

1. Présentation de l'API OMdB (<https://www.omdbapi.com/>)

L'API OMDB permet aux développeurs d'accéder à une base de données de films et de séries télévisées. Pour avoir accès à cette base de données, les développeurs doivent s'inscrire et obtenir une clé qui sera utilisée dans les paramètres de l'URL lors des requêtes (le paramètre « apikey »). Une fois cette clé obtenue, il est possible de rechercher des informations sur des films ou des séries télévisées de trois (3) façons distinctes:

- 1) En effectuant une recherche par titre. Pour cela, il faut utiliser le paramètre « t » et spécifier le titre du film ou de la série que l'on cherche. Par exemple, pour faire une recherche sur le film « Pulp Fiction », on utilise l'URL suivante :

<http://www.omdbapi.com/?apikey=b310fc2a&t=pulp+fiction>

Voici le fichier JSON qui est retourné :

```
{
  Title: "Pulp Fiction",
  Year: "1994",
  Rated: "R",
  Released: "14 Oct 1994",
  Runtime: "154 min",
  Genre: "Crime, Drama",
  Director: "Quentin Tarantino",
  Writer: "Quentin Tarantino, Roger Avary",
  Actors: "John Travolta, Uma Thurman, Samuel L. Jackson",
  Plot: "The lives of two mob hitmen, a boxer, a gangster and his wife, and a pair of diner bandits intertwine in four tales of violence and redemption.",
  Language: "English, Spanish, French",
  Country: "United States",
  Awards: "Won 1 Oscar. 78 wins & 75 nominations total",
  Poster: "https://m.media-amazon.com/images/H/VV5BNQNMHMDI+ZTVuNTBjZi00MTI1LWU5JmZlY2IyZVJhZjE3VzI5MjIjXkE5YkFgcGdeOXVhZjZkZjZjOSNzIj00._V1_SX300.jpg",
  Ratings: [...],
  Metascore: "94",
  imdbRating: "8.9",
  imdbVotes: "2,852,525",
  imdbID: "tt0110912",
  Type: "movie",
  DVD: "28 Aug 2002",
  BoxOffice: "$107,928,762",
  Production: "N/A",
  Website: "N/A",
  Response: "True"
}
```

- 2) En effectuant une recherche par ID IMDB (il s'agit d'un identifiant unique attribué à chaque film et série télévisée). Pour cela, il faut utiliser le paramètre "i" et spécifier l'ID IMDB du film ou de la série que l'on cherche. Par exemple, pour rechercher le film « Reservoir Dogs » avec l'ID tt0105236, on utilisera l'URL suivante :

<http://www.omdbapi.com/?apikey=b310fc2a&i=tt0105236>

```

Title: "Reservoir Dogs",
Year: "1992",
Rated: "R",
Released: "02 Sep 1992",
Runtime: "99 min",
Genre: "Crime, Thriller",
Director: "Quentin Tarantino",
Writer: "Quentin Tarantino, Roger Avary",
Actors: "Harvey Keitel, Tim Roth, Michael Madsen",
Plot: "When a simple jewelry heist goes horribly wrong, the surviving criminals begin to suspect that one of them is a police informant.",
Language: "English",
Country: "United States",
Awards: "12 wins & 23 nominations",
Poster: https://m.media-amazon.com/images/H/H/5B/2E/xdmeEwNjIYmoZc0508YjA5JlTK2HjktZ7jEjZV2QxHjA1kIeYxKfGcGdeQXyVHTQxNzHzNDI%\_V1\_SX388.jpg,
Ratings: [-],
Metascore: "79",
imdbRating: "8.3",
imdbVotes: "1,019,301",
imdbID: "tt0105236",
Type: "movie",
DVD: "18 Mar 2003",
BoxOffice: "$2,832,029",
Production: "N/A",
Website: "N/A",
Response: "True"
}

```

- [https://www.omdbapi.com/?apikey=b310fc2a&s="love"](https://www.omdbapi.com/?apikey=b310fc2a&s=)

```

* Search: [
  * { - },
  * { - },
  * { - },
  * { - },
  * { - },
  * {
    Title: "P.S. I Love You",
    Year: "2007",
    imdbID: "tt0431308",
    Type: "movie",
    Poster: https://m.media-amazon.com/images/H/MV5BNTg2MDg4HjISNV5BMl5BanBnXkFtZTcwMzQ0ZmUw@@.\_V1\_SX300.jpg
  },
  * {
    Title: "I Love You, Man",
    Year: "2009",
    imdbID: "tt1155056",
    Type: "movie",
    Poster: https://m.media-amazon.com/images/H/MV5BMjU4HjISNTFyYjV5BMl5BanBnXkFtZTcwMTQ1NTMzMg@@.\_V1\_SX300.jpg
  },
  * {
    Title: "Love & Other Drugs",
    Year: "2010",
    imdbID: "tt0758752",
    Type: "movie",
    Poster: https://m.media-amazon.com/images/H/MV5BMTgxOTczODEyYjV5BMl5BanBnXkFtZTcwMDc0MDY4Mw@@.\_V1\_SX300.jpg
  },
  * {
    Title: "Love, Death & Robots",
    Year: "2019-",
    imdbID: "tt9561862",
    Type: "series",
    Poster: https://m.media-amazon.com/images/H/MV5BYTNiYTNkNTAtYzE3ZS08ZDQ1LWUwZTY1ZjI1NzYzZWJjYjU4XkEyXkFqcGdeQXVyODk4OTc3MTU@.\_V1\_SX300.jpg
  },
  * {
    Title: "Punch-Drunk Love",
    Year: "2002",
    imdbID: "tt0272338",
    Type: "movie",
    Poster: https://m.media-amazon.com/images/H/MV5BMVE1OTY4HjgtYjcwNjE0YjV5BMl5BanBnXkFtZTcwMjQ0ZmUw@@.\_V1\_SX300.jpg
  }
],
totalResults: "29940",
Response: "True"
}

```

Dans ce type de fichier, le champ "totalResults" indique combien de résultats il y a en totalité et le champ « Response » indique si la recherche s'est effectuée avec succès. Dans l'exemple ci-haut, la liste envoyée ne contient que 10 films (la page 1 par défaut), mais il y aurait en fait 20940 films qui correspondent à la recherche.

Outre les paramètres « t » (titre), « i » (ID OMdb) et « s » (search), il existe d'autres paramètres qui peuvent être utilisés pour affiner les recherches :

By ID or Title

Parameter	Required	Valid Options	Default Value	Description
i	Optional*		<empty>	A valid IMDb ID (e.g. tt1285016)
t	Optional*		<empty>	Movie title to search for.
type	No	movie, series, episode	<empty>	Type of result to return.
y	No		<empty>	Year of release.
plot	No	short, full	short	Return short or full plot.
r	No	json, xml	json	The data type to return.
callback	No		<empty>	JSONP callback name.
v	No		1	API version (reserved for future use).

*Please note while both "i" and "t" are optional at least one argument is required.

By Search

Parameter	Required	Valid options	Default Value	Description
s	Yes		<empty>	Movie title to search for.
type	No	movie, series, episode	<empty>	Type of result to return.
y	No		<empty>	Year of release.
r	No	json, xml	json	The data type to return.
page <small>New!</small>	No	1-100	1	Page number to return.
callback	No		<empty>	JSONP callback name.
v	No		1	API version (reserved for future use).

Au niveau de la sécurité, nous déduisons que L'API ne contient pas de donnée sensible et qu'il n'y a donc pas de risque pour les données privées. De plus, même si l'utilisation de l'API ne requiert pas d'authentification (une simple clé suffit), il existe des limites de requêtes pour éviter les abus, les attaques DDOS et le « scraping ». Bref, même s'il faut rester vigilant et respecter les bonnes pratiques de sécurité en général, la sécurité n'est pas un enjeu très important pour cette API.

Au niveau de la documentation, les fonctionnalités de l'API sont bien décrites. Les informations sur les données qu'elle permet de récolter dans le cadre d'une recherche sur un film en particulier sont disponibles, mais elles ne sont pas présentées clairement. De plus, il n'y a pas d'exemple pour les recherches de chaîne de caractère ni d'informations précises concernant les données que ce type de recherche permet de récolter (il faut tester l'API pour le savoir)

2. Présentation du projet (<https://github.com/jsforgues/MovieWorld>)

MovieWorld est une application React qui utilise l'API OMdB. Elle permet de rechercher des films, des séries ou des épisodes d'une série en fonction de certains paramètres de recherche et elle permet d'obtenir plusieurs informations concernant ces œuvres (réalisateur, année, pays, etc.).

Une fois sur la page principale, l'utilisateur peut choisir d'effectuer deux types de recherche :

- 1) ou bien avec une chaîne de caractère dans la barre de recherche (une fois que les résultats sont retournés sur la page, l'utilisateur peut ensuite choisir de filtrer les résultats en fonction de leurs types : films, séries ou épisodes).
- 2) Ou bien avec une référence IMBD dans le TextBox IMBD. Dans ce cas, si la référence IMBD existe, un seul titre sera retourné.

Une fois la recherche complétée, les « cards » de chaque film sont retournées sur la page principale. L'utilisateur peut ensuite choisir d'obtenir plus de détail sur un film en particulier en cliquant sur le bouton « Details » de la carte, ce qui le mènera sur une page qui contient plusieurs informations sur le film.

Les différentes parties du code :

Le fichier app.js utilise le composant React Router qui définit les routes de l'application : il y a deux routes qui sont utilisées pour le moment :

- Route "/" qui affiche la page de recherche du film
- Route "/movie/:abc" qui affiche la page de description du film

Le fichier movieCard.js contient une fonction nommée MovieCards qui prend en paramètre un objet movie. La fonction affiche ensuite un composant React Card avec le titre, l'année, le type du film et une image du film. Il y a également un bouton qui permet aux utilisateurs de voir les détails du film. La fonction affiche aussi des étiquettes de couleurs différentes en fonction du type de film.

Le fichier MovieSearch.js comprend des fonctions pour effectuer des recherches par nom, par ID IMDB et par année. Les résultats de la recherche sont stockés dans le tableau de films et affichés à l'utilisateur. Il y a également une fonction pour rafraîchir l'écran et pour effectuer une recherche par défaut.

Ce fichier contient aussi la mise en page de la page moviesearch qui comprend les barres de recherche et le conteneur qui affichent les différentes « cards ».

Le fichier `Moviedescription.js` contient une fonction React qui affiche une description d'un film en particulier. Il utilise la fonction `useParams` pour récupérer le titre du film et l'utilise pour faire une requête à l'API. Il stocke ensuite les données dans une variable appelée `movieDes` qui est ensuite utilisée pour afficher les informations dans la page web.

Ce fichier contient aussi la mise en page de la page `moviedescription`. La section d'affichage des films comprend une image du film et des informations détaillées sur le film, telles que le titre, la note, la durée et le réalisateur. Il y a également un aperçu du film.

Les points à améliorer au niveau de l'application :

- 1) Refaire la structure de la mise en page afin de prendre en compte les différentes tailles de fenêtres et ainsi assurer un bon affichage des éléments, quelle que soit la dimension de la fenêtre. Pour le moment lorsqu'un utilisateur redimensionne la taille de sa fenêtre, les différentes parties de la mise en page se chevauchent.
- 2) Exploiter plus avantageusement les possibilités graphiques offertes par Semantic UI. Pour le moment, les propriétés graphiques des éléments sont actuellement définies à plusieurs endroits, en utilisant à la fois Semantic UI et les fichiers CSS conventionnels, ce qui complique les choses inutilement. Pour améliorer l'aspect global de l'application, il sera nécessaire de mieux comprendre la relation entre les éléments de Semantic UI et les fichiers CSS conventionnels et de modifier l'affichage en conformité avec les meilleures pratiques.
- 3) Avoir plus d'icônes intéressantes. Remplacer certains mots par des icônes (ex. : le nom des pays par des icônes de drapeaux, le nom des genres comme « comédie » ou « action » par des icônes correspondants, etc.) ou encore implémenter les fonctions de Semantic UI pour afficher les drapeaux de pays.
- 4) Implémenter de nouvelles fonctions de recherche. La sélection de films en fonction de l'année n'est pas encore implémentée. Ce sera la moins compliquée à implémenter, car l'API prend en charge les URL avec le paramètre de l'année (« y »).
- 5) Il serait aussi intéressant d'implémenter des méthodes de recherche en fonction de paramètres qui ne sont pas pris en charge par l'URL, par exemple une recherche en fonction des réalisateurs, des acteurs, des notes imbd, des pays ou de la décennie. Peut-être que dans ce cas-ci il serait aussi préférable de trouver un API plus performant. Il existe de nombreux API semblables à celui que nous avons choisi.
- 6) Implémenter un nouvel API qui pourrait venir compléter la description des films de notre API. Par exemple, un API qui retourne des photos d'acteur. Nous sommes à la recherche

d'un API comme celui-ci : <https://unsplash.com/developers> , mais qui pourrait aussi nous retourner des photos d'acteurs.

- 7) Implémenter dans le code deux routes définies dans la partie "BrowserRouter" : une route qui renvoie vers une page qui affiche qu'aucun élément n'a été trouvé quand la recherche ne retourne rien et une route qui affiche « page 404 » lorsque l'API n'est pas accessible.