

Projet MEAN (MongoDB, Express, Angular, Node.js) - Note de TP (25% du module)

Pierre Pompidor

Vous devez créer une application web permettant l'échange **gracieux de services et de biens**.
(Un peu à l'image du site <https://www.allovoisins.com>).

La création libre d'un compte - identification via une adresse email et un mot de passe - permet d'offrir le prêt d'un bien et/ou l'utilisation de ses compétences à d'autres membres du site. L'emprunt des biens (valué par exemple à sa valeur neuve divisé par x), et l'utilisation des compétences d'un membre, seront définis par demies journées (un service pourra par exemple être valué à 50 euros). A terme (c'est à dire après laps de temps à définir), le compte d'un membre pourra être désactivé (via une notification envoyée à l'administrateur), après avertissement, si le membre est trop "consommateur" et que ses biens et/ou ses compétences ne sont pas assez "consommés".

Fonctionnalités à implémenter côté serveur :

Fonctionnalités obligatoires :

- Création d'une base de données MongoDB comportant au moins sept collections :
 - une collection **Membres** : email, MDP, nom, prénom, rôle, ville, adresse, téléphone, ...
 - une collection **Biens** : idBien, nom, descriptif, lienPhoto, prixNeuf, ...
 - une collection **Services** : idService, descriptif, ...
 - une collection **Utilisations** : idUtilisation, email, bienOuService, idBienOuService, date (demi-j.), ...
 - une collection **Disponibilités** : idDisponibilité, bienOuService, idBienOuService, numSemaine, numJour
 - une collection **DescriptifBiens** : idBien, motClé
 - une collection **DescriptifServices** : idService, motClé

Attention, la mise en place d'un bonus entraînera sans doute la création d'autres collections et d'autres attributs. Dans un premier temps, les collections et leurs documents seront créés via des fichiers JSON importés par la commande *mongoimport*.

- Implémentation d'un serveur **Node.js** assurant les services web suivants :
 - **inscription d'un nouveau membre**
 - **création d'un bien**
 - **suppression d'un bien**
 - **emprunt d'un bien**
 - **création d'un service**
 - **suppression d'un service**
 - **utilisation d'un service**
 - **recherche par mots clefs et par dates d'un bien ou d'un service**
 - notification à l'administrateur des déséquilibres
(ce qui permettra à celui-ci d'envoyer des avertissements ou de désactiver un compte)

Attention il est interdit d'utiliser l'ODM *Mongoose*, sinon la partie serveur ne sera pas notée.

Fonctionnalités facultatives :

- **Gestion des agendas** de prêts de biens ou d'utilisation de compétences (services) :
l'utilisateur indique sur quelles demies journées (par simplicité) ses biens ou ses services sont disponibles (une sélection graphique serait bienvenue)
l'emprunt d'un bien ou l'utilisation d'un service est alors signalée sur l'agenda.

- Gestion de la géolocalisation :
la recherche affiche sur une carte des marqueurs correspondants aux utilisateurs les plus proches (notion de proximité à définir), offrant le bien ou le service attendu.

Fonctionnalités à implémenter côté client :

L'application **Angular** doit au moins comporter quatre modules gérant les fonctionnalités correspondantes :

- à l'administrateur (avec mot de passe prédéfini mais modifiable) ;
- aux membres de la communauté ;
- aux biens ;
- aux services.

et, via ceux-ci, gérer les interfaces (de sélection et d'affichage des résultats) correspondantes aux services gérés par le serveur Node.js.

Attention il est obligatoire de créer des modules et d'utiliser le gestionnaire de projets *Angular CLI*, sinon la partie client ne sera pas notée.

Méthodologie de travail et barème (non contractuel) :

- MongoDB : création initiales des sept collections : au maximum 2 points
- Node.js (au maximum 6.5 points) :
 - ~~service web d'inscription : 1 point~~
 - ~~service web création d'un bien : 0.5 point~~
 - ~~service web emprunt d'un bien : 0.5 point~~
 - ~~service web suppression d'un bien : 0.5 point~~
 - ~~service web création d'un service : 0.5 point~~
 - ~~service web utilisation d'un service : 0.5 point~~
 - ~~service web suppression d'un service : 0.5 point~~
 - service web de calcul des déséquilibres : 0.5 point
 - service(s) web de recherche (bien et service) pour toute combinaison de critères : 2 points
- Angular (au maximum 6.5 points) :
 - Formulaire d'inscription et mise en oeuvre de l'authentification : 1 point
 - Sélection des actions et affichage des résultats (biens et services) : $6 \times 0.5 = 3$ points
 - Formulaire de recherche et affichage des résultats : 1 point
 - Gestion des déséquilibres : 1 point
 - utilisation du routeur : 0.5 point
- Gestion des agendas (Mongo + Node + Angular) : jusqu'à 4 points
- Géolocalisation : jusqu'à 4 points

Vous aurez compris qu'un seul bonus suffit pour avoir une très bonne note.

Planning :

Chronologie prévisionnelle des 4 séances de cours/TP (je ferai un point d'avancement à la fin de chaque TP) :

- Séance 1 : création des collections de tests, mise en oeuvre de quelques services web (affichage du JSON de la liste des membres dans le navigateur, et de la sélection de biens ou de services suivant quelques critères)
- Séance 2 : mise en oeuvre des services web importants (au minimum inscription d'un membre, recherche de biens ou de services suivant plusieurs mots-clefs et une date (ou une période))
- Séance 3 : test des codes d'exemple d'Angular, affichage de la liste des membres, création d'un formulaire
- Séance 4 : implémentation des fonctionnalités importantes (inscription d'un nouveau membre, formulaire de recherche)

La gestion de l'authentification et des bonus seront a priori faites en dehors des séances de TP.

Le rendu des projets sera effectué lors de la dernière semaine de cours de décembre.