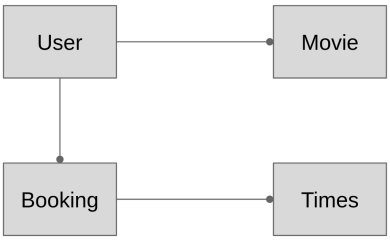
TP Flask, REST et OpenAPI

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

1. TP vert



1. Améliorez certains points d'entrée du service `Movie` pour proposer une API REST de niveau 3 et mettez à jour la spécification de votre API.
2. Complétez le microservice `Movie` avec les points d'entrée de votre choix et mettez à jour la spécification openAPI en conséquence.
3. Testez votre microservice avec Postman (https://www.postman.com/).
4. Ecrivez le microservice `Showtime` à partir de la spécification OpenAPI téléchargeable sur Moodle et testez votre service avec Postman.
5. Coder le service `Booking` à partir de la spécification OpenAPI téléchargeable sur Moodle et testez votre service avec Postman.
6. Construire la spécification openAPI du service `User` de façon à ce qu'il utilise les services `Booking` et `Movie`.
7. Ecrivez le microservice correspondant et testez votre service avec Postman.

IMPORTANT	Pour coder `Booking` vous aurez besoin d'appeler le service `Showtime`. Pour cela vous devez installer le paquet `requests` et utiliser la fonction `get`. Voir ici

2. TP bleu

Améliorer l'application en utilisant l'API REST de la base de données IMDB https://imdb-api.com/API

IMPORTANT	il faut vous créer un compte (gratuit) pour obtenir un token pour faire des requêtes

3. TP rouge

- Initiez vous à `Docker` et `DockerCompose`.

- Le but est de créer un conteneur par service et d'utiliser DockerCompose pour deployer le tout.

TIP	Créer un `DockerFile` dans le répertoire de chaque service et ensuite créer un fichier `docker-compose.yaml` à la racine de tous les services.