

NodeJS, CommonJS, ExpressJS, Response

Finland Visit 2015

April 21, 2015

NodeJS

- ▶ Server side runtime for JavaScript
- ▶ Can run any JS, but mainly used for “web applications”
- ▶ Run JS scripts from Codio terminal using
\$ node script_name.js

CommonJS Modules

- ▶ Node has a large package/library ecosystem
- ▶ Useful for common functionality - just use a pre-written module
- ▶ Follows CommonJS standard
- ▶ Modules are managed by the “Node Package Manager” NPM
- ▶ Pre-installed on Codio, run from terminal using
`$ npm install npm_package_name`
- ▶ Adds the package code to a `node_modules` directory
- ▶ Can then `require('npm_package_name');` in your script

Request Module

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

- ▶ An HTTP client written in JavaScript
- ▶ Can be used by your scripts to access other HTTP resources on the web
- ▶ *Very useful!*
- ▶ Main methods correspond to HTTP verbs:
 - ▶ GET `request.get(url, callback);`
 - ▶ PUT `request.put();`
 - ▶ POST `request.post();`
 - ▶ DELETE `request.delete();`
- ▶ Be careful, because callbacks are asynchronous

Express Module

```
var express = require('express');  
var app = express();
```

- ▶ An HTTP server written in JavaScript
- ▶ The “opposite” of the `request` module
- ▶ Can be used by your scripts to provide an HTTP endpoint on the web
- ▶ *Very useful!*
- ▶ Main methods correspond to HTTP verbs:
 - ▶ `GET app.get('/', function(req, res) {});`
 - ▶ etc.
- ▶ Most often used for serving HTML websites, but can be used directly for APIs etc.
- ▶ The basis for the `restify` NPM module we will look at tomorrow