



Récupérer dans l'app la moyenne et le max calculés côté serveur

1. On swipe le navigation drawer pour choisir d'aller vers le statfragment
2. Avant de s'afficher, celui-ci lance statvaluesupdater...
3. ...qui lance une AsyncTask ...
 1. Se connecte au serveur via l'api REST et le nouveau webservice "stat"
 2. <http://172.20.1.209:8081/myconsumption/stat/mean/5429817ae4b025d69733f9f8>
 3. <http://172.20.1.209:8081/myconsumption/stat/max/5429817ae4b025d69733f9f8>
 4. Récupère les valeurs et les ajoute dans la db locale sqllite de l'appli Android
4. ... Une fois terminé, un callback indique au statfragment que les valeurs sont disponibles dans la db locale de l'appli
5. On les affiche

Problèmes :

1. Ne vérifie pas si une valeur existe déjà dans la db locale Android -> ce n'est pas une "sync" mais juste un "get" => offline mode not supported
2. Les stats ne sont pas stockées dans une db côté serveur mais calculée en live -> mongodb ?

Ce que je projette de faire :

- Fixer un cahier des charges du TFE pour voir si ça correspond aux attentes de Pr Guy Leduc
- Consommation moyenne : offline mode & mongodb

Questions :

- What's next?
- Utilisation de la lib Android Support sans passer par *Maven*
- Pas moyen de faire fonctionner le retrieveur chez moi. Properties.xml (my-consumption business/resources) :
 - `<!--<entry key="mongo_host">mxcsrv20.manex.biz</entry>-->`
 - `<entry key="mongo_host">127.0.0.1</entry>`
 - `<entry key="mongo_port">27017</entry>`
 - `<entry key="mongo_db">starfishr_myconsumption</entry>`
- Accès au git de l'extérieur ?
- Gestion des exceptions, quelle est la "norme" en java ? Parce que l'app de Patrick peut planter complètement assez facilement
 - Il fait souvent des try catch avec catch vides
 - Il ne fait jamais de throw new exception machin chose
 - Il n'a pas créé d'exceptions propres à son programme
- Bugs : utiliser les "issues" de gitlab ?
- **Fixer les rendez-vous des prochaines réunions**