



Assignment 8

Due: Wed 18.12.2019; 18:00h (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: 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/omm-ws1920/tree/master/assignments/08-nodejs/skeleton/ScreenshotApp>

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!":

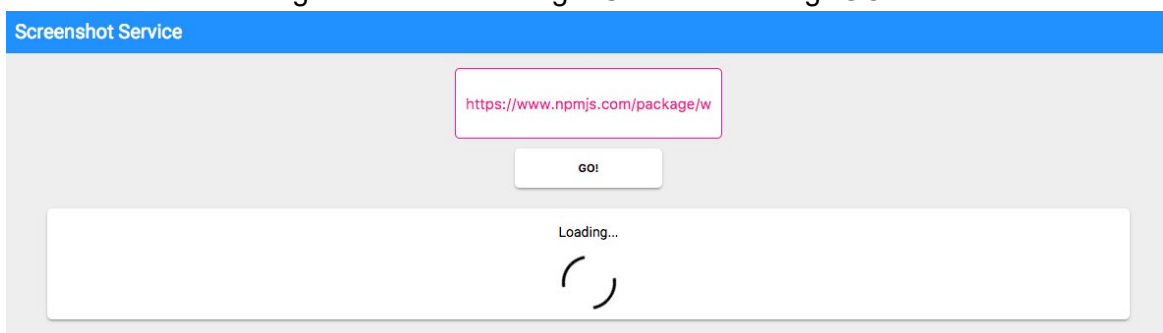
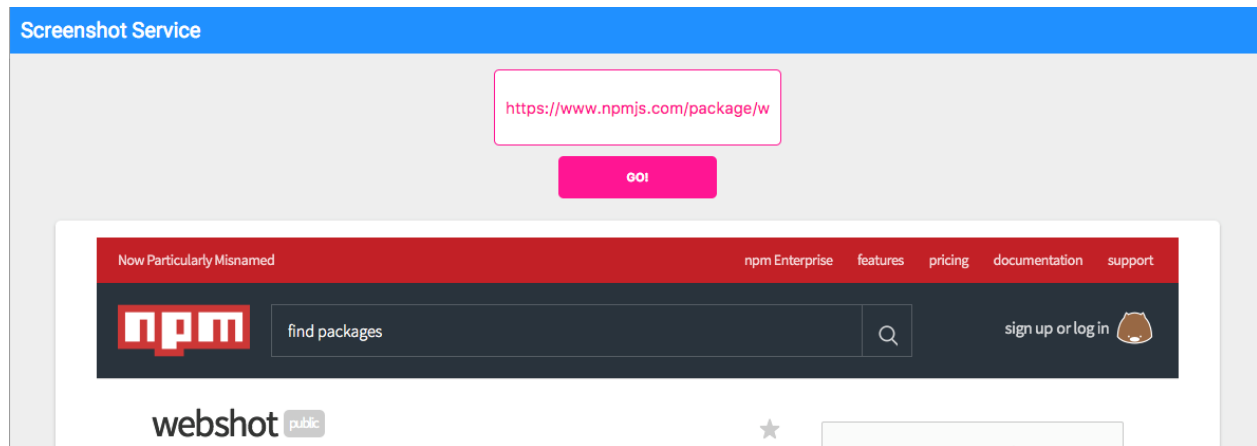


Figure 2: The Screenshot has been created.



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:

- First, require the 'shoot' module in app.js. The file-path is routes/shoot.js.
- Mount the 'shoot' module to the path '/shoot' in app.js.
- Start the app with 'npm start'
- Open your browser and go to <http://localhost:3000/shoot> - you should see the webpage with the form and the button.
- 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>
- 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 2: 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 3: 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 as ZIP file via Uni2Work. 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 [@Aleksa](#) and [@Andre](#), remember that they also want to enjoy their weekends 😊

It also makes sense to ask the question in our #omm-ws1920 channel. Maybe fellow students can help or benefit from the answers, too!