

Backlog RM - Anomalie #16838

[Ecran principal] Temps de chargement de dossiers virtuels

12/04/2021 10:46 - Cyril VAZQUEZ

Statut:	R&D - Terminé	Début:	12/04/2021
Priorité:	2-Sérieux	Echéance:	
Assigné à:			
Catégorie:			
Version cible:	2.7		
Tags RM:	2.7.3		

Description

Lorsqu'une activité comporte beaucoup de dossiers virtuels (+10000) le chargement de l'arborescence est très long.

Analyse

Affichage par presenter recordsManagement/welcome/welcomePage

Voir délai pour l'appel au service 'filePlan/filePlan/readTree'

Voir délai de construction côté presenter html

Voir délai de construction du html par le JS

Selon les résultats:

* ne charger en back que le premier niveau et ajouter des routes pour charger le contenu d'un répertoire lorsque l'utilisateur l'ouvre

* charger toutes les niveaux mais seulement construire le html avec le JS lorsque l'utilisateur l'ouvre

Historique

#1 - 14/04/2021 11:29 - Cyril VAZQUEZ

Pb lié à la traduction:

Quand l'instruction de fusion a pour cible non pas un noeud déjà présent mais un fragment, il n'est pas possible de traduire avant fusion. La traduction a donc été faite après, mais sur la répétition de milliers d'éléments cela pose problème:

```
<?merge data /chemin/template.html ?>
```

Comme template est inclus au doc au moment de la fusion, il ne peut être traduit avant fusion.

Solution:

Nouvelle fonctions dans le système de templating pour ajouter des fragments au document sans les inclure dans l'arbre DOM

Modification de la fonction de traduction pour permettre de traduire ces fragments avant fusion

Modification du système de fusion pour utiliser en priorité les fragments déjà présents au lieu de les charger des fichiers à partir de l'URI précisée dans l'instruction de fusion

résultat = lorsque la fusion d'un tableau utilise une cible qui est un fragment, le fragment déjà présent et traduit est utilisé

#2 - 14/04/2021 12:25 - Cyril VAZQUEZ

Second problème lié à la récursivité lors de la construction de l'arbre.

La lecture en DB renvoie un tableau de dossiers.

Le contrôleur fait appel à `laabs::buildTree` pour construire l'arbre.

`laabs::buildTree` fait appel à `laabs::buildBranch` récursivement

Création d'un `buildTree2` qui n'est pas récursif mais utilise un nouveau tableau indexé par id d'objet

#3 - 16/04/2021 11:55 - Alexandre GOLDSTEIN

- Fichier Screenshot from 2021-04-16 11-50-24.png ajouté

Tests réalisés à ce jour avec 10000 dossiers : on a 5 sec de délai sur la branche contenant la correction, le spinner ne s'arrete pas sur la branche develop (les dossiers ne sont jamais affichés)

Tests également fait avec 2000 dossiers : on a 2,22 sec de délai d'affichage sur la branche corrective, et 6.02 sec en comparaison sur la branche develop

#4 - 16/04/2021 14:46 - Alexandre GOLDSTEIN

- Statut changé de A traiter à A livrer

#5 - 19/04/2021 15:10 - Cyril VAZQUEZ

- Version cible changé de 2.7 à 2.7.3

#6 - 22/04/2021 12:15 - Arnaud PAUGET

- Statut changé de A livrer à Livré

#7 - 09/06/2021 14:45 - Emmanuel DILLARD

- Projet changé de Maarch RM - Product Backlog à Backlog RM

- Version cible changé de 2.7.3 à 2.7

- Tags RM 2.7.3 ajouté

#8 - 03/08/2021 16:48 - Emmanuel DILLARD

- Statut changé de Livré à R&D - Terminé

Fichiers

Screenshot from 2021-04-16 11-50-24.png	3,3 ko	16/04/2021	Alexandre GOLDSTEIN
---	--------	------------	---------------------