

# Rapport

Dans ce projet nous avons réalisé une application qui communique avec une API. On a choisi the movieDB API, c'est une API simple et facile à utiliser et bien documentée. Nous allons récupérer des films depuis the movieDB via l'API et afficher les films dans notre site web. Nous avons utilisés les technologies suivantes: Docker, JavaScript, JQuery, CSS, HTML, Spring Framework. Pour rendre l'application plus interactive nous avons utilisé OWL CAROUSEL, pour faire défiler les films.

## Présentation de l'API the movieDB api

Une API (application programming interface ou « interface de programmation d'application ») est une interface logicielle qui permet de « connecter » un logiciel ou un service à un autre logiciel ou service afin d'échanger des données et des fonctionnalités.

Les API offrent de nombreuses possibilités, comme la portabilité des données, la mise en place de campagnes de courriels publicitaires, des programmes d'affiliation, l'intégration de fonctionnalités d'un site sur un autre ou l'open data. Elles peuvent être gratuites ou payantes.

Dans notre cas nous utilisons the movieDB API qui est gratuite et facile à utiliser, ce service API de the movieDB est destiné à ceux qui souhaitent utiliser les images et/ou les données de the movieDB, émissions de télévision ou acteurs dans leur application. Cet API nous permet de récupérer et utiliser les données et/ou images de the movieDB manière programmatique.

## Microservices

Notre application reçoit des requêtes depuis un navigateur, l'application traite la requête en interrogeant l'API, une fois la réponse reçue, l'application parse la réponse de manière lisible et compréhensible par n'importe quel utilisateur et renvoie la réponse.

## Controlleur

Notre application est basée principalement sur la classe `RestController`. Avec cette classe nous interrogeons l'API et récupérer les données et les stocker dans nos modèles. La récupération des données se fait avec la class `RestTemplate` fourni par Spring Framework, il faut noté que les données récupérées sont en format json de ce fait il faut les mappées pour cela nous nous servons d'une autre class `ResponseEntity` qui est générique de la framework spring. Et l'ensemble des données sont envoyées dans la page html de l'application.

## Modèles

Les modèles utilisés pour réaliser l'application sont les suivantes:

**Movie** Cet modèle nous permet de représenter les données d'un film telles que: le nom du film, le nombre de vote, la description et la date de sortie ...

**MovieGenre** Tous les films ne sont pas pareils de ce fait nous utilisons cette classe pour récupérer les films selon le genre par exemple: on peut récupérer via l'API les films comédies les plus populaires et récent. Cette classe est composée des attributs qui sont des listes de modèle Movie.

**DiscoverMovie** Une fois qu'on les films qu'on souhaite il faut stocker le nombre de films, le nombre page ainsi que la liste des films, c'est ainsi entre en jeux la classe **DiscoverMovie**. Cette dernière représente l'ensemble des données d'un genre de films les plus récents.

### **Installation**

Pour ce qui est l'installation il faut regarder le fichier `README.mddu` project.