



## Assignment 8 (HF, major subject)

Due: Wed 21.12.2016; 20:00h (1 Week)

### Goals

This assignment trains you in...

- Using middleware and routing in Express apps
- Connecting the front-end to the back-end of an app
- Securing paths with Passport

### Task 1: Screen Scraping

**Difficulty: Easy**

Write 300 words on the concept of “screen scraping” by answering 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 answer in the file “task1.txt”.

### Task 2: Screenshot Module

**Difficulty: Medium**

Extend a basic NodeJS app with a feature that allows the user to take screenshots of any website. All the user has to do is enter a URL, and click a submit button. The web app will then receive the URL, generate the screenshot and respond with the URL of the screenshot image.

We provide a skeleton for this task: <https://github.com/MIMUC-MMN/assignments-16-17/tree/master/08%20-%20nodejs%20authentication>

Your remaining job is to ...

- a) include the module into your NodeJS-Express app
- b) extend a static HTML file that shows the user interface. This page asynchronously requests the screenshots and displays them once they have been created.

Figure 1: After entering a URL and clicking “GO!”:

The figure shows a screenshot of a web browser window titled "Screenshot Service". At the top, there is an input field containing the URL "https://www.npmjs.com/package/w". Below the input field is a white button with the text "GO!". Underneath the button, the word "Loading..." is displayed next to a circular progress bar that is partially filled. The rest of the page is a plain light gray color.



Figure 2: The Screenshot has been created.

To further illustrate the workflow, we made a short demo video: <https://youtu.be/oFoTeVAFBuc>

### Here's a rough step-by-step guide about what you have to do:

- a) First, require the 'shoot' module in app.js. The file-path is routes/shoot.js.
- b) Mount the 'shoot' module to the path '/shoot' in app.js.
- c) Start the app with 'npm start'
- d) Open your browser and go to <http://localhost:3000/shoot> - you should see the webpage with the form and the button.
- e) Extend the code in shoot/index.html. When the user submits the form, we prevent the default action and instead perform an asynchronous GET request to <http://localhost:3000/shoot/screenshots/create> (or simply with relative path "screenshots/create"). You have to pass an URL query parameter "url". The value of the parameter should be the website that you want to take a screenshot of. The final becomes: <http://localhost:3000/shoot/screenshots/create?url=google.de>
- f) The service (our shoot module) responds with a JSON object like this:

```
{  
  "status": "ok",  
  "message": "re-used screenshot",  
  "path": "/shoot/screenshots/testde.png",  
  "responseTime": 0  
}
```

You can now use the response to create an image and set its "src" attribute to the path you can find in the answer. For example:

```

```

That's it! You should now be able to create screenshots of arbitrary websites. Let us know if there's anything we can help you with.

Include all your code files in the submission, but please **exclude the node\_modules folder**.



### Task 3a: Require Authentication for /shoot

Difficulty: Medium

This is an extension of Task 2.

We now want to restrict access to the '/shoot' URL path. Users who are not logged in should not be able to create and retrieve screenshots. If a non-logged in user visits

<http://localhost:3000/shoot> they are automatically redirected to <http://localhost:3000>

Change the code in app.js to achieve this.

### Task 3b: OPTIONAL Task for Professionals.

Difficulty: Hard

Imagine you want to allow the users to create screenshots that only he or she can retrieve. So, if User A creates a screenshot for bing.com, User B does not have access to the screenshot.

To achieve this functionality, a couple of changes are necessary in the 'shoot' module (routes/shoot.js)

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.