

Projet : Votre première application réelle Node.js sur Heroku !!!

1. Téléchargez le fichier [node-sample.zip](#). C'est une application Node.js client/serveur déjà fonctionnelle, permettant simplement la connexion d'utilisateurs et d'un administrateur, et leur gestion simple. Elle peut déjà s'exécuter en local, mais aussi sur Heroku, directement.

- La seule limitation est la **modification** de la base de données d'utilisateurs, qui ne sera pas effective sur Heroku : les modifications du fichier `database.json` qui la contient ne seront pas persistantes après les 30 minutes d'inactivité permises par la version gratuite des comptes Heroku. Vous pouvez utiliser, lors du déploiement sur Heroku de votre application, des services en ligne de gestion de fichiers JSON : [IBM Cloudant](#), [Amazon S3](#), ou une simple base de données [MongoDB](#) (déconseillé par simplicité).
- Il vous est **fortement** conseillé de commencer votre application en vous basant sur celle-ci, étant donné le peu de documentations à jour sur les modules `express` et `passport` pour la connexion client/serveur avec authentification, et leur communication via websockets. C'est la raison pour laquelle la plupart des offres d'emploi pour un expert Node.js sont bien plus rémunératrices que les autres.

2. Choisissez, par groupes de 3 ou 4 étudiants maximum, un projet entre (a), (b) ou (c).

a. Gestion de réservations sur un plan de salle d'un restaurant :

- L'application permet à tout employé de sélectionner pour un client donné : une date de réservation, une heure d'arrivée (parmi les services possibles dans la journée) et une table. Il en va de même pour les clients qui n'ont pas réservé, cela fonctionne de la même manière.
- Seul l'administrateur (les gérants) peut modifier le plan de salle et les heures possibles d'arrivée dans une journée.
- Bien entendu, si plusieurs employés ou gérants agissent en même temps sur les réservations, tout doit être dynamiquement mis à jour pour tout le monde qui est connecté, grâce aux websockets.
- Le système doit être suffisamment simple à utiliser pour permettre à un employé utilisant une tablette tactile d'effectuer une réservation sur le plan de salle.

b. Gestion de tchats publics :

- Chaque utilisateur connecté peut rejoindre un tchat existant et participer à la discussion publique.
- Seul l'administrateur peut créer ou supprimer des tchats.

- Une interface mobile serait appréciée pour pouvoir tchater sur smartphone via le navigateur (pas d'application statique à installer).

c. Gestion de documents collaboratifs :

- Chaque utilisateur peut modifier n'importe quelle partie d'un document collaboratif existant.
- Seul l'administrateur peut créer ou supprimer des documents collaboratifs.
- Un système de versioning (simple sauvegarde des documents à un chaque instant) serait un atout pour revenir à un état antérieur. Un visionnage par diff est un plus.