

JEE Projet Médiathèque

Christopher Rousseau

22 décembre 2017

Contents

Médiathèque	2
Modalités	2
Authentification	3
Front	3
Partie administration	4
Partie utilisateur	4
Groupes de 3	5
Travail à rendre	5
Projet	6
Rapport	6

Ce sujet est susceptible d'évoluer durant quelques jours.

Médiathèque

Pour ce projet, vous réaliserez un ensemble de composants visant à aider à la mise en place d'un service Médiathèque / ludothèque, permettant à la fois au personnel de gérer les stocks, mais aussi aux utilisateurs d'accéder à des services mis en place par la médiathèque.

La médiathèque doit pouvoir gérer des ressources des types suivants :

- DVD
- CDs audio
- Jeux vidéos
- Jeux de sociétés
- Livres

Modalités

Afin de réaliser ce projet, vous aurez la possibilité de travailler par groupes de 2 à 3, une partie de ce projet sera donc optionnelle pour les plus petits groupes.

Le principal objectif du projet est de vous apprendre à utiliser et gérer plusieurs ressources / composants et de les interconnecter :

- Back office
- API
- SOAP
- Interface utilisateur
- Partie administration
- Services externes

Les technologies autorisées pour ce projet sont :

- Spring Boot (imposé pour la partie back office)
- Spark Java
- React.js

- vue.js
- Angular.js
- Angular

Elles seront imposées pour certaines parties et libres, sous réserve de justifications, pour d'autres.

Authentication

Afin de gérer vos utilisateurs, vous devrez créer un service d'authentification. Ce service devra gérer les rôles utilisateurs (utilisateur/administrateur)

Cette authentification comprendra :

- Une adresse email comme identifiant
- Un mot de passe avec une règle cohérente (affichée sur la page d'authentification)

Prenez garde de ne pas stocker les mots de passe en clair dans la base de données, et à vous protéger d'éventuelles injections SQL.

Le format de l'identifiant devra être une adresse email valide.

Vous pourrez vous aider de ce tutoriel spring : <https://spring.io/guides/gs/securing-web/>

Front

Dans cette partie, vous devrez choisir une technologie afin de réaliser l'interface utilisateur, vous aurez le choix entre :

- react.js
- angular
- vue.js
- angular.js

Vous aurez à justifier votre choix en comparant les différentes technologies.

Partie administration

La partie administration devra comprendre les fonctionnalités suivantes :

- Gestion des utilisateurs (Listing + CRUD)
- Gestion des stocks (Listing + CRUD)
- Listing des emprunts en cours avec couleur et warning si l'emprunt en cours est plus long que 7 jours
- Création et gestion d'annonces (Listing + CRUD avec un petit editeur WYSIWYG)
- Export (PDF ou json ou XML ou csv) listing utilisateurs
- Export (PDF ou json ou XML ou csv) stocks avec état
- Export (PDF ou json ou XML ou csv) ressources empruntées avec emprunteur

Le type d'export est au choix, vous pouvez en implémenter un ou plusieurs.

Un de vos partenaires n'utilise pas le même système d'information que vous, il a prévu une migration future, mais doit continuer de maintenir sa base utilisateur en attendant. Son système accepte d'importer des données en utilisant une requête SOAP, il vous faudra donc créer deux services SOAP afin d'exporter les données dont il a besoin.

- Listing des utilisateurs(non administrateurs) de la base
- Listing des ressources empruntées (non rendues) pour un utilisateur donné

Partie utilisateur

La partie utilisateur devra comporter les informations suivantes :

- une page d'accueil accessible en mode connecté et non-connecté
- Un formulaire d'authentification
- Accès at modification des informations de l'utilisateur connecté
- Accès aux informations sur les emprunts de l'utilisateur connecté

Pour la page d'accueil, vous rendrez disponibles quelques informations utiles :

- Un "widget" météo intégré sous forme d'iframe <http://www.infoclimat.fr/api-previsions-meteo.html?id=2988507&cntry=FR>
- Un encart presse (type d'implémentation : <http://www.lemurdelapresse.com/>)
- Les annonces de la médiathèque (créées dans la partie administration)

Groupes de 3

Cette partie est optionnelle pour les groupes de moins de 3 personnes.

Afin de simuler un service de commandes / achats fournisseurs, vous créerez une API REST avec SparkJava ainsi qu'une interface utilisateur (CRUD simplifié + listing) sans authentification. Ce service fonctionnera à la manière d'une centrale d'achat.

Vous utiliserez des données volatiles, vous ne stockerez donc rien en base de données, mais devrez veiller à la cohérence des données ainsi qu'à leur format.

L'API devra permettre d'ajouter des articles des mêmes types que l'application principale et de les exporter sous forme de listes au format JSON.

Vous devrez implémenter une méthode simple d'import de liste d'articles dans votre application principale afin d'ajouter les différents éléments à votre stock (via le fichier JSON généré). Cet import devra être implémenté dans la partie administration de votre application.

Il faudra donc penser à gérer le nombre d'article commandé.

Vous aurez également à livrer des tests, avec un rapport d'exécution. (couverture de tests, rapport HTML...) Les tests que vous aurez à implémenter seront :

- Tests unitaires
- Tests d'intégration
- Tests sélénium Webdriver

Voici quelques documents qui pourraient vous aider à réaliser ces derniers.

<http://www.seleniumhq.org/projects/webdriver/>

Pour une execution headless de vos tests sélénium :

<http://elementalselenium.com/tips/46-headless-ghostdriver>

<http://www.assertselenium.com/headless-testing/getting-started-with-ghostdriver>

<https://github.com/detro/ghostdriver>

Vous décrirez brièvement la stratégie de tests que vous avez mis en oeuvre lors du projet.

Travail à rendre

Le travail à rendre prendra la forme d'une archive au format **.zip** et comprendra votre projet ainsi qu'un rapport détaillés ci-dessous.

Vous enverrez votre projet au plus tard le Dimanche 28 Janvier à l'adresse suivante `c.rousseau@attineos.com`. N'oubliez pas de préciser la composition de votre groupe dans le corps de l'email.

Si la pièce jointe est trop volumineuse, pensez aux solutions d'hébergement de contenu temporaires.

Projet

L'archive devra comprendre les sources de votre projet, ainsi que un / des fichier(s) `.war` deployables sur un serveur web.

Rapport

Le rapport sera au format PDF et devra comprendre les informations suivantes :

- Vos interfaces de service (vous pourrez les décrire sous forme de tableaux,
- Un diagramme UML
- Un schéma relationnel
- La justification du choix des technologies pour la partie interface utilisateur (UI/UX)
- Le détail implémentation de l'authentification
- Un manuel partie Administration
- Un manuel utilisateur
- Une notice d'exécution / installation

Question Bonus (optionnelle) : Dans quel cas utiliseriez vous SOAP plutôt que REST lors du design d'une API et inversement ?