Auteur : Spinelli Isaia et

Simonet Yoann

Prof : [Dutoit Fabien](https://mse.hes-so.ch/consultation/horaires/?annee=2019&trimestre=1&id=8329&type=1)

Assist. : Christophe Greppin

Date : 11.10.2019

Classe : SYM-A

Laboratoire 2 : Protocoles applicatifs

Systèmes mobiles

Table des matières

[Introduction - 2 -](#_Toc21169614)

[Réponse aux questions - 2 -](#_Toc21169615)

[Conclusion - 6 -](#_Toc21169616)

[Difficultés rencontrées - 6 -](#_Toc21169617)

[Compétences acquises - 6 -](#_Toc21169618)

[Résultats obtenus - 6 -](#_Toc21169619)

## Introduction

## Manipulation

Dans le cadre de cette manipulation vous pouvez rester sur une solution simple « en mémoire », vous indiquerez toutefois dans votre rapport les limitations de cette façon de faire et proposerez des outils et techniques mieux adaptés (sans forcément réaliser l’implémentation). (page 8)

### Activité Asynchrone

### Activité Différée

Pour cette activité, il a fallu réaliser des requêtes différées. En l'absence de connexion avec le serveur, l'application fonctionne normalement, sans que l'utilisateur n'éprouve une gêne quelconque. Dès que la connexion avec le serveur est rétablie, les informations qui avaient été fournies par l'utilisateur sont transmises au serveur.

Pour faire ceci, lors d’une demande d’envoi d’une requête nous créons un thread si ce n’est pas déjà fait. Celui-ci checkera toutes les 5 secondes s’il y a une connexion à internet. Quand la connexion sera bien présente, il enverra toutes les requêtes précédemment envoyées. Donc quand il y aucune demande de requête, le thread ne prend aucune ressource CPU.

Afin de tester cette activité, nous avons coupé notre wifi et envoyé plusieurs requêtes. Ensuite, nous avons réactivé le wifi pour voir toutes les réponses des requêtes dans les LOGs.

Cette méthode pourrait être problématique si le réseau est stable et qu’on modifie plusieurs fois le même élément car l’ordre d’arrivée des requêtes n’est pas garanti.

On pourrait utiliser le multiplexage de toutes les connexions vers un même serveur en une seule connexion. Plus d’information sur le principe ici : <https://connect.ed-diamond.com/GNU-Linux-Magazine/GLMF-115/Multiplexage-des-connexions-SSH>.

### Activité Sérialisation

### Activité Compressé

### Activité GraphQL

## Questions

### Traitement des erreurs

### Authentification

### Threads concurrents

### Ecriture différée

### Transmission d’objets

### Transmission compressée

## Conclusion

### Difficultés rencontrées

Lecture de la donnée.

### Compétences acquises

### Résultats obtenus

Date : 10.11.19

Nom de l’étudiant : Spinelli Isaia et Simonet Yoann