01UDFOV/01TXYOV Applicazioni Web I / Web Applications I [2021/2022]

BigLab 2A: APIs with Express

What Are We Building This Week?

This week you will create a **basic back-end for your FilmLibrary**. To do so, you will use the Express framework. The back-end must implement a series of **APIs** to support the main features of the web-based FilmLibrary you developed in BigLab1: **create**, **read**, **update**, and **delete** the films. The data will be persistently stored in an SQLite database.

Step-by-step Instructions

- Design a set of APIs to support the main features of your web-based FilmLibrary. The APIs should allow the application to:
  - **Retrieve** the list of all the available films.
  - **Retrieve** a list of all the films that fulfill a given filter (i.e., the same filters described in BigLab1).
  - **Retrieve** a film, given its "id".
  - **Create** a new film, by providing all relevant information – except the "id" that will be automatically assigned by the back-end.
  - **Update** an existing film, by providing all the relevant information, i.e., all the properties except the "id" will overwrite the current properties of the existing film. The "id" will not change after the update.
  - **Mark** an existing film as favorite/unfavorite.
  - **Delete** an existing film, given its "id".

  Data *passed to* or *received from* the API should be in **JSON format**. You must list the designed APIs, together with a short description of the parameters and the exchanged entities, in the README.md file that is included in your repository. Be sure to identify which are the collections and elements you are representing. You might want to follow this structure for reporting each API:

  ```
  [HTTP Method] [URL, optionally with parameter(s)]
        [One-line about what this API is doing]
        [Sample request, with body (if any)]
        [Sample response, with body (if any)]
        [Error response(s), if any]
  ```

- Implement the designed HTTP APIs with Express. The films must be **stored persistently** in an SQLite database. To this end, you can use the database "films.db" that is included in your group's BigLab2 repository. The database contains two tables, "users" and "films", with each user that may have one or more films. The film entries have all the fields already described in BigLab1. During this week, consider only the information of the "films" table. In other words, when you are retrieving films you should not consider the value of the "user" column. When you are creating new films, you should assign all of them to the same user (e.g., user with id=1). The information about users will be used during the last week when you will implement the login functionality.

- Test the realized API with the **REST Client extension** for Visual Studio Code or similar tools.

Hints:

1. *Carefully read the general specification of the BigLab before starting to design of your APIs:* *https://polito-wa1-aw1-2022.github.io/materials/labs/BigLab2/BigLab2.pdf*
2. *To visualize and edit the database, you can exploit:*
   a. *"DB Browser for SQLite", available at https://sqlitebrowser.org/*
   b. *SQLiteStudio, available at https://sqlitestudio.pl/*
3. *For handling dates and times, you can exploit Day.js*