



SafetyNet

Spécifications techniques de SafetyNet Alerts

Stack Technique	2
Spécifications fonctionnelles et techniques	2
URLS	3
Endpoints	4

Stack technique

- Spring Boot
- Maven de préférence, il est aussi possible d'utiliser Gradle
- Code versionné sur un repo Git
- Bibliothèque Java permettant de parser du JSON (par exemple, Jackson ou Gson)
- Tests unitaires avec JUnit
- Couverture de code mesurée avec la librairie JaCoCo
- Logging (traces d'exécution) avec Log4j ou tinylog

Spécifications fonctionnelles et techniques

- Le serveur d'alertes SafetyNet démarre.
- Tous les endpoints url sont fonctionnels.
- Tous les endpoints tracent leurs requêtes et leurs réponses :
 - Les réponses réussies sont enregistrées au niveau Info.
 - Les erreurs ou exceptions sont enregistrées au niveau Erreur.
 - Les étapes ou calculs informatifs sont enregistrés au niveau Débogage.
- Maven est fonctionnel, exécute des tests et une couverture de code.
- Tous les endpoints sont couverts par les tests.
- La compilation génère un rapport de test Surefire sur les résultats du test JUnit.
- Le build inclut un rapport de couverture JaCoCo et archive une couverture de code de 80 %.
- SafetyNet Alerts est développé avec une architecture MVC.
- La base de code adhère aux principes SOLID :
 - Principe de responsabilité unique
 - Principe d'ouverture/fermeture
 - Principe de substitution de Liskov
 - Principe de séparation des interfaces
 - Principe d'inversion de dépendance

URLs

`http://localhost:8080/firestation?stationNumber=<station_number>`

Cette url doit retourner une liste des personnes couvertes par la caserne de pompiers correspondante. Donc, si le numéro de station = 1, elle doit renvoyer les habitants couverts par la station numéro 1. La liste doit inclure les informations spécifiques suivantes : prénom, nom, adresse, numéro de téléphone. De plus, elle doit fournir un décompte du nombre d'adultes et du nombre d'enfants (tout individu âgé de 18 ans ou moins) dans la zone desservie.

`http://localhost:8080/childAlert?address=<address>`

Cette url doit retourner une liste d'enfants (tout individu âgé de 18 ans ou moins) habitant à cette adresse. La liste doit comprendre le prénom et le nom de famille de chaque enfant, son âge et une liste des autres membres du foyer. S'il n'y a pas d'enfant, cette url peut renvoyer une chaîne vide.

`http://localhost:8080/phoneAlert?firestation=<firestation_number>`

Cette url doit retourner une liste des numéros de téléphone des résidents desservis par la caserne de pompiers. Nous l'utiliserons pour envoyer des messages texte d'urgence à des foyers spécifiques.

`http://localhost:8080/fire?address=<address>`

Cette url doit retourner la liste des habitants vivant à l'adresse donnée ainsi que le numéro de la caserne de pompiers la desservant. La liste doit inclure le nom, le numéro de téléphone, l'âge et les antécédents médicaux (médicaments, posologie et allergies) de chaque personne.

`http://localhost:8080/flood/stations?stations=<a list of station_numbers>`

Cette url doit retourner une liste de tous les foyers desservis par la caserne. Cette liste doit regrouper les personnes par adresse. Elle doit aussi inclure le nom, le numéro de téléphone et l'âge des habitants, et faire figurer leurs antécédents médicaux (médicaments, posologie et allergies) à côté de chaque nom.

`http://localhost:8080/personInfolastName=<lastName>`

Cette url doit retourner le nom, l'adresse, l'âge, l'adresse mail et les antécédents médicaux (médicaments, posologie et allergies) de chaque habitant. Si plusieurs personnes portent le même nom, elles doivent toutes apparaître.

`http://localhost:8080/communityEmail?city=<city>`

Cette url doit retourner les adresses mail de tous les habitants de la ville.

Endpoints

Nous aurons besoin des endpoints suivants :

`http://localhost:8080/person`

Cet endpoint permettra d'effectuer les actions suivantes via Post/Put/Delete avec HTTP :

- Ajouter une nouvelle personne
- Mettre à jour une personne existante (pour le moment, supposons que le prénom et le nom de famille ne changent pas, mais que les autres champs peuvent être modifiés)
- Supprimer une personne (utilisez une combinaison de prénom et de nom comme identificateur unique)

`http://localhost:8080/firestation`

Cet endpoint permettra d'effectuer les actions suivantes via Post/Put/Delete avec HTTP :

- Ajout d'un mapping caserne/adresse
- Mettre à jour le numéro de la caserne de pompiers d'une adresse
- Supprimer le mapping d'une caserne ou d'une adresse

`http://localhost:8080/medicalRecord`

Cet endpoint permettra d'effectuer les actions suivantes via Post/Put/Delete HTTP :

- Ajouter un dossier médical
- Mettre à jour un dossier médical existant (comme évoqué précédemment, supposez que le prénom et le nom de famille ne changent pas)
- Supprimer un dossier médical (utilisez une combinaison de prénom et de nom comme identificateur unique)