**AJAX**

**DEADLINE:** 31/05/2020

**FOLDER STRUCTURE**

|  |  |
| --- | --- |
| FL13\_HW16/\*     └─ task/**\*\***  └─ homework/\*     └─ app/\*  └─ index.html\*  └─ .eslintrc.js  └─ js\*  └─ index.js\*  └─ styles\*  └─ index.css\*     └─ server/\* | \* - required  \* - tentative (is not required)  \*\* - restricted (no need to submit) |

**TASK**

Using AJAX, you need to implement simple application to manage users.

1. As a user I want to see a form with *‘name’*, *‘username’* inputs and *‘Add New User’* button. When button is clicked – app sends request to server and tries to create a new user with proper values of inputs as well. While request is processed button should be disabled. In case of success response update user list.
2. After page loading I want to see a list of existing users under this form. While request for getting users is in process there must be some *notification* or *loader* *(‘Loading….’* will be enough) to inform about execution process.
3. Each list element must be represented as*: ‘UserId’* field, *‘name’* and *‘username’* inputs, *‘Update’* and *‘Delete’* buttons.

During update/delete requests proper button should be disabled (you can block all UI or just current user element, but it is not necessary). After getting response update user list and enable buttons.

Server part has been implemented already and provides such API's:

* GET **/users** – returns list of users

*Request Options*: no

*Response*:

Code: 200 / Body: JSON – list of users

* POST **/users** – creates new user

*Request Options*:

{

name: ‘string’,

username: ‘string’

}

*Parameter content type:* application/json

*Response*:

Code: 201 / Body: empty

* PUT **/users/:userId** – updates existing user

*Request Options*: Body:

{

name: ‘string’,

username: ‘string’

}

*Parameter content type:* application/json

*Response*:

Code: 204 / Body: empty

* DELETE **/users/:userId** – deletes user

*Request Options*: no

*Header: ‘Authorization’ :* ‘admin’

*Response*:

Code: 204 / Body: empty

**IMPORTANT!** **Before you begin working:**

* install **Node.js** (https://nodejs.org/ )
* go to **server** folder
* open **terminal(or cmd)**
* run **npm install** command
* run **npm run server** to launch server

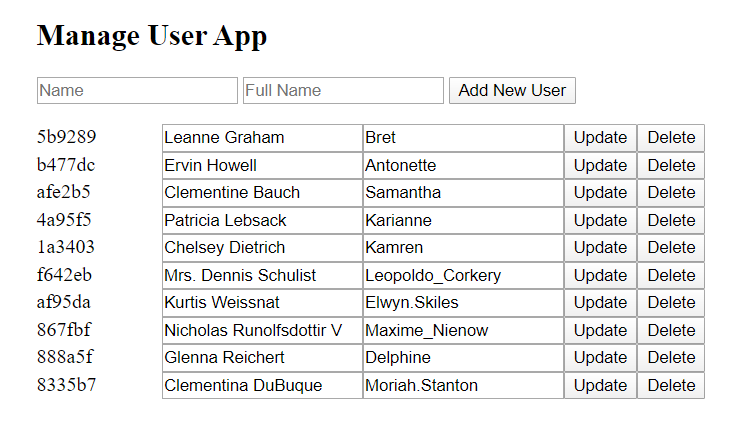
(there should be message: *Server is listening on port: 3000*)

* to stop running server press **Ctrl+C**

*Note: after restarting server all data will be reverted to its initial state*

**EXAMPLE**

**TIP: Don’t spend your time on styling, just create simple but efficient app.**

*It might look like this*:****

**RESTRICTIONS**

* Do not use **fetch** for this homework.

**BEFORE SUBMIT**

* In order to use npm package manager install nodejs (https://nodejs.org/ )
* Install eslint to check your code (npm install -g eslint)
* - open a terminal(or cmd)
* - go to src folder
* - run eslint (i.e. eslint credits\_handler.js)
* Code should be without ‘errors’
* Code should be clean, readable and well formatted
* Remove all redundant comments
* Test all your functions

**SUBMIT**

* The **FL13\_HW16** folder without **task** folder should be uploaded to github repository “**FL-13**” into **master** branch.

**USEFUL LINKS**

* <https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX/Getting_Started>
* <https://www.codecademy.com/articles/what-is-rest>