

Mandaye - Bug #14575

rajouter un delai pour la detection d'une authentification réussie ou échouée

12 janvier 2017 10:35 - Serghei Mihai

Statut:	Fermé	Début:	12 janvier 2017
Priorité:	Normal	Echéance:	
Assigné à:	Serghei Mihai	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
Certaines applications métier (encore Ermes) utilisent du javascript pour construire toute la page. Le chargement de ces scripts peut être longue et les elements nécessaires à la détection de l'authentification ne sont pas encore chargés quand le page.onLoadFinished est executé.			
Un délai d'attente pour l'execution du corps de onLoadFinished permet de regler cela.			

Révisions associées

Révision eee5c873 - 09 février 2017 09:48 - Serghei Mihai

wait for page loading to check authentication (#14575)

Log an error if both auth success and failure functions return true.

Historique

#1 - 12 janvier 2017 10:36 - Serghei Mihai

- Fichier 0001-add-delay-for-execution-of-onLoadFinished-event.patch ajouté
- Patch proposed changé de Non à Oui

J'ai testé avec un délai de 400ms et ça fonctionne.

#2 - 12 janvier 2017 10:41 - Frédéric Péters

Pas fan, magie trop fragile. Je verrais plutôt la possibilité de définir une fonction qui doit être vraie avant d'avancer (et exécuter cette fonction toutes les 200ms pendant max 2 secondes, par exemple).

#3 - 12 janvier 2017 16:18 - Serghei Mihai

- Fichier 0001-wait-for-page-loading-to-complete-before-authenticat.patch ajouté

Alors je verrais cette fonction dans le auth_checker de l'application qui en as besoin. Ensuite relance de la fonction de vérification de l'authentification à coup de setInterval.

Je viens de tester sur Ermes et ça fonctionne, mais ça devrait être testé plus sérieusement.

#4 - 13 janvier 2017 15:31 - Frédéric Péters

```
}  
  // check authentication  
  check_auth();
```

Il y a là un problème d'indentation.

Mais sur le fond, quand même, je préférerais qu'un module puisse déclarer sa propre fonction "en état de vérifier l'authent ?", plutôt qu'exécuter le code de vérification de l'authent et de demander à celui-ci de gérer non plus seulement "bon / pas bon" mais également un "ne sais pas trop encore" un peu foireux.

#5 - 23 janvier 2017 17:35 - Serghei Mihai

C'est mon idée: le module de l'application dispose d'une fonction supplémentaire en plus de auth_success ou password_change_required qui dit "pas bon" si impossible de vérifier l'authentification. Et quand c'est bon, ça laisse la main à auth_success et compagnie.

Le nom que je propose dans le patch de plus haut n'est probablement pas terrible, à changer, mais l'idée est la même.

#6 - 23 janvier 2017 17:51 - Frédéric Péters

Peut-être que je me perds dans cette nouvelle lecture du patch (c'était il y a dix jours) mais sommairement, là, que l'expiration soit passée (counter > 10) ou que ça soit prêt (loaded), le code de vérification de l'authent est appelé, et doit donc pour moi gérer trois cas (bon / pas bon / oops ça avait pas chargé).

#7 - 26 janvier 2017 15:02 - Frédéric Péters

Retour sur la discussion du jour, un peu adapté suite à la lecture du code.

On définit aujourd'hui `window.auth_success`, on doit pouvoir également définir `window.auth_failure`. Quand les deux sont présents, s'il n'y a pas un retour positif d'un des deux, on attend un peu genre ¼ de seconde avant de réessayer. (en bout de course si ça n'arrive pas on aura [#14606](#) pour tuer l'affaire).

Aussi, dans les consignes d'écriture de ces fonctions d'évaluations, il faut noter qu'il est mieux de tester l'existence d'un élément, plutôt que l'absence.

ex:

```
window.auth_success = function(){
  if ($('#INDEX_TBL_LOGIN').length < 1)
    return true;
  return false;
}
```

Retournerait que ça marche en cas de présence de mettons un bug qui retournerait une page blanche; on devrait plutôt avoir l'inverse, genre (en inventant le sélecteur) :

```
window.auth_success = function(){
  if ($('#INDEX_TBL_LOGOUT').length)
    return true;
  return false;
}
```

#8 - 27 janvier 2017 15:53 - Serghei Mihai

- Fichier `0001-wait-for-page-loading-to-check-authentication-14575.patch` ajouté

- Statut changé de Nouveau à En cours

- Assigné à mis à Serghei Mihai

Patch suivant ce principe (sur l'exemple d'Archimed) mais avec un délai d'attente de 0.1 secondes.

Au passage je découvre que l'événement `onLoadFinished` s'exécute 2 fois.

#9 - 01 février 2017 13:51 - Josué Kouka

Sur une application comme Ermes, on peut directement taper sur des ws :

```
resp = logged_requests.post('https://mediatheques.montpellier3m.fr/DEFAULT/Ermes/Recherche/logon.svc/logon', {
  data: { 'username': 'whatever', 'password': 'whatever' } })
```

Reponse en cas d'echec

```
{ "errors": [{"data": {"__type": "ObjectCollection:#Archimed.Serialization", "badPasswordCount": 1, "doCheckAdditional": false, "checkAdditionalResult": true}, "id": "TPA6+H6ii1lZT0oyliNXCLuaPkU=", "msg": "Le mot de passe saisi est incorrect, vérifiez la syntaxe et réessayez.", "type": "InvalidCredentials"}], "message": "", "success": false, "d": "Anonymous" }
```

Je pense qu'avec

Reponse en cas de succès

```
{ "errors": [], "message": "", "success": true, "d": "entrouvert" }
```

Je pense qu'en ajoutant une logique simple appel ws pour les applications utilisant des ws (Fred l'avait évoqué une fois je pense) ça devrait marcher. (je fais le test puis je mets à jour le ticket)

#10 - 01 février 2017 16:47 - Serghei Mihai

Ce que m'embête c'est de devoir gerer les cookies en provenance de requests comme le faisait l'ancien mandaye. Et de mes souvenirs ce n'était pas évident.

L'utilisation de phantomjs semblait apporter la solution de ne plus rien avoir à gerer à ce niveau, juste transférer les cookies.

#11 - 06 février 2017 15:38 - Serghei Mihai

Suite à la discussion de ce matin, patch à jour prenant en compte la définition de la fonction auth_failure dans chaque application. Testé sur Ermes et Teamnet.

#12 - 06 février 2017 15:41 - Frédéric Péters

Patch absent.

#13 - 06 février 2017 15:44 - Serghei Mihai

- Fichier 0001-wait-for-page-loading-to-check-authentication-14575.patch ajouté

#14 - 06 février 2017 15:58 - Frédéric Péters

Vraiment > 1 et pas >= 1 dans if (\$('a.account_logoff').length > 1) et if (\$('#input[type="password"]#code').length > 1) ?

```
if (input.auth_failure !== undefined) {  
  // check authentication failure only if the function is defined  
  if (!input.auth_success && !input.auth_failure)
```

Je pense que ça gagnerait à un commentaire bien plus explicite, que le "on fait le taf uniquement si c'est pas undefined" est déjà bien exprimé par la première ligne; que la seconde condition par contre, elle exprime le fait qu'en présence des deux fonctions, si aucune n'est vérifiée, on lache l'affaire et le setInterval viendra rejouer plus tard.

Aussi, sur la situation où on aurait input.auth_success et input.auth_failure, je me dis qu'on doit traiter ça comme une situation d'erreur (et idéalement logger ça de manière suffisamment critique pour qu'on soit informé parce que ça veut sans doute dire que l'application a changé de comportement).

#15 - 06 février 2017 16:23 - Serghei Mihai

Frédéric Péters a écrit :

Aussi, sur la situation où on aurait input.auth_success et input.auth_failure, je me dis qu'on doit traiter ça comme une situation d'erreur (et idéalement logger ça de manière suffisamment critique pour qu'on soit informé parce que ça veut sans doute dire que l'application a changé de comportement).

Dans ce cas on doit revoir la façon dont on écrit les messages en sortie de phantom. Actuellement tout va dans stdout qui est lu par le code python et interprété comme json.

Benj donnait une piste ce matin : <https://github.com/ariya/phantomjs/blob/master/examples/stdin-stdout-stderr.js> . On devrait passer les messages à logger dans stderr et le stdout uniquement pour le json à interpréter.

#16 - 06 février 2017 16:34 - Frédéric Péters

Non; à nouveau ne mélangeons pas, pour le micmac de sortie, on a [#14881](#). Ici, je dis qu'il faudrait arrêter le traitement sur une erreur, et que celle-ci soit logguée du côté Python, quand le code affirme que ça a à la fois été un succès et un échec.

#17 - 06 février 2017 17:31 - Serghei Mihai

- Fichier 0001-wait-for-page-loading-to-check-authentication-14575.patch ajouté

Ok.

Je vois l'erreur retournée dans le json sous la forme {"result": "error", "reason": "..."} et logguée ensuite. Message à modifier.

#19 - 07 février 2017 16:08 - Serghei Mihai

- Fichier 0001-wait-for-page-loading-to-check-authentication-14575.patch ajouté

Patch refactorisé, principalement la fonction de vérification de l'échec.

#20 - 07 février 2017 17:23 - Serghei Mihai

- Fichier 0001-wait-for-page-loading-to-check-authentication-14575.patch ajouté

Patch rebasé sur master.

#21 - 08 février 2017 14:43 - Serghei Mihai

- Fichier 0001-wait-for-page-loading-to-check-authentication-14575.patch ajouté

Après des tests sur d'autres projets en prod comme le conservatoire de Vincennes et Teamnet à Meyzieu je découvre que si la fonction `auth_failure` n'est pas définie pour chaque application (et dans ce cas je lui attribue `undefined`) phantomjs l'interprete comme `null` et la condition d'arrêt de la fonction `check_auth` n'est pas respectée. Cela a comme effet une boucle pour les codes saisis éronnés.

En attendant de trouver une explication et solution à ce phenomène je définis des fonctions `auth_failure` pour duonet et teamnet.

Testé avec les 3: archimed, duonet et teamnet.

#22 - 08 février 2017 14:56 - Frédéric Péters

Après des tests sur d'autres projets en prod comme le conservatoire de Vincennes et Teamnet à Meyzieu je découvre que si la fonction `auth_failure` n'est pas définie pour chaque application (et dans ce cas je lui attribue `undefined`) phantomjs l'interprete comme `null` et la condition d'arrêt de la fonction `check_auth` n'est pas respectée. Cela a comme effet une boucle pour les codes saisis éronnés.

Quand même du mal à piger cette explication sans précision sur la condition dont tu parles.

Tu dis que le "return undefined;" du code; quand on passe dedans et bien malgré tout `input.auth_failure == null` ?

#23 - 08 février 2017 14:59 - Serghei Mihai

- Fichier 0001-wait-for-page-loading-to-check-authentication-14575.patch supprimé

#24 - 08 février 2017 14:59 - Serghei Mihai

- Fichier 0001-wait-for-page-loading-to-check-authentication-14575.patch ajouté

#25 - 08 février 2017 15:04 - Serghei Mihai

Frédéric Péters a écrit :

Tu dis que le "return undefined;" du code; quand on passe dedans et bien malgré tout `input.auth_failure == null` ?

C'est ça.

Je detecte ça en voyant le code du if :

```
if (input.auth_failure !== undefined) {  
  ...  
}
```

s'executer.

#26 - 08 février 2017 15:23 - Frédéric Péters

Et donc, pour comprendre, isoler les choses, jouer avec les types, genre :

```
var page = require('webpage').create();  
var input = Object();  
  
page.open('http://localhost', function(status) {  
  input.auth_failure = page.evaluate(function() {  
    return undefined;  
  });  
  if (input.auth_failure == true) console.log('==true');  
  if (input.auth_failure == false) console.log('==false');  
  if (input.auth_failure == undefined) console.log('==undefined');  
  if (input.auth_failure == null) console.log('==null');  
  if (input.auth_failure === undefined) console.log('===undefined');  
  if (input.auth_failure === null) console.log('===null');  
  if (input.auth_failure != undefined) console.log('!=undefined');  
  if (input.auth_failure != null) console.log('!=null');  
  if (input.auth_failure !== undefined) console.log('!=undefined');  
  if (input.auth_failure !== null) console.log('!=null');  
  
  phantom.exit();  
});
```

Plutôt que dire "on essaiera plus tard de comprendre".

#27 - 08 février 2017 18:21 - Serghei Mihai

- Fichier 0001-wait-for-page-loading-to-check-authentication-14575.patch ajouté

Patch à jour pour gérer le cas d'undefined.

Je modifie également le fichier d'authentification pour Archimed pour qu'il n'attende pas l'événement onLoad car il attend que tous les scripts de l'application finissent leur boulot et les fonctions retourneront null.

#28 - 09 février 2017 07:31 - Frédéric Péters

```
if result.get('error') :
    logger.error('Error occurred: %s' % result.get('reason'))
```

On peut juste logger le message; "error occurred" c'est implicite. (et si la raison seule n'est pas top, on a la main dans le js pour mieux l'écrire).

À part ça, si les tests sont concluants, je me dis qu'on peut y aller.

#29 - 09 février 2017 09:49 - Serghei Mihai

- Statut changé de En cours à Résolu (à déployer)

Ok.

```
commit eee5c87346e73431200f9f07d29e7ed45bce3e09
Author: Serghei Mihai <smihai@entrouvert.com>
Date: Mon Jan 23 16:20:07 2017 +0100
```

```
wait for page loading to check authentication (#14575)
```

```
Log an error if both auth success and failure functions return true.
```

#30 - 26 février 2017 23:38 - Serghei Mihai

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0001-add-delay-for-execution-of-onLoadFinished-event.patch	1,03 ko	12 janvier 2017	Serghei Mihai
0001-wait-for-page-loading-to-complete-before-authenticat.patch	3,75 ko	12 janvier 2017	Serghei Mihai
0001-wait-for-page-loading-to-check-authentication-14575.patch	3,99 ko	27 janvier 2017	Serghei Mihai
0001-wait-for-page-loading-to-check-authentication-14575.patch	4,4 ko	06 février 2017	Serghei Mihai
0001-wait-for-page-loading-to-check-authentication-14575.patch	5,27 ko	06 février 2017	Serghei Mihai
0001-wait-for-page-loading-to-check-authentication-14575.patch	5,08 ko	07 février 2017	Serghei Mihai
0001-wait-for-page-loading-to-check-authentication-14575.patch	5,03 ko	07 février 2017	Serghei Mihai
0001-wait-for-page-loading-to-check-authentication-14575.patch	7,21 ko	08 février 2017	Serghei Mihai
0001-wait-for-page-loading-to-check-authentication-14575.patch	5,6 ko	08 février 2017	Serghei Mihai