TP SPRING

STÉPHANE RIGAUT

Montreuil, le 22/10/2019

BNP PARIBAS

La banque d'un monde qui change

Sommaire

- Prérequis
- Application fil rouge
  - GITHUB & Vérification du squelette
  - Intégration IDE
  - Lancement de l'application
  - On développe
    - Caractéristiques d'un super-héros
    - Informations supplémentaires
    - API REST
  - Structure du code attendu

Prérequis

- IntelliJ
- JDK 8
- Maven

Application fil rouge

- Référentiel de super-héros (API REST auto-générées par la librairie spring-boot-starter-data-rest)
  - Créer un super-héros
  - Afficher le détail d'un super-héros
  - Lister l'ensemble des super-héros
  - Supprimer un super-héros

GITHUB & Vérification du squelette

- Récupérer le squelette : git clone https://github.com/Stephhhh/superhero.git
- Vérifier que le fichier pom.xml existe
  - Les dépendances sont correctes
    - Rest Repositories
    - Spring Data JPA
    - H2 Database
    - SpringFox
    - Lombok
  - Le plugin spring-boot-maven-plugin est présent. Il permet de créer un livrable exécutable (jar, war …). Il se trouve par défaut dans le répertoire du projet dans /target

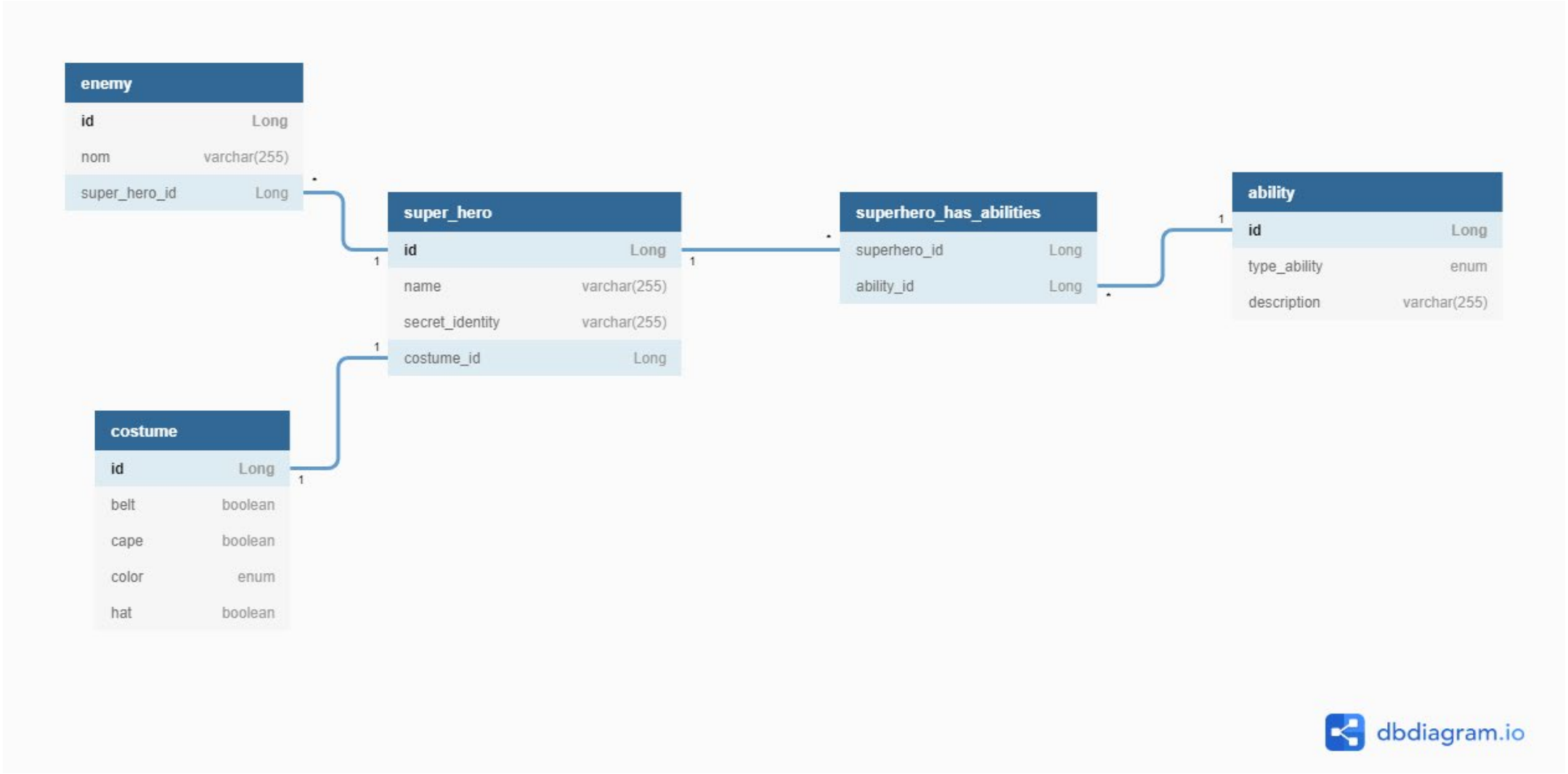
Intégration IDE

- Extraire dans un répertoire le contenu
- Ouvrir l'IDE
- File -> Open
- Choisir le dossier contenant le squelette

Lancement de l'application

- Lancer l'application
  - Aller dans la classe principale (SuperheroApplication)
  - Clique droit -> debug
- Ouvrir un navigateur et aller sur l'url http://localhost:8080
- Vérifier les endpoints (interfaces disponibles)
  - http://localhost:8080/swagger-ui.html
  - http://localhost:8080/h2

Caractéristiques d'un super-héros



Caractéristiques d'un super-héros

- Color (BLUE, GREEN, ORANGE, RED) [enum]
- TypeAbility (FLY, RUN_FAST, USE_ELEMENT) [enum]

- Bonnes pratiques
  - Penser à découper en plusieurs objets plutôt que de mettre tous les attributs dans une seule classe
  - Encapsulation
    - Les attributs d'une classe ne peuvent pas être directement manipulés de l'extérieur de la classe
      - Utiliser le mot clé private pour définir les attributs de la classe
      - Définir des méthodes d'accès à ces attributs (getter) avec le mot clé public (ou pour ce TP, utiliser spécialement lombok avec @Getter)

Informations supplémentaires

- Utiliser des Set et non des List (Hibernate gère mieux les cycles récursifs)
  - https://docs.jboss.org/hibernate/orm/3.2/api/org/hibernate/collection/PersistentBag.html
- Indiquer les relations avec les propriétés
  - Costume [OneToOne, JoinColumn, CascadeType.ALL]
  - Capacités [ManyToMany, JoinTable, CascadeType.PERSIST, CascadeType.MERGE]
  - Enemies [OneToMany, JoinColum, CascadeType.ALL]
- N'oublier pas de mettre @JsonIdentityInfo(generator = ObjectIdGenerators.UUIDGenerator.class, property="@UUID") sur les classes filles (Set)
  - Cela permet d'éviter les boucles infinies pour la sérialisation vers JSON (Jackson)

API REST

- Ajouter deux APIs REST en utilisant SuperHeroRepositoryRest
  - Pour obtenir un super-héros
  - Pour obtenir tous les super-héros

Structure du code attendu