

# React Task 4 - Async Redux.

---

This is the fourth part of the application.

In this assignment, you should continue to work on your project with a slight modification.

This module designed to teach how to work with asynchronously Redux.

## Prerequisite:

Firstly, you need to install the `redux-thunk` module in your project using `npm`:

1. Go to your project directory
2. Install `redux-thunk`

```
npm install --save redux-thunk
```

3. Run your project

```
npm run start
```

## Project structure requirements

1. Create a new folder and `jsx` file for new components:
  - `src / components / PrivateRouter / PrivateRouter.jsx`
2. Rename `CreateCourse` folder and file:
  - `src / components / CreateCourse / CreateCourse.jsx => CourseForm / CourseForm.jsx`
3. Create new `js` files for thunk functions:

```
src
|-- store
|  |-- user
|  |   |-- thunk.js //new file
|  |   |-- //old files
|  |-- courses
|  |   |-- thunk.js //new file
|  |   |-- // old files
|  |-- authors
```

```
|      |-- thunk.js //new file
|      |
|      |__ //old files
|__ ...
```

4. Admin credentials. These credentials provide role **ADMIN**.

```
email: admin@email.com
password: admin123
```

5. APIs from SWAGGER for Module 4:

- /courses/add [POST]
- /courses/{id} [PUT]
- /authors/add [POST]
- /courses/{id} [DELETE]
- /logout [DELETE]
- /users/me [GET]



#### WARNING:

For fetching requests you should add **Authorization** header with user's token

**Authorization: token**

## Criteria (25 points max)

### APIs

- [3 points] - Use **Thunk** for asynchronous actions.
- [2 points] - Create request to get current user if localStorage contains token. (See **/users/me [GET]** in the SWAGGER);

### PrivateRoute Component

- [2 points] - Create new component **PrivateRouter**.  
Use **PrivateRoute** for routes:
  - **/courses/add**;
  - **/courses/update/:courseId**.  
(**Private route** in COMPONENTS).

### CourseForm Component (update mode)

- [1 point] - **Update Course** form should be opened by route **/courses/update/:courseId**.
- [1 point] - Show route only for user with **ADMIN** role.
- [1 point] - Show information about course in fields with the same name.

- [2 points] - Send updated course to the back-end. (See `/courses/{id}` in the SWAGGER).
- [1 point] - Save updated course to the **store** after success response.  
(**Update course** in COMPONENTS).

### CourseForm Component (create mode)

- [1 point] - Show `/courses/add` route only for user with **ADMIN** role.
- [2 points] - Send new course to the back-end. (See `/courses/add` in the SWAGGER).
- [2 points] - Send new author to the back-end. (See `/authors/add` in the SWAGGER).
- [1 point] - Save new course to the **store** after success response.  
(**Add new course** in COMPONENTS).

### CourseCard Component

- [1 point] - Show buttons `Delete course` and `Update course` when role - ADMIN.
- [2 points] - Sent delete request to the back-end. (See `/courses/{id}` in the SWAGGER).
- [1 point] - Delete the course from the **store** after success API response.  
(**Course card with ADMIN role** in COMPONENTS.pdf).

### Header

- [2 points] - Modify **Logout** functionality. Call endpoint for logout on a backend.(See `/logout` in the SWAGGER)  
(**Header** in COMPONENTS).

Please find detailed description of components and functionality in the file COMPONENTS