NODE JS

* Install node js from <https://nodejs.org/it/download/>
* $Sudo npm install –g express-generator
* $express *<name\_app>* --hogan –c less
* $cd *<name\_app>* && npm install
* $npm install –g nodemon
* $npm install ejs
* $nodemon *<name\_app>*
* incase we want to just run it🡪$DEBUG=*<name\_app>*:\*npm start

FULLSTACK node.js mongodb react

Create a package.json file with all the project and npm information

* npm init

install the following dependencies

* npm install –save express
* npm i –S mongodb
* npm i –S react react-dom

let install the developer dependencies that we will use only in the developer environment

* npm install –save-dev webpack
* npm i –D babel-cli babel-loader babel-preset-es2015 babel-preset-stage-2 babel-preset-react

Now instead of restarting everything letàs use the nodemon dependency

* npm i –D nodemon
* npm I –D eslint eslint-plugin-react babel-eslint

now let’s create a source code and public directory:

* mkdir src
* mkdir public
* mkdir api

let’s create empty files for the structure:

* touch api/index.js
* touch public/index.html
* touch src/index.js

into the package.json file change the content inside the curly bracket of “scripts” to:

* {“start”: “nodemon –exec babelweb-node server.js –ignore public/”,
* “dev”: “webpack -wd”
* }

at root level add a file named “webpack.config.js” and add the following code:

|  |  |
| --- | --- |
| module.exports = { | |
|  | entry: './src/index.js', |
|  | output: { |
|  | path: \_\_dirname + '/public', |
|  | filename: 'bundle.js' |
|  | }, |
|  | module: { |
|  | loaders: [ |
|  | { |
|  | test: /**\.**json$/, |
|  | loader: 'json-loader' |
|  | }, |
|  | { |
|  | test: /**\.**js$/, |
|  | loader: 'babel-loader' |
|  | } |
|  | ] |
|  | } |
|  | }; |

Then add also in the root folder a file named “.babelrc” with the following code

|  |  |
| --- | --- |
| { | |
|  | "presets": ["react", "es2015", "stage-2"] |
|  | } |

Then also create in the root folder a file named “.eslintrc.js” with the following code

|  |  |
| --- | --- |
| module.exports = { | |
|  | "parser": 'babel-eslint', |
|  | "env": { |
|  | "browser": true, |
|  | "commonjs": true, |
|  | "es6": true, |
|  | "node": true |
|  | }, |
|  | "extends": ["eslint:recommended", "plugin:react/recommended"], |
|  | "parserOptions": { |
|  | "ecmaFeatures": { |
|  | "experimentalObjectRestSpread": true, |
|  | "jsx": true |
|  | }, |
|  | "sourceType": "module" |
|  | }, |
|  | "plugins": [ "react" ], |
|  | "rules": { |
|  | "indent": ["error", 2], |
|  | "linebreak-style": ["error","unix"], |
|  | "quotes": ["error","single"], |
|  | "semi": ["error","always"], |
|  | "no-console": ["warn", { "allow": ["info", "error"] }] |
|  | } |
|  | }; |

NOTE: if you want to run the file you should type ./node\_modules/.bin/babel-node server.js

If you want to avoid it you can go on the command line and write

* vi ~/.bash\_profile

and insert

* export PATH-$PATH:./node\_modules/.bin

rerun the termina now you can run just writing babel-node <name\_app>.js

add a config.js file and add the following code to set up the port of the server

const env = process.env;

export default {

port: env.PORT || 8080

};

NOTE: to run into the dev env:

* npm run dev

and add the bundle.js file that is in the public folder in the html file (usually at the end of footer into the body)

<script src="/bundle.js" charset="utf-8"></script>