



Training & Certification

Lab 3.1 - Deliver Data from a Library API

The labs-1 folder contains the following files:

- `data.js`
- `package.json`
- `validate.js`

The `data.js` file contains the following:

```
'use strict'
const { promisify } = require('util')
const { randomBytes } = require('crypto')
const timeout = promisify(setTimeout)

async function data () {
  await timeout(50)
  return randomBytes(10).toString('base64')
}

module.exports = data
```

The `data.js` file exports a function that returns a promise (an `async` function) that resolves to a random BASE64 string. This function represents some kind of asynchronous data source.

The `package.json` file contains the following:

```
{
  "name": "labs-1",
```

```
"scripts": {  
  "start": "echo \"TODO: SET THE START SCRIPT\" && exit 1"  
}
```

Using any Node core library and/or web framework create an HTTP server that meets the following criteria:

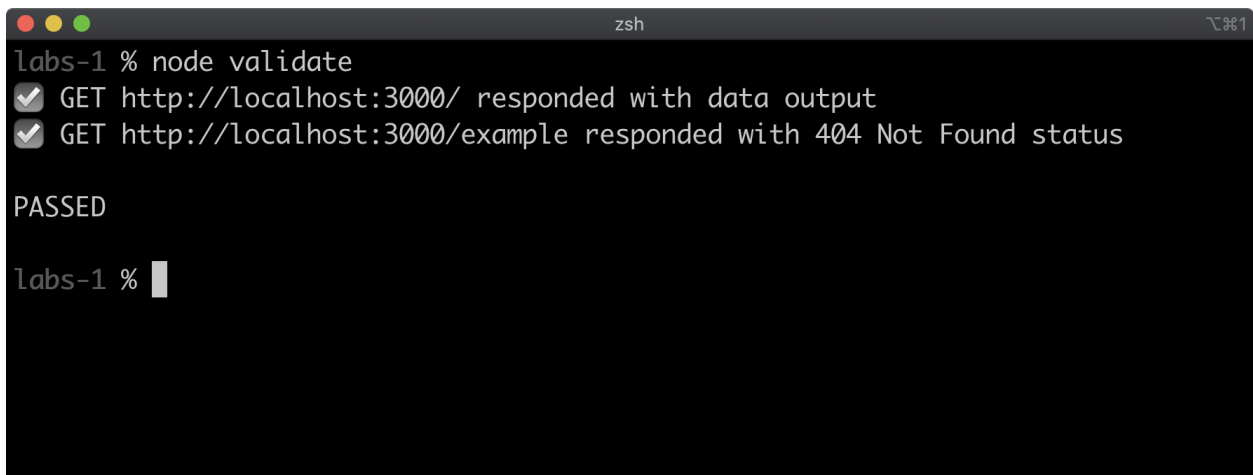
- Listens on localhost
- Listens on port 3000
- Responds to HTTP GET requests to / with data from the `data` function as exported from the `data.js`
- Responds with a 404 to GET requests to any other route

The `package.json` `start` script must contain a command to start the server.

Run the following command to check whether the created server meets the criteria:

```
node validate
```

If the server was correctly implemented, the output of this command should be as follows:

A terminal window titled 'zsh' with a dark background. The prompt is 'labs-1 %'. The command 'node validate' has been executed. The output shows two checkmarks: '✓ GET http://localhost:3000/ responded with data output' and '✓ GET http://localhost:3000/example responded with 404 Not Found status'. Below this, the word 'PASSED' is displayed. The prompt 'labs-1 %' is followed by a cursor.

```
labs-1 % node validate  
✓ GET http://localhost:3000/ responded with data output  
✓ GET http://localhost:3000/example responded with 404 Not Found status  
  
PASSED  
  
labs-1 %
```