Web Applications [2024/2025]

Lab 11: Loading data from the server

You will **update the client part** of the web application developed so far to invoke some of the **APIs** designed and implemented on the server side. Specifically, the list of films displayed in the web application must be gathered from the server.

1. Load the film list from the server

Modify the client part of the web application so that:

- When the home page of the application loads, the available **films are retrieved from the server** by invoking the corresponding API endpoint. The retrieved data must be **stored into the application's state**. Film static data must be deleted from the application code.
- When the user clicks on a filter, the list of **filtered films is gathered from the server** by invoking the corresponding API endpoint. The films must be filtered directly on the server side.

  Note: use only a single state to store the list of films. This list will contain either all films or the filtered films. This is different from the previous lab where the state always contained all the films, whereas the filter happened on the client only when a given filter was activated.
  The reason why it is better to use a single state is that when the add, update, and delete operations will be implemented by communicating them to the server, having a duplicate state carries a high risk of introducing bugs, because one must always remember to update both states, but only use the correct one in each component.

Beware: currently, all the other operations (i.e., add, update, and delete) are still not linked to the server (i.e., they act on the local copy of the data). Also, add and update will have no visible effects if the list of films is reloaded when the app navigates away from the form, back to the "/" route or a filter route.

2. *Optional*: Handle slow server responses

- Handle a slow server response, in particular in the home page when the application loads. Show a placeholder value ("waiting", and/or a loading icon) when you are still waiting for the data from the server.
  - You can make the server response slower by using the setTimeout function in the server code handling the corresponding API.

Hints:

1. ***Remind (again!)****: in this lab, the previously implemented add, edit and delete operations work, but every time a filter is selected (thus the list of films is retrieved from the server) the add/edit/delete operations will be lost! This is perfectly fine considering that you are not communicating these changes to the server.*