SC-T-213-VEFF - Web Programming I

Lab Assignment 5 – Server-side JavaScript

Reykjavík University | Deadline: **11th March 2019, 23:59**

The topic of this assignment is: **Writing server-side JavaScript using Node.js**.

1 Part 1: Node Module

Write two JavaScript files. The first one, *math.js*, shall contain a Node.js module providing two functions. The first function, called *doDivision(a,b)*, shall return the result of a divided by b. The second function, called *stringifyDivision(a,b)*, shall return a string describing the divison in the format "*a* divided by *b* is *result*". For example, stringifyDivision(4,2) shall return "4 divided by 2 is 2" (there is no need to anyhow round the result). stringifyDivision shall make use of doDivision. The second file, *mathUser.js*, shall import the module and demonstrate its functionality by calling the two functions and printing their return values to the console. It is sufficient to call each function once (with arbitrary values).

2 Part 2: Basic HTTP Server

Write an HTTP server (in a file called *httpServer.js*) using Node.js (and in-built modules) that listens to requests on IP 127.0.0.1, port 3000. It shall answer two kinds of requests:

1. If the URL is `http://127.0.0.1:3000/divide?a=[value]&b=[value]` and the HTTP method is GET, the server shall return the string that the function stringifyDivision in Section 1 would produce. For example, if you call `http://127.0.0.1:3000/divide?a=4&b=2`, the server shall return "4 divided by 2 is 2". The content type of the return value shall be text/plain, and the HTTP status code shall be 200.

2. If any other URL is called, the server returns a text string saying "This operation is not supported." (with content-type text/plain and HTTP status code 405).

The server shall never throw any exceptions/crash based on user requests. For example, if the user leaves out one of the parameters in `http://localhost:3000/divide`, the server shall fall back to the second case.

3 Requirements and Hints

For the overall assignment, the following requirements shall be fulfilled:

1. There are no restrictions on the JavaScript version.

2. You are only allowed to use Node.js and its built-in modules. No external modules/libraries (such as Express) are allowed.

We give the following hints:

- If you need any pointers on how to begin, start by going through the *learnyounode* tutorials.
- For Part 2, look at the http and url modules of Node.js
- For the URL in Part 2, remember that the part after the question mark (`?a=[value]&b=[value]`) is called the *query* part of a URL.

Submission

The lab is submitted via Canvas. The following deliverable needs to be included:

1. A single zip file containing the three JavaScript files *math.js*, *mathUser.js*, and *httpServer.js*.