

SciQLOP - Task #2115

User story # 2112 (Closed): Taches provenant du sprint 8

Analyse bug de mise à jour des requetes

27/09/2017 11:23 AM - Anonymous

Status: Closed	Start date: 29/09/2017
Priority: Normal	Due date:
Assignee:	% Done: 0%
Category:	Estimated time: 0.00 hour
Target version:	Spent time: 0.00 hour
revision: r0	blocked: No
Description	
Lorsque qu'une requête est terminée, la requetes suivante s'exécute mais sur un à priori d'état qui n'est plus le bon. Il faudrait que la prochaine requete se mette à jour en fonction des nouveaux éléments de la première requêtes exécutés	

History

#1 - 29/09/2017 10:09 AM - Anonymous

- Status changed from New to In Progress
- Assignee set to Anonymous
- Start date set to 29/09/2017

#2 - 29/09/2017 11:04 AM - Anonymous

concernant l'analyse du bug de la mise à jour de la requete (edited)

[10:09]

je rappel, on lance n requetes sur une variable

[10:10]

les n requetes ont un variable range et un variable cache range final basé sur l'état initial de la variable

[10:10]

le problème est que si une requete intermédiaire a fonctionné (edited)

[10:10]

du fait qu'on modifie alors l'état de la variable

[10:11]

la requete suivante n'est plus en phase avec ce nouvel état de la variable

[10:11]

mais avec l'ancien

[10:11]

bref, ça crée des trou dans les données et donc dans la visu aussi (edited)

[10:11]

l'idée évoque en réunion

[10:11]

était de mettre à jour ces requetes

[10:11]

dès que la précédente a fonctionné. (edited)

[10:12]

je t'avais dis que j'en ferai l'analyse puis que je reviendrais vers toi

[10:12]

et donc, c'est fait

[10:12]

la réponse est que ce n'est pas possible

[10:12]
car en fait ya deux niveaux de gestion des requetes

[10:12]
ce problème ce voit au niveau du VC

[10:13]
mais il existe aussi au niveau du worker

[10:13]
donc en mettant à jour la request du VC

[10:13]
il faudrait que je notifie aussi le worker de se mettre à jour

[10:13]
sauf que lui à peut être déjà démarré quelques requetes

[10:13]
sur l'état suivant

[10:14]
De plus, je ne suis pas trop fan de la proposition évoqué en réunion

[10:15]
car elle a le défaut d'assumer qu'une requete est fausse si la précédente fonctionne

[10:15]
c'est à dire, qu'on a une API qui considère que l'échec est plus probable que le succès

[10:15]
Je propose donc la solution suivante

[10:15]
qui est très proche

[10:15]
au lieu de baser la nouvelle requete (edited)

[10:16]
sur l'état initiale de la variable

[10:16]
on la base sur l'état final de la variable en considérant que la requete précédente a fonctionné (edited)

[10:16]
ce qui permet un succès sans trop rien faire tant que les requetes fonctionnent

[10:17]
et donc dans le cas ou une requetes échou

[10:17]
c'est là, qu'on applique un traitement différent

[10:18]
par exemple si la requetes n échoue, on peut annuler toutes les requetes > n. Puis éventuellement relancer les n requetes suivantes avec des paramètres corrigés (edited)

[10:18]
en se basant sur n-1 (l'état initial de la variable)

[10:19]
le principe principale étant que la requetes n+1 se base sur la n au lieu de se baser sur l'état initial de la variable (edited)

#3 - 09/10/2017 11:37 PM - Anonymus

- Status changed from In Progress to Closed