

Javascript : exercices séance 9->12

Les séances 9 à 12 sont consacrées à perfectionner votre apprentissage par la réalisation d'un projet.

Il est attendu de votre part que vous maîtrisiez :

- Javascript et l'organisation du code en pseudo-objets et pseudo-classes.
- JQuery.
- JSON et Genson.
- Le démarrage d'un serveur utilisant Jetty en mode embedded, configuré avec une ou plusieurs servlets.
- Les requêtes Ajax.
- La programmation d'une Servlet.
- L'utilisation de JWT pour signer l'authentification d'un utilisateur.
- Les cookies.
- HttpSession.

Ceci compose aussi la matière de l'examen.

De plus, vous pouvez profiter du projet pour découvrir Bootstrap, Select2, jQuery UI et jQuery datatables, mais ce n'est pas obligatoire.

Plusieurs sujets sont proposés, vous en choisissez un seul. Cependant, pour tous les projets, il faut :

- Permettre l'enregistrement et l'authentification des utilisateurs, en utilisant JWT. Utilisez un cookie pour retenir la chaîne JWT et les HttpSession pour éviter de valider la signature à chaque requête.
- Utiliser des pseudo-classes et des pseudo-objets en Javascript.
- Utiliser jQuery, en réutilisant vos fonctions form<->json quand cela s'y prête.
- Le serveur doit être en Java, utiliser Jetty en mode embedded et être pensé en stateless. Les données seront enregistrées dans des fichiers textes contenant du JSON.

Ce projet ne sera pas évalué. Vous pouvez donc explorer et prendre des risques sans que cela ne prête à réelle conséquence. Nous insistons une fois encore sur l'aspect auto-formation : l'examen pourra vous demander de faire n'importe quel point ci-dessus, et vu le temps pour le réaliser vous ne pourrez pas espérer vous auto-former sur le moment même.

Projet A : Dessinez-c'est gagné !

Votre application sera capable de :

- Enregistrer un nouvel utilisateur.
- Authentifier un utilisateur existant.
- Lorsque l'utilisateur est authentifié, il peut soit dessiner, soit jouer :
- S'il dessine :

- Une interface à la paint (en très simplifié) lui permet de faire un dessin. Chaque trait à une couleur. Chaque trait retient l'ensemble des points par lequel la souris passe. L'ordre des traits est retenu.
- Il peut enregistrer un ou plusieurs mots-clefs : c'est ce que le joueur doit deviner par le dessin.
- S'il joue :
 - Le back-end choisit un dessin au hasard.
 - Les traits sont dessinés dans leur ordre successif, avec un peu d'attente entre chaque trait (100ms p.ex.)
 - Il rentre dans un champ texte des propositions. Le serveur l'accepte si la proposition correspond à un des mots clefs enregistrés par le dessinateur.
- Imaginez un système de points : au plus vite c'est trouvé, au plus grand les points p.ex. Ajoutez une page 'Hiscore' listant les utilisateurs classés par leurs points du plus grand au plus petit.
- Faites preuve d'imagination pour améliorer tout cela.

Projet B : Chat

Votre application sera capable de :

- Enregistrer un nouvel utilisateur.
- Authentifier un utilisateur existant.
- Un utilisateur authentifié se voit proposer la liste des chatrooms en cours, et a aussi la possibilité d'en créer une nouvelle. Dans ce cas, il doit donner le nom de sa chatroom.
- Le serveur ne sera pas totalement stateless : il devra retenir les 10 derniers messages de chaque chatroom. Il n'y a pas de persistance, hormis les utilisateurs enregistrés.
- Quand un utilisateur joint une chatroom, il voit la liste des 10 derniers messages. Cette liste sera rafraîchie automatiquement toutes les secondes.
- L'utilisateur peut ajouter un message à la chatroom dans laquelle il est. Ce message est préfixé de son login : [machin] dit : bonjour à tous !
- Enrichissez votre interface, p.ex. envoi d'emoji, protection d'une chatroom pour un mot de passe, envoi d'un message privé, etc...
- Au niveau du serveur, chaque fois qu'un utilisateur demande le contenu d'un chatroom, reprenez le moment (new Timestamp()) où cela se passe. L'utilisateur fait donc partie de la liste des utilisateurs de ce chatroom, et l'instant de son dernier rafraîchissement est connu. Le serveur peut donc en profiter pour déterminer quels sont les utilisateurs qui ont leur effectué leur dernier rafraîchissement il y a plus de 10 secondes et les supprimer de la chatroom. De plus, il peut communiquer la liste des utilisateurs en plus des dix derniers messages.
- Faites preuve d'imagination pour améliorer tout cela.

Projet C : IMDB

Votre application sera capable de :

- Enregistrer un nouvel utilisateur.
- Authentifier un utilisateur existant.
- Un utilisateur authentifié se voit proposer la liste des films dans le système.

- Il peut en sélectionner un pour l'éditer, ou bien en ajouter un nouveau.
- Le formulaire d'édition d'un film contient des champs de titre, année de parution, synopsis, etc. Les données associées à des personnes sont gérées dans une interface séparée.
- L'interface de gestion des personnes permet : de sélectionner un ou plusieurs personnes, mais aussi de modifier les données associées (nom, prénom, date de naissance, bio etc).
- Ainsi pour spécifier le réalisateur d'un film, il faudra un bouton ; en cliquant sur le bouton, une boîte de dialogue s'ouvre et qui permet de sélectionner une personne parmi la liste des réalisateurs et des boutons permettant d'en créer un nouveau et/ou d'éditer sa fiche.
- La persistance se fait au niveau du serveur, un seul fichier pour tous les films et un seul pour toutes les personnes.
- Faites preuve d'imagination pour améliorer tout cela.

Projet D : ?

Votre application sera capable de :

- Enregistrer un nouvel utilisateur.
- Authentifier un utilisateur existant.
- La persistance se fera dans des fichiers JSON, gardez cela le plus simple possible.
- Vous décidez vous-même des fonctionnalités, en respectant toutes les contraintes de la première page. C'est sans doute une bonne idée de valider avec le prof que 1. vous ne passez pas à côté des objectifs d'apprentissage et 2. vous ne vous embarquez pas dans quelque chose de trop complexe ou irréalisable. Si vous avez une bonne idée, le prof se réserve le droit de vous l'emprunter pour les années prochaines !