

Rattrapage crédits étrangers

Projet : TodoList

1. Backend

J'ai choisi de développer le backend de mon application en Javascript, car je connais déjà bien ce langage et il répondait aux besoins du projet.

J'ai utilisé NodeJS pour le Javascript côté serveur et la librairie Express qui fournit les blocs élémentaires nécessaires à la réalisation d'un backend : routage, récupération des headers HTTP ou des paramètres GET ou POST via des middlewares.

Pour la base de données, j'ai choisi PostgreSQL car c'est une base de données que j'ai appris à utiliser dans les derniers mois et la popularité de cette dernière est croissante. On utilisera le module npm 'pg' afin d'assurer la communication backend - base de données.

J'ai commencé par faire un diagramme UML des éléments nécessaires à l'application : cela m'a permis ensuite de construire le schéma de la base de données et les propriétés des modèles du backend.

Concernant l'authentification des utilisateurs j'ai opté pour un modèle basé sur des access token et refresh token : l'utilisateur se connecte une première fois avec ces identifiants ce qui lui génère un couple access_token / refresh_token. L'access token sera passé dans le header HTTP 'x-todo-access-token' et permettra au backend de reconnaître l'utilisateur à l'origine de chaque appel. L'access token est stocké dans le cache du serveur (utilisation de la librairie node-cache), quand le serveur s'arrête, il n'est donc plus valide.

Le refresh token permet d'obtenir un nouvel access token pour l'utilisateur.

Ce mécanisme permet d'identifier l'utilisateur à chaque fois qu'il relance l'app sans taper ses identifiants mais par manque de temps je n'ai pas eu le temps d'implémenter cette feature dans le client.

J'ai donc créé les modèles suivants : user, todolist, task, access_token, refresh_token et leur méthode associée : get, create, update, delete .

Ensuite, j'ai créé des routes pour utilisant ces modèles et dont le but est de s'assurer de la validité des paramètres de la requête, appeler la méthode du modèle concernée et renvoyer le résultat. J'ai choisi de ne pas prendre en compte la sécurité et donc il y'a des failles facilement corrigées mais j'ai préféré me concentrer sur les fonctionnalités.

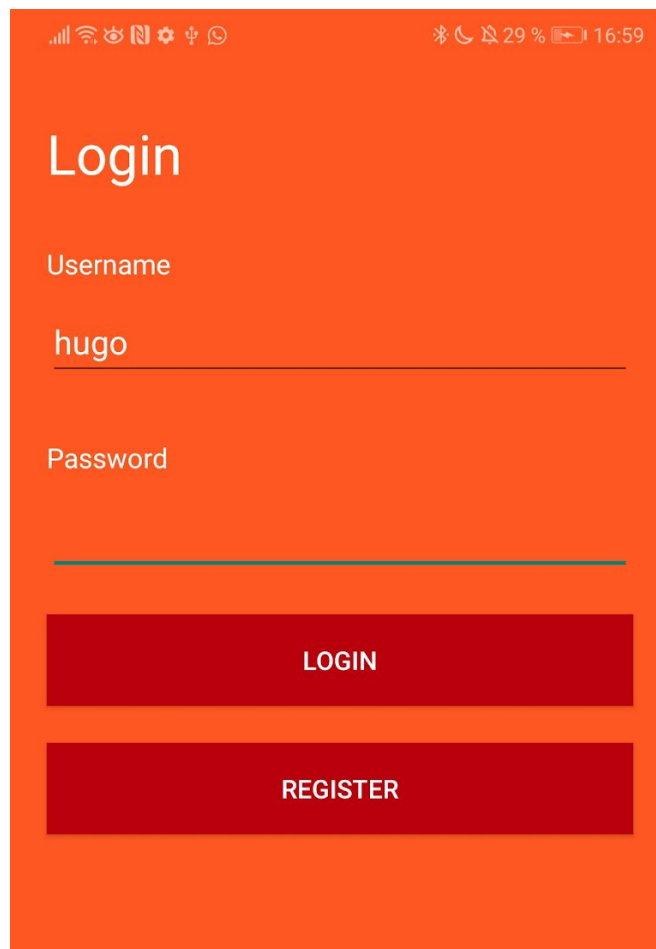
L'API renvoie des résultats au format json avec un clef "status" qui indique la réussite ou non de l'opération.

2. Application Android

Je n'avais jamais fait d'application Android avant ce projet. J'ai donc commencé par suivre le MOOC afin de me familiariser avec l'environnement, les librairies, la façon de faire et les objets propres à Android. J'ai choisi de faire une application qui communique en temps réel avec sa base de données distante. Donc, à chaque opération sur l'application, un appel réseaux est effectué. Ce n'est pas le plus optimal, car cela entraîne une latence et rend l'application inutilisable hors ligne. Il faudrait développer une base de données (par exemple SQLite) qui vient se synchroniser au serveur à des moments ponctuels pour une expérience utilisateur optimale.

Pour communiquer avec mon API Rest, j'ai utilisé la librairie "Retrofit" qui permet de simplifier la récupération des données et génère des objets Java à partir des réponses du serveur.

L'application est très simple, la page d'accueil permet de se connecter ou de créer un compte.



hugo

LOGIN

REGISTER



29 % 16:59

Register

Username

Password

Email

REGISTER

Ensuite, la page “principale” qui permet de cliquer sur une todolist afin de la modifier.

F SFR 40 % 18:30

test

Hdudbdv

Create Todolist

CREATE

La vue détaillée des todolist, on peut filtrer par task “done” ou “non, ajouter ou supprimer une task.

F SFR 32 % 17:16

Hdudbdv

☐ Hide done

Bevueeue DELETE DONE

Bevueeuevbivivdis DELETE DONE

Bevueeuevbivivdisbsv DELETE DONE

Add task

ADD TASK

3. Conclusion

J'ai apprécié faire ce projet, car il m'a permis d'avoir un aperçu de la programmation mobile Android. Je n'envisage pas aujourd'hui de faire carrière comme développeur mobile je suis certain que cela me sera tout de même utile professionnellement. En effet, l'entreprise dans laquelle je travaille (Withings) possède une grande équipe de développeurs mobile avec qui je suis amené à collaborer régulièrement. J'ai envie maintenant de découvrir la programmation iOS pour voir les différences entre les deux plateformes.