

Spring Project



Vigliengo Angel

Présentation du contexte

L'application est un système de réservation de billets pour des événements. Elle permet aux utilisateurs de parcourir les événements disponibles, de voir les détails de chaque événement et de réserver des billets pour ces événements. Les utilisateurs peuvent également créer un compte et se connecter pour gérer leurs réservations et afficher leur historique de réservation.

L'application est développée en utilisant le framework Spring Boot pour faciliter la mise en place rapide et la configuration de l'application. J'ai utilisé une base de données MySQL pour stocker les données des événements et des utilisateurs.

On utilise Spring Boot Security JPA pour gérer l'authentification et l'autorisation des utilisateurs. Cela permet de gérer les sessions et les cookies de manière sécurisée. J'ai également intégré facilement la couche de persistance de données avec JPA pour stocker et récupérer les informations des utilisateurs.

L'interface utilisateur est conçue avec Thymeleaf pour la présentation et une navigation facile. J'ai essayé d'intégrer également Spring WebFlow pour créer un flux de navigation facile et flexible pour l'utilisateur, cependant j'ai rencontré quelque problème, donc j'ai préféré ne pas l'utiliser. J'ai laissé dans le code là où j'en étais dans l'implémentation de Spring WebFlow.

J'ai mis en place AspectJ en créant la classe LoggingAspect qui permet de logger différentes informations liées à l'exécution des méthodes du package `fr.limayrac.avSpringProject.controller`, telles que les arguments d'entrée, les valeurs de retour, les exceptions levées et le temps d'exécution.

Voici les différentes méthodes que comporte LoggingAspect :

logBefore: Cette méthode est exécutée avant l'exécution de toutes les méthodes du package `fr.limayrac.avSpringProject.controller`. Elle permet de logger le nom de la méthode qui va être exécutée et ses arguments.

logAfterReturning: Cette méthode est exécutée après l'exécution avec succès de toutes les méthodes du package `fr.limayrac.avSpringProject.controller`. Elle permet de logger le nom de la méthode exécutée et la valeur de retour.

logAfterThrowing: Cette méthode est exécutée lorsque n'importe quelle méthode du package `fr.limayrac.avSpringProject.controller` lance une exception. Elle permet de logger le nom de la méthode qui a lancé l'exception et le message d'erreur.

logExecutionTime: Cette méthode est exécutée autour de toutes les méthodes du package `fr.limayrac.avSpringProject.controller`. Elle permet de mesurer le temps d'exécution des méthodes et de logger cette information.

```

@Aspect
@Component
public class LoggingAspect {

    private final Logger logger = LoggerFactory.getLogger(this.getClass());

    @Before("execution(* fr.limayrac.avSpringProject.controller.*.*(..))")
    public void logBefore(JoinPoint joinPoint) {
        logger.info("Entering method: " + joinPoint.getSignature().getName());
        logger.info("Input arguments: " + Arrays.toString(joinPoint.getArgs()));
    }

    @AfterReturning(pointcut = "execution(* fr.limayrac.avSpringProject.controller.*.*(..))",
returning = "result")
    public void logAfterReturning(JoinPoint joinPoint, Object result) {
        logger.info("Exiting method: " + joinPoint.getSignature().getName());
        logger.info("Output: " + result);
    }

    @AfterThrowing(pointcut = "execution(* fr.limayrac.avSpringProject.controller.*.*(..))",
throwing = "exception")
    public void logAfterThrowing(JoinPoint joinPoint, Exception exception) {
        logger.error("Exception in method: " + joinPoint.getSignature().getName());
        logger.error("Exception message: " + exception.getMessage());
    }

    @Around("execution(* fr.limayrac.avSpringProject.controller.*.*(..))")
    public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {
        long startTime = System.currentTimeMillis();

        Object result = joinPoint.proceed();

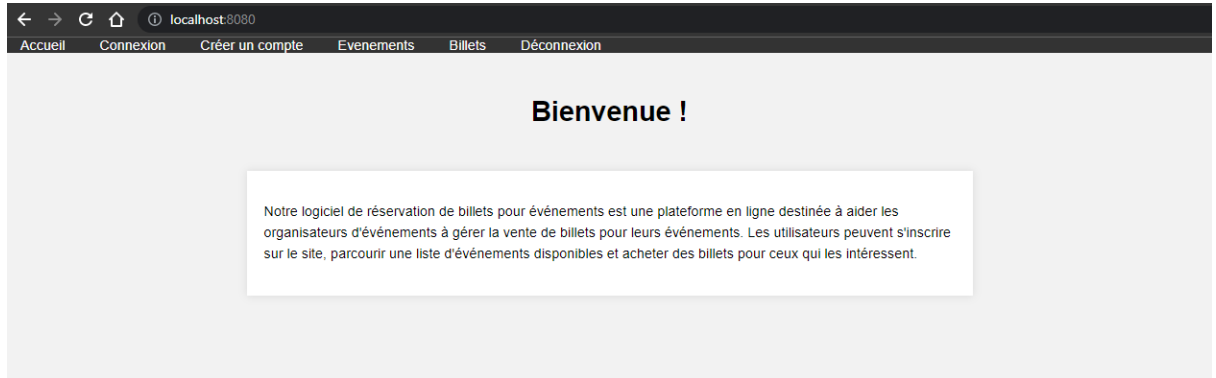
        long endTime = System.currentTimeMillis();
        logger.info("Execution time: " + (endTime - startTime) + " milliseconds");
        return result;
    }
}

```

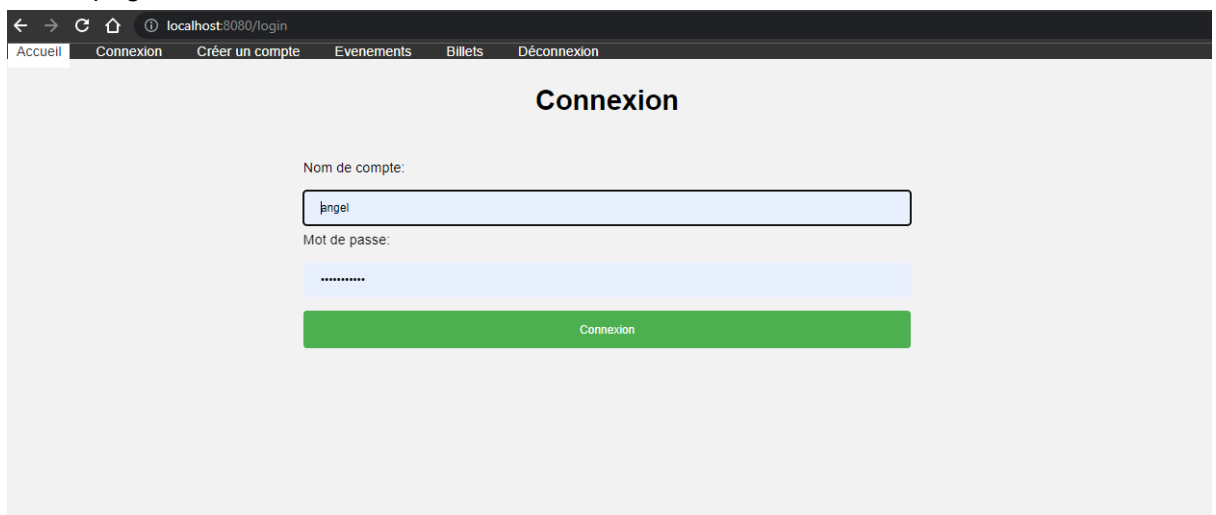
Présentation des actions possible

Voici les différentes url ainsi que les actions possibles:

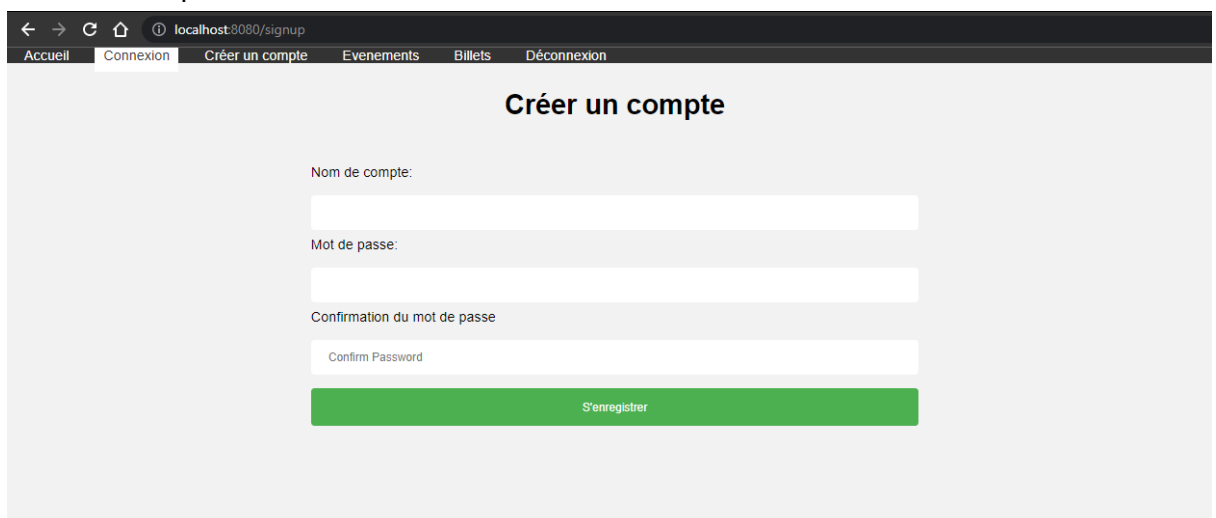
- Accueil : une simple page d'accueil qui permet d'accueillir l'utilisateur



- Connexion : une page permettant de se connecter afin de pouvoir accéder aux autres pages du site



- Créer un compte :

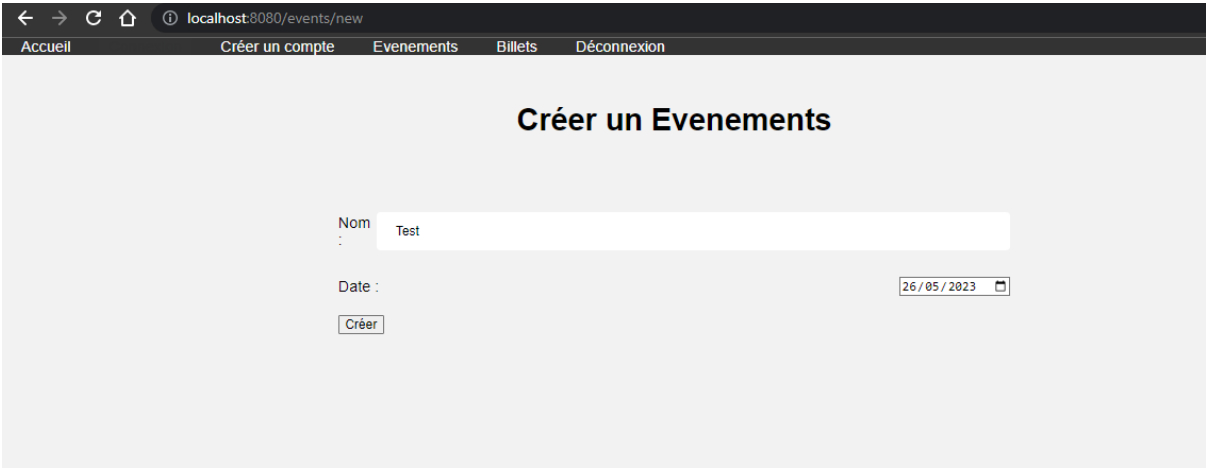
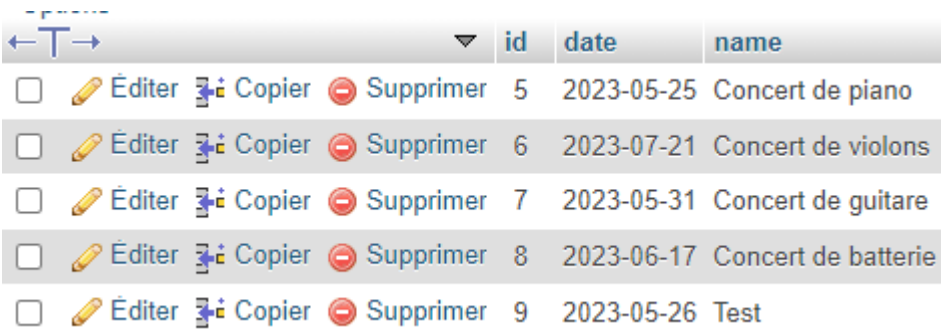


- Listes des événements : permet d'afficher la liste des evenements disponibles en bdd, sur cette page on peut voir une onglet créer une nouvelle evenement



Nom	Date
Concert de piano	2023-05-25
Concert de violons	2023-07-21
Concert de guitare	2023-05-31
Concert de batterie	2023-06-17

- Création d'un événement : cette page est un formulaire permettant la création d'événement. Un événement contient un nom, une date et un id

				id	date	name
<input type="checkbox"/>	Éditer	Copier	Supprimer	5	2023-05-25	Concert de piano
<input type="checkbox"/>	Éditer	Copier	Supprimer	6	2023-07-21	Concert de violons
<input type="checkbox"/>	Éditer	Copier	Supprimer	7	2023-05-31	Concert de guitare
<input type="checkbox"/>	Éditer	Copier	Supprimer	8	2023-06-17	Concert de batterie
<input type="checkbox"/>	Éditer	Copier	Supprimer	9	2023-05-26	Test

- Une fois envoyer on est redirigé vers la listes des événements

Nom	Date
Concert de piano	2023-05-25
Concert de violons	2023-07-21
Concert de guitare	2023-05-31
Concert de batterie	2023-06-17
Test	2023-05-26

- Liste des billets : cette page permet de lister les billets disponibles en bdd, sur cette page on peut voir une onglet créer un nouveau billet

Type	Prix	Événement
Première classe	100.00	Concert de piano
Première classe	100.00	Concert de piano
Familiale	120.00	Concert de guitare
Classique	50.00	Concert de piano
Classique	50.00	Concert de piano
Classique	50.00	Concert de piano
Classique	50.00	Concert de piano
Première classe	150.00	Concert de batterie

- Création d'un billet : cette page est un formulaire permettant la création d'un billet. Un billet contient un type, un prix, un evenement et un id

localhost:8080/tickets/new

Accueil Connexion Créer un compte Evénements Billets Déconnexion

Créer un billet

Type Familiale

Prix 100

Événement Test

Créer

				id	price	type	event_id
<input type="checkbox"/>	Éditer	Copier	Supprimer	4	100.00	Première classe	5
<input type="checkbox"/>	Éditer	Copier	Supprimer	5	100.00	Première classe	5
<input type="checkbox"/>	Éditer	Copier	Supprimer	7	120.00	Familiale	7
<input type="checkbox"/>	Éditer	Copier	Supprimer	9	50.00	Classique	5
<input type="checkbox"/>	Éditer	Copier	Supprimer	10	50.00	Classique	5
<input type="checkbox"/>	Éditer	Copier	Supprimer	11	50.00	Classique	5
<input type="checkbox"/>	Éditer	Copier	Supprimer	12	50.00	Classique	5
<input type="checkbox"/>	Éditer	Copier	Supprimer	13	150.00	Première classe	8
<input type="checkbox"/>	Éditer	Copier	Supprimer	14	100.00	Familiale	9

- Une fois envoyer on est redirigé vers la listes des billets

Type	Prix	Événement
Première classe	100.00	Concert de piano
Première classe	100.00	Concert de piano
Familiale	120.00	Concert de guitare
Classique	50.00	Concert de piano
Classique	50.00	Concert de piano
Classique	50.00	Concert de piano
Classique	50.00	Concert de piano
Première classe	150.00	Concert de batterie
Familiale	100	Test

- Déconnexion : lorsqu'on clique sur Déconnexion quand on est connecté on est redirigé vers l'accueil

Notre logiciel de réservation de billets pour événements est une plateforme en ligne destinée à aider les organisateurs d'événements à gérer la vente de billets pour leurs événements. Les utilisateurs peuvent s'inscrire sur le site, parcourir une liste d'événements disponibles et acheter des billets pour ceux qui les intéressent.

- Si on clique sur Evenements ou Billets lorsqu'on est pas connecté on est redirigé vers la page de Connexion

Nom de compte:

angel

Mot de passe:

Connexion

