



# W1 - Javascript Avancé

---

W-JSC-502

## Piscine MERN J04

---

React partie 1

3.0



# Piscine MERN J04

repository name: Piscine\_MERN\_Jour\_04  
repository rights: ramassage-tek  
language: Node.js



- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).

## COMPÉTENCES à ACQUÉRIR

- Javascript
- Node.js
- MongoDB
- Express.js
- React.js



## DÉTAILS ADMINISTRATIFS

- Sauf indication contraire, chaque exercice doit être rendu sous la forme `ex_numéro_exercice.js` à la racine du répertoire de rendu (par exemple : `ex_42.js` pour l'exercice numéro 42).

## INTRODUCTION

Pour cette quatrième journée, nous allons travailler avec notre serveur Node.js, notre base de données MongoDB, le framework appelé [Express.js](#) et le framework [React.js](#).



Nous vous laissons un grand choix quant à l'implémentation des fonctionnalités demandées ; c'est à vous de décider et de défendre vos choix lors de la soutenance.

## RESTRICTIONS

« [Express.js](#) » sert d'API côté serveur pour :

1. Traiter les requêtes reçues
2. Communiquer avec la base de données MongoDB
3. Envoyer la réponse au client.

« [React.js](#) » représente le client, c'est avec ce framework que l'utilisateur interagit, et c'est celui-ci qui envoie les requêtes à l'API [Express.js](#).

L'application à réaliser aujourd'hui doit impérativement, pour toutes actions (requêtes), utiliser cette communication entre [React.js](#) et [Express.js](#).



L'API doit être [RESTful](#).



Tous les modèles doivent avoir les opérations [CRUD](#) (Create, Read, Update, Delete).



## EXERCICE 0 (0 POINT)

Se renseigner sur [React.js](#) et lire intégralement le sujet avant de commencer.

## EXERCICE 1 (0 POINT)

Vous devez installer via NPM les différents paquets nécessaires.



Vous devez rendre le fichier « package.json » à jour pour que votre correcteur puisse installer automatiquement les paquets que vous aurez utilisé.

## EXERCICE 2 (2 POINT)

Commencez par créer un espace membre.



Reprenez votre exercice de « connexion / inscription » du jour 03 de la piscine MERN.

Les champs de votre base de données décrivant un membre doivent être :

- Un id : de type « int ».
- Un login : de type « string », ayant une longueur comprise entre 5 et 20 caractères ; ce login doit être unique dans la collection.
- Un email : de type « string », qui doit être unique dans la collection et **avec validation par regex pour que l'email soit considéré comme valide.**
- Un password : de type « string » ; celui-ci doit être hashé avec sha1 dans la collection.
- Un type : de type booléen ; celui-ci détermine si le membre est administrateur ou non.

Pour l'inscription, l'utilisateur doit fournir un login, un email, un mot de passe et une confirmation de mot de passe.

Pour la connexion, l'utilisateur se connecte avec son login et son mot de passe.



## EXERCICE 3 (5 POINTS)

Votre espace membre devra donner accès à un blog.

Chaque membre qui va s'inscrire aura son blog accessible à l'adresse suivante : <http://localhost:4242/<login>> où `<login>` est le nom du membre.

Avec son propre blog, chaque membre doit avoir accès à un CRUD « billet » permettant de :

- Voir tous les billets de son blog
- Créer un nouveau billet avec :
  - Un titre
  - Le contenu du billet
- Voir le détail d'un billet
- Éditer un billet
- Supprimer un billet

De plus l'utilisateur devra pouvoir :

- Voir les commentaires associés à ses billets
- Supprimer les commentaires liés à ses billets



Attention, sur son propre blog, un utilisateur ne peut pas créer ou modifier de commentaires.

## EXERCICE 4 (6 POINTS)

Sur le blog des autres, chaque membre peut :

- Voir tous les billets du blog (sur la page d'accueil <http://localhost:4242/<login>/>)
- Voir le détail de chaque billet (titre, contenu, et commentaires)
- Créer des commentaires sur les billets

La route doit être la suivante pour voir les détails des billets : [http://localhost:4242/<login>/<id\\_billet>/](http://localhost:4242/<login>/<id_billet>/)



Une fois connecté, un membre ne doit pas être obligé de se reconnecter pour naviguer de son blog à ceux des autres.



## **EXERCICE 5 (1 POINT)**

---

Sur la page d'accueil : <http://localhost:4242>, tous les blogs existants doivent être listés.

## **BONUS (4 POINTS)**

---

Il doit être possible pour chaque membre, sur chaque blog, de pouvoir répondre aux commentaires, et de pouvoir répondre aux réponses des commentaires, et cela sans limite de niveaux ; c'est-à-dire que chaque réponse postée peut obtenir une ou plusieurs autres réponses.