Organisation du code :

Le projet comportera deux applications. La première sera dédiée au client et comportera une interface graphique (HTML, CSS, …). Celle-ci communiquera à la deuxième application grâce à des requêtes Ajax. La deuxième application agit comme le back-end de notre application, elle recevra des tâches à effectuer venant du client (c’est-à-dire, l’administrateur système dans le cas de notre projet) utilisant la première application. Ce back-end effectuera alors les tâches qui lui ont été confiées et fera des requêtes à la base de données PostgreSQL si nécessaire.

Voici un schéma représentant ces 2 applications :

C:\Users\Corenthin\Downloads\Untitled Diagram (2).png

Organisation des modules Node.js :

C:\Users\Corenthin\Downloads\module.png

C:\Users\Corenthin\Downloads\légende.png

*Italique* = librairie / module externe

Le back-end Node.js comportera une structure précise : l’app.js sera la pièce centrale du back-end. Il se chargera d’initier en premier la connexion avec la base de données PostgreSQL. Pour cela, il utilisera database.js qui sera un fichier de configuration, celui-ci possède la configuration nécessaire pour se connecter à la base de donnée. Le but étant de permettre la modification des informations de la base de donnée, si besoin est, et sans devoir modifier app.js. Pour permettre une connexion PostgreSQL, on utilisera la librairie externe « pg ». En plus de cela, app.js fera appel à des fichiers modèle pour le Json. À titre d’exemple, on utilisera le modèle user.js qui définira quelles sont les informations d’un utilisateur.

Enfin, routes.js contiendra les différentes méthodes qui seront exécutées en fonction de la requête Ajax envoyée par l’application cliente.