

# Rapport Projet Applications Web orientées Services

Binôme : Thibaud Moutsita, Aymeric Roinel

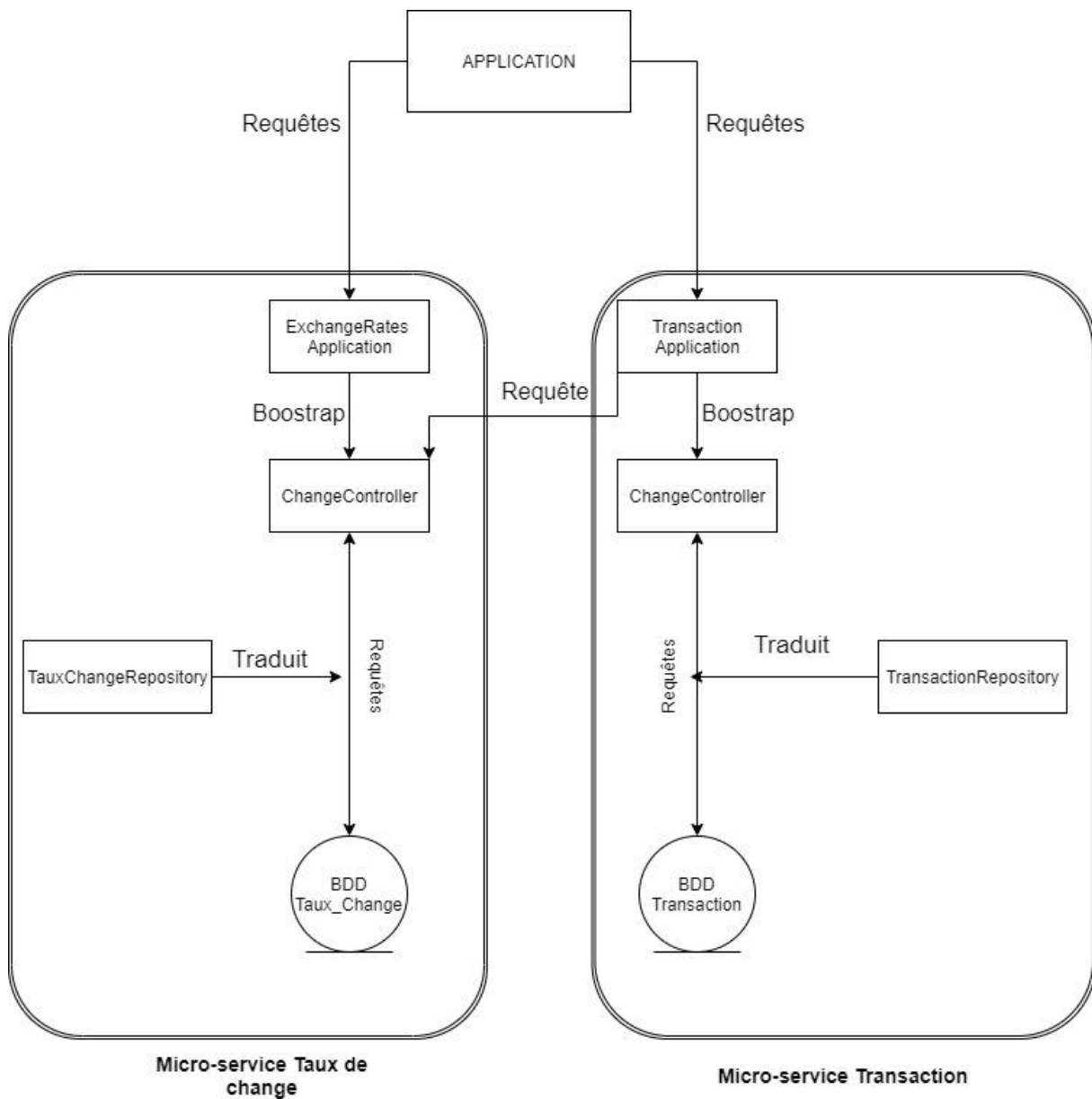
URL du Github : <https://github.com/Ayrox99/Projet-Applications-Web-orient-es-Services>

[Compiler/exécuter le projet.](#)

Voir README.md

[Documentation technique.](#)

Schéma d'architecture :



### Choix techniques :

Nous sommes restés sur une base de données H2, base sur laquelle nous préchargeons donc des données via des requêtes dans le fichier *data.sql*. Cela n'est pas idéal mais malheureusement nous n'avons pas réussi à faire fonctionner un autre type de base.

Concernant l'architecture de l'application, nous sommes restés sur une architecture classique, sans Docker ou MiniKube.

Pour les requêtes de mise à jour des bases de données, n'ayant pas trouvé la documentation pour mettre à jour directement dans la base de données, nous sommes passés par un chemin différent : nous récupérons l'entité à modifier (et donc elle est sauvegardée en tant qu'objet), puis nous la supprimons de la base, pour ensuite la modifier en tant qu'objet et la réinjecter.

Dernier point à souligner. Lorsque nous rentrons une date, par exemple « 2018-09-20 », cela est considéré comme « 2018-09-20 00:00:00 ». Par conséquent les dates sont affichés (problème purement d'affichage, tout fonctionne correctement en BDD) avec un jour en moins (pour notre exemple cela donnerait « 2018-09-19 »). Pour pallier à cela nos fonctions incrémentent d'un jour avant les affichages, pour un confort de compréhension.

### Bilan du projet :

Ce projet a été très enrichissant à de nombreuses échelles :

- En premier lieu évidemment d'un point de vue informatique et Web Service nous avons beaucoup appris, avec toutes les documentations liées aux méthodes, aux classes, à l'architecture de l'application. D'un point de vue résolution de bugs aussi, car les erreurs que nous avons pu avoir étaient toujours très difficiles à régler (il y en avait peu mais elles étaient toujours très difficiles), très spécifiques et il était difficile de trouver de l'aide sur internet.
- Du côté humain aussi, car nous travaillions pratiquement tout le temps ensemble (pour augmenter notre efficacité à lire et comprendre les documentations, pour se former plus rapidement). En particulier après une année sans forcément beaucoup avoir la possibilité de faire connaissance avec les différentes personnes de la promotion.

Au niveau des difficultés rencontrées, nous avons eu de gros problèmes lorsque nous avons tenté d'utiliser autre chose qu'une base H2 (postGres and MySQL). C'est bien dommage car en dehors du *pom.xml* et du fichier *application.properties*, il n'y a normalement rien à changer, JPA faisant office de traducteur universel avec les bases de données. Comme dit plus haut, 90% des erreurs que nous avons dû corriger était très difficiles de par leur spécificité.

Au niveau des satisfactions :

- Une fois la documentation sur les URI comprise et les bugs éliminés, nous avons pu avancer très vite et proposer une multitude de filtres/requêtes.
- Nous trouvons notre front-end avec l'interface graphique très réussi (en toute modestie !).