

# Projet Jeux Olympiques 2024

## Documentation technique

Créateur: Hugo ROGER

### Introduction

#### Technologies utilisées :

Côté backend : Java 8 avec le framework Spring, PostgreSQL 15 pour la base de données.

Côté frontend : HTML, CSS, Javascript, Bootstrap, template Thymeleaf de Java Spring, jQuery.

Le site est hébergé sur Heroku. La base de données est hébergée sur la plateforme Aiven.

Utilisation de l'API Stripe pour les paiements, des bibliothèques pdfbox et zxing pour la génération de PDF et de QR code.

---

### Sécurité :

#### Authentification

Utilisation de Spring Security.

La classe entité User est liée à l'AuthenticationManager de Spring Security, permettant de gérer les sessions et authentifications.

L'administrateur doit entrer un login et un mot de passe (qui sera encrypté par la fonction de hachage BCrypt).

Il y a 3 rôles : client, employee et admin.

Il est impossible pour un utilisateur d'avoir 2 sessions ouvertes simultanément.

#### Validation des données

Les champs suivants sont validés systématiquement dans le backend dès la création d'une entité ou de leur passage dans un DTO:

Pour les utilisateurs:

- Prénom utilisateur (non nul, enlèvement des espaces inutiles, sanitisation contre injection XSS, string de taille limitée)
- Nom utilisateur (non nul, enlèvement des espaces inutiles, sanitisation contre injection XSS, string de taille limitée)
- Email (non nul, unique, format valide, sanitisation contre injection XSS, string de taille limitée)
- Mot de passe (encodage, doit être entre 8 et 20 caractères et doit contenir au moins une majuscule, une minuscule, un chiffre et un caractère spécial parmi les suivants: `#?!@$%^&*+,-`)

Pour les offres:

- Nom d'offre (non nul, enlèvement des espaces inutiles, sanitisation contre injection XSS, string de taille limitée)
- Description d'offre (non nul, enlèvement des espaces inutiles, sanitisation contre injection XSS, string de taille limitée)
- Prix/montant (non nul, forcer à avoir deux décimales, valeur bornée, doit être un nombre). (s'applique aussi aux achats)

Dans le backend, ils sont mappés en tant que value objects.

Côté frontend, il y a aussi de la validation des formulaires de connexion, d'inscription et de création d'offres pour ces paramètres.

---

Modèle de base de données :

cf modèle de base de données

---

Achats :

Utilisation simple de l'API Stripe en mode test pour faire des paiements mockés. Il n'y a pas d'implémentation d'interface pour demander les coordonnées bancaires ou bien d'authentification par Stripe.

---

## Evolution futures :

On pourra envoyer par email les e-billets lorsque l'achat sera effectué.

Il serait aussi pertinent de demander une confirmation de création de compte par email.

On pourra aussi utiliser le système de vérification d'e-billet (notamment s'il y a développement d'application mobile).