

Authentic 2 - Bug #55127

Avant de faire l'évaluation de "show_condition" par ast, lorsque l'on ajoute des variables (service_ou_slug, service_slug et service au context d'évaluation), définir ces variable à None si le paramètre service n'est pas défini dans la requête.

23 juin 2021 18:12 - Nicolas Roche

Statut:	Fermé	Début:	23 juin 2021
Priorité:	Normal	Echéance:	
Assigné à:	Nicolas Roche	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		

Description

```
"show_condition": {  
  "ma_mire": "'service_slug=="agents'",
```

L'évaluation échouant lorsque les variables ne sont pas définies,

Le but de ce ticket est surtout de supprimer la génération de traces lorsque l'on arrive directement sur une mire de connection, alors que l'on a prévu de l'afficher ou non, suivant le service qui nous a redirigé.

Révisions associées

Révision da6013b6 - 01 juillet 2021 12:24 - Nicolas Roche

views: evaluation context error will success (#55127)

Historique

#1 - 23 juin 2021 18:21 - Nicolas Roche

- Fichier 0001-views-pass-service-parameter-to-show-evaluation-cont.patch ajouté
- Tracker changé de Support à Bug
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Désolé pour le titre du ticket, je n'étais vraiment pas inspiré.
J'ai écrit un test, mais il est tiré par les cheveux.

#3 - 23 juin 2021 18:25 - Frédéric Péters

- Projet changé de Authentik à Authentic 2

#4 - 23 juin 2021 18:57 - Benjamin Dauvergne

evaluate_condition a un paramètre on_raise pour déterminer une valeur par défaut si l'expression échoue, c'est ça qu'il faut utiliser.

#5 - 24 juin 2021 10:29 - Nicolas Roche

- Fichier 0001-views-do-not-log-on-evaluation-context-failure-55127.patch ajouté

Ok.

#6 - 24 juin 2021 10:48 - Benjamin Dauvergne

Je t'embête encore un poil, je me dis que puisque qu'on a un attribute self.enabled on pourrait s'en servir comme valeur par défaut, ça permet d'avoir les deux comportements :

- condition qui échoue -> caché
- condition qui échoue -> visible

À défaut d'en passer par là, une valeur par défaut à True me paraît plus sûre; lors du debug d'une condition avoir la mire de login qui disparaît par défaut c'est un poil embêtant.

#7 - 24 juin 2021 11:11 - Serghei Mihai

Benjamin Dauvergne a écrit :

Je t'embête encore un poil, je me dis que puisque qu'on a un attribut `self.enabled` on pourrait s'en servir comme valeur par défaut, ça permet d'avoir les deux comportements :

- condition qui échoue -> caché
- condition qui échoue -> visible

Si `self.enabled` est faux on n'arrive même pas à l'exécution de `evaluate_condition`:

https://git.entrouvert.org/authentic.git/tree/src/authentic2/utills/__init__.py#n179

À défaut d'en passer par là, une valeur par défaut à `True` me paraît plus sûre; lors du debug d'une condition avoir la mire de login qui disparaît par défaut c'est un poil embêtant.

Je pense aussi que c'est mieux de ne pas perdre un bloc de connexion.

#8 - 24 juin 2021 11:23 - Serghei Mihai

Au delà de la valeur de fallback en cas d'échec d'évaluation de la condition il faut quand même logger quelque part le fait que les variables n'existent pas.

On peut logger en warning au lieu de error dans:

```
try:
    return evaluate_condition(show_condition, ctx)
except Exception as e:
    logger.error(e)
```

pour éviter les mails.

#9 - 24 juin 2021 11:30 - Frédéric Péters

pour éviter les mails.

S'il y a des erreurs il faut les voir (dans un mail ou dans sentry, comme aujourd'hui), personne ne va aller les chercher dans les logs.

Et si on considère que c'est le comportement correct d'afficher (ou cacher) le bloc en cas d'exception à l'évaluation, je trouve qu'il n'y a pas à encombrer les logs (de quelque chose qu'on considère correct).

#10 - 24 juin 2021 14:55 - Benjamin Dauvergne

Pour moi c'est une fonctionnalité à utiliser à la marge, en gros c'est bien sympa de cacher/afficher un truc au bon public, mais dans le cas où ça ne marche le mieux c'est de laisser les gens pouvoir agir, donc comme je le disais à minima la valeur devrait être `True`.

C'est moins grave d'afficher un bouton en trop que de bloquer tout le monde (je trouve); déplacer le fonctionnement du flag `enabled` c'était la cerise sur le gâteau (une fois que l'expression de condition d'un bouton agent fonctionne on le désactive).

#11 - 24 juin 2021 16:01 - Nicolas Roche

- Fichier `0001-views-manage-default-evaluation-result-on-context-fa.patch` ajouté

Ok.

Dans mon cas d'usage, si l'utilisateur arrive de nulle part sur la page de login, alors il aura la mire ADFS réservée aux agents, pas de souci.

Perso le patch me semble quand même contre intuitif.

Si le `backend.enabled()` renvoie `False`, alors dans `src/authentic2/views.py::login()`

```
authenticators = utils.get_backends('AUTH_FRONTENDS')
```

ne renvoie rien et donc on ne passe pas sur

```
# Create blocks
for authenticator in authenticators:
    ...
```

```
if authenticator.shown(ctx=show_ctx)
```

par conséquent on pourrait simplifier et dans `src/authentic2/authenticators.py::BaseAuthenticator::shown()` mettre :

```
# Evaluation failure will success
return evaluate_condition(show_condition, ctx, on_raise=True)
```

Et du coup je réalise que je ne comprend pas la dernière remarque de Benjamin :

déplacer le fonctionnement du flag `enabled` c'était la cerise sur le gâteau (une fois que l'expression de condition d'un bouton agent fonctionne on le désactive).

"déplacer" ça voulais dire qu'il fallait modifier le comportement de `utils.get_backends()` ?

#12 - 24 juin 2021 16:49 - Benjamin Dauvergne

Nicolas Roche a écrit :

Et du coup je réalise que je ne comprend pas la dernière remarque de Benjamin :

déplacer le fonctionnement du flag `enabled` c'était la cerise sur le gâteau (une fois que l'expression de condition d'un bouton agent fonctionne on le désactive).

"déplacer" ça voulais dire qu'il fallait modifier le comportement de `utils.get_backends()` ?

Oui.

#13 - 24 juin 2021 18:10 - Nicolas Roche

- Statut changé de *Solution proposée* à *En cours*

"déplacer" ça voulais dire qu'il fallait modifier le comportement de `utils.get_backends()`

ok, ça marche presque mais ça casse les 2 tests `test_login_autorun` sur l'oidc et le saml (200 au lieu de 302 ; on n'est plus auto-logged).

Du coup je jette l'éponge et revient sur :

À défaut d'en passer par là, une valeur par défaut à `True` me paraît plus sûre;

#14 - 24 juin 2021 18:21 - Nicolas Roche

- Fichier `0001-views-evaluation-context-error-will-success-55127.patch` ajouté

- Statut changé de *En cours* à *Solution proposée*

Voilà.

#15 - 01 juillet 2021 12:15 - Nicolas Roche

- Fichier `0001-views-evaluation-context-error-will-success-55127.patch` ajouté

J'ai ajouté un test de connexion, via un service défini, mais pas celui porté par la condition, pour montrer que la condition marche dans les deux sens.

#16 - 01 juillet 2021 12:22 - Benjamin Dauvergne

- Statut changé de *Solution proposée* à *Solution validée*

#17 - 01 juillet 2021 12:24 - Nicolas Roche

- Statut changé de *Solution validée* à *Résolu (à déployer)*

```
commit da6013b67b58d8d1289bfa0496621351e6ed8e7e
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date: Wed Jun 23 18:16:47 2021 +0200
```

views: evaluation context error will success (#55127)

#18 - 01 juillet 2021 22:16 - Frédéric Péters

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

Fichiers

0001-views-pass-service-parameter-to-show-evaluation-cont.patch	2,98 ko	23 juin 2021	Nicolas Roche
0001-views-do-not-log-on-evaluation-context-failure-55127.patch	2,78 ko	24 juin 2021	Nicolas Roche
0001-views-manage-default-evaluation-result-on-context-fa.patch	4,5 ko	24 juin 2021	Nicolas Roche
0001-views-evaluation-context-error-will-success-55127.patch	2,82 ko	24 juin 2021	Nicolas Roche
0001-views-evaluation-context-error-will-success-55127.patch	3,4 ko	01 juillet 2021	Nicolas Roche