Plan

- ❑ STRUCTURE DES PROJETS SPRING BOOT
- ❑ EXECUTION DE PROJET SB
- ❑ LES ANNOTATIONS SB
- ❑ ORGANISATION DE NOS PROJETS
- ❑ CONNECTION AUX BASES DE DONNEES AVEC JPA
- ❑ MISE EN PRATIQUE(PROJET)
- ❑ DEPLOIEMENT

Structure des projets Spring Boot

Exécution des projets Spring Boot

Les Annotations de Spring Boot

Annotations SB

@Configuration

Annotations SB

@ComponentScan

Annotations SB

@EnableAutoConfiguration

Annotations SB

@SpringBootApplication

@Bean

Annotations SB

@Component

Annotations SB

@Autowired

Annotations SB

@RestController

@RequestMapping

Annotations SB

@GetMapping

Annotations SB

@PostMapping

Annotations SB

@PutMapping

Annotations SB

@DeleteMapping

Annotations SB

@RequestParam

Annotations SB

@PathVariable

Annotations SB

@Service

Annotations SB

@Repository

Organisation de nos projets Spring Boot

Connexion aux bases de données(JPA)

JPA- Starter

- Starter

```
<dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

JPA- H2

- Starter

```
<dependency>
   <groupId>com.h2database</groupId>
   <artifactId>h2</artifactId>
   <scope>runtime</scope>
</dependency>
```

- Configuration

```
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=password
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.jpa.show-sql=false
spring.jpa.properties.hibernate.format_sql=true
```

JPA- MySql

- Starter

```
<dependency>
   <groupId>mysql</groupId>
   <artifactId>mysql-connector-java</artifactId>
   <scope>runtime</scope>
</dependency>
```

JPA- MySql

- Configuration

```
spring.datasource.url=jdbc:mysql://adress:port/db_name?useSSL=false
spring.datasource.username=
spring.datasource.password=
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect. MySQL8Dialect
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=false
spring.jpa.properties.hibernate.format_sql=true
```

JPA- Oracle

- Starter

```
<dependency>
   <groupId>com.oracle.ojdbc</groupId>
   <artifactId>ojdbc8</artifactId>
   <scope>runtime</scope>
</dependency>
```

JPA- Oracle

- Configuration

```
spring.datasource.url= jdbc:oracle:thin:@//adress:port/dbname
spring.datasource.username=
spring.datasource.password=
spring.datasource.driverClassName=oracle.jdbc.OracleDriver
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.Oracle12cDialect
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=false
spring.jpa.properties.hibernate.format_sql=true
```

JPA- Postgress

- Starter

```
<dependency>
   <groupId>org.postgresql</groupId>
   <artifactId>postgresql</artifactId>
   <scope>runtime</scope>
</dependency>
```

JPA- Postgress

- Configuration

```
spring.datasource.url=jdbc:postgresql://adress:port/dbname
spring.datasource.username=
spring.datasource.password=
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=false
spring.jpa.properties.hibernate.format_sql=true
```

Annotations JPA

Annotations JPA

@Entity

Annotations JPA

@Table

Annotations JPA

@Column

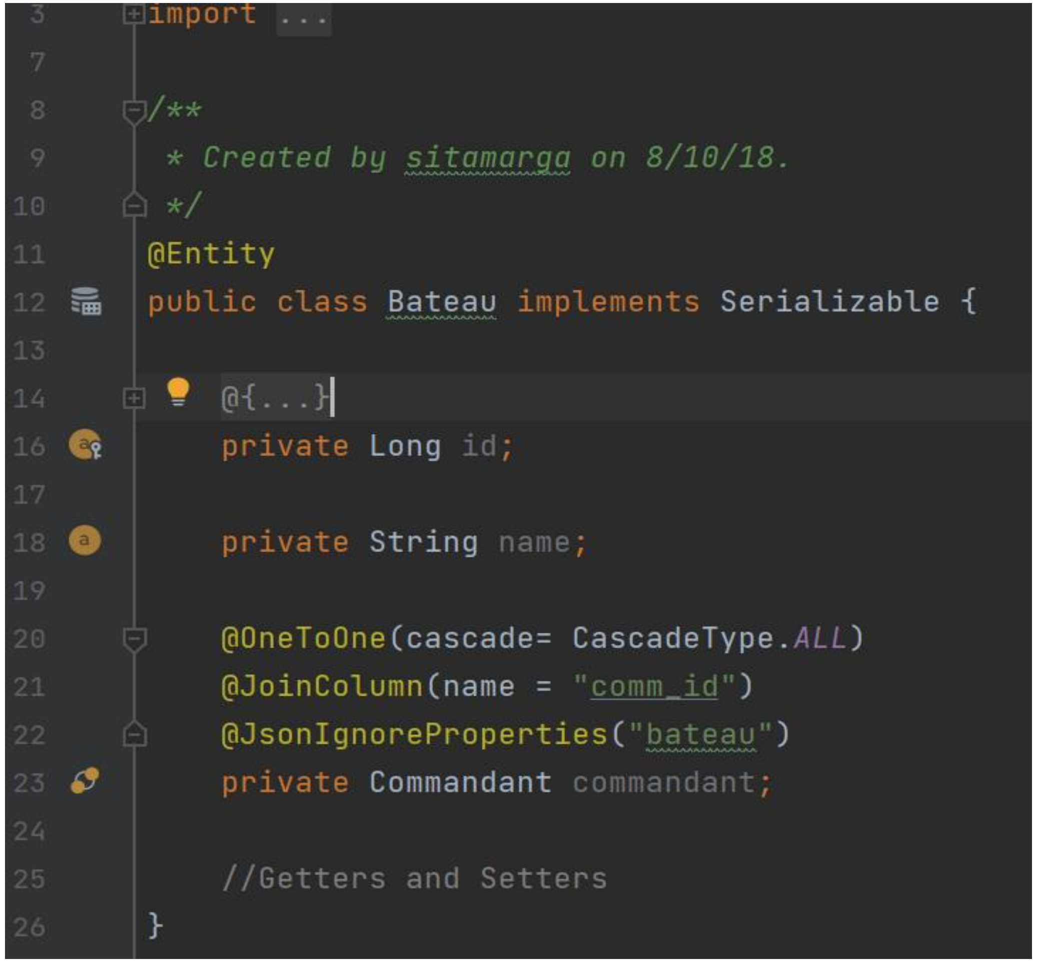
Annotations JPA

@Id

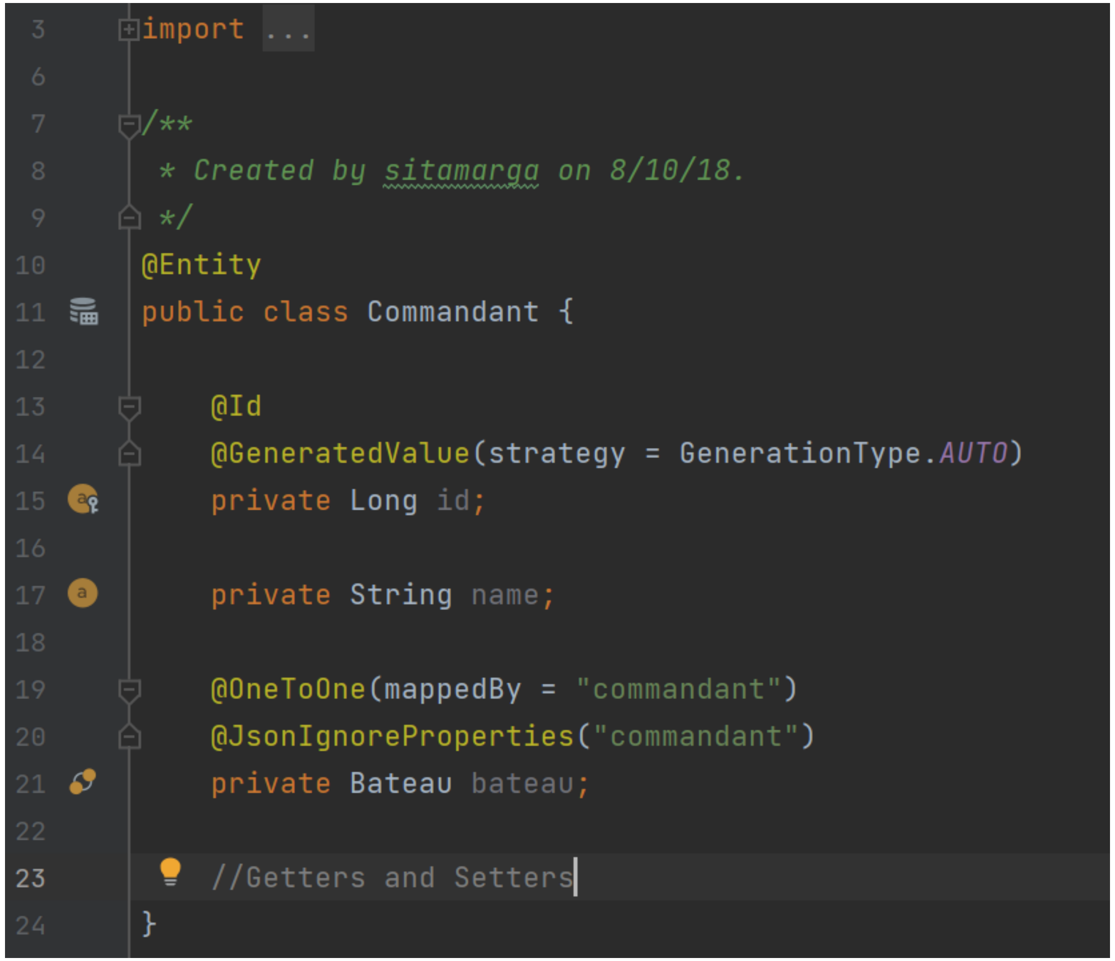
Annotations JPA

@OneToOne

Annotations JPA



```java
import ...

/**
 * Created by sitamarga on 8/10/18.
 */
@Entity
public class Bateau implements Serializable {

    @{...}
    private Long id;

    private String name;

    @OneToOne(cascade= CascadeType.ALL)
    @JoinColumn(name = "comm_id")
    @JsonIgnoreProperties("bateau")
    private Commandant commandant;

    //Getters and Setters
}
```

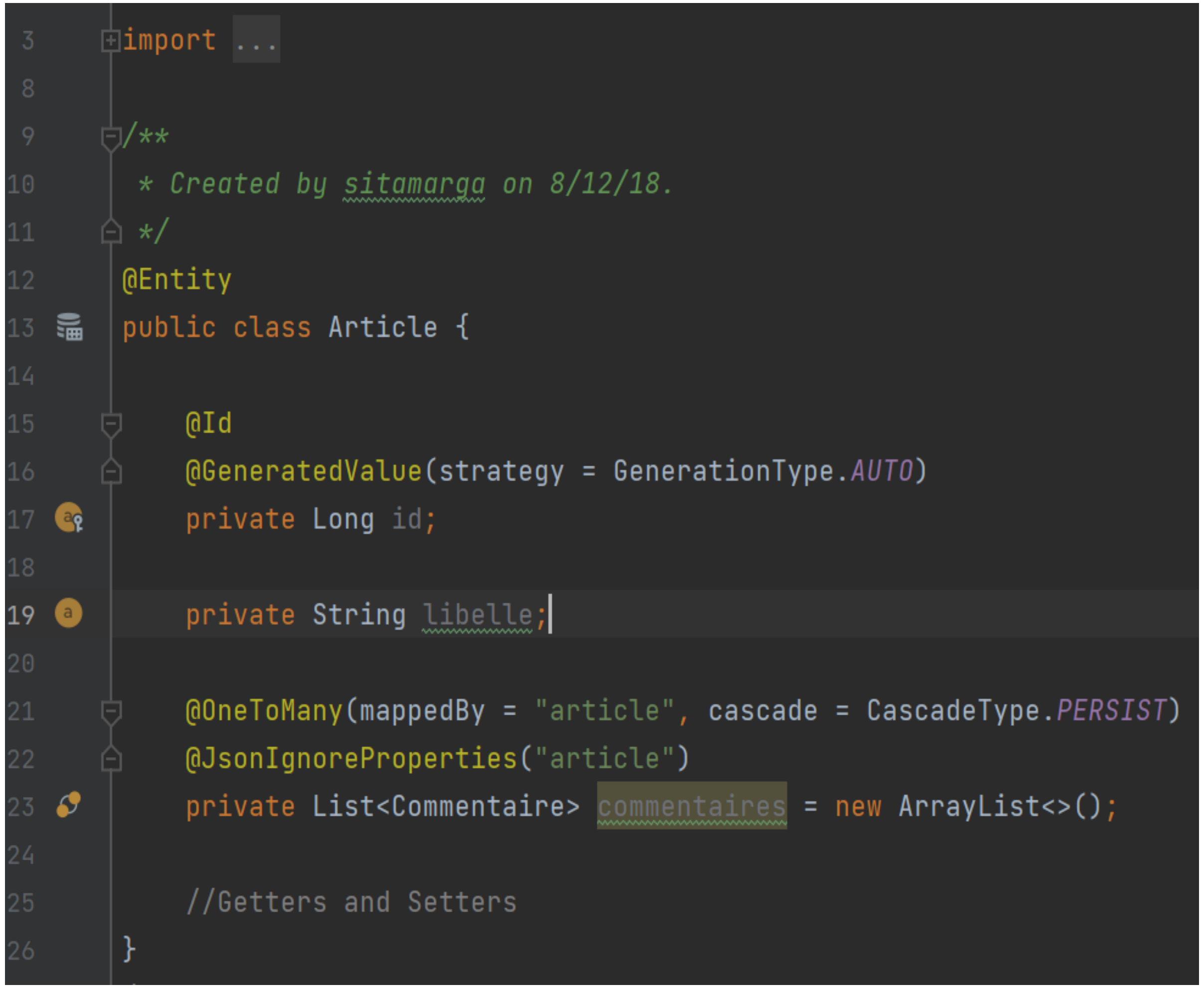


```java
import ...

/**
 * Created by sitamarga on 8/10/18.
 */
@Entity
public class Commandant {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    private String name;

    @OneToOne(mappedBy = "commandant")
    @JsonIgnoreProperties("commandant")
    private Bateau bateau;

    //Getters and Setters
}
```

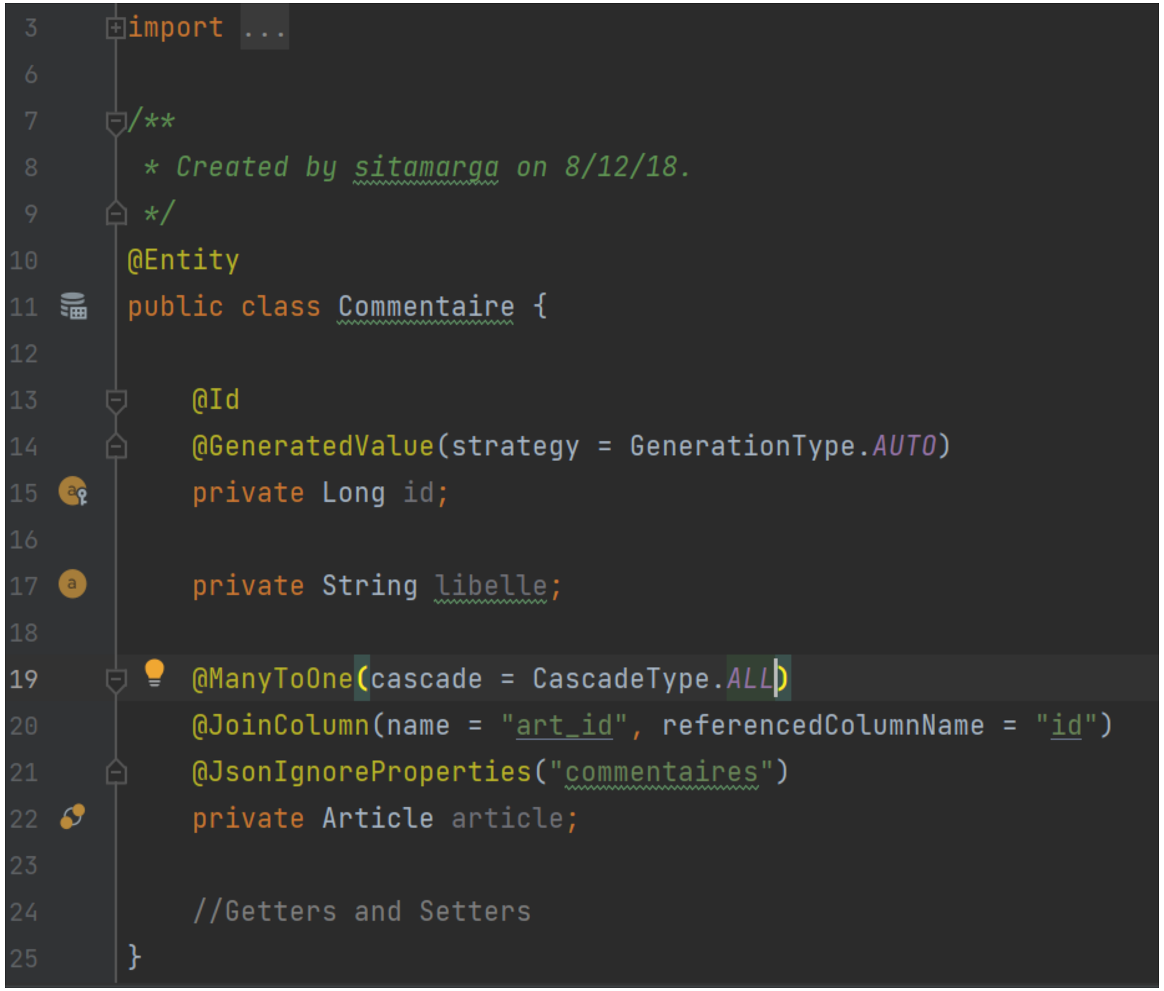
Annotations JPA

@OneToMany
@ManyToOne

Annotations JPA



```java
import ...

/**
 * Created by sitamarga on 8/12/18.
 */
@Entity
public class Article {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    private String libelle;

    @OneToMany(mappedBy = "article", cascade = CascadeType.PERSIST)
    @JsonIgnoreProperties("article")
    private List<Commentaire> commentaires = new ArrayList<>();

    //Getters and Setters
}
```

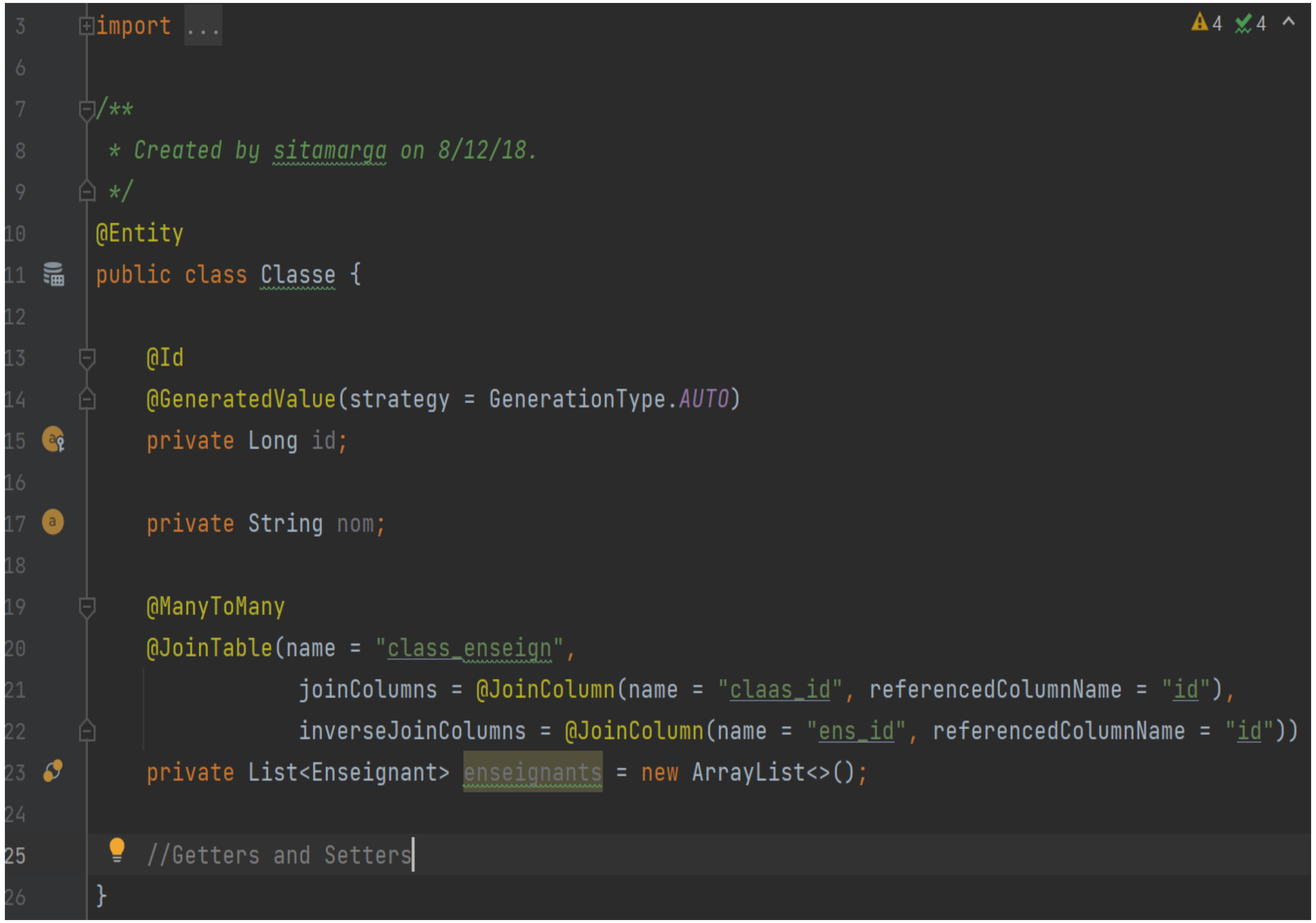


```java
import ...

/**
 * Created by sitamarga on 8/12/18.
 */
@Entity
public class Commentaire {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    private String libelle;

    @ManyToOne(cascade = CascadeType.ALL)
    @JoinColumn(name = "art_id", referencedColumnName = "id")
    @JsonIgnoreProperties("commentaires")
    private Article article;

    //Getters and Setters
}
```

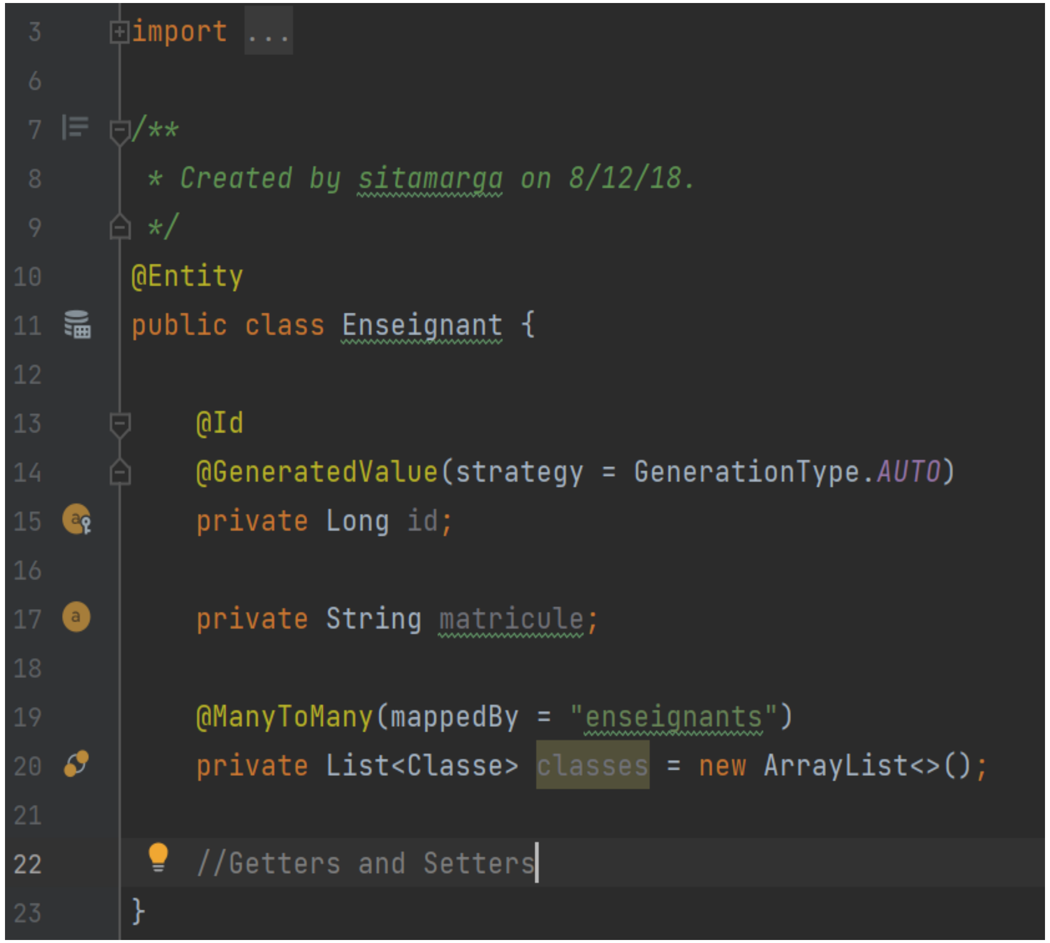
Annotations JPA

@ManyToMany

Annotations JPA



```java
import ...

/**
 * Created by sitamarga on 8/12/18.
 */
@Entity
public class Classe {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    private String nom;

    @ManyToMany
    @JoinTable(name = "class_enseign",
            joinColumns = @JoinColumn(name = "claas_id", referencedColumnName = "id"),
            inverseJoinColumns = @JoinColumn(name = "ens_id", referencedColumnName = "id"))
    private List<Enseignant> enseignants = new ArrayList<>();

    //Getters and Setters
}
```



```java
import ...

/**
 * Created by sitamarga on 8/12/18.
 */
@Entity
public class Enseignant {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    private String matricule;

    @ManyToMany(mappedBy = "enseignants")
    private List<Classe> classes = new ArrayList<>();

    //Getters and Setters
}
```

Requettes JPA (JPQL)

Projet de démo

Deploiement

Fin