Compte-rendu Tp Spring avec BDD et front end :

Les modification à faire pour géré la Base de Donnée :
J'ai eu plusieurs difficulté pour ajouter une base de donnée.
J'ai eu d'abord l'initialisation en utilisant application.properies pressent dans les ressource et un fichier sql pour l'implémenter . Mais ça na jamais marché.
Du coup j'ai changé de stratégie, et j'ai initialiser ma bdd dans la partie application avec :

```
@Autowired
private JdbcTemplate jdbcTemplate;
private static final Logger log = LoggerFactory.getLogger((Book.class));
public static void main(String[] args) {
    SpringApplication.run(SpringDataApplication.class, args);
}

@PostConstruct
private void initDb(){
    String sqlStatement[] = {
            "INSERT INTO book (title, author,text) VALUES('Siddhartha', 'Herm
             "INSERT INTO book (title, author, text) VALUES('Le Loup des step

    };
    Arrays.asList(sqlStatement).forEach(sql ->{
        jdbcTemplate.execute(sql);
    });
```

ET avec ça ma base de donnée est crée et initialisé.
Et les autres modification à faire sont seulement une interface qui extends CrudRepository<Book,Integer>.
On peut ici y définir des méthode qui n'existe pas par défaut.

```
@Repository
public interface IBookDAO extends CrudRepository<Book,Integer> {

    List<Book> findAll() ;
    Book findByTitle(String title);
    Book findById(int id);
    List<Book> findAllByAuthor(String author);
    void deleteById(int id);
}
```

Et il reste seulement à modifier le contrôleur pour appeler les méthode de l'interface pour récupérer et modifier les données de la bdd.
On peut voir ici qu'on recupère un Book en entrait, sachant que aucun id n'ai donnée, et on peut l'enregistré dans l'interface, ici la variable dao. Et elle retour l'id crée pour l'affiché sur l'interface.

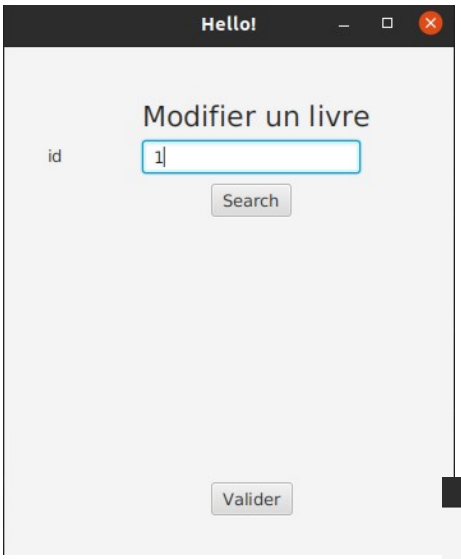
```
@PutMapping(value = "/books", produces="text/plain")
public int replaceBook(@RequestBody Book book){
    //Book b =new Book(book.getTitle(),book.getAuthor(),book.getText());
    dao.save(book);
    return book.getId();
}
```

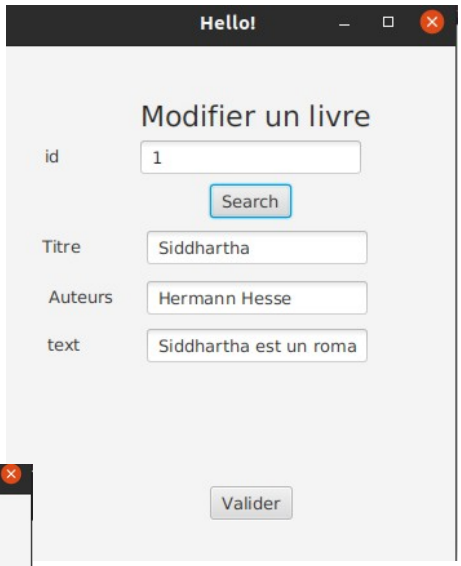
Maintenant pour l'interface Front-end, j'ai pour le début repris simplement, mon interface javaFX que j'ai fais pour le cour de web service.
Donc pour toute la partie graphique, très peut de modification à faire. Par contre au niveau de l'application il y avais des modification à faire, notamment sur la partie application qui appel l'Api.
Il se trouve que j'ai eu énormément de problème pour ajouter un livre. Je n'arriver pas à faire passer un élément dans le body de la requête, seulement dans les queryParam. J'ai donc longuement regarder sur internet. Après être passer de forum inintéressant et vidéo youtube en italien. J'ai trouver un site : https://www.tabnine.com/code/java/methods/java.net.HttpURLConnection/getResponseMessage
où j'y est trouver un moyen d'envoyer des élément dans le body, il fallait juste que j'utilise une autre librairie pour faire des requête http.
Ça marcher mais impossible de récupérer la réponse de l'Api, j'en ai besoin lors de l'ajout d'un livre pour récupérer l'id du livre.
Sachant que sur PostMan je récupérer bien l'id donc ça ne pouvais pas venir de là.
J'ai alors cherché pour réussir à récupérer la réponse.
Et je suis alors tombé sur mon site salvateur par hazard :https://zetcode.com/java/getpostrequest/
Et j'y est vue cette ligne : request.setEntity(new StringEntity("My test data"));
Qui envoyé un body à la requete en utilisant la librairie que j'utilisai au départ, et qui soit disant partant, était bien plus facile à utilisé.
Il sufisait donc que dans request.setEntity() je mette mon JSON dans le format d'une string .
C'est donc la dit fonction :

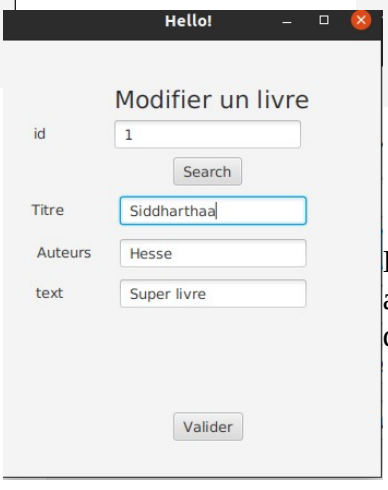
```
String data = "{\n \"title\":\""+ titre+"\",\n  \"author\": \""
        + author+"\",\n  \"text\":\""+ text+"\"\n}";
byte[] out = data.getBytes(StandardCharsets.UTF_8);

return service.path("/books").entity(out,MediaType.APPLICATION_JSON).put(new GenericType<String>(){});
```

L'interface pour la modification donne donc ceci :
en premier on recherche le livre avec l'id, se qui nous donne les info sur le livre Siddhartha :







Et on peut donc modifier les info et appuer sur valider et elle seront modifer dans la bdd :