

JavaScript - Semaine 5 /6 – Séance 1/3

Préambule

Durant cette semaine 5, nous allons nous consacrer à l'utilisation des sessions et de res.locals.

Consultez la théorie concernant les sessions pour les exercices de cette semaine.

Exercice 1 – Connexion Utilisation (Login)

Sur un site Web, on peut traditionnellement s'enregistrer et/ou se connecter pour avoir accès à plus de contenu et/ou effectuer des opérations autre que la simple consultation.

Commençons par réaliser la connexion (login) d'un utilisateur pour notre site des Exoplanètes.

Les étapes à réaliser sont :

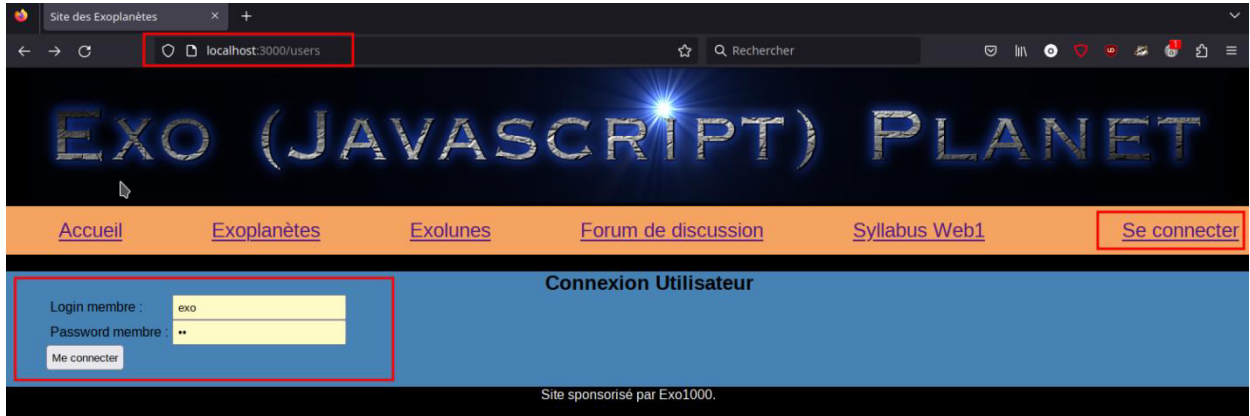
1. Ajouter un lien "se connecter" en haut à droite du site
 - a. Un petit float:right en CSS devrait vous aider à placer ce lien à droite
2. Ajouter un contrôleur users qui possédera les 2 routes suivantes :
 - a. / : afficher un formulaire login et mot de passe quand un utilisateur clique sur le lien "se connecter"
 - b. /login : vérifier que le mot de passe est bien "js"
 - i. si oui créer une variable de session contenant le login et renvoyer vers /members contenant simplement le texte "Bonjour <login>".
 - ii. sinon rediriger vers la page de login
 - iii. Si l'utilisateur est déjà connecté, redirigez-le directement vers /members
3. Créer un contrôleur members qui possédera la route
 - a. / : vérifier que la variable de session "login" est présente
 - i. si oui afficher la page d'accueil membre à savoir "Bonjour <login> ! "
 - ii. sinon renvoyer vers la page de login

Indications utiles :

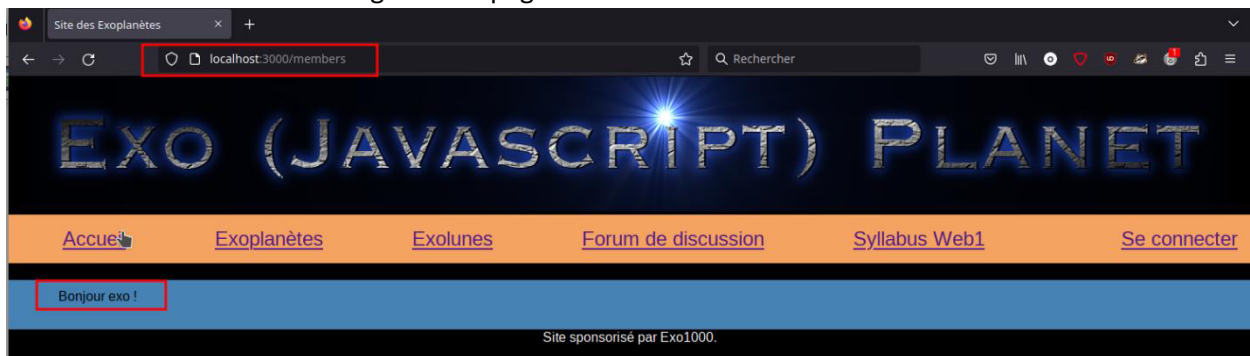
1. Il n'y a pour l'instant aucun lien pour cette partie avec la base de données. Le mot de passe est volontairement "hardcodé" pour se concentrer sur la nouvelle matière à savoir les sessions.
2. Il est nécessaire d'utiliser une variable de session pour cet exercice pour stocker le login de l'utilisateur

Aperçu :

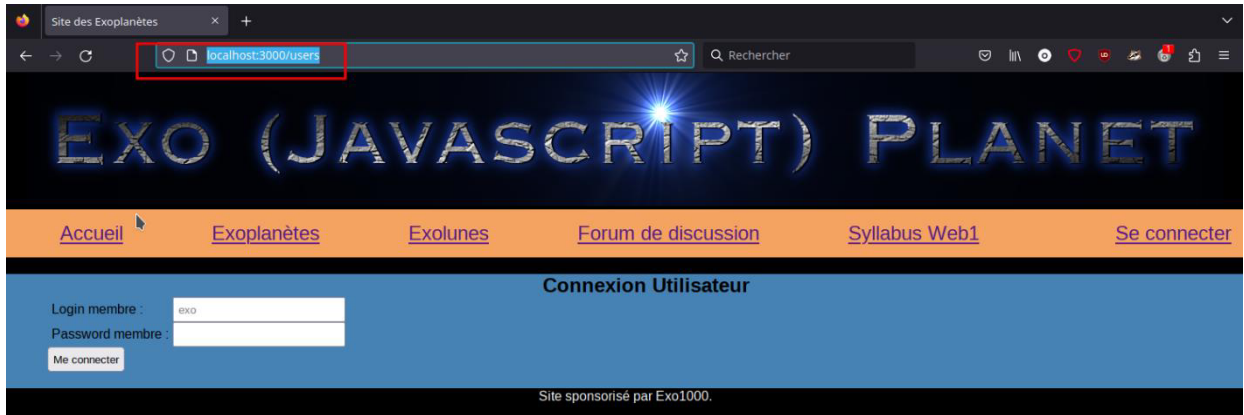
Je me connecte avec l'utilisateur "exo" et le mot de passe "js".



Je suis connecté et donc redirigé vers la page des membres.



Si j'entre un mot de passe autre que "js", je suis redirigé vers la page de login.



Exercice 2 – Session de test ou tests des sessions

Nous allons tester notre exercice 1 pour bien comprendre le mécanisme des sessions.

Réalisez les étapes suivantes dans l'ordre :

1. Connectez-vous avec un premier compte (login: exo1, mdp : js)

OK ? :

2. Vous êtes maintenant dans la page de l'espace membre (<http://localhost:3000/members> et le message "Bonjour exo1" apparaît

OK ? :

3. Via les “outils de développement Web” du navigateur, allez dans l’onglet “Stockage”, vous voyez un cookie.

OK ? :

Que représente ce cookie ? Une idée de sa valeur ?

4. Copiez l’URL de l’espace membre (<http://localhost:3000/members>) dans un nouvel onglet de votre navigateur. Vous voyez “Bonjour exo1”.

OK ? :

Pq le site ne vous renvoie pas vers la page de login ?

5. Copiez l’URL de l’espace membre (<http://localhost:3000/members>) dans un nouvel onglet **d’un autre navigateur**.

Que se passe-t-il ? Pq ?

6. J'arrête le serveur node (CTRL-C) et le relance (npm start)
7. Je rafraichis la page membre dans le premier navigateur (où j’avais le message “Bonjour exo1”)

Que se passe-t-il ? Pq ?

Exercice 3 – Déconnexion Utilisateur

A peine connecté, on va déjà tenter de déconnecter notre utilisateur !

Les étapes à réaliser sont :

1. Dans la page d'accueil membre(vue), ajoutez un formulaire HTML avec un simple bouton afin de se déconnecter.
2. Le contrôleur gérant le logout détruira la session et redirigera vers la page de login
3. Vérifiez bien que l'utilisateur ne sait plus accéder à l'espace membre en étant déconnecté (notamment en copiant/collant l'URL de l'espace membre dans un nouvel onglet).

Exercice 4 – Inscription Utilisateur

Essayons de réaliser l'inscription d'un utilisateur pour le site des Exoplanètes.

Les étapes à réaliser :

1. Ajoutez un lien "s'enregistrer" en haut à droite du site renvoyant vers /users/register
2. Le lien "s'enregistrer" affichera un formulaire demandant les informations suivantes :
 - a. Nom
 - b. Prénom
 - c. Email
 - d. Mot de passe
 - e. Confirmation de mot de passe
3. Ajoutez une table "users" dans la base de données avec Datagrip pour pouvoir persister les données du compte.
 - a. user_id : INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT
 - b. name : VARCHAR(100) NOT NULL
 - c. firstname : VARCHAR(100) NOT NULL
 - d. email : VARCHAR(100) NOT NULL UNIQUE (*)
 - e. password : VARCHAR(60) NOT NULL
4. Le formulaire renverra vers /users/add via la méthode POST (car envoi de mot de passe de l'utilisateur).
5. La route /users/add fera dans un premier temps un simple ajout (INSERT) dans la base de données. Les données, étant gérées par le modèle, vous créerez une fonction d'ajout d'un membre dans un **nouveau modèle User.js** que vous appellerez dans la route /users/add ! Une fois l'ajout effectué, vous pouvez renvoyer l'utilisateur sur la page de login.

* : le mot clé UNIQUE provoquera une vérification de la part de la base de données afin que chaque email soit unique -> impossible d'avoir 2 emails identiques pour ce champ.

Remarques :

1. **Ne faites encore aucune vérification (doublons, confirmation mot de passe, email déjà existant, ...), ni cryptage de mot de passe pour cet exercice. Essayez simplement de réaliser l'ajout d'un membre !**