

Université d'Orléans
2019-2020

Malik
SULTAN Li-San
Bineta diagne
Romain
Yoann

Rapport

Muséum d'Histoire Naturelle d'Orléans



Sommaire

Introduction

Présentation du projet

But du projet

Outils utilisés

Conduite du projet

Répartition des tâches

Communication, fréquence de travail etc..

Fonctionnement du projet

Modèle de données

Architecture du projet

Description des pages

Conclusion

Amélioration à apporter

Problèmes rencontrés

Introduction

Au cours de notre année en Master 2 MIAGE, ils nous a été demandé de réaliser un projet en Interopérabilité des SI afin de mettre en pratique ce que nous avons appris.

Dans un premier temps, nous vous présenterons le projet qui nous a été demandé et nous vous expliquerons quels outils nous avons utilisé pour le mettre en place.

Puis nous vous parlerons de la conduite du projet, comment nous avons réparti les tâches au sein du groupe, nos outils pour partager notre projet etc..

Ensuite nous vous présenterons rapidement notre modèle de données, nous décrirons également les pages de notre application.

Enfin, nous discuterons sur les possible évolution de ce projet et des problèmes rencontrés avant de dire ce que nous a apporté ce projet.

Présentation du Projet

But du projet

Ce projet consiste à modéliser le SI du nouveau muséum d'histoire naturelle d'Orléans, le MOBE.

Ce musée a une collection d'objet qui augmente chaque année avec de nouvelles oeuvres achetées.

Ces **oeuvres** peuvent être exposées dans l'une des **20 salles du musée** pendant une certaine **période**, sinon elles sont **remisées dans la réserve** jusqu'à la prochaine **exposition**.

Un exemple :

le musée a acheté le **15 septembre 2015** un **tyrannosaure**, nommé **Rex**, stocké à la livraison dans la **réserve** ;

le **1er janvier 2016**, il a été exposé dans le **grand hall du musée** ;

le **1er septembre 2016**, Rex a été déplacé dans la **grande salle 1**, pour l'exposition "**les dinosaures**".

le **15 mars 2017**, il est **revenu dans le grand hall**.

Du **20 décembre 2018 au 1 avril 2019**, il est retourné en réserve pour être **restauré**.

Il est à nouveau exposé dans la grande salle 1 depuis.

Des **expositions thématiques** sont organisées régulièrement (eg "les dinosaures") mais pas en permanence.

Dans ce cas, certaines oeuvres peuvent être **prêtées** par d'autres musée pour être exposées.

Le musée met également en place la **vente de billets électroniques en ligne** pour les expositions.

Après le paiement, les utilisateurs récupèrent un **QR code** sur leur mobile ; en arrivant dans le hall d'entrée du musée, des tourniquets automatiques avec des scanners de QR code leur permettent de rentrer directement sans faire la queue à la caisse.

Chaque QR code ne peut évidemment être utilisé qu'une seule fois.

L'application est donc divisé en deux partie: une **partie admin** et une **partie client**.

La partie admin (Client JFx) demande d'exposer des oeuvres dans des salles du musées pendant une période définie. Les oeuvres peuvent être restaurées, mais aussi prêtées par d'autres musées. Il est possible que des oeuvre puissent être acheté également.

La partie client (Client Web JS) permet à un visiteur de pouvoir acheter des billets électroniques en ligne. Ce paiement permet de fournir un QR code qui sera utilisé dans le hall d'entrée du musée.

Des tourniquets automatiques avec des scanners de QR code leur permettent de rentrer directement sans faire la queue à la caisse.

Chaque QR code ne peut évidemment être utilisé qu'une seule fois.

Conduite du projet

Répartition du travail

Pour la réalisation de ce projet, nous avons formé une équipe de 5 personnes. Afin de répartir au mieux nos tâches de travail nous avons dans un premier temps pensé à ces deux parties (Admin et Utilisateur).

Nous avons jugé que la partie Utilisateur était la plus importante. En effet, il y a une partie Qr code, paiement etc..

Comme nous étions cinq, nous avons décidé que ce serait dans un premier temps trois personnes qui travaillent sur la partie Utilisateur et deux personnes sur la partie Admin. Ainsi, Bineta, Malik et Yoann étaient sur la partie Utilisateur et Li-San et Romain étaient sur la partie Admin.

Précision pour nom sur le git différent :
Sultan Li-San: ls-info

Outils utilisés

Afin de réaliser ce projet divers outils ont été nécessaires à la fois pour la répartition des tâches, parler de l'avancement du projet mais surtout réaliser ce projet.

Pour la communication au sein du groupe, nous avons choisi le logiciel discord. Nous avons créé un serveur de chat afin de se répartir les tâches, s'entraider et définir les points importants de ce projet.

Pour le rapport, nous avons utilisé un document partagé via google. Nous avons également créé un autre document pour mettre des ordres de priorité pour ce projet.

Puis pour réaliser ce projet, nous avons utilisé principalement le logiciel IntelliJ pour développer en java. Notre application tourne en java 11, nous avons pu appliquer du jfx 11 à l'aide des dépendances maven pour le jfx 11 et du fichier module-info.

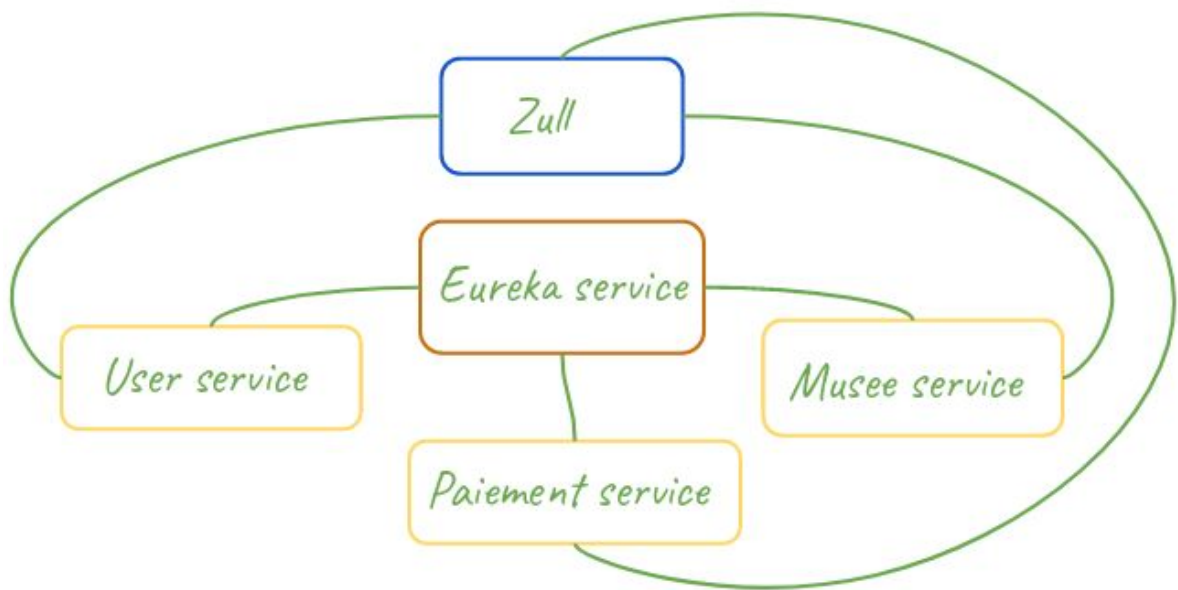
Enfin le dépôt du projet se trouve sur bitbucket.

Fonctionnement du projet

Pour ce projet, nous avons choisi l'api REST.

Pour lancer le projet dans l'ordre il faut s'aider du fichier README.

C'est à dire, dans un premier temps il faut lancer Eureka service, puis Zull puis Userservice puis musee-service ensuite paiement-service. La partie jfx se lance en dernier.



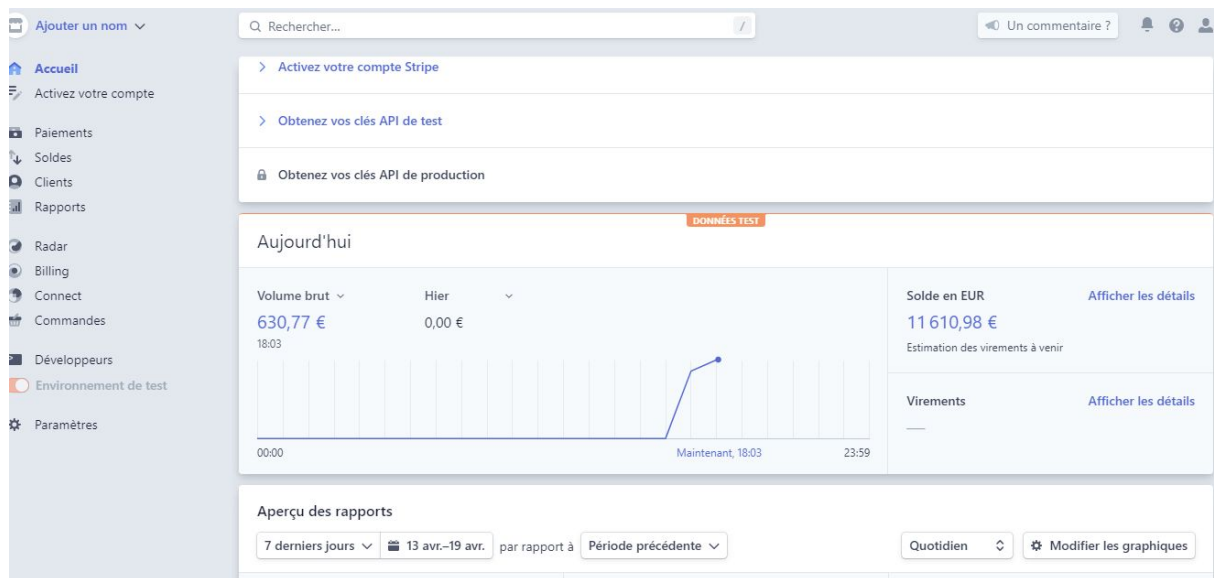
Détail des liens entre les projets

User service sert à l'authentification tandis que musee service est réservé pour les fonctionnalités créations d'expositions, création des salles, des oeuvres etc... Paiement service est utilisé pour la partie paiement.

Pour générer le paiement nous avons utilisé l'api stripes. Nous l'avons pour l'instant utilisé en mode test mais il peut basculer en mode production.

Il y a également deux clé : l'une est du côté angular et l'autre est du côté spring.

pour tester l'api coté angular voici les données de test les données seront récupérées sur le compte stripes creer par malik.

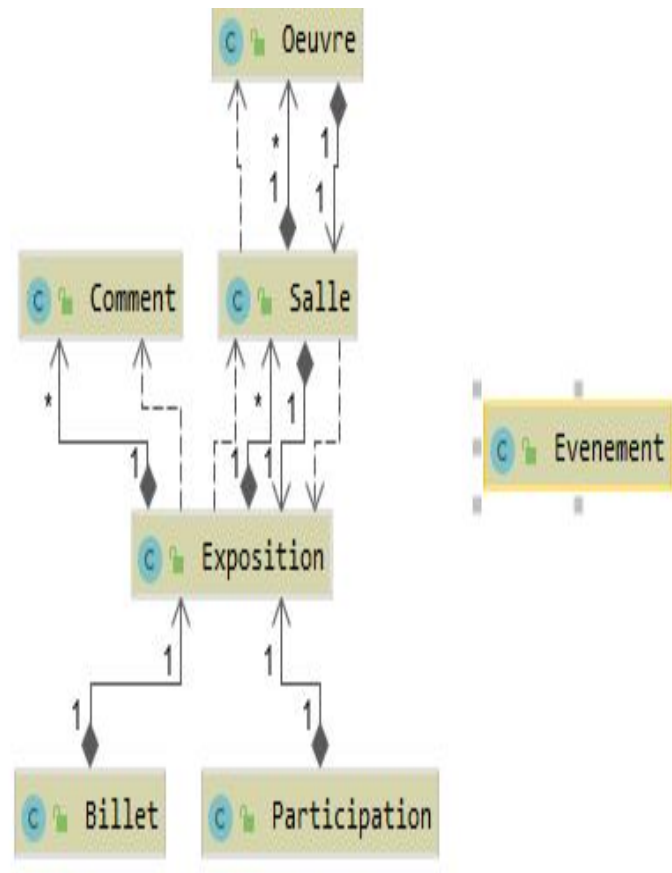


<https://stripe.com/docs/testing>

donc au moment du paiement

<https://medium.com/oril/easy-way-to-integrate-payments-with-stripe-into-your-spring-boot-angular-application-c4d03c7fc6e>

Modèle de données



Architecture du projet

<http://localhost:8765/api/paiement/make/>

<http://localhost:8765/api/user/service/>

<http://localhost:8765/api/musee/>

Urls	GET	POST	PUT	DELETE
Dans Musée service				
/expositions	récupérer toutes les expositions	Ajouter une exposition body: exposition		
/expositions/{expositionId}	récupérer une exposition par son Id		Modifier une exposition	Supprimer une exposition par son id
/commentaires/{idexpo}		Ajouter un commentaire à une exposition		

<code>/participations"</code>		Participer à une exposition		
<code>/salles/{idsalle}/exposition</code>	Renvoie l'exposition d'une salle			
<code>/expositions/{id}/salle/{idsalle}</code>		Ajouter une salle à une exposition		
<code>/expositions/commentaire/{id}"</code>	Récupérer les commentaires d'une exposition			
<code>"/expositions/{id}/commentaires</code>	Récupérer les commentaire d'une exposition selon son Id			
<code>/oeuvres</code>	Récupérer toutes les oeuvres	Ajouter une oeuvre bod: oeuvre		
<code>/oeuvres/{oeuvreId}</code>	Récupérer une oeuvre par son Id			
<code>/reserves"</code>	Récupérer toutes les oeuvre dans le réserve			
<code>/exposes"</code>	Récupérer toutes les oeuvre exposées			
<code>/reserves/{id}</code>			Mettre une oeuvre en réserve	
<code>/restauration/id</code>			Mettre une oeuvre en restauration	
<code>/restauration/oeuvres</code>	Récupérer les oeuvres qui sont en restauration			
<code>/prets</code>	Récupérer les oeuvres empruntés		Enregistrer une oeuvre comme empruntée body : oeuvre	
<code>/salles/{id}/oeuvres</code>	Récupérer les oeuvres d'une			

	salle			
<code>/expositions/{id}/oeuvres</code>	Récupérer les oeuvres d'une exposition			
<code>/salles/{salleid}/oeuvres/{oeuvresid}</code>		Mettre une oeuvre dans une salle		
<code>/oeuvres/{oeuvresid}/salles/{salleid}"</code>		Lié une salle à une oeuvre		
<code>/oeuvres/{oeuvresid}/sortiereserve"</code>	Ajout d'une notification quand une oeuvre est sortie du réserve			
<code>billet/{id}/send/{email}</code>		Envoyer le billet dont on a mis l'id à l'email		
<code>billet/{id}/validate</code>		Valider un billet		
<code>/salles</code>	Récupérer toutes les salles	Ajouter une salle body : salle	Modifié une salle	
<code>/salles/{salleId}</code>	Récupérer une salle par son Id			
<code>/expositions/{idexpo}/salles</code>	Récupérer les salles d'une exposition			
<code>/Salles/disponible</code>	Récupérer la liste des salles disponible			
<code>/qrcode/billet/{id}</code>	Récupérer le QR Code d'un billet à partir de son ID			
<code>/qrcode/billet/{id}/send/{email}</code>	Envoyer le QR Code d'un billet par mail à une adresse mail			
<code>/qrcode/billet/{id}/validate</code>	Permet de valider un billet (le marquer comme étant déjà utilisé)			

Dans Paiement-service : Racine = /payment				
/charge		Envoyer le paiement sur le compte stripe		

Description des pages

Partie Admin:

Connexion

Pour la partie admin, vous aurez une page de connexion demandant votre login et votre mot de passe. Il y a également un bouton de connexion. Bien entendu cette partie vérifie si votre login/mot de passe est correcte avec SpringSecurity et une requête get. Il envoie également une erreur si les champs ne sont pas saisis.

Menu

Une fois connecté, vous accédez à une page menu. Cette page est composée de plusieurs boutons.

Le bouton **Consulter les salles** vous dirigera vers la partie *Consulter les Salles*.

Le bouton **Consulter les oeuvres** vous dirigera vers la partie *Consulter des Oeuvres*.

Le bouton **Consulter une exposition** vous dirigera vers la partie *Consulter une exposition*.

Le bouton **Créer une exposition** vous dirigera vers la partie *Création d'exposition*.

Le bouton **Organiser une exposition** vous dirigera vers la partie *Organisation d'une exposition*.

Le bouton **Ajouter une salle** vous dirigera vers la partie *Ajouter une Salle*.

Le bouton **Acheter une oeuvre** vous dirigera vers la partie *Acheter une Oeuvre*.

Le bouton **Emprunter une oeuvre** vous dirigera vers la partie *Louer une Oeuvre*.

(Le bouton **Rendre une oeuvre** vous dirigera vers la partie *Rendre une Oeuvre*.)

Vous avez la possibilité de vous **déconnecter** ce qui vous permettra de vous rediriger vers la page de connexion.

Consulter les Salles

(A compléter si détail des oeuvre dans les salles)

Cette page permet de voir les salles créées précédemment.

Il y a un bouton qui permet de retourner au menu principal.

Consulter des oeuvres

Sur cette page vous pouvez consulter les oeuvres du musée.

Elle permet de tracer une oeuvre, savoir si elle est en restauration, dans quelle salle elle est exposée, si elle est loué et non acheté etc..

C'est aussi la possibilité pour vous de rendre une oeuvre, l'exposer, la mettre en réserve ou restauration en fonction du bouton que vous choisirez.

Il y a un bouton qui permet de retourner au menu principal.

Consulter une exposition

Cette page permet de voir les expositions en cours.

Elle vous montre la liste des salles concernées. En fonction des salles que vous choisissiez, vous verrez apparaître les oeuvres présente dans ces salles.

Il y a un bouton qui permet de retourner au menu principal.

Organisation de l'exposition

Cette page propose une liste des salles disponibles. Il y a également une liste des exposition.

Il y a un bouton pour exposer.

Il y a un bouton qui permet de retourner au menu principal.

Création d'exposition

Cette partie va contenir deux pages.

La première vous demandera le nom de l'exposition que vous voulez créer ainsi qu'une catégorie à ajouter mais également la date de début et la date de fin de cette exposition.

Vous pourrez également mettre une description concernant cette exposition.

Puis vous avez la possibilité de continuer ou de quitter pour revenir dans le menu.

Si vous continuez, vous allez devoir choisir le bouton Choisir salle pour avoir la seconde page.

(Cette page vous demandera de sélectionner des oeuvres et de choisir une salle afin de les exposer dans celle-ci.

Bien entendu comme une exposition peut se dérouler dans plusieurs salle, vous aurez la possibilité de continuer. Mais vous pouvez également choisir de terminer.

Si vous continuez, vous aurez accès à la même page mais les oeuvres ainsi que la salle choisi précédemment seront supprimés dans les listes. Vous aurez une fois encore les deux boutons pour choisir de nouvelles oeuvres et salles ou terminer.)

Ajouter Salle

Cette page permet d'ajouter des salles d'exposition. Dans le sujet il est dit qu'il y a 20 salles mais le musée peut s'agrandir. En pensant à cette éventualité, la création de salle est donc plus pratique.

Il y a un bouton qui permet de retourner au menu principal.

Rendre une oeuvre

Cette page présente une liste d'oeuvre louées.

Il y a un bouton qui permet de retourner au menu principal.

Location d'oeuvres

Cette page propose de saisir une oeuvre que l'on souhaite emprunter.

Il y a un bouton Emprunter pour valider la location de cette oeuvre.

Il y a un bouton qui permet de retourner au menu principal.

Achat d'oeuvre

Sur cette page vous pourrez saisir le nom de l'oeuvre que vous souhaitez acheter.

Le bouton acheter permet de valider l'achat.

Il y a un bouton qui permet de retourner au menu principal.

Quelques images de l'application javaFx:

Description 1: Page de connexion de l'application , l'utilisateur doit se connecter avec son pseudo et mot de passe (tepa,tepa)

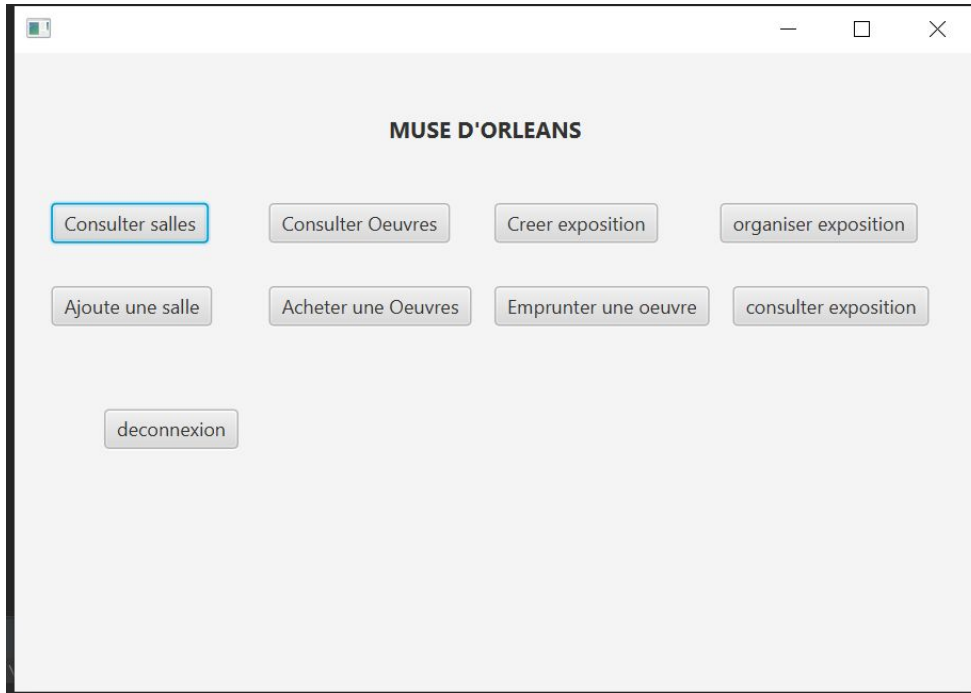


ACCUEIL

pseudo

mot de passe

Description 2 : Page d'accueil de l'application avec une liste d'option



Description 3 : Afficher tous les oeuvres

recherche

liste oeuvres

id	nom	date
17	acheter	Fri Apr 10 1...
18	emprunt	Fri Apr 10 1...
19	emprunter	Fri Apr 10 1...
20	test	Sat Apr 11 ...
21	test	Sat Apr 11 ...
22	TEST	Sat Apr 11 ...
23	test	Sat Apr 11 ...
24	testi	Sat Apr 11 ...
25	test	Sat Apr 11 ...
26	test	Sat Apr 11 ...

Donne oeuvre

id
nom
date
achat

☐ expose
☐ reserves
☐ restauration
☒ tout

expose

rendre

mettre en reserves

restauration

historique

retour au menu

Description 4 : Afficher les oeuvres en restauration

recherche

liste oeuvres

id	nom	date
17	acheter	Fri Apr 10 1...
18	emprunt	Fri Apr 10 1...

Donne oeuvre

id
nom
date
achat

☒ expose
☐ reserves
☐ restauration
☐ tout

expose

rendre

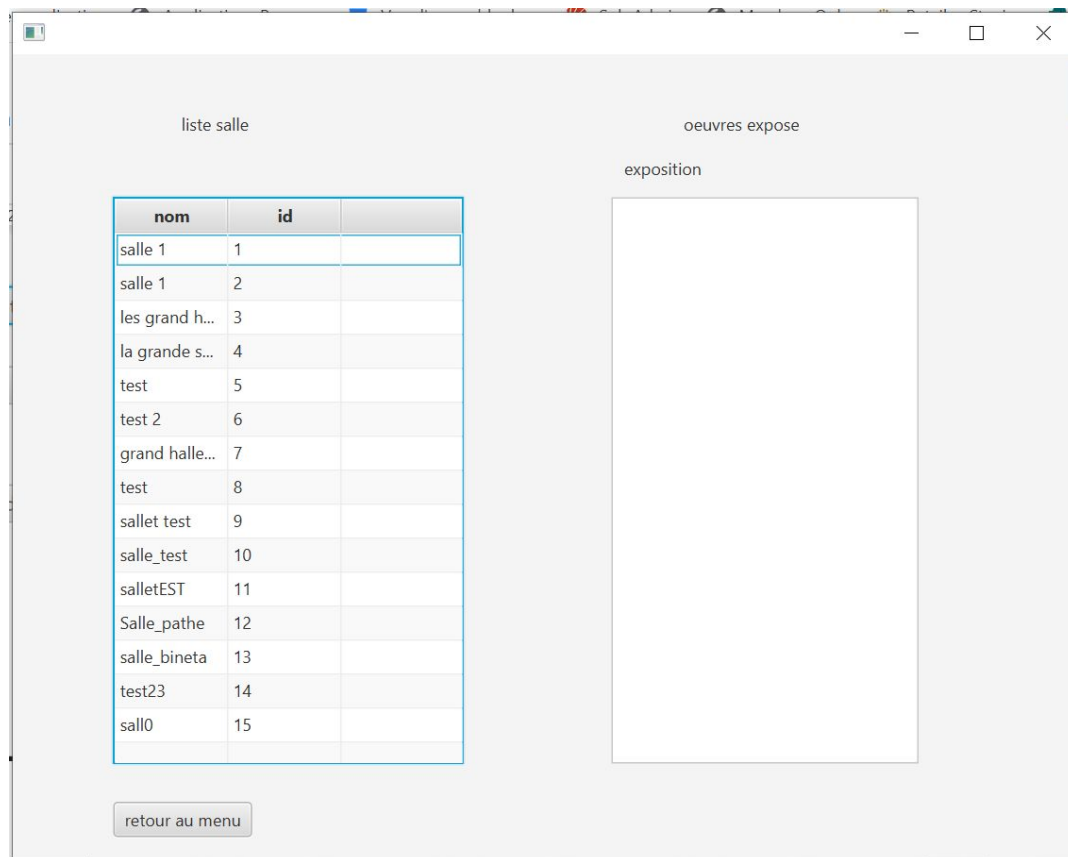
mettre en reserves

restauration

historique

retour au menu

Description 5 : Liste des Salles



NB : cette liste de description n'est pas exhaustive , elle montre juste quelque partie de notre application.

Description Application angular :

se mettre dans le dossier angular et taper

-npm install

-npm start

se diriger vers localhost:4200

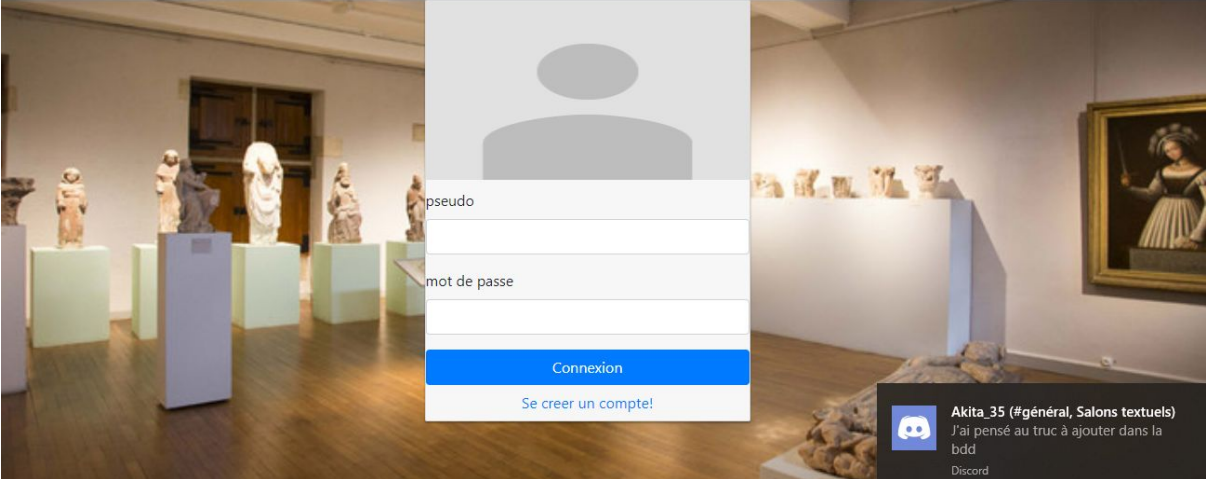
la vue de connexion , utilise spring security en mode basic


voila ce qu'il faut mettre dans le header de la requête http au moment de la connexion

```
const headers = new HttpHeaders({
  user ? {
    authorization: 'Basic ' + btoa(user.username + ':' +
user.password)
  } : {}
});
```


MUSEE D'HISTOIRE NATURELLE D'ORLEANS

Accueil Inscription Connexion






pseudo

mot de passe

Connexion

[Se creer un compte!](#)



Akita_35 (#général, Salons textuels)
J'ai pensé au truc à ajouter dans la bdd
Discord

MUSEE D'HISTOIRE NATURELLE D'ORLEANS

Accueil binette Deconnexion

Toutes les expositions

#	titre	categorie	date debut	date fin	paiement	Detail
1	attaques	sci fi	2020-03-18	2020-03-19	paiement	Detail
2	attaques	sci fi	2020-03-17	2020-03-19	paiement	Detail

I faut cliquer sur détail pour voir les commentaires laisser sur cette exposition ou paiement pour prendre son billet

ici on arrive à la vue de paiement

MUSEE D'HISTOIRE NATURELLE D'ORLEANS

Accueil binette Deconnexion

Le billet vaut 25 euros

Credit Card #	Expiration Month	Expiration Year	CVC	Submit
---------------	------------------	-----------------	-----	--------

MUSEE D'ORLEANS

email bintadiagne93@gmail.com

OK

Accueil


binette Deconnexion

Paiement accepté. Merci pour votre achat.

Votre billet pour accéder au musée vous a été envoyé par mail.
 Vous devrez être en sa possession lors de votre visite.

Votre billet d'accès au musée


Boîte de réception x



testmiage45@gmail.com
Bonjour, vous trouverez en ci-joint votre billet.

16:46 (il y a 39 minutes)


☆



testmiage45@gmail.com
À moi ▾

17:24 (il y a 0 minute)

☆ ↶ ⋮



↶ Répondre

➡ Transférer

Conclusion

Pour conclure, ce projet nous a permis de mettre en pratique ce que nous avons pu voir en cours et en tp.

Au cours de ce projet, nous avons bien évidemment rencontrés quelques difficultés que nous avons pu résoudre. Ce projet nous a également apporté beaucoup de compétences.

Aux sujets de problèmes rencontrés nous en avons bien évidemment eu quelques uns :

Tout d'abord (qr code)

Il y avait également le lien entre le javafx et la jdk 11. En effet, la jdk 11 ne supporte plus les anciennes versions avec javafx. Nous avons donc ajouté des dépendance maven pour

un javaFx 11 mais il fallait également ajouté un fichier “module-info” dans le projet afin qu’il puisse fonctionner correctement.

La fonction d’authentification avec le spring Security et la jdk 11 n’était pas évidente à réaliser. Nous avons ([Resoudre l’authentification explication](#))
[zull avec springboot](#)

Enfin, ce projet nous a permis de savoir comment gérer des qr code avec une partie paiement ce qui était nouveau pour nous.

Ce que nous a apporté le projet:

Mettre en pratique ce que nous avons vu en CM/td/tp

Angular

gérer des qr code

partie paiement

Springsecurity

design avec jfx