

Labo #3

API REST avec Node, Express, DAO, JSON et base de données MySQL

Reprendre votre Labo2.

Vous allez créer un **Api Rest** nommé **api_rest_permis**(le nom du projet).

Vous allez créer un **serveur Node avec express**, un **controleurPermis** et un **modelPermis** qui va s'occuper de la base de données (ce sera un DAO). Votre serveur de base de données sera **MySQL** la base de données se nommera **bdpermis** et votre table **permis**. La même que celle de votre labo2. Vous allez changer la route **app.all("/permis", ...** pour

```
app.get("/permis", ...  
app.post("/permis", ...  
app.put("/permis/:id", ...  
app.delete("/permis/:id", ...
```

Les réponses de l'API seront retournées au client en format JSON.

Permission d'accès à votre API

Pour pouvoir utiliser votre api les **membres** doivent créer un compte sur votre site (nom, prénom, courriel et mot de passe) et recevront une clé (générée par **sha1**) nommée **apiKey** ou **apiCle**. Cette clé vous allez l'envoyer comme réponse à son enregistrement. Lorsqu'un membre désire par exemple obtenir la liste de vos permis il va envoyer une requête par **get** de la manière suivante (directement via l'url du navigateur ou par une clique sur un bouton):

localhost :8282/permis?apiKey=la clé de l'utilisateur

Le port 8282 est ici comme titre d'exemple.

Lorsque la requête va tomber dans la route du serveur **app.get("/permis", ...** vous allez récupérer la donnée de **apiKey** et valider cette clé dans la table membres. Si la clé existe vous allez retourner la liste des permis, sinon vous allez retourner un message explicite pour que le membre vérifie sa clé ou qu'il crée un compte.

Chaque requête d'un membre doit être accompagnée de son **apiKey**.

À faire

Chaque requête sera envoyée au serveur et à la bonne route pour gérer cette requête. **NE PAS utiliser un tableau et après faire les requêtes au tableau.** Utilisez Bootstrap pour vos interfaces utilisateur.

Implémenter et tester les 4 routes suivantes :

```
app.get("/permis", ...  
app.post("/permis", ...  
app.put("/permis/:id", ...  
app.delete("/permis/:id", ...
```

REMISE

Le projet sera déposé dans LEA. Lors de la remise placez l'url de votre vidéo des tests ainsi que le nom des membres de votre équipe. Pour les tests suivre l'ordre décrite plus haut. Vous devez parler dans votre vidéo sans à voir à montrer votre code (sauf si celui-ci ne fonctionne pas), seulement les tests.

ÉVALUATION

Toute consigne non respectée vous fera perdre 10% de la note finale.

Les 4 routes complètement fonctionnelles **80%** (20% par route, si non fonctionnelle 10%)

La gestion des permissions via apiKey **15%**

Ergonomie de votre api **5%**

Attention à votre vidéo de tests. Vous devez illustrer l'utilisation des 4 routes.