the root of this module. This is done like this:
   ```
   router.use('/',express.static(path.join(__dirname,
   '../spotifysearch')));
   ```
8. Export the router:
   ```
   module.exports = router;
   ```
9. Now there is one final step to make everything work. We have to import our module and mount it to a proper path within app.js.
   a. Import the module with var spotify = require('./routes/spotify')
   b. Mount it with app.use('/spotify', spotify);
10. Start the app and use your browser to navigate to http://localhost:3000/spotify

If everything worked correctly, you should see the following when you start typing in the box:



localhost:3000/spotify/

foo

When you submit your solution, please **exclude** the node_modules folder.

Submission

Please turn in your solution via UniWorX. You can form groups of up to three people.

We encourage you to sign up for Slack! All you need is a CIP account and an email address that ends in “@cip.ifi.lmu.de”. Ask us if you don’t know how to get them.

If you have questions or comments before the submission, please contact one of the tutors. They are on Slack @tobi.seitz, @peterjuras and @thomas-weber. Remember, that they also want to enjoy their weekends ☺

It also makes sense to ask the question in our #mmn-ws1617 channel. Maybe fellow students can help or benefit from the answers, too!

CodeLabs

As always, everyone is invited to participate in the CodeLabs on Wednesday between 6 and 8 pm. We encourage you to go there if you struggle with installation or nasty bugs that just don’t seem to go away. The times are flexible, so you can join anytime during the CodeLab hours.