

# Projet Internet Movie DataBase - Difficile

## Description

L'objectif de ce TP est de mettre en base de données des informations concernant le cinéma et de mettre au point une petite application permettant d'effectuer des recherches dans les données en fonction de demandes d'un utilisateur.

Ces données sont stockées dans le fichier **films.json** qui présente au format JSON des données sur des films et acteurs. A noter qu'il vous faudra d'abord analyser la structure du fichier afin de comprendre la structure des données.

## Objectifs

Le but est de développer 2 applications basées sur l'API JPA :

- La première pour mettre en base ces données sur le cinéma.
- La seconde pour naviguer dans les données grâce à des interactions avec l'utilisateur.

**Objectif n°1 :** Analyser le fichier afin d'en comprendre la structure

**Objectif n°2 :** Réaliser le diagramme de classes ainsi que le MPD. Vous réaliserez un document de conception contenant les 2 diagrammes. Ce document sera comité avec le projet dans un répertoire appelé « conception » et situé à la racine.

**Objectif n°3 :** Réaliser l'application permettant de parser le fichier JSON et de mettre en BDD les données (en utilisant JPA). Les informations de connexion à la base de données devront être stockées dans un fichier de configuration. Pour accéder à ses données, l'utilisation d'un singleton est recommandée.

**Objectif n°4 :** Réaliser des optimisations afin que l'insertion soit la plus rapide possible. Vous pourrez par exemple mettre en place des caches mémoire en utilisant là encore le design pattern singleton.

**Objectif n°5 :** Réaliser une seconde application dotée d'un menu et permettant à l'utilisateur de réaliser des recherches dans les données. Voici le menu proposé :

1. Affichage de la filmographie d'un acteur donné
2. Affichage du casting d'un film donné
3. Affichage des films sortis entre 2 années données
4. Affichage des films communs à 2 acteurs/actrices donnés.
5. Affichage des acteurs communs à 2 films donnés
6. Affichage des films sortis entre 2 années données et qui ont un acteur/actrice donné au casting
7. Fin de l'application

Les interactions avec l'utilisateur peuvent se faire avec le Scanner.

## Instructions techniques complémentaires

- Créez un nouveau projet **STS/GitHub/Maven** nommé **traitement-data**
- Créez un répertoire **conception** dans votre projet
  - Réalisez un petit document de conception, converti au format PDF, contenant le diagramme de classes ainsi que le MPD de votre projet
  - Ce document est comité au même titre que vos classes Java.
- Réalisez les développements.
- Mettez en place les classes techniques permettant de garantir une bonne réutilisation du code technique !
- Développez l'application de restitution des données.
- Bon courage !

## Annexes – exemples avec du code pour vous inspirer

### Exemple 1 : Pour récupérer la date de naissance de Chris Pratt

1. Faire un Inspect sur « Born » dans la page HTML

The screenshot shows the IMDb page for Chris Pratt (nm0695435). The browser's developer tools are open to the 'Elements' tab, highlighting the HTML structure of the 'Born' section. The HTML code shown is:

```
<tr>...</tr>
<tr>...</tr>
<tr>
  <td id="overview-top" class="name-overview-widget__section">
    <div id="resume-teaser">...</div>
    <div class="txt-block" id="name-bio-text">...</div>
    <div id="name-born-info" class="txt-block">
      <h4 class="inline">Born:</h4> == $0
      <time datetime="1979-6-21">...</time>
      " in "
      <a href="/search/name?birth_place=Virginia,%20Minnesota,%20USA&ref=nm_ov_bth_place">Virginia, Minnesota, USA</a>
    </div>
  </td>
</tr>
<tr>...</tr>
</tbody>
</table>
</div>
<script>...</script>
```

2. Code Java :

```
// Connexion au document
Document doc = Jsoup.connect("https://www.imdb.com/name/nm0695435/").get();

// Sélection de l'attribut datetime de la balise time située après l'élément
d'identifiant name-born-info
String birthDate = doc.select("#name-born-info > time").attr("datetime");
```

## Exemple 2 : Pour récupérer la liste des films dans lesquels il a joué

1. Faire un Inspect sur la page HTML pour trouver la balise contenant la liste des films

The screenshot shows the IMDb Filmography page for James Reece. The browser's developer tools are open, displaying the HTML structure. A yellow arrow points from the first step of the list to the `div.filmo-category-section` element in the DOM tree. The DOM tree shows a list of film entries under the `div.filmo-category-section` element, including "Cowboy Ninja Viking", "The Electric State", "Garfield", "Guardians of the Galaxy Vol. 3", "Untitled Super Mario Project", "The Guardians of the Galaxy: Holiday Special", and "La liste finale". The film list is rendered as a table with columns for the film title, role, and year.

Film	Rôle	Année
Cowboy Ninja Viking	Duncan	
The Electric State		
Garfield	Garfield (voice)	2023
Guardians of the Galaxy Vol. 3	Peter Quill / Star-Lord	2023
Untitled Super Mario Project	Mario (voice)	2023
The Guardians of the Galaxy: Holiday Special	Peter Quill / Star-Lord	2022
La liste finale (TV Series)	James Reece	2022

The DOM tree shows the following structure for the film list:

```
<div id="filmo-category-section" class="filmo-category-section">
  <div class="filmo-row odd" id="actor-tt1801045">
    <div class="year_column"> &nbsp; </div>
    <b>
      <a href="/title/tt1801045/?ref=nm_flmg_act_1">Cowboy Ninja Viking</a>
    </b>
    " ("
    <a href="https://pro.imdb.com/title/tt1801045?rf=cons_nm_filmo&ref=cons_nm_filmo" class="in_production">announced</a>
  ) "
  <br>
  " Duncan "
</div>
```

2. Analyser la structure HTML d'une div contenant les données d'un film pour voir comment vous allez pouvoir en extraire les données. A ce stade il faut être astucieux...

```
<div class="filmo-row odd" id="actor-tt1801045">
  <span class="year_column"> &nbsp; </span>
  <b>
    <a href="/title/tt1801045/?ref=nm_flmg_act_1">Cowboy Ninja Viking</a>
  </b>
  " ("
  <a href="https://pro.imdb.com/title/tt1801045?rf=cons_nm_filmo&ref=cons_nm_filmo" class="in_production">announced</a>
) "
  <br>
  " Duncan "
</div>
```

### 3. Mettre en place le code Java

```
// Connexion au document
Document doc = Jsoup.connect("https://www.imdb.com/name/nm0695435/").get();

// Récupération de la première div avec pour classe filmo-category-section
Element filmSectionElt = doc.select(".filmo-category-section").first();

// Récupération de toutes les div de l'élément précédent ayant la classe
// filmo-row

Elements filmElts = filmSectionElt.select(".filmo-row");

for (Element elt: filmElts) {

    // Récupération de l'URL du film
    String movieUrl = elt.select("b > a").attr("href");

    // Récupération du nom du film
    String movieName = elt.select("b > a").text();

    // Select le premier élément br
    Element last = elt.select("br").first();

    // Select le nœud situé après br au format String
    String characterName = last.nextSibling().toString();

    System.out.println(movieUrl+" - "+movieName+" - "+ characterName);
}
```

