

Node.js, Express Framework, MySQL

Installing node.js

1. Download Node.js: <https://nodejs.org>
2. Run the installer and follow the instructions
3. Reboot the computer
4. Access the command line and test the project creation

Creating the first node.js project

1. Open the command line
2. Create a folder for the project: `mkdir olaMundo`
3. Access the folder: `cd olaMundo`
4. Initialize package.json: `npm init`
5. Answer the questions to create the project (ENTER means you agree). Example:

```
C:\olaMundo>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (olamundo)
version: (1.0.0)
description: primeira aplicacao
entry point: (index.js)
test command:
git repository:
keywords:
author: Eu
license: (ISC)
About to write to C:\olaMundo\package.json:

{
  "name": "olamundo",
  "version": "1.0.0",
  "description": "primeira aplicacao",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Eu",
  "license": "ISC"
}
Is this OK? (yes)
```

6. Create the file `index.js` on the project folder and paste the following code:

```
const http = require('http'); // Loads the HTTP module
const server = http.createServer( // Creates a server instance
  // Send na HTTP response with content "Ola mundo":
  function(request, response) {
    response.writeHead(200, {'Content-Type': 'text/html'}); // Starts
    the response as an html page
    response.write('<html><body><h1>Ola Mundo</h1></body></html>');
    response.end(); // Ends the response
  }
);
server.listen(3000); // The server listens on port 3000
console.log('Servidor a correr na porta 3000');
```

7. Start the server: on the command line, execute the following command `node index.js`
8. Open the browser in `localhost:3000` and verify the message is presented

Use Express Framework

1. (If you are using UPT's network) Define the proxy server. In the command line:

```
set http_proxy=http://proxy.uportu.pt:3128
set https_proxy=http://proxy.uportu.pt:3128
```

2. Install the express module using the option `--save` to store the dependency on the project

```
npm install express --save
```

3. Create the `app.js` file to configure the app server with the following code:

```
const express = require('express');
const app = express();
const server = app.listen(3000, function(){
  console.log("servidor a funcionar na porta %s", server.address().port);
});
```

4. Create 4 routes with different behaviours:

- a. `"/`: Shows "Raiz"
- b. `"/ajuda"`: Shows "Ajuda"
- c. `"/hora"`: Shows the current date/time on the format `dd/mm/yyyy hh:MM`

Note: `Date()` function documentation: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Date

- d. `"/antigos"`: Allows accessing static HTML/CSS/JS files (copy the folder `"solução_html"` with the solution to the HTML/CSS/JS exercises)

Use MySQL with node.js

1. Install (with `--save`) modules `sequelize` and `mysql`
2. Using `mysql` (don't Forget to create the connection file), build a webpage that allows viewing all the information on the cars stored on the database (don't forget to create routes). Example:

Marca	Modelo	Cilindrada	Fotografia
Audi	A3	2000	
Ford	Fiesta	1300	
Ford	Focus	1600	
Renault	Clio	1200	

3. Using `sequelize`, build a webpage that allows viewing all the information about the brands (marca) on the database. Example:

Cod	Marca	Nome	Marca
1		Audi	
2		Ford	
3		Renault	
6		Fiat	
7		mitsubishi	