

Exercice4

Le but est de réaliser de manière aisée des services Rest. Il est demandé de faire un projet Spring Boot avec une configuration de web dédiée pour l'API formation et register:

Etape 1 : création du projet

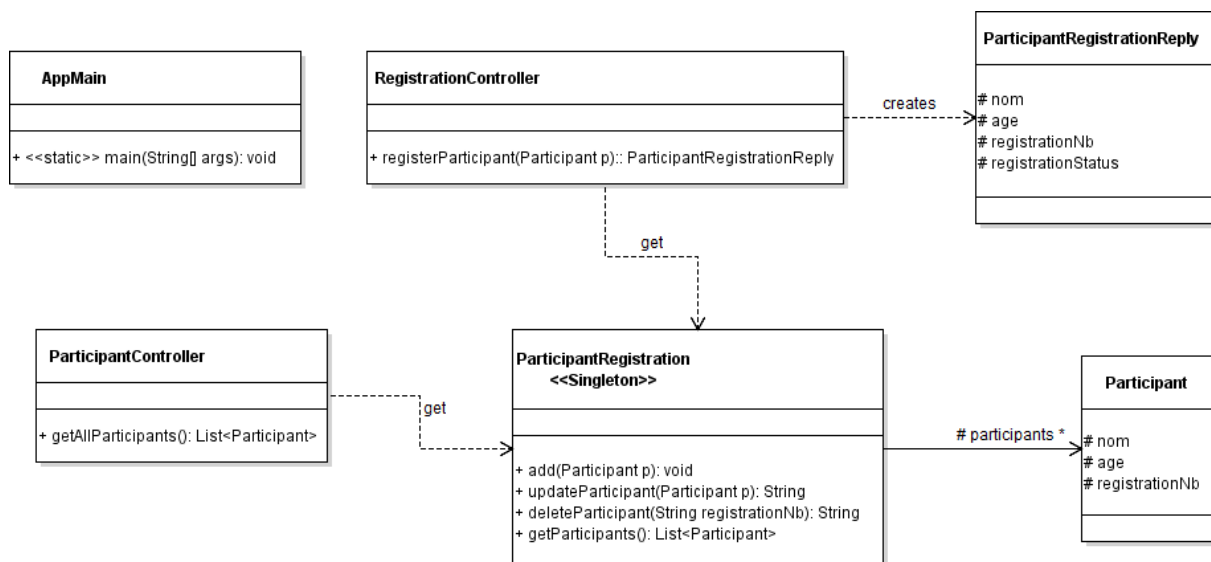
Il est demandé de visualiser les dépendances Maven à partir de l'Initializer avec les starters suivants :

- spring-boot-starter
- spring-boot-starter-test
- spring-boot-starter-web
- spring-boot-starter-test

Générer le projet et importer le dans le workspace courant

Etape 2 : Conception d'une application REST

Le projet a pour but de conserver les enregistrements de participants à une formation. Dans ce but, un concepteur propose le diagramme de classe ci-dessous.



2 services Rest sont proposés :

- 1) `ParticipantController` offre un service sur http Get afin de lister les participants déjà enregistrés. L'url choisie est `/formation/participants`,
- 2) `RegistrationController` offre les 3 services d'un CRUD
 - a. Sur http Post pour enregistrer un nouveau participant. L'url choisie est `/register/participant`, le nouveau participant étant sérialisé dans le corps de la requête,
 - b. Sur http Put pour mettre à jour un participant déjà connu. L'url choisie est `/register/participant`, le participant étant sérialisé dans le corps de la requête,
 - c. Sur http Delete pour supprimer un enregistrement de participant. L'url choisie est `/unregister/participant/{regNb}`, où `regNb` est le numéro d'enregistrement fourni lors de l'ajout d'un participant via la méthode Post.

Pour pouvoir effectuer ce développement, un participant est un POJO contenant 2 attributs :

- Un nom considéré unique pour une formation
- Un âge
- Un numéro d'enregistrement,

L'ensemble des enregistrements est conservé dans un objet de classe

`ParticipantRegistration` considéré comme singleton dans la suite de l'exercice.

Etape 3 : Préparation du livrable

Le projet a pour but de conserver les enregistrements de participants à une formation. Dans ce but,

Il est important de déclencher les phases essentielles du cycle de vie d'un projet Maven qui sont :

- `mvn clean`
- `mvn install`

Démarrer Spring boot avec `mvn spring-boot:run`

Par un navigateur accéder à l'url de base `/formation/participants` pour invoquer la méthode `Get` du `ParticipantController`.

Puis par l'utilisation d'un plug in REST pour votre navigateur préféré, il est demandé d'ajouter des participants et de les modifier ou supprimer pour en voir l'effet par le 1^{er} service.

Il est possible d'utiliser SOAPUI pour effectuer des tests similaire, il sera alors important d'accéder au WADL de ces services.