



Assignment 8 (HF, major subject)

Due: Wed 16.12.2015; 12:00h (10 days)

Goals

After doing these exercises,

- You know how to create a simple API with Express.
- You will have dealt with intellectual property and fair use in the specific use case of screen scraping.
- You will have MongoDB installed on your system (or have signed up for an [alternative service](#) providing access to a mongo database for you).

Task 1: Screen Scraping API

Difficulty: Intermediate

We want to create a web API that generates screen-shots from arbitrary sites for you (similar to, e.g., <http://www.shrinktheweb.com/>). There is an npm module that we can use for this: [webshot](#). It uses the WebKit-based [PhantomJS](#) to access remote websites and renders them.

To make things easier, download our code [skeleton from GitHub](#). It contains all the Express boilerplate and complete code on the front-end.

Then install webshot from the root of the directory that contains the skeleton:

```
npm install --save webshot
```

This fetches all the dependencies and makes webshot available while saving the dependency to the package.json file.

```
var webshot = require('webshot');
```

Here's the procedure that we suggest.

1. Create a new module: routes/shoot.js (similar to /routes/index.js)
2. Require the webshot module as shown above.
3. Require the path module.
4. Use the express.static middleware to serve the /screenshots directory.
5. In the root route of this module, we handle the screenshot requests.
 - a. The target URL for the screenshot is passed as a parameter in the GET request. You can access this information this way: req.query.url
 - b. Create a filename for the screenshot. We propose you use the target URL and add a '.png' to it. This will allow us to re-use already created screenshot files later on.
 - c. Perform the call to the webshot function. It takes three parameters: targetURL, outputPath (filename with absolute path, use path.join() for this), and a callback function.



- d. The front-end expects a JSON containing the path to the image file. Respond with a JSON like this:
- ```
{
 "status": "ok",
 "path": "/shoot/screenshots/google.de.png",
 "message": "re-used screenshot"
}
```
- e. [Optional:] Make sure the user passed a valid target URL.
- f. [Optional:] Use the `fs` module to check whether the screenshot file already exists. If it does, send its path back to the client rather than creating a new screenshot.
6. Require your new module in `routes/index.js` and use it on the `/shoot` route.

Put all your source code files **except the node\_modules directory** in the folder "task1".

## Task 2: Discuss Screen Scraping

Difficulty: Intermediate

Write 300 words on screen scraping. Answer the following questions:

1. What is screen scraping?
2. Who uses screen scraping?
3. Are we allowed to show screen shots of arbitrary web sites on our own web site?

Put your text in the folder "task2".

## Task 3: MongoDB set up

Difficulty: Easy

For the next tutorial we will need MongoDB. To avoid installation during the tutorial, we ask you to download and install it locally on your machine.

**General link to download MongoDB:** <https://www.mongodb.org/downloads#production>

Windows: <https://docs.mongodb.org/master/tutorial/install-mongodb-on-windows/>

Mac OSX: <https://docs.mongodb.org/master/tutorial/install-mongodb-on-os-x/>

Linux, e.g. Ubuntu: <https://docs.mongodb.org/master/administration/install-on-linux/>

CIP Pool: The CIP pools do not have MongoDB at the moment. You can sign-up for the free version of <https://mongolab.com/> and access the database there.

To make sure it works, type `mongo --version` in a terminal.

## Submission

Put the solution to Task1 – Task2 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](#).