

# Rapport de projet J2E

## 1. Présentation du projet

Le but de ce projet est d'implémenter un service web de gestion d'évènements pour des entreprises spécialisées dans ce domaine. Le client doit pouvoir créer, modifier et supprimer des évènements et les participants correspondants.

Pour cela notre application se basera sur les technologies Hibernate, SpringBoot et Thymleaf pour la partie serveur et client ainsi qu'une base de donnée PostgreSQL. Il sera donc nécessaire pour faire fonctionner ce projet d'instancier une base de données.

## 2. Notice Utilisateur

Dans le fichier `src/main/java/colloque/ressource/application.properties` il est possible de définir la base de données à utiliser ainsi que les identifiants associés. Selon la version de PostgreSQL, le port associé ne sera pas obligatoirement 5433, il est possible de le modifier dans ce-même fichier.

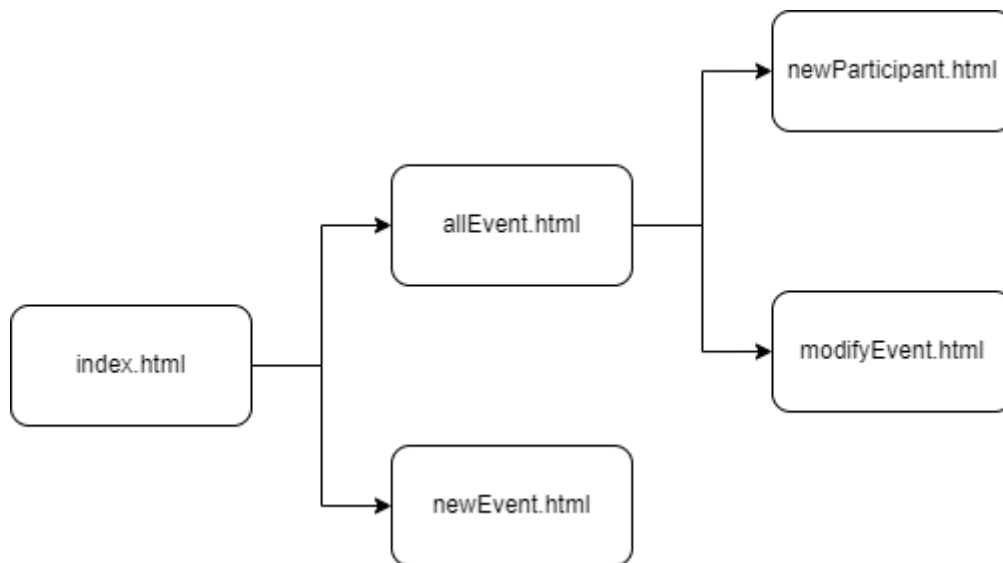
```
spring.datasource.driver-class-name=org.postgresql.Driver
spring.datasource.url=jdbc:postgresql://localhost:5433/test
spring.datasource.username=postgres
spring.datasource.password=postgres
spring.jpa.show-sql=true
spring.datasource.initialization-mode=always
```

La compilation et le lancement de notre application se fait via Maven. La version que nous utilisons est la 11. Vous pouvez changer de version pour celle qui est la plus adaptée à votre environnement de programmation dans le fichier `pom.xml` se trouvant à la racine de projet.

Lorsque l'application est lancée elle est disponible sur le port 8080. Pour y accéder taper `localhost :8080` dans votre navigateur.

### 3. Architecture du projet

L'arborescence de page de notre application est la suivante :



En tapant localhost:8080 dans votre navigateur vous êtes automatiquement redirigé sur la page index.html. Depuis cette page on peut choisir de créer un nouvel évènement (newEvent.html) ou de consulter la liste des évènements actuellement existants (allEvent.html).

Depuis la liste d'évènements, on peut supprimer ou modifier un évènement existant via des boutons ou ajouter un participant à cet évènement (newParticipant.html). Le bouton de suppression redirige directement vers la page allEvent.html et le bouton de modification vers modifyEvent.html contenant un formulaire de modification.

### 4. Limite de notre projet

Notre application gère bien la création, suppression et modification d'évènements et de participants, cependant elle ne permet pas une connexion sécurisée. L'application ne gère pas non plus les limites d'inscription aux évènements. Pour ce faire nous aurions pu nous aider des fonctionnalités « springboot-starter-security » pour les connexions et des balises Thymleaf optionnelles pour indiquer le dépassement de la limite d'inscrits.

La partie qui nous a le plus posé problème est la création de nouveaux participants liés à un évènement. Notamment la récupération des identifiants d'évènements entre le frontend et le backend.

### 5. Répartition du travail

Pour nous répartir le travail, nous avons choisi de travailler chacun sur un controller. Thomas sur le ParticipantController et Théo sur le EvenementController. Une fois cela fait, nous avons travaillé sur le même ordinateur à la correction des différents bugs et la liaison des deux controllers.