w.c.s. - Development #58276

Limiter au maximum le chevauchement d'une soumission et d'un autosave au niveau Javascript

28 octobre 2021 18:41 - Benjamin Dauvergne

Statut:	Fermé	**Début:**	28 octobre 2021
Priorité:	Normal	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

J'ouvre un ticket pour la partie JS proposée dans #58208, pour ne pas mélanger.

Révisions associées

Révision 53c6f2d5 - 28 décembre 2021 13:31 - Benjamin Dauvergne

js: prevent autosave while user is active (#58276)

Révision a1a66e78 - 28 décembre 2021 13:31 - Benjamin Dauvergne

js: launch autosave on tab change (#58276)

Révision 72fa3ca7 - 28 décembre 2021 13:31 - Benjamin Dauvergne

js: delay submit until autosave ajax request is finished (#58276)

Révision 7a13c6eb - 28 décembre 2021 13:31 - Benjamin Dauvergne

forms: abort autosave after 200 ms (#58276)

Historique

#1 - 28 octobre 2021 18:47 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#2 - 28 octobre 2021 18:47 - Benjamin Dauvergne

- Fichier 0003-js-disable-submit-buttons-during-autosave-58276.patch ajouté

- Fichier 0004-js-launch-autosave-on-tab-change-58276.patch ajouté

- Fichier 0001-js-prevent-autosave-while-user-is-active-58276.patch ajouté

- Fichier 0002-js-abort-current-autosave-ajax-request-on-submit-582.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

#3 - 28 octobre 2021 18:52 - Benjamin Dauvergne

Plusieurs choses à prendre ou laisser :

- rallonger le délais avant autosave à 8 secondes et le débouncer chaque fois qu'un évènement clavier,souris ou écran tactile arrive, pour éviter de faire un autosave pendant la saisie,
- annuler la requête AJAX en cours sur un submit (pas super efficace, la requête ayant déjà pu être reçue et donc finira d'être traitée par w.c.s.)
- mieux mais plus impactant au nieau fonctionnel (et rendant le point précédant inutile), désactiver les boutons de soumission pendant l'autosave et afficher un message ("en cours de sauvegarde", à localiser)
- lancer l'autosave sur un changement d'onglet (rare évènement nous disant que la saisie par l'usager est en pause)

#5 - 28 octobre 2021 20:34 - Frédéric Péters

> mieux mais plus impactant au niveau fonctionnel (et rendant le point précédant inutile), désactiver les boutons de soumission pendant l'autosave et afficher un message ("en cours de sauvegarde", à localiser)

non pour ça, sûr.

#6 - 28 octobre 2021 23:46 - Benjamin Dauvergne

- Fichier 0004-js-delay-submit-until-autosave-ajax-request-is-finis.patch ajouté

- Fichier 0001-js-prevent-autosave-while-user-is-active-58276.patch ajouté

- Fichier 0003-js-launch-autosave-on-tab-change-58276.patch ajouté

- Fichier 0002-js-abort-current-autosave-ajax-request-on-submit-582.patch ajouté

Ok, alors mieux certainement: capter les clicks sur les boutons submit, si un autosave est en cours on conserve la référence au bouton et on bloque le submit ainsi que les autosave. Quand l'appel autosave se conclut on reclique sur le bouton et le submit se fait (j'aurai bien vu un overlay semi-transaprent disant que la soumission est en cours, mais c'est un goût personnel).

#9 - 30 octobre 2021 15:43 - Frédéric Péters

Ok 0001, 0002 (même si je me dis sans vérifier que ça ne va pas nécessairement avoir un effet côté traitement de la requête sur le serveur), 0003.

Sur 0004,

> Ok, alors mieux certainement: capter les clicks sur les boutons submit, si un autosave est en cours on conserve la référence au bouton et on bloque le submit ainsi que les autosave.

Je viens de regarder les logs et la majorité est <200ms (donc l'autosave_timeout_id pourrait être 200ms) mais il y en a quand même qui prennent plus de temps et ça va être frustrant.

(en passant, cette mesure à 200ms est déjà utilisée pour retarder le submit en cas de formulaire avec du géocodage inversé et des champs à synchroniser, donc jugée ok là).

Donc je dirais que ça m'irait d'avoir le patch attendant 200ms max.

Mais peut-être (voire bien sûr ?) les autosave à problème sont les longs ?

Ça m'irait totalement arbitrairement que côté serveur l'autosave qui voit qu'il a pris plus de 200ms il décide de finalement ne pas enregistrer.

~~~

Tout autre chose on pourrait peut-être envisager d'exploiter last_update_time, avec dans autosave() un store conditionné là-dessus, façon :

```
    def autosave(...):
        dt = datetime.datetime.now()
        ...
        formdata.store(max_datetime=dt)
```

et le .store() il ferait in fine que le UPDATE SQL ajoute un "WHERE last_update_time < dt".

#10 - 30 octobre 2021 15:56 - Frédéric Péters

> Donc je dirais que ça m'irait d'avoir le patch attendant 200ms max.

Via statistics.quantiles, sur le temps des autosave hier sur wcs.node1.prod,

- 10% : 0.04
- 20% : 0.042
- 30% : 0.044
- 40% : 0.047
- 50% : 0.05
- 60% : 0.071
- 70% : 0.097
- 80% : 0.15
- 90% : 0.27
- 95% : 0.455
- 99% : 2.605
- 99,9% : 3.3654
- 99,99% : 6.70147

(allez, on pourrait poser 300ms et choper 90%).

#11 - 06 novembre 2021 14:27 - Benjamin Dauvergne

- Fichier 0002-js-launch-autosave-on-tab-change-58276.patch ajouté

- Fichier 0001-js-prevent-autosave-while-user-is-active-58276.patch ajouté
~~~

- Fichier 0003-js-delay-submit-until-autosave-ajax-request-is-finis.patch ajouté

- Fichier 0004-js-augment-autosave-delay-if-autosave-fails-58276.patch ajouté

- Fichier 0005-forms-abort-autosave-after-200-ms-58276.patch ajouté

Frédéric Péters a écrit :

> 0002 (même si je me dis sans vérifier que ça ne va pas nécessairement avoir un effet côté traitement de la requête sur le serveur), 0003.

J'ai viré 0002, avec 0004 ça ne servait plus à rien avec le 0004 qui attend la fin de la requête et comme tu le remarques ça n'a aucun effet coté serveur de toute façon.

> > Ok, alors mieux certainement: capter les clicks sur les boutons submit, si un autosave est en cours on conserve la référence au bouton et on bloque le submit ainsi que les autosave.
>
> Je viens de regarder les logs et la majorité est <200ms (donc l'autosave_timeout_id pourrait être 200ms) mais il y en a quand même qui prennent plus de temps et ça va être frustrant.

Je ne suis pas chaud pour changer la durée actuelle finalement, c'est une scorie de mes tests, je resui revenu à 5 secondes.

> Mais peut-être (voire bien sûr ?) les autosave à problème sont les longs ?

Le problème c'est le chevauchement, forcément c'est plus fréquent avec un autosave ou une soumission classique qui dure longtemps.

> Ça m'irait totalement arbitrairement que côté serveur l'autosave qui voit qu'il a pris plus de 200ms il décide de finalement ne pas enregistrer.

Je vois mais ce n'est pas aussi correct que d'attendre le retour réseau, là tu voudrais que coté serveur on essaie autant que faire se peut de ne prendre que 200s ms (c'est pas évident sans signal timer et même avec un signal ça ne garantit rien et ça rend la logique compliquée) et que coté client on prenne cette info pour n'attendre que 200ms sauf qu'on ne sait pas quand la requête aura été reçue et le timer démarré, on pourra toujours avoir une erreur.

Donc je veux bien arrêter les autosave qui prennent plus de 200ms de façon assez vague en utilisant SIGALRM, mais comme ça n'inclut par les délais réseaux ça ne garantit pas 200ms coté client, donc 0004 (0003 maintenant) est essentiel.

> Tout autre chose on pourrait peut-être envisager d'exploiter last_update_time, avec dans autosave() un store conditionné là-dessus, façon :
>
> et le .store() il ferait in fine que le UPDATE SQL ajoute un "WHERE last_update_time < dt".

J'ai tenté ça, ça ne marche pas (je ne soumet pas le patch), parce qu'autosave ne se base pas sur le contenu du formdata du draft, uniquement sur ce qui est dans session.magictokens et le contenu de formdata.data est écrasé avec ça. Connaître le last_update_time du formdata ne me donne pas le last_update_time de la session qui est le véritable objet à suivre. En attendant la seule bonne correction pour ça c'est de ne pas modifier les données en session, donc #58208. Ou #24635 (remplacer magictokens par une utilisation directee de formdata.data) dans un futur plus lointain.

TL;DR

- correction dans 0001 autosave_is_running était mis à true trop tôt
- retour à un délai de 5 secondes entre autosave
- 0002 supprimé (XMLHttpRequest.abort())
- ajout de 0004 pour augmenter le délai en cas d'erreur d'autosave jusqu'à un max de 5 minutes.
- ajout de 0005: arrêter un autosave qui prend plus de 200ms coté serveur en utilisant un timer via signal.setitimer/SIGALRM.

#12 - 06 novembre 2021 14:46 - Frédéric Péters

> (c'est pas évident sans signal timer et même avec un signal ça ne garantit rien et ça rend la logique compliquée)

Je ne pensais vraiment pas à quelque chose de compliqué, de la manière la plus bête même :

```
@@ -1351,6 +1351,9 @@ class FormPage(Directory, FormTemplateMixin):
         if not session.has_form_token(get_request().form.get('_ajax_form_token')):
             return result_error('obsolete ajax form token (late check)')

+        if time.time() - get_request().t0 > 0.2:
+            return result_error('too long')
+
```

```
        try:
            self.save_draft(form_data, page_no)
        except SubmittedDraftException:
```

#13 - 06 novembre 2021 15:50 - Benjamin Dauvergne

Voilà branche à jour, je suppose que ce n'est intéressant que si effectivement la partie avant save_draft prend du temps (pour des appels à des data source externes), par contre ça n'améliorera pas l'attente coté client puisque de toute façon on va attendre avant de décider d'abandonner, le signal permettait de répondre au client le plus vite possible en cas de lenteur.

Il faudrait quand même donner un coup d'œil à #58208 car si on n'écrit plus en session, si on sait qu'un draft existe (draft_formdata_id existe dans les données du magictoken) alors on n'est pas obligé d'attendre la fin de l'autosave puisqu'il ne peut plus y avoir d'écriture en session par celui-ci, en gros ça voudrait dire qu'après le premier autosave réussi ou la première transition de page, l'attente entre submit/autosave ne serait plus nécessaire.

#14 - 12 novembre 2021 14:50 - Frédéric Péters

- ajout de 0004 pour augmenter le délai en cas d'erreur d'autosave jusqu'à un max de 5 minutes.

```
+  var autosave_delay_max = 5 * 60 * 1000; // 5 minutes
```

I y a une limite max à 60 secondes pour le traitement d'une requête par uwsgi, le 5 minutes ici est bizarre.

```
+          if (jqxhr.status != 200) { // network error
```

De manière générale je ne capte pas trop ce qui est cherché comme solution par ce patch.

#15 - 22 novembre 2021 10:43 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

#16 - 23 novembre 2021 11:08 - Benjamin Dauvergne

- Statut changé de En cours à Solution proposée

Frédéric Péters a écrit :

> I y a une limite max à 60 secondes pour le traitement d'une requête par uwsgi, le 5 minutes ici est bizarre.
> De manière générale je ne capte pas trop ce qui est cherché comme solution par ce patch.

L'idée c'est de ne pas matraquer le serveur d'autosave si il timeout ou part en erreur, alors que des dizaines de personnes ont des pages ouvertes; ça n'a pas de rapport avec le timeout de traitement des requêtes.

PS: j'ai ajouté un commentaire dans le patch (sur la branche) pour donner cette explication.

#17 - 21 décembre 2021 11:10 - Benjamin Dauvergne

- Fichier 0002-js-launch-autosave-on-tab-change-58276.patch ajouté

- Fichier 0001-js-prevent-autosave-while-user-is-active-58276.patch ajouté

- Fichier 0003-js-delay-submit-until-autosave-ajax-request-is-finis.patch ajouté

- Fichier 0004-forms-abort-autosave-after-200-ms-58276.patch ajouté

J'ai viré le patch 0004 si ça peut faciliter une validation, les plus important c'est le décalage du submit et la limitation à 200ms.

#18 - 28 décembre 2021 13:31 - Frédéric Péters

- Statut changé de Solution proposée à Résolu (à déployer)

Validé poussé,

```
commit 7a13c6eb6a467c8a67c675a985800f1845cc7f08
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Wed Nov 3 19:18:51 2021 +0100

    forms: abort autosave after 200 ms (#58276)

commit 72fa3ca77bf004f087d297a914ba4bd2b761db2f
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Thu Oct 28 23:43:38 2021 +0200
```

```
    js: delay submit until autosave ajax request is finished (#58276)

commit a1a66e7807f7e967b511c8cb9a0beb3078e5c69f
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Thu Oct 28 18:38:36 2021 +0200

    js: launch autosave on tab change (#58276)

commit 53c6f2d51d49f63c7101bc3a6d531a455edfa371
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Wed Oct 27 09:10:07 2021 +0200

    js: prevent autosave while user is active (#58276)
```

#19 - 28 décembre 2021 15:16 - Frédéric Péters

- Statut changé de Résolu (à déployer) à Solution déployée

#21 - 27 février 2022 04:42 - Transition automatique

Automatic expiration

Fichiers

0003-js-disable-submit-buttons-during-autosave-58276.patch	1,22 ko	28 octobre 2021	Benjamin Dauvergne
0004-js-launch-autosave-on-tab-change-58276.patch	1,15 ko	28 octobre 2021	Benjamin Dauvergne
0001-js-prevent-autosave-while-user-is-active-58276.patch	2,56 ko	28 octobre 2021	Benjamin Dauvergne
0002-js-abort-current-autosave-ajax-request-on-submit-582.patch	1,71 ko	28 octobre 2021	Benjamin Dauvergne
0004-js-delay-submit-until-autosave-ajax-request-is-finis.patch	1,97 ko	28 octobre 2021	Benjamin Dauvergne
0001-js-prevent-autosave-while-user-is-active-58276.patch	2,56 ko	28 octobre 2021	Benjamin Dauvergne
0003-js-launch-autosave-on-tab-change-58276.patch	1,15 ko	28 octobre 2021	Benjamin Dauvergne
0002-js-abort-current-autosave-ajax-request-on-submit-582.patch	1,71 ko	28 octobre 2021	Benjamin Dauvergne
0002-js-launch-autosave-on-tab-change-58276.patch	1,15 ko	06 novembre 2021	Benjamin Dauvergne
0001-js-prevent-autosave-while-user-is-active-58276.patch	2,65 ko	06 novembre 2021	Benjamin Dauvergne
0003-js-delay-submit-until-autosave-ajax-request-is-finis.patch	1,67 ko	06 novembre 2021	Benjamin Dauvergne
0004-js-augment-autosave-delay-if-autosave-fails-58276.patch	2,13 ko	06 novembre 2021	Benjamin Dauvergne
0005-forms-abort-autosave-after-200-ms-58276.patch	3,62 ko	06 novembre 2021	Benjamin Dauvergne
0002-js-launch-autosave-on-tab-change-58276.patch	1,15 ko	21 décembre 2021	Benjamin Dauvergne
0001-js-prevent-autosave-while-user-is-active-58276.patch	2,65 ko	21 décembre 2021	Benjamin Dauvergne
0003-js-delay-submit-until-autosave-ajax-request-is-finis.patch	1,67 ko	21 décembre 2021	Benjamin Dauvergne
0004-forms-abort-autosave-after-200-ms-58276.patch	2,37 ko	21 décembre 2021	Benjamin Dauvergne