

# myFilms - Proiect Java Spring

Student Marius-Augustin Nițu

## 1. Introducere

Proiectul, myFilms, reprezintă o aplicație web pentru gestionare unei colecții de filme folosind Java Spring, o baza de date MySQL și un API public. Acesta dorește să implementeze funcționalitatea comună pe care IMDb, Letterboxd o oferă: posibilitatea de a crea și gestiona o lista proprie cu filmele dorite.

## 2. Tehnologii utilizate

Java - limbajul principal de programare utilizat pentru dezvoltarea aplicației.

Bibliotecile și framework-urile Java folosite:

- The Spring Framework – dezvoltarea aplicației web folosind arhitectura Model-View-Controller (MVC).
- Java Database Connectivity (JDBC) – gestionarea și executarea de SQL Queries pentru baza mea de date MySQL.
- Jackson – prelucrarea JSON-urilor obținute în formatul dorit.

## 3. Proiectarea Aplicației

### 3.1 Arhitectura Model-View-Controller

Framework-ul Spring a fost folosit pentru a realiza Arhitectura Model-View-Controller a aplicației. Modelul este reprezentat de clasa Film (obiecte pe care doresc să le gestionez în colecția mea). View-urile sunt diversele pagini HTML (pagina principală, pagina de adăugare) folosite pe baza cererilor gestionate de Controller.

### 3.2 Baza de date

Baza de date MySQL a fost proiectată folosind MySQL Workbench cu scopul de a asigura persistența datelor în aplicația myFilms. Această conține doar informația esențială pentru un obiect Film, structura sa fiind următoare:

myFilms	
🔑 id	int
title	varchar?
year	int?
rating	int?

MySQL Workbench - instrument utilizat pentru crearea și administrarea bazei de date, precum și pentru gestionarea serverului MySQL local, la care aplicația se conectează.

### 3.3 Conexiunea și gestionarea bazei de date

Pentru conexiunea și gestionarea bazei de date am folosit Java Database Connectivity (JDBC). Următoare funcție inserează în baza de date informația reprezentativă a unui obiect Film folosind un SQL Query:

```
public void addFilm(String title, int year, int rating) { 1 usage
    /* Insert into the database. */
    String sqlQuery = "insert into films (title, release_year, rating) values(?, ?, ?)";
    jdbcTemplate.update(sqlQuery, title, year, rating);
}
```

Asemănător au fost realizate și funcționalitățile de selectare, actualizare și ștergere din baza de date a obiectelor Film folosind simple SQL Query.

### 3.4 OMDb API

[OMDb API](#) este un API-ul public folosit pentru a efectua cereri și obține JSON-uri reprezentative pentru filmele dorite. Aceste JSON-uri sunt prelucrate folosind Jackson pentru a construi obiectele Film. În urmă înregistrări am obținut o cheia pe care o folosesc să fac cereri sub următoarea formă:


- [www.omdbapi.com/?apikey=\[key\]&t=\[title\]&y=\[year\]](http://www.omdbapi.com/?apikey=[key]&t=[title]&y=[year])

## 4. Funcționalitatea Aplicației


### 4.1 Pagina Principala

În cadrul paginii principale se poate vizualiza colecția de filme precum și gestiona. Sunt prezente funcționalitatea de a adăuga un film, sau de a vizualiza colecția de filme sortată în funcție de an de lansare / rating. Fiecare film din colecție conține opțiunea de modificare și ștergere.

[Add Film](#) [Sort by Year](#) [Sort by Rating](#)



**The Shawshank Redemption - 1994**  
**Plot:** A banker convicted of uxoricide forms a friendship over a quarter century with a hardened convict, while maintaining his innocence and trying to remain hopeful through simple compassion.  
**Genre:** Drama  
**Director:** Frank Darabont  
**Actors:** Tim Robbins, Morgan Freeman, Bob Gunton  
**Rating:** 10  
[Update](#) [Delete](#)



**The Matrix - 1999**  
**Plot:** When a beautiful stranger leads computer hacker Neo to a forbidding underworld, he discovers the shocking truth--the life he knows is the elaborate deception of an evil cyber-intelligence.  
**Genre:** Action, Sci-Fi  
**Director:** Lana Wachowski, Lilly Wachowski  
**Actors:** Keanu Reeves, Laurence Fishburne, Carrie-Anne Moss  
**Rating:** 10  
[Update](#) [Delete](#)

### 4.2 Adăugare unui film nou

Un film poate să fie adăugat din pagină principală. În pagina nouă, utilizatorul are un formular pe care îl completează cu datele filmului dorit. Datele sunt validate, un mesaj de eroare este altfel afișat și un nou obiect Film este introdus în baza de date și creat folosind datele obținute de la API. Filmul în formularul de adăugare completat, respectiv afișarea sa în pagina principală.

**Add Film**

[Add Film](#)



### **The Shawshank Redemption - 1994**

**Plot:** A banker convicted of uxoricide forms a friendship over a quarter century with a hardened convict, while maintaining his innocence and trying to remain hopeful through simple compassion.

**Genre:** Drama

**Director:** Frank Darabont

**Actors:** Tim Robbins, Morgan Freeman, Bob Gunton

**Rating:** 10

Update

Delete

## **4.3 Sortarea în funcție de an de lansare / rating**

Funcționalitatea de afișarea a colecției de filme sortată în funcție de an de lansare sau rating este prezentată în pagină principală sub formă a două butoane. La apelul acestora, Controller-ul efectuează cererea și reactualizează pagină principală.

## **5. Testarea Aplicației**

Pentru aplicația myFilms, am utilizat două tipuri principale de testare pentru a asigura calitatea și funcționalitatea acesteia:

- Testarea unitară cu JUnit - am folosit JUnit pentru a verifica corectitudinea anumitor aspecte individuale din aplicație (corectitudinea răspunsului JSON primit de la API, construcția obiectului Film folosind JSON-ul obținut). Astfel, am identificat și remediat eventualele erori în etapele incipiente ale dezvoltării.
- Testarea funcțională - am testat manual din punct de vedere funcțional aplicația pentru a evalua dacă aceasta respectă cerințele dorite. Am verificat interacțiunile utilizatorului cu aplicația (adăugarea, modificarea și ștergerea filmelor) și am simulat scenarii reale pentru a asigura că funcționalitatea este completă.

## **6. Îmbunătățiri**

Următoarele îmbunătățiri pot să fie aduse:

- Responsive Design – partea de front-end a fost proiectată pentru o rezoluție standard, fără a lua în considerare dispozitivele mobile, tablete.
- Optimizare în cod - pentru a reduce numărul de interogări SQL, cereri API care se efectuează pentru toate obiectele la orice modificare.
- Sistem de autentificare - astfel încât fiecare utilizator să aibă propria sa colecție de filme pe care o gestionează.
- Funcționalitate de căutare/filtrare mai avansată - astfel încât utilizatorul să poate găsi cu ușurință un film din listă.
- Redirecționare către pagină IMDb asociată fiecărui film.