

# Document d'Architecture

Projet\_JEE  
L3 DIFS

Arthur Malié

## Infrastructure d'exécution

Versions utilisées :

- IDE utilisé : IntelliJ IDEA 2021.1.1 (Ultimate Edition)
- Serveur embarqué dans l'application : Apache Tomcat/9.0.45
- Java 11.0.8 (adopt-openj9-11)

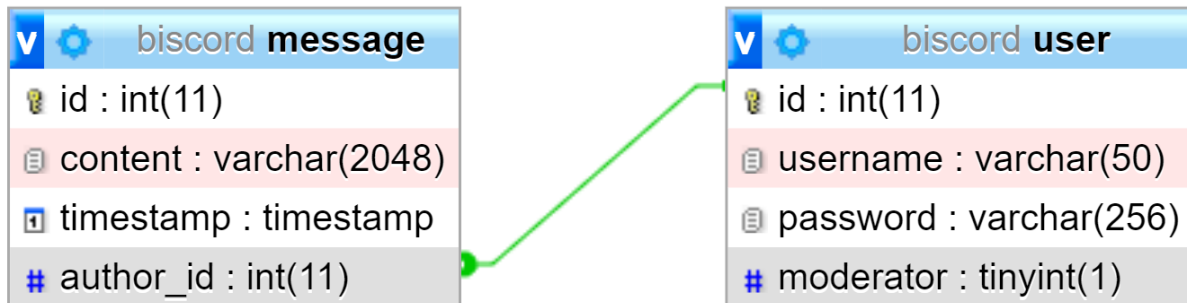
## Structure de persistance

Le SGBD à utiliser est un serveur MariaDB-10.4.18

La base de données comprend deux tables : Message et User.

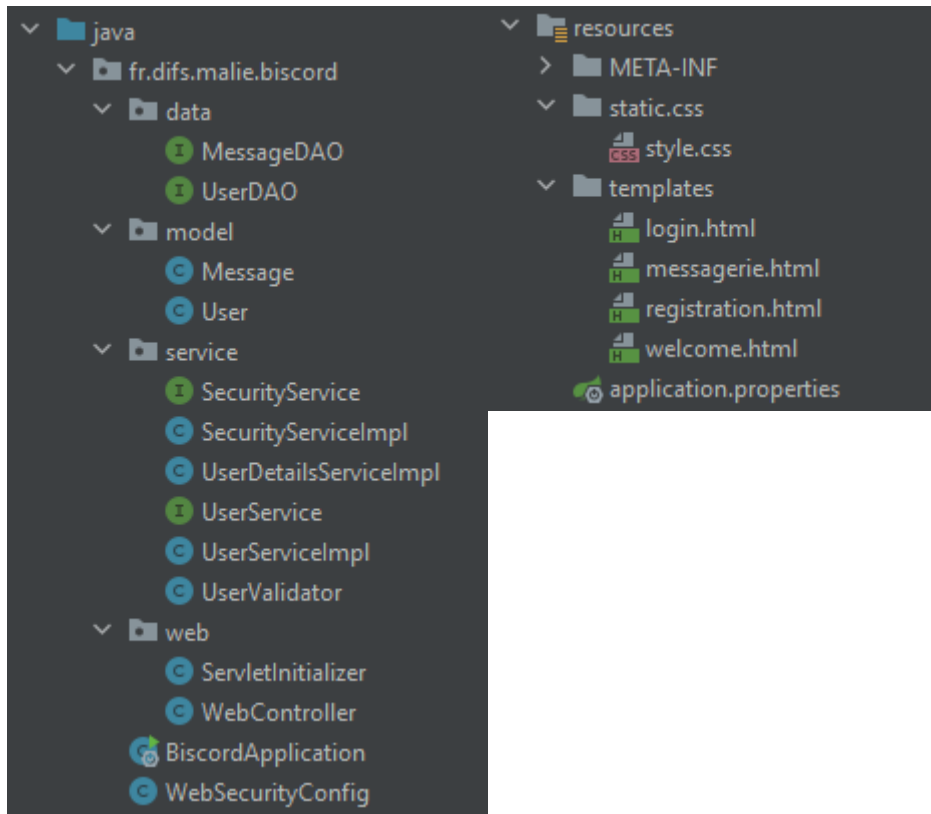
Un Message possède un identifiant, un contenu, une date/heure, et l'identifiant de l'auteur du message.

Un User possède un identifiant, un nom d'utilisateur, un mot de passe (chiffré par l'application) et un booléen indiquant s'il est modérateur.



# Architecture logicielle

Arborescence du projet :



Pour le développement de cette application, j'ai utilisé Maven et le framework Java Spring, ainsi que les dépendances suivantes :

- Spring data JPA : facilite grandement l'utilisation de JPA via des requêtes CRUD auto-générées, l'exécution dynamique des requêtes, la configuration de JpaRepositories, annotations, et bien plus encore.
- Thymeleaf : c'est un moteur de template, permettant la génération de vues pour les applications web basées sur le modèle MVC. Il fournit une intégration complète de Spring.
- Spring security : c'est un puissant framework d'authentification et de contrôle d'accès / autorisation. Il est le standard pour sécuriser les applications basées sur Spring. Il s'intègre également facilement à Spring data.

## Modèle

Le modèle représente l'état de l'application, c'est-à-dire les données .

Les classes constituant le modèle sont celles du package Model : Message et User. Ces classes définissent les entités de la base de données que l'application aura à traiter.

Ainsi que les interfaces du package data : MessageDAO et UserDAO qui héritent de JpaRepository, ce qui nous permet de collectionner les messages et users et d'exécuter des requêtes CRUD sur ces collections.

## Vue

La vue présente l'interface utilisateur.

Les fichiers constituant la vue sont situés dans le dossier ressources/templates. Ce ne sont pas des JSP, mais des fichiers HTML, dû à l'utilisation de Thymeleaf.

- welcome.html : page d'accueil
- login.html : page du formulaire de connexion
- registration.html : page du formulaire de création d'un nouveau compte utilisateur
- messagerie.html : la messagerie de Biscord. C'est là que sont affichés l'ensemble des messages, et le formulaire d'envoi d'un nouveau message.

## Contrôleur

La couche contrôleur comprend donc toutes les classes restantes. Il sert d'interface entre la Vue et le Modèle.

Le package web contient la classe WebController qui joue le rôle principal de contrôleur. Il intercepte les requêtes, les traite, et les envoie à la couche modèle appropriée pour le traitement des données. Une fois que le modèle renvoie les données au contrôleur, le contrôleur peut les envoyer à la vue correspondante.

Le package security contient toutes les classes concernant la sécurité du site web (authentification, autorisation, connexion, validation, inscription, ...) à l'exception de la classe WebSecurityConfig. Cette classe permet la configuration de la sécurité du site web, notamment en définissant les autorisations d'accès pour chaque page du site.