Projet Industriel
2015-2016




# RASE: Radio Sociale pour la détection d'épidémies

## DOSSIER TECHNIQUE

*Participants :*
**Gaetan Robert LESCOUFLAIR**
**Kathleen JEAN-BAPTISTE**

*Encadrant :*
**Amosse EDOUARD**

# Table des figures



# Table des tableaux

# Table des matières

# Historique des versions

**Tableau 1- Historique des versions**

| Version | Date | Modifications | Révisé par |
|---|---|---|---|
| **1.0** | 11/03/2016 | Première version du document | Gaëtan R. Lescouflair |
| **1.2** | 20/05/2016 | Amélioration du document | Gaëtan R. Lescouflair / Kathleen Jean-Baptiste |

# Table des abréviations

**Tableau 2-Acronymes**

| Acronyme | Définition |
|---|---|
| **RaSE** | Radio Sociale pour la détection d'épidémies |
| **MVC** | Modèle Vue Contrôleur |
| **MOA** | Maitre d'Ouvrage |
| | |

# Introduction

Ce document représente le dossier technique du projet dans le cadre de notre projet industriel RaSE. Cette application comprend deux versions. Une version mobile qui va fonctionner sur les appareils mobiles (Tablette etTéléphone Android) et une version web qui contient un front end et un back end pour l'administration du Système.

## But du document

Ce document a pour objectif de fournir les informations indispensables à la compréhension du fonctionnement et de l'utilisation de notre système.

## Références

- Fiche du projet
- Cahier des charges
- Dossier projet industriel

## Contenu du document

Ce document est destiné à présenter le projet RaSE, les fonctionnalités liées au projet, les étapes d'utilisation de la plateforme. On en distingue les points suivants :

- La description générale du produit.
- Les exigences fonctionnelles et matérielles du Système.
- La présentation des outils utilisés.
- Le processus de déploiement de l'application

# Description générale du produit

## Mission

L'objectif de ce projet c'est de réaliser une application permettant de détecter et de suivre l'évolution des cas d'épidémies afin de permettre aux organismes de santé de faire leur intervention à temps au niveau des zones touchées, aussi de sensibiliser la population sur une éventuelle d'épidémies

## Description du projet

Le projet RaSE consiste à l'utilisation des technologies disponibles, facilement utilisables et accessibles à une majorité de personnes et d'en concevoir un système de surveillance et de détection d'épidémies. La solution

qui va être réalisée est une plateforme sociale spatio-temporelle d'échange de données entre les professionnels de santé issues des reports des cas de maladies par rapport à des épidémies surveillées. Ce qui permet d'effectuer un suivi, de détecter une épidémie tout en pouvant les visualiser sur fond d'écran cartographique et d'obtenir des statistiques pour fournir une réponse adéquate à une situation qui se développe dans une zone bien déterminée.

Il comprend plusieurs modules :

1- Module d'administration
2- Module de gestion de report
3- Module d'analyse
4- Module d'alerte
5- Module de visualisation

# Explication globale du fonctionnement

Déjà mentionné dans la partie de l'introduction, le système fonctionne sur deux plateformes : Une plateforme web et une plateforme mobile. Pour utiliser la version mobile, nous devons avoir des appareils Android tandis que la version web requiert l'utilisation d'un navigateur pour y accéder.

Une connexion internet est obligatoire pour avoir accès aux différentes fonctionnalités du système.

## Partie Mobile

La version mobile est utilisée pour faire un report de cas de maladie avec un nombre limité de champs dans le but de faciliter la saisie pour les personnels de santé ou les personnes avisées et une visualisation des cas reportés sur un fond cartographique. La partie visualisation inclut aussi des filtres en fonction de plusieurs critères de recherche. (Ajouter ou modifier si il y a lieu).

Pour qu'un utilisateur puisse accéder à la fonctionnalité de report sur les appareils mobiles , il doit avoir un compte utilisateur qu'il peut créér lui-même ou peut être fourni par l'administrateur du système après que l'administrateur ait activé le compte de celui-ci. Pour chaque groupe d'individu destiné à faire un report de cas, un niveau de confiance lui est attribué afin de mieux contrôler le déclenchement d'une épidémie.

## Partie Web

En plus des fonctionnalités qui sont incluses dans la partie mobile, la version web contient bien d'autres encore. Comme les modules d'administration du système, de gestion d'une ontologie dont la finialité est d'obtenir un vocabulaire commun lors d'un report, de paramétrage, d'alerte,d'envoi de notifications pour les souscripteurs.

L'accès aux fonctionnalités de la partie web necessite un poste de travail muni d'un navigateur web et d'une connexion internet. Un compte utilisateur qui est activé par l'administrateur du sysstème est aussi obligatoire.

# Exigences fonctionnelles et matériels du Système

La mise en place d'un ensemble exigences matériels et logiciels est nécessaire pour le bon fonctionnement du système. Nous expliquerons par la suite, les caractéristiques relatives à la plateforme web necéssitant un serveur central et des postes clients, et la plateforme mobile pour les téléphones et/ou tablettes Android.

## Serveur Central

| Caracteristiques matérielles et logicielles | |
|---|---|
| **Processeur** | Intel core Quad CPU |
| **Mémoire Disque** | 200 Gb |
| **Mémoire RAM** | 8 Gb |
| **Système d'exploitation** | Windows Server 2008 |
| **Installations requises** | Netbeans 8.0.2 |

## Poste Client

Les postes clients ont besion d'un navigateur web. Le logiciel testé est fonctionnel sur Internet Explorer, Firefox et Google Chrome.

## Appareils Mobiles

| Caractéristiques du Mobile | |
|---|---|
| **Système d'exploitation** | Android 4.0 ou + |
| **Dimension** | 4' ou plus |
| **Mémoire Disque** | 16 Gb |
| **Mémoire RAM** | 1 Gb |

## Technologie

**-Web**

Java EE
HML5
CSS3
Javascript
AngularJS

**-Mobile**

Android

**-Dependances**

Google Map

Json : Sérialisation et de-sérialisation d'objets en JSON

## Environnement de développement

Netbeans 8.0.2 : IDE de développement du serveur et de la partie web

MongoDB : Base de données

Android Studio : IDE de développement de la parite mobile du système

## Limitation d'accès au système

Le logiciel sera implémenté dans des sites publics ou privés qui sont dans le domaine de la santé afin d'apporter une amélioration dans les services de surveillance d'épidémiologique. Chaque personne souhaitant effectuer une ou plusieurs tâches sur le système aura accès à un compte d'utilisateur activé par un administrateur du système.

# Présentation des outils utilisés

| Outils | Description |
|---|---|
| **MongoBD 2.4** | Est un système de gestion de base de données NoSQL open-source. |
| **NetBeans 8.1** | Est un IDE open-source permettant de faire des applications web en utilisant le langage java. |
| **Android Studio 2.1** | Est un IDE open-source qui permet de faire du développement mobile. |

# Processus de déploiement de l'application

## Référence des manuels d'installation

| Outils | Liens, Annexes ou Fichiers |
|---|---|
| | |
| | |
| | |
| | |

# Modèle de données et Dictionnaire de données

## Définition

Le dictionnaire de données est une façon pour décrire un ensemble de données géré dans une base de données à savoir le nom de la table, le nom et le type de données de chaque attribut de la table dont la finalité est de donner une meilleure compréhension et manipulation de la base par n'importe quelle personne souhaitant faire une intervention au niveau de la base.

## Description de l'approche

Les données sont stockées dans des collections dans une base de données MongoDB. Ce dictionnaire de données définit le nom de chaque collection ainsi que les attributs pour chacune des collections se trouvant dans la base de données.

## Modèle de données

### *Description du modèle de données*

| Type du modèle | Description | Information recueillies |
|---|---|---|
| **Maladie** | Représentation d'une maladie | Id, Nom, Description, Seuil, TypeSeuil, liste de Categorie, Liste de Symptomes |
| **Symptome** | Représentation d'un symptôme | Id, Nom et Description |
| **Categorie** | Représentation d'une catégorie de maladie | Id, Nom et Description |
| **CasMaladie** | Représentation d'un cas de maladie | Id, Maladie, Liste de Symptomes, Utilisateur, Mention, Sexe, Groupe d'age, Date, Note, Nombre de cas, Location, Zone |
| **MentionResponsabilite** | Représentation des diverses mention | Id, Nom et Description, Groupe de l'individu |
| **Departement** | Représentation d'un département | Id, Properties, Type, Geometry |
| **Commune** | Représentation d'une commune | Id, Properties, Type, Geometry |
| **Demographie** | Représentation d'une section communale | Id, Properties, Type, Geometry |
| **Adresse** | Représentation d'une adresse | Rue, Commune, Département |

| | | |
|---|---|---|
| **Zone** | Représentation d'une localité | Section communale, Département, Commune, Arrondissement |
| **Geometry** | | Coordonnées , Type |
| **ProprietesDepartement** | Représentation des informations | |
| **ProprietesCommune** | | |
| **ProprietesDocument** | | |
| **Personne** | Représentation des informations sur | Id, Prénom, Nom, Mail, Password, Role, Telephone, Adresse, Actif, IP |
| **Utilisateur** | Donne des informations supplémentaires pour un utilisateur. | Mention de responsabilité |
| **Administrateur** | Donne des informations supplémentaires pour un administrateur | Typeadmin |
| **GroupeIndividu** | Représente des informations sur un groupe d'individu | Id, Nom du Groupe, Description |
| **Role** | Représentation des rôles pour les utilisateurs et administrateurs | Id, Privilège |
| **Souscription** | Représente des informations sur les souscripteurs souhaitant recevoir des notifications | Id, Nom, Prénom, Email, Titre, Code, Actif |

**Tableau 3- Modèle de données**

## Présentation des champs du modèle

## Description détaillé du modèle et du dictionnaire de données

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **Administrateur** | utilisateur | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| typeAdmin | | **String** | **String** | |
| | | | | |
| | | | | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **Adresse** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| rue | | **String** | **String** | |
| commune | | **String** | **String** | |
| departement | | **String** | **String** | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **CasMaladie** | cas_maladie | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| id | | **String** | **String** | |
| maladie | | **Maladie** | **Object** | **Maladie** |
| listeSymptome | | **List<Symptome>** | **Array** | **Symptome** |
| user | | **Utilisateur** | **Object** | **Utilisateur** |

| **mention** | | **MentionResponsabilite** | **Object** | **MentionResponsabilite** |
|---|---|---|---|---|
| **sexe** | **{Femme,Homme, Mixe}** | **String** | **String** | |
| **groupeAge** | **{Moins_de_5 ,Plus_de_5 }** | **String** | **String** | |
| **date** | | **String** | **String** | |
| **location** | | **List<Double>** | **Array** | |
| **note** | | **int** | **integer** | |
| **nombreCas** | | **int** | **int** | |
| **zone** | | **Zone** | **Object** | **Zone** |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **Categorie** | categorie | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| id | | **String** | **String** | |
| nom | | **String** | **String** | |
| description | | **String** | **String** | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **Commune** | geo_commune | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| _id | | | | |
| properties | | **ProprietesCommune** | **Object** | **ProprietesCommune** |
| type | | | | |
| geometry | | **MultiGeometry** | **Object** | **MultiGeometry** |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **Demographie** | geo_section_demographie | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| _id | | | | |
| properties | | **ProprietesDemographie** | **Object** | **ProprietesDemographie** |
| type | | | | |
| geometry | | **Geometry** | **Object** | **Geometry** |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **Departement** | geo_departement | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| _id | | | | |
| properties | | **ProprietesDepartement** | **Object** | **ProprietesDepartement** |
| type | | | | |
| geometry | | **MultiGeometry** | **Object** | **MultiGeometry** |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **Geometry** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| type | | **String** | **String** | |
| coordinates | | **Double [ ] [ ] [ ]** | **Arrray** | |
| | | | | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **GroupeIndividu** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| _id | | **String** | **String** | |
| nomGroupe | | **String** | **String** | |
| description | | **String** | **String** | |
| listeMention | | **List<MentionResponsabilite>** | **Array** | **MentionResponsabilite** |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **HistoriqueDetection** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **Maladie** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **MentionResponsabilite** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **MultiGeometry** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **Parametre** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **Personne** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **ProprietesCommune** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **ProprietesDemographie** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **ProprietesDepartement** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **Role** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **Souscription** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| | | | | |

| | | | | |
|---|---|---|---|---|
| | | | | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **Symptome** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **Utilisateur** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| Nom Classe | Nom Collection | Description |
|---|---|---|
| **Zone** | | |

| Nom attribut | Description de l'attribut | Type attribut classe | Type attribut collection | Reference |
|---|---|---|---|---|
| | | | | |
| | | | | |

| | | | | |
|---|---|---|---|---|
| | | | | |

# Spécification du web service

## Web Service REST

Le web service fournit et reçoit des informations sur forme JSON. Le web service est réalisé avec les contrôleurs de Framework Sping.

## Les Méthodes http Utilisées

| Méthode | Opération effectuée sur le Serveur | Qualité |
|---|---|---|
| **GET** | Lecture de la ressource | |
| **POST** | Créer une nouvelle ressource | |
| **PUT** | Modifier une nouvelle ressource | |
| **DELETE** | Supprimer une ressource | |
| **HEAD** | Envoie ou Retourne l'entête de réponse HEADER et/ou pas de corps de réponse | |

## Description des ressources

Url de base : http://localhost :8080/api/v1/

| Ressource | Méthodes | URI | Description |
|---|---|---|---|
| **administrateur** | GET, POST, PUT, DELETE | Base/administrateur | |
| **categorie** | GET, POST, PUT, DELETE | Base/categorie | |
| **maladie** | GET, POST, PUT, DELETE | Base/maladie | |

| | | | |
|---|---|---|---|
| **symptome** | GET, POST, PUT, DELETE | Base/symptom | |
| **mention** | GET, POST, PUT, DELETE | Base/metion | |
| **casMaladie** | GET, POST, PUT, DELETE | Base/ casMaladie | |
| **utilisateur** | GET, POST, PUT, DELETE | Base/utilisateur | |
| **departement** | GET, POST, PUT, DELETE | Base/departement | |
| **commune** | GET, POST, PUT, DELETE | Base/commune | |
| **sectionCommunale** | GET, POST, PUT, DELETE | Base/sectionCommunale/ | |
| **parametre** | GET, POST, PUT, DELETE | Base/parametre | |
| **souscription** | GET, POST, PUT, DELETE | Base/souscription | |
| **zonelist** | GET | Base/zonelist | |

## Description détaillé des Ressources

### *Nom Ressource : administrateur*

Méthode : GET

| | |
|---|---|
| **URI** | **Base/administrateur/** |
| **Description** | |

Méthode : GET

| | |
|---|---|
| **URI** | **Base/administrateur/{id}** |
| **Description** | |

Méthode : GET

| | |
|---|---|
| **URI** | **Base/administrateur/{email}** |
| **Description** | |

Méthode : POST

| URI | Base/administrateur/ |
|---|---|
| Description | |

Méthode : PUT

| URI | Base/administrateur/{id} |
|---|---|
| Description | |

Méthode : DELETE

| URI | Base/administrateur/{id} |
|---|---|
| Description | |

### *Nom Ressource : casMaladie*

Méthode : GET

| URI | Base/casMaladie/ |
|---|---|
| Description | |

Méthode : GET

| URI | Base/casMaladie/{id} |
|---|---|
| Description | |

Méthode : GET

| URI | Base/casMaladie/{email} |
|---|---|
| Description | |

Méthode : POST

| URI | Base/casMaladie/ |
|---|---|

| Description | |
|---|---|

Méthode : PUT

| URI | Base/casMaladie/{id} |
|---|---|
| Description | |

Méthode : DELETE

| URI | Base/casMaladie/{id} |
|---|---|
| Description | |

### *Nom Ressource : categorie*

Méthode : GET

| URI | Base/categorie/ |
|---|---|
| Description | |

Méthode : GET

| URI | Base/categorie/{id} |
|---|---|
| Description | |

Méthode : POST

| URI | Base/ categorie / |
|---|---|
| Description | |

Méthode : PUT

| URI | Base/ categorie /{id} |
|---|---|
| Description | |

Méthode : DELETE

| URI | Base/categorie /{id} |
|---|---|
| Description | |

*Nom Ressource : departement*

Méthode : GET

| URI | Base/departement /{id} |
|---|---|
| Description | |

*Nom Ressource : commune*

Méthode : GET

| URI | Base/commune/{id} |
|---|---|
| Description | |

| Corps requête | |
|---|---|
| Corps réponse | |

*Nom Ressource : sectionCommunale*

Méthode : GET

| URI | Base/sectionCommunale/{id} |
|---|---|
| Description | |

*Nom Ressource : zonelist*

Méthode : GET

| URI | Base/zonelist/ |
|---|---|
| Description | |

*Nom Ressource : demographie*

Méthode : GET

| URI | Base/ demographie/{x}/{y} |
|---|---|
| Description | |

*Nom Ressource : groupeIndividu*

Méthode : GET

| URI | Base/ groupeIndividu/ |
|---|---|
| Description | |

Méthode : GET

| URI | Base/ groupeIndividu/{id} |
|---|---|
| Description | |
| Entête | |
| Corps requête | |
| Corps réponse | |

Méthode : POST

| URI | Base/ groupeIndividu / |
|---|---|
| Description | |
| Entête | |
| Corps requête | |
| Corps réponse | |

Méthode : PUT

| URI | Base/ groupeIndividu/{id} |
|---|---|
| Description | |
| Entête | |
| Corps requête | |
| Corps réponse | |

Méthode : DELETE

| URI | Base/ groupeIndividu /{id} |
|---|---|
| Description | |
| Entête | |
| Corps requête | |
| Corps réponse | |

### *Nom Ressource : maladie*

Méthode : GET

| URI | Base/maladie/ |
|---|---|
| Description | |

Méthode : GET

| URI | Base/ maladie/{id} |
|---|---|
| Description | |

Méthode : GET

| URI | Base/ maladie/{email} |
|---|---|
| Description | |

Méthode : POST

| URI | Base/ maladie / |
|---|---|
| Description | |

Méthode : PUT

| URI | Base/maladie/{id} |
|---|---|
| Description | |

Méthode : DELETE

| URI | Base/maladie/{id} |
|---|---|
| Description | |

### *Nom Ressource : mention*

Méthode : GET

| URI | Base/mention/ |
|---|---|
| Description | |

Méthode : GET

| URI | Base/mention/{id} |
|---|---|
| Description | |

Méthode : GET

| URI | Base/mention/{email} |
|---|---|
| Description | |

Méthode : POST

| URI | Base/ mention / |
|---|---|
| Description | |

Méthode : PUT

| URI | Base/mention/{id} |
|---|---|
| Description | |

Méthode : DELETE

| URI | Base/ mention /{id} |
|---|---|
| Description | |
| Entête | |

### *Nom Ressource : parametre*

Méthode : GET

| URI | Base/parametre/ |
|---|---|
| Description | |
| Entête | |

Méthode : GET

| URI | Base/parametre/{id} |
|---|---|
| Description | |

Méthode : GET

| URI | Base/ parametre/{nom}/nom |
|---|---|
| Description | |

Méthode : POST

| URI | Base/parametre / |
|---|---|
| Description | |

Méthode : PUT

| URI | Base/parametre/{id} |
|---|---|
| Description | |

Méthode : DELETE

| URI | Base/parametre/{id} |
|---|---|
| Description | |

### *Nom Ressource : role*

Méthode : GET

| URI | Base/role/ |
|---|---|
| Description | |

Méthode : GET

| URI | Base/role/{id} |
|---|---|
| Description | |

Méthode : POST

| URI | Base/role/ |
|---|---|
| Description | |

Méthode : PUT

| URI | Base/role/{id} |
|---|---|
| Description | |

Méthode : DELETE

| URI | Base/role/{id} |
|---|---|
| Description | |

*Nom Ressource : souscription*

Méthode : GET

| URI | Base/souscription/ |
|---|---|
| Description | |

Méthode : GET

| URI | Base/souscription/{id} |
|---|---|
| Description | |

Méthode : POST

| URI | Base/administrateur/ |
|---|---|
| Description | |

Méthode : PUT

| URI | Base/souscription/{id} |
|---|---|
| Description | |

Méthode : DELETE

| URI | Base/souscription/{id} |
|---|---|
| Description | |

### *Nom Ressource : symptome*

Méthode : GET

| URI | Base/ symptome / |
|---|---|
| Description | |

Méthode : GET

| URI | Base/ symptome /{id} |
|---|---|
| Description | |

Méthode : POST

| URI | Base/ symptome / |
|---|---|
| Description | |

Méthode : PUT

| URI | Base/symptome/{id} |
|---|---|
| Description | |

Méthode : DELETE

| URI | Base/ symptome /{id} |
|---|---|
| Description | |

### *Nom Ressource : utilisateur*

Méthode : GET

| URI | Base/utilisateur/ |
|---|---|
| Description | |

Méthode : GET

| URI | Base/utilisateur/{id} |
|---|---|
| Description | |

Méthode : GET

| URI | Base/ utilisateur /{email} |
|---|---|
| Description | |

Méthode : POST

| URI | Base/utilisateur/ |
|---|---|
| Description | |

Méthode : PUT

| URI | Base/utilisateur/{id} |
|---|---|
| Description | |

Méthode : DELETE

| URI | Base/utilisateur/{id} |
|---|---|
| Description | |

# Extrait de code et description de fonctionnalité

## Exemple de classe model maladie

### Fichier source : Maladie.java

```
package ht.mbds.haiti.rase.model.model;

import com.fasterxml.jackson.annotation.JsonCreator;
import com.fasterxml.jackson.annotation.JsonIgnore;
import com.fasterxml.jackson.annotation.JsonProperty;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.DBRef;
import org.springframework.data.mongodb.core.mapping.Document;

/**
 *
 * @author MyPC
 */
@Document(collection = "maladie")
public class Maladie implements Serializable
{
    @Id
    private String id;
    private String nom;
    private String description;
    private double  seuil;
    private String typeSeuil;

    @DBRef
    private Categorie categorie;
    @DBRef
    private List<Symptome> symptomes= new ArrayList<Symptome>();

    public Maladie() {
    }

    @JsonCreator
    public Maladie(@JsonProperty("nom") String nom, @JsonProperty("description") String
description, @JsonProperty("seuil") double seuil,  @JsonProperty("typeSeuil") String
typeSeuil,@JsonProperty("categorie") Categorie categorie) {
        this.nom = nom;
        this.description = description;
        this.seuil = seuil;
        this.typeSeuil =typeSeuil;
        this.categorie = categorie;
    }

/*
    public Maladie(long _id, String nom, String description, int seuil) {
        this.id = _id;
        this.nom = nom;
```

```
        this.description = description;
        this.seuil = seuil;
    }

*/


@JsonIgnore
public boolean isNew() {
    return id == null;
}


public String getId() {
    return id;
}

public void setId(String _id) {
    this.id = _id;
}


public String getDescription ()
{
    return description;
}

public void setDescription (String description)
{
    this.description = description;
}

public String getNom ()
{
    return nom;
}

public void setNom (String nom)
{
    this.nom = nom;
}

public double getSeuil() {
    return seuil;
}

public void setSeuil(double seuil) {
    this.seuil = seuil;
}

public String getTypeSeuil() {
    return typeSeuil;
}

public void setTypeSeuil(String typeSeuil) {
    this.typeSeuil = typeSeuil;
}

public Categorie getCategorie() {
    return categorie;
```

```
    }

    public void setCategorie(Categorie categorie) {
        this.categorie = categorie;
    }

    public List<Symptome> getSymptomes() {
        return symptomes;
    }

    public void setSymptomes(List<Symptome> symptomes) {
        this.symptomes = symptomes;
    }



    public void addSymptome (Symptome newSymptome){
        symptomes.add(newSymptome);
    }

    public void removeSymptome(Symptome removeSymptome){
        symptomes.remove(removeSymptome);
    }


    @Override
    public String toString() {
        return "Maladie{" + "id=" + id + ", nom=" + nom + ", description=" + description + ",
seuil=" + seuil + '}';
    }



}
```

# Exemple de classse repository

## Fichier source : MaladieRepositry

```
package ht.mbds.haiti.rase.model.repository;

import ht.mbds.haiti.rase.model.model.Maladie;
import org.springframework.data.repository.Repository;
import java.util.List;
import org.springframework.data.mongodb.repository.MongoRepository;

import org.springframework.data.mongodb.repository.Query;
/**
 *
 * @author MyPC
 */
public interface MaladieRepository extends MongoRepository<Maladie,String> {


    public Maladie findByNom(String nom);

}
```

Exemple de classe service

Ficher source de l'interface : MaladieService.java

```
package ht.mbds.haiti.rase.service;

import ht.mbds.haiti.rase.model.model.Maladie;
import java.util.List;

/**
 *
 * @author MyPC
 */
public interface MaladieService {

    // maladie

    public Maladie findMaladieById(String id);

     public Maladie findMaladieByNom(String nom);

     public List<Maladie> findMaladieAll();

     public Maladie saveMaladie ( Maladie maladie);

     public void deleteMaladie (Maladie maladie);

     public void deleteMaladie (String id);

}
```

**Fichier source de l'implementation de l'interface : MaladieServiceImpl.java**

```
package ht.mbds.haiti.rase.service;

/**
 *
 * @author MyPC
 */

import ht.mbds.haiti.rase.model.model.Maladie;
import ht.mbds.haiti.rase.model.repository.MaladieRepository;
import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

@Service
@Transactional
public class MaladieServiceImpl implements MaladieService {

    @Autowired private MaladieRepository maladieRepository;

    @Override
    public Maladie findMaladieById(String id) {
      return maladieRepository.findOne(id);
    }

    @Override
    public Maladie findMaladieByNom(String nom) {
        return maladieRepository.findByNom(nom);
    }

    @Override
    public List<Maladie> findMaladieAll() {
       return maladieRepository.findAll();
    }

    @Override
    public Maladie saveMaladie(Maladie maladie) {
       return maladieRepository.save(maladie);
    }

    @Override
    public void deleteMaladie(Maladie maladie) {
       maladieRepository.delete(maladie);
    }

    @Override
    public void deleteMaladie(String id) {
       maladieRepository.delete(id);
    }

}
```

# Exemple de classe rest

## Fichier source : MaladieController.java

```
package ht.mbds.haiti.rase.rest;

/**
 *
 * @author gaetan
 */

import ht.mbds.haiti.rase.model.model.Maladie;
import ht.mbds.haiti.rase.service.MaladieService;
import ht.mbds.haiti.rase.utils.Message;
import ht.mbds.haiti.rase.utils.SimpleMessage;
import javax.validation.Valid;
import static org.springframework.http.MediaType.APPLICATION_JSON_VALUE;
import static org.springframework.http.MediaType.APPLICATION_XML_VALUE;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseStatus;
import org.springframework.web.bind.annotation.RestController;


@RestController
@RequestMapping(value="/v1/maladie")
public class MaladieController {

    //message success
    static final String success_message_maladie_create = "La maladie a été créée avec succès";
    static final String success_message_maladie_update = "La maladie a été modifiée avec
succès";
    static final String success_message_maladie_delete = "La maladie a été supprimée avec
succès";
    // message fail
    static final String fail_message_maladie_create = "Echec de création de la maladie";
    static final String fail_message_maladie_update = "Echec de modification de la maladie";
    static final String fail_message_maladie_delete = "Echec de suppression de la maladie";

    @Autowired private MaladieService maladieService;

    @RequestMapping(method=RequestMethod.GET, produces=APPLICATION_JSON_VALUE)
    public Maladie[] getMaladieArray() {
        return maladieService.findMaladieAll().toArray(new Maladie[]{});
    }

     @RequestMapping(value="/{Id}", method=RequestMethod.GET,
produces={APPLICATION_JSON_VALUE})
    public Maladie getMaladieById(@PathVariable("Id") String Id) {
        Maladie maladie = maladieService.findMaladieById(Id) ;
        return maladie;
    }
```

```
   @RequestMapping(method=RequestMethod.POST, consumes={APPLICATION_JSON_VALUE})
    public Message createMaladie(@Valid @RequestBody Maladie malaide) {
        Message<Maladie> message= null;
        try
        {
            Maladie savedMaladie = maladieService.saveMaladie(malaide);
            message = new Message<Maladie>(success_message_maladie_create,true,savedMaladie);
            return message;
        }
        catch(Exception ex)
        {
            message = new Message<Maladie>(fail_message_maladie_create,false,null);
            return message;
        }

    }


    @RequestMapping(value="{Id}", method=RequestMethod.PUT,
consumes={APPLICATION_JSON_VALUE},produces={APPLICATION_JSON_VALUE})
    public SimpleMessage updateMaladie(@PathVariable("Id") String maladieId,
                           @RequestBody Maladie maladie) {
        SimpleMessage message =null;
        try
        {
            maladie.setId(maladieId);
            maladieService.saveMaladie(maladie);
            message= new SimpleMessage(success_message_maladie_update,true);
            return message;
        }
        catch(Exception ex)
        {
            message= new SimpleMessage(fail_message_maladie_update,true);
            return message;
        }


    }

    @RequestMapping(value="{Id}",
method=RequestMethod.DELETE,produces={APPLICATION_JSON_VALUE})
    public SimpleMessage deleteMaladie(@PathVariable("Id") String maladieId) {
         SimpleMessage message =null;
         try
         {
             maladieService.deleteMaladie(maladieId);
             message= new SimpleMessage(success_message_maladie_delete,true);
             return message;
         }
         catch(Exception ex)
         {
             message= new SimpleMessage(fail_message_maladie_delete,true);
             return message;
         }

    }


}
```

# Conclusion

# Bibliographie et URL

# Annexes