

# RESSOURCES RELATIONNELLES

DOCUMENTATION TECHNIQUE - V1.0



Agence ROSE

## Table des matières

OBJECTIFS ET DESCRIPTION GENERALE DU PROJET .....	4
Description et périmètre du projet .....	4
Equipe .....	4
Pilotage .....	5
Suivi .....	5
DOCUMENTS DE REFERENCE .....	5
TERMINOLOGIE .....	5
CONCEPTION TECHNIQUE .....	5
Schéma d'architecture applicative .....	5
BASE DE DONNEES .....	6
Modèle logique de données .....	6
Choix des technologies .....	7
BACKEND DES APPLICATIFS .....	8
Architecture.....	8
Choix des technologies .....	9
API et token.....	9
Tests.....	10
Prérequis pour l'installation en local .....	10
FRONTEND DES APPLICATIFS .....	10
Architecture.....	10
Choix des technologies .....	11
RGAA.....	12
Prérequis pour l'installation en local .....	12
ANNEXES .....	13

<b>Cahiers de tests.....</b>	<b>13</b>
<b>Echantillon de recette de jeux de tests.....</b>	<b>13</b>

## Objectifs et description générale du projet

### Description et périmètre du projet

Dans le cadre de la mise en œuvre de la politique du Gouvernement, le ministère des Solidarités et de la Santé veut mettre en place une plateforme de partage de ressources entre citoyens.

Selon une enquête de la Fondation de France, *"en 2020 plus de 7 millions de Français se trouvent en situation d'isolement"*, un chiffre qui a progressé de 3 millions en 10 ans. L'épidémie mondiale du covid n'ayant pas arrangé cette situation, l'enjeu d'une telle solution repose sur le besoin de cohésion sociale dont chacun a besoin.

Cette plateforme vise également à prévenir la précarité de l'information sur les domaines du social, de la santé publique et de d'organisation du système de santé.

Les applicatifs développés s'adressent à tous les citoyens, sans distinction. Ceux-ci devront alors être adaptés à tous en mettant l'accent sur leur ergonomie et simplicité d'utilisation. Le projet présentera une application mobile et une version site web desktop.

### Equipe

L'agence ROSE est une agence de communication digitale, créée en 2022, se situant au 264 Boulevard Godard, 33300 Bordeaux

Président directeur général : Jean- Michel Pédégé.

Tél : 06.12.34.56.78

Contact : [contact@rose.fr](mailto:contact@rose.fr)

Numéro immatriculation : W123456789

Rôle	Parti
Pilotage, product owner, chef de projet	Simon Vallero
Développement des applicatifs backend	Emma Scheuber, Olivia Lahary
Développement des applicatifs frontend	Rachel Tcha, Olivia Lahary

## Pilotage

### Suivi

Avancées projet : <https://trello.com/ressources-relationnelles-partie-2>

### Documents de référence

Cahier des charges fonctionnel : [Lien vers le CDC](#)

## Terminologie

API : Application Programming Interface (Interface de Programmation d'Application)

BDD : Base De Données

IHM : Interface Homme Machine

JWT : Json Web Token

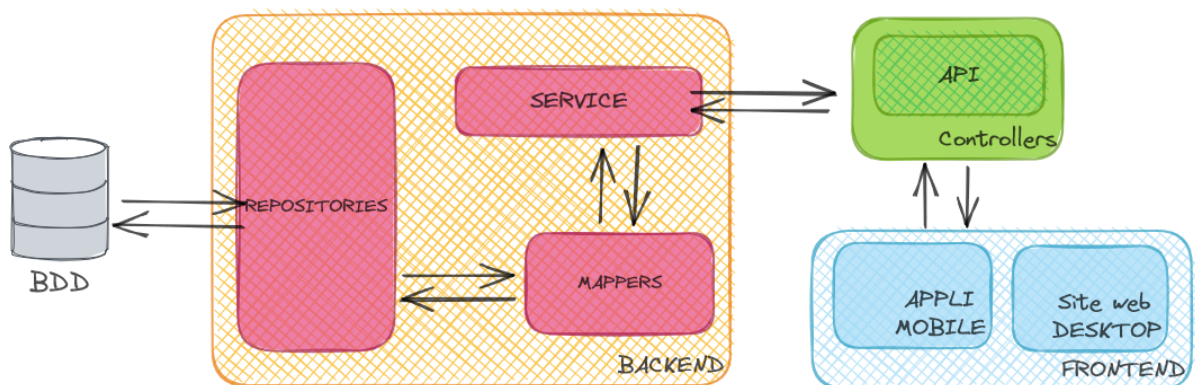
REST : Representational State Transfer

TU : Test Unitaire

URL : Uniform Resource Locator (Localisateur Uniforme de Ressource)

## Conception technique

### Schéma d'architecture applicative



L'architecture de l'application se décompose en quatre éléments :

- La base de données

- Le backend
- L'API
- Le frontend

Chaque élément est une porte d'entrée vers un autre, ce qui fait qu'ils sont isolés. Par exemple le frontend ne peut pas communiquer avec le backend directement. Ce fonctionnement nous permet d'apporter plus de sécurité à l'application.

Ressources Relationnelles est une application qui expose ses données sous forme d'API. C'est-à-dire que le backend définit des points d'entrée qui sont accessibles par le frontend via des URLs.

Si le frontend veut afficher un écran, il doit communiquer à l'API sa demande en appelant l'URL correspondante aux données qu'il souhaite montrer. L'API va pouvoir authentifier la demande qu'elle reçoit grâce à un système de token. Cela protège donc des appels qui ne proviennent pas des IHM de Ressources Relationnelles.

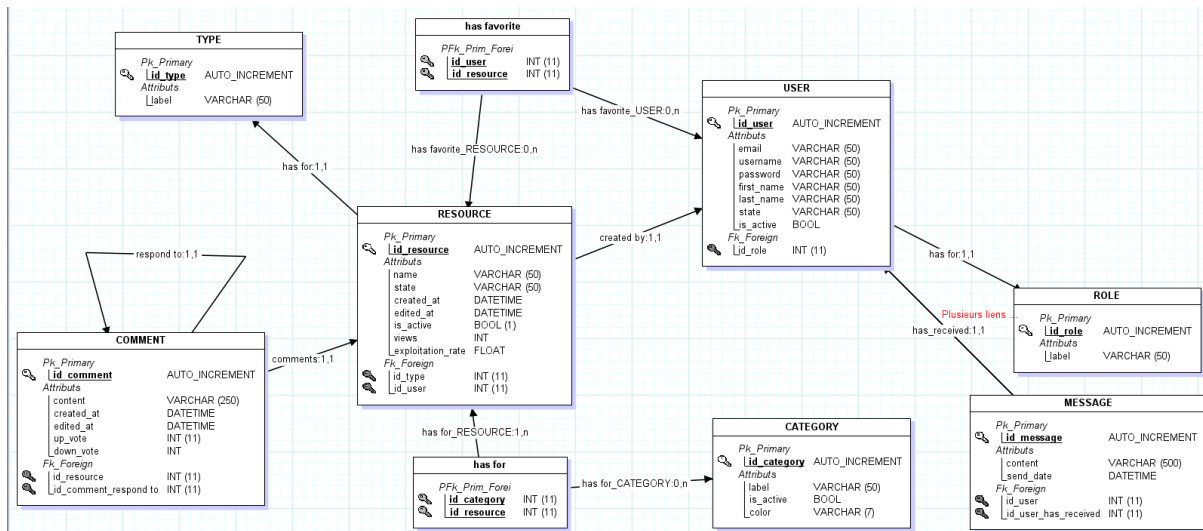
Une fois ce contrôle passé, l'API va donc transmettre la demande au backend en appelant les méthodes concernées.

La base de données n'est accessible que par les repositories situés dans le backend. Ce qui réduit significativement sa vulnérabilité.

## Base de données

La base de données est une base MariaDB version 10. Ce SGBD est open source donc gratuit et possède un support communautaire poussé. Il répond aussi aux attentes de sécurité du client.

## Modèle logique de données



La table resource contient deux clés étrangères :

- id\_user : le user qui a ajouté la ressource
- id\_type : le type de la ressource (format de document)

La table user contient une clé étrangère :

- id\_role : le rôle du user (admin, citoyen, ...)

La table comment contient deux clés étrangères :

- id\_user : le user qui a créé le commentaire
- id\_comment\_responds\_to : si le commentaire est une réponse à un autre alors cet id sera celui du commentaire au-dessus (possiblement null).

Deux tables de jointure existent :

- has\_favorite : elle contient les id de user et de resource afin de lister les ressources favorites d'un user.
- has\_for : elle contient des id de resource et de category afin de lister les ressources appartenant à une catégorie.

La table message contient deux clés étrangères :

Cette table sera utilisée quand l'évolution des messages entre utilisateurs sera implémentée.

- id\_user : le user qui envoie le message
- id\_user\_has\_received : le user qui reçoit le message

Choix des technologies

La base de données est une base MariaDB version 10.

Ce SGBD est open source donc gratuit et possède un support communautaire poussé. Il répond également aux attentes de sécurité du client.

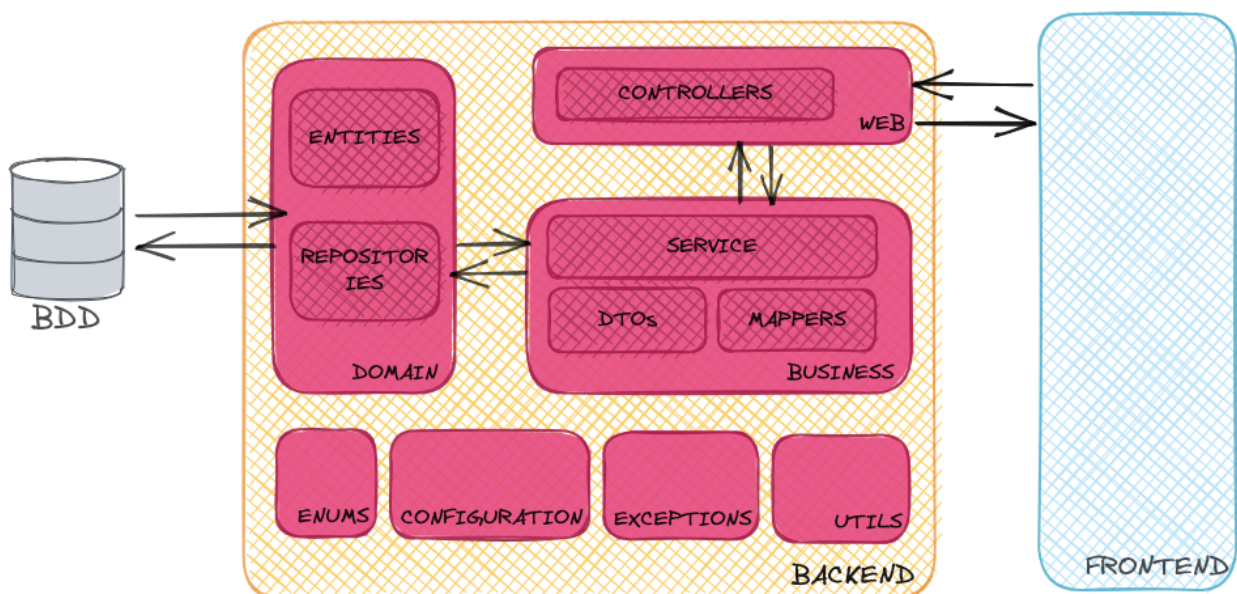
## Backend des applicatifs

### Architecture

Dans sa demande, le ministère préconise une architecture MVC. Celle-ci serait simple à mettre en place, par contre elle serait difficile à adapter aux évolutions futures. Plus le projet évoluera, plus il deviendra difficile à maintenir. Un autre inconvénient est qu'elle oblige à manipuler un grand nombre de fichier.

Pour pallier ces problèmes nous avons mis en place, avec l'aval du ministère, une architecture en couche, qui sera gérée selon le modèle du framework Spring Boot. Celle-ci a plusieurs avantages :

- La séparation de la logique métier des autres parties. Ce qui permet de la rendre indépendante en réduisant les impacts des changements des autres parties du backend.
- Le respect des principes d'une logique orienté objet, ce qui améliore la maintenabilité et la stabilité du projet.



Ainsi, selon l'architecture mise en place, le backend se compose de plusieurs couches qui ne peuvent communiquer qu'avec une seule autre dans un sens et



son retour. C'est-à-dire que la couche web, après avoir reçu la demande du frontend ne peut communiquer qu'avec la couche business, qui elle-même transmet la demande à la couche domain. Puis le chemin inverse afin de transmettre la réponse au frontend.

Dans la couche domain, les repositories sont les seuls éléments de l'application à pouvoir contacter la base de données grâce à un ORM.

La couche business contient le métier de l'application, c'est-à-dire le traitement des données. Les services se composent des méthodes et les mappers traduisent les entités et les DTOs.

Les points d'entrées du backend sont exposés dans la couche web. Ce sont dans les controllers qu'ils sont définis.

Dans la couche business, nous pouvons appeler d'autres couches internes au backend :

- Utils : elle contient les méthodes utilisées dans plusieurs services. Par exemple la vérification du format d'une adresse mail.
- Exception : permet de gérer les exceptions renvoyées.
- Configuration : toute configuration de l'application. Comme l'authentification ou le swagger.
- Enums : permet de définir les enums.

### Choix des technologies

Le backend est codé en Java 17 avec apache-maven 3.8.7 et le framework Spring Boot 3. Cette configuration nous permet une gestion des dépendances simplifiée mais aussi de nous concentrer sur le développement du métier sans avoir trop de configurations à gérer. De plus nous pourrions générer le projet en .WAR ou .JAR

Pour communiquer avec la base de données, le backend utilise l'ORM Hibernate qui est adapté à Java. C'est une solution multiplateforme, open-source qui nous offre un support communautaire important.

### API et token

Un token (JWT) est une chaîne de caractères encryptés qui contient des informations comme l'identité d'un user authentifié, ses autorisations, ...

À l'authentification d'un utilisateur sur Ressources Relationnelles, un token est généré puis attribué à l'utilisateur. A chaque requête qui sera envoyée à l'API, le

serveur décodera le token de l'utilisateur afin de confirmer qu'il a bien accès aux informations qu'il demande. Cela permet, par exemple, de faire connaître au backend quel est le rôle de l'utilisateur qui requête sans avoir à contacter la base de données.

Un token est composé de trois éléments :

- Le header : contient le hash qui est utilisé pour crypter le token
- Le payload : contient des attributs cryptés du token
- La signature : contient le header et le payload concaténés avec un secret

Pour utiliser un JWT, nous avons implémenté la dépendance spring-boot-starter-security de Spring Boot qui respecte les normes REST.

Cela est important car ces normes sont appliquées à la conception de l'API. Par exemple, les communications au sein de l'application sont stateless, c'est-à-dire que les informations côté client ne sont jamais stockées. Les requêtes sont traitées séparément, indépendamment.

## Tests

Des tests unitaires sont écrits dans le répertoire de tests. Pour les développer nous utilisons JUnit, qui est un framework de tests unitaires spécialement conçu pour Java.

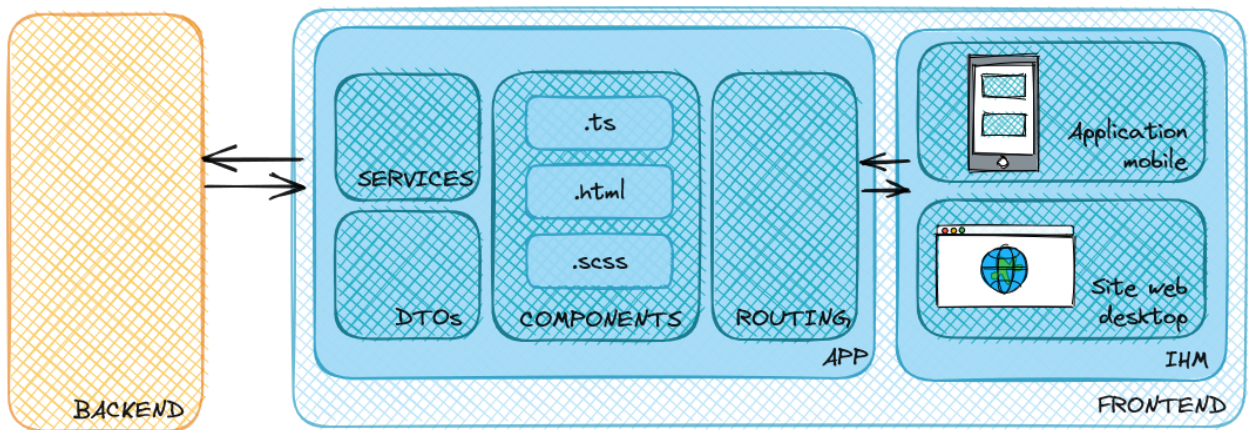
Les TU sont réalisés pour chaque fonctionnalité développée. En mettant l'accent sur les tests aussi souvent que possible nous rendons l'application plus stable performante.

## Prérequis pour l'installation en local

- apache-maven-3.8.7 [maven.apache.org/download.cgi](https://maven.apache.org/download.cgi)
- JDK17 [oracle.com/java/technologies/javase/jdk17-archive-downloads.html](https://oracle.com/java/technologies/javase/jdk17-archive-downloads.html)
- lombok.jar [projectlombok.org/download](https://projectlombok.org/download)

## Frontend des applicatifs

## Architecture



Le framework Angular nous permet de développer les deux applicatifs en même temps. Son cœur est appelé ici « App ». Celui-ci est composé de composants qui sont des éléments qui composent les écrans des IHM. Chaque composant est géré par trois fichiers, le TypeScript qui équivaut au controller du backend, le html qui est la vue et le scss qui est la feuille de style. Les composants peuvent même se combiner entre eux, ce qui nous permet une certaine souplesse dans les développements.

Les Services sont les fichiers qui contiennent la logique métier des composants. Les DTOs, sont les objets que nous manipulons dans le code.

Enfin le routing est l'élément qui nous permet d'attribuer nos URLs au front (celles qui sont visibles par l'utilisateur).

### Choix des technologies

Pour le frontend mobile et le site web nous utilisons le Framework Angular en version 15. Ce framework nous permet de travailler avec les langages TypeScript, HTML et CSS.

Il utilise une approche basée sur les composants pour la construction d'interfaces utilisateur. Ce qui rend l'application ouverte aux évolutions et facilite sa maintenance.

Angular est un choix qui est très compatible avec le langage orienté objet, aujourd'hui beaucoup d'applications sont conçues avec le couple Java/Angular.

De plus, Angular bénéficie de mises à jour régulières et d'un support grandissant grâce à une grande communauté. Ce choix nous permet donc de concevoir une application plus robuste sur le long terme.

Pour l'application mobile nous allons compléter Angular par un autre framework : Ionic. Il est open source qui permet de créer des applications mobiles hybrides en utilisant Angular et les mêmes langages. Ce framework offre un ensemble de composants d'interface utilisateur et permet aux applications de s'exécuter autant sur iOS que sur Android.

Son intégration est facile avec Angular, c'est un Framework qui permet de tirer parti des fonctionnalités avancées d'Angular, tels que la gestion de l'état pour le développement des applications mobiles.

## RGAA

Le respect des RGAA (Référentiel général d'Accessibilité pour les Administrations) est un document officiel pour permettre une plus grande accessibilité pour les personnes en situation de handicap.

Pour respecter ce référentiel il a été établis l'utilisation de balises HTML appropriées pour notre code, ainsi que le renseignement des alternatives textuelles pour les images et éléments graphiques et non textuels.

Le visuel fourni par le ministère de travail a été légèrement changé pour coller au mieux à ce référentiel, les couleurs ont été retravaillées pour rester dans l'idée du thème mais les contrastes ont été un peu plus marquée pour fournir une lisibilité optimale.

Nous avons opté pour une structure de menu cohérente pour faciliter la navigation entre les différentes pages du site.

Nous prévoyons une veille technologique pour surveiller les évolutions des RGAA afin d'assurer le respect des normes les plus récentes en matière d'accessibilité numérique.

## Prérequis pour l'installation en local

Node.js [nodejs.org/en](https://nodejs.org/en)

AngularCLI (dans un terminal) : `npm install -g @angular/cli` Suivre les instructions et accepter le routing + installer avec scss.

## Annexes

### Cahiers de tests

#### Plan de test

Création d'un compte	<input type="checkbox"/>
Ajout de commentaire et réponse	<input type="checkbox"/>
Ajout d'une ressource	<input type="checkbox"/>
Édition d'une ressource	<input type="checkbox"/>
Modification des informations personnelles de l'utilisateur	<input type="checkbox"/>
Ajout de catégories	<input type="checkbox"/>
Édition de catégories	<input type="checkbox"/>
Suppression de catégories	<input type="checkbox"/>
Lister les ressources	<input type="checkbox"/>
Supprimer/Archiver une ressource	<input type="checkbox"/>
Activer une ressource	<input type="checkbox"/>
Désactiver une ressource	<input type="checkbox"/>
Ajouter une catégorie	<input type="checkbox"/>
Éditer une catégorie	<input type="checkbox"/>
Supprimer/archiver une catégorie	<input type="checkbox"/>
Lister les catégories	<input type="checkbox"/>
Recherche de ressource	<input type="checkbox"/>
Filtrer les ressources par type	<input type="checkbox"/>
Système de vote (like)	<input type="checkbox"/>
Système de vote (dislike)	<input type="checkbox"/>
Attribution de rôles	<input type="checkbox"/>
Voir les stats	<input type="checkbox"/>

Tâches réalisées	0
NB de tâches	22
% de tâche tester	0%



#### Echantillon de recette de jeux de tests

Titre du test : Inscription d'un nouvel utilisateur :

Objectif du test : Vérification qu'un nouvel utilisateur puisse s'inscrire avec succès sur l'application

Prérequis : Application web en ligne et accessible

L'utilisateur est sur la page d'inscription de l'application.

Scénario du test :

Etapes :

1. L'utilisateur remplit le formulaire le formulaire d'inscription avec les informations requises :
  - Prénom : John
  - Nom : Doe
  - Adresse mail : [john.doe@exemple.fr](mailto:john.doe@exemple.fr)
  - Mot de passe : \*\*\*\*\*
  - Confirmation mot de passe : \*\*\*\*\*

## 2. L'utilisateur clique sur s'inscrire

Les résultats attendus :

Le système vérifie que tous les champs sont remplis et que les informations sont valides

L'application enregistre les informations utilisateur dans la base de données

L'application affiche un message de succès

L'utilisateur est redirigé vers la page de connexion de l'application.

Cas de tests alternatifs :

1. L'utilisateur remplit le formulaire avec une adresse mail déjà utilisée par un autre utilisateur.

Résultats attendus

Le système détecte que l'adresse email est déjà utilisée et affiche un message d'erreur indiquant la nature de cette erreur.

2. L'utilisateur remplit le formulaire d'inscription avec un mot de passe ne respectant pas les critères de sécurité

Résultats attendus

Le système détecte que le mot de passe ne respecte pas les critères de sécurité, et affiche un message d'erreur indiquant la nature de l'erreur

3. L'utilisateur laisse un des champs obligatoires vide dans le formulaire d'inscription

Résultats attendus

Le système détecte les champs obligatoires vides et affiche un message d'erreur indiquant la nature de l'erreur.