01UDFOV/01TXYOV Applicazioni Web I / Web Applications I [2023/2024]

Lab 2: Database integration

In this lab, you'll integrate your JavaScript application with a local database. In the first part, you'll implement functions to retrieve data from the database, and in the second part, you'll make modifications to the data stored in the database.

1. Retrieve data from the database.

The database includes a collection of films with fields detailed in the first lab. Download the `films.db` database from the following link:
https://github.com/polito-webapp1/lab-2024/blob/main/lab02-node-database/films.db

Modify the program from the previous lab (you can either build upon your existing solution or use the Lab 1 solution as a starting point: https://github.com/polito-webapp1/lab-2024/tree/main/lab01-node) by adding the following features as asynchronous methods to the **FilmLibrary** for retrieving data from the database:

a. Retrieve all the stored films and return a Promise that resolves to an array of Film objects.
b. Retrieve all favorite films and return a Promise that resolves to an array of Film objects.
c. Retrieve all films watched today and return a Promise that resolves to an array of Film objects.
d. Retrieve films whose watch date is earlier than a given date (received as a parameter). Return a Promise that resolves to an array of Film objects.
e. Retrieve films whose rating is greater than or equal to a given number (received as a parameter). Return a Promise that resolves to an array of Film objects.
f. Retrieve films whose title contains a given string (received as a parameter). Return a Promise that resolves to an array of Film objects.

Finally, confirm the proper functioning of the implemented methods by calling them and printing the results.

2. Modify the data stored in the database.

Before proceeding with this exercise, make a copy of the local database file, as the following methods will permanently modify its content.

Add the following features as methods to the **FilmLibrary** object:

a. **Store** a new movie into the database. After completion, print a confirmation/failure message.
b. **Delete a movie** from the database (using its ID as a reference). After completion, print a confirmation/failure message.
c. **Delete the watch date** of all films stored in the database. After completion, print a confirmation/failure message.

Notes:

1. As covered in the lectures, you can connect to an SQLite database using the following module: **sqlite3** (https://www.npmjs.com/package/sqlite3)
2. To browse the content of the database, you can use one of the two following tools:
   a. Download the Visual Studio Code *SQLite extension* (you can search for it in VSCode extension hub or through the following link): https://marketplace.visualstudio.com/items?itemName=alexcvzz.vscode-sqlite
   b. Download the application *DB Browser for SQLite*: https://sqlitebrowser.org/dl/