
TP 01

Délai de reddition : deux semaines (le 21.03.2022 à 23h59 max)

Exercice 1 - Callback

Le but de cet exercice est d'implémenter le pattern : Request/Acknowledge/**Callback**

Dans cet exercice nous supposons que le client fait appel au web service pour une requête qui demande un très long calcul.

À implémenter

Implémenter la classe **ClientMain** qui :

- Envoie une requête GET au service avec l'url :
 - Domain/Application/**service/calcul**
- Dans le Header «Location » de la requête, le client met l'URL du service qui recevra la réponse.

Implémenter le web service « **Service** » qui :

- Reçoit la requête de la part du client et retourne un status « 303 », ensuite il lance un Thread Callback en lui passant l'url (se trouvant dans le header « Location ») du service qui recevra la réponse :
 - `new Callback(url).start();`

Implémenter la classe **Callback** qui :

- Envoie un POST au web service ServiceCallback lorsque la réponse est prête.
 - Pour simplifier, la payload du POST contiendra le message « la réponse envoyée » que le ServiceCallback affichera dans la console.

Fournis

Le web service **ServiceCallback** avec l'url :

- Domain/Application/**serviceCallback/resultat**

Qui reçoit la réponse lorsqu'elle est prête et l'affiche dans la console.

Exercise 2 - Poll

Dans cet exercice il faut implémenter le pattern : Request/Acknowledge/**Poll**

Nous supposons que le client fait appel au web service pour une requête qui ne peut pas être résolue immédiatement.

À implémenter

La classe **ClientMain** qui doit :

- Envoyer une requête au service1 avec l'url :
 - Domain/Application/**service1/number**
- Vérifier le status de la réponse.
- Récupérer l'url du service qui va répondre (service2) dans le header « Location ».
- Rediriger la requête vers service2.
- Vérifier le status de la réponse du service2, et répéter la requête tant que la réponse n'est pas encore prête.

Le web service **Service1** est censé répondre à une requête qui demande un certain temps de calcul, pour cela, il lance un `CalculeThread` qui va mettre à jour la « `DataBase` » une fois le calcul terminé :

- Lance le Thread new `CalculeThread().start`
- Retourner le status `accepted` pour informer le client que la requête est prise en compte, mais qu'il faudrait solliciter le `service2`, pour cela il faudrait retourner l'url du `service2` qui va prendre en charge la requête.

Le web service **Service2** :

- Vérifier si la réponse est prête : `new DataBase().reponsePrete()`
- Retourner la réponse lorsqu'elle est prête.

Fournis

DataBase une classe qui implémente la méthode `reponsePrete()` qui retourne un boolean lorsque la requête est prête, dans cet exercice elle retourne **true** après un nombre définit de requêtes.

Exemple de résultat :

Requête	pas	encore	prête,	veuillez	patienter.....
Requête	pas	encore	prête,	veuillez	patienter.....
Requête	pas	encore	prête,	veuillez	patienter.....
Requête	pas	encore	prête,	veuillez	patienter.....
Requête	pas	encore	prête,	veuillez	patienter.....
Requête	pas	encore	prête,	veuillez	patienter.....
Requête	pas	encore	prête,	veuillez	patienter.....
Requête	pas	encore	prête,	veuillez	patienter.....
Requête	pas	encore	prête,	veuillez	patienter.....
Requête	pas	encore	prête,	veuillez	patienter.....
Requête	pas	encore	prête,	veuillez	patienter.....
Requête	pas	encore	prête,	veuillez	patienter.....
Requête	pas	encore	prête,	veuillez	patienter.....
Requête	pas	encore	prête,	veuillez	patienter.....
La requête	est	prête	après	16	Poll

Bon travail !

Questions : Assistant Baptiste Bartolomei – B4.24
Baptiste.bartolomei@hesge.ch