Authentic 2 - Development #25567

django 1.11 : un utilisateur inactif n'est plus authentifié

02 août 2018 11:04 - Emmanuel Cazenave

Statut:	Fermé	**Début:**	02 août 2018
Priorité:	Normal	**Echéance:**	
Assigné à:	Emmanuel Cazenave	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Planning:**	

Description

Dans le même genre que #25468, un changement dans django.contrib.auth : https://code.djangoproject.com/ticket/25232 https://github.com/django/django/commit/e0a3d937309a82b8beea8f41b17d8b6298da2a86

Un utilisateur inactif n'est plus authentifié (alors qu'avant il ne pouvait plus se loguer, mais si il avait une session en cours, il restait authentifié).

Ça casse ce test : http://git.entrouvert.org/authentic.git/tree/tests/test_views.py#n22

Une solution simple est d'implémenter dans authentic.backends.models_backends la méthode get_user plutôt de que se reposer sur contrib.auth.

Demandes liées:

Lié à Authentic 2 - Development #21489: Fonctionner avec Django 1.11 (et 1.8)	**Fermé**	**06 avril 2018**

Révisions associées

Révision 2c208379 - 12 août 2018 20:11 - Emmanuel Cazenave

django 1.11: keep an inactive user authenticated (#25567) (#21489)

Historique

#1 - 02 août 2018 11:04 - Emmanuel Cazenave

- Lié à Development #21489: Fonctionner avec Django 1.11 (et 1.8) ajouté

#2 - 02 août 2018 14:31 - Benjamin Dauvergne

Ça me va de changer le comportement pour celui de Django 1.11, c'est pas déconnant de déconnecter un utilisateur inactif, ici il va falloirt voir quand même comment finir le logout tout en désactivant l'utilisateur (je penche pour un flag en session qui indiquera à la vue de logout de désactiver l'utilisateur à la fin du logout); bien sûr ça ne marchera pas si l'utilisateur est connecté dans plusieurs navigateur (la perfection n'est pas de ce monde).

#3 - 02 août 2018 15:09 - Emmanuel Cazenave

Sur le fond pas de problème pour moi mais par contre est-ce qu'on pourrait décorréler ça de la migration django 1.11, en faire un ticket à part puisque c'est un changement de comportement ?

Pour le maintien du comportement actuel en 1.11, le fix que je décrivais est dans wip/dj111 et fonctionne.

#4 - 04 août 2018 12:26 - Benjamin Dauvergne

Ce qui me fait chier c'est qu'il va falloir défaire cette modification pour revenir au comportement qu'on souhaite en fait, je préférerai qu'on corrige le test.

#5 - 09 août 2018 11:13 - Emmanuel Cazenave

La modif en question (dcc38c5c85ec6d687c2a9143d12619357fe171e3 dans wip/dj111) :

```
diff --git a/src/authentic2/backends/models_backend.py b/src/authentic2/backends/models_backend.py
index 61ce7103..335b29c4 100644
--- a/src/authentic2/backends/models_backend.py
+++ b/src/authentic2/backends/models_backend.py
@@ -66,6 +66,13 @@ class ModelBackend(ModelBackend):
            else:
                user_login_failure(user.get_username())
```

```
+    def get_user(self, user_id):
+        UserModel = get_user_model()
+        try:
+            return UserModel._default_manager.get(pk=user_id)
+        except UserModel.DoesNotExist:
+            return None
+
```

J'ai juste fait un copié/coller du get_user django/contrib/auth/backends.py tel qu'il est en django 1.8

Je comprends pas bien ton plan : corriger le test facile mais ne pas toucher à authentic2/backends/models_backend.py et passer à django 1.11 ça donne un comportement bizarre quand un utilisateur supprime son compte (sans avoir tout compris à ce qu'il se passe, j'observe qu'on se retrouve sur la page d'accueil avec encore son nom/prénom dans la mire de connexion).

#6 - 09 août 2018 13:44 - Benjamin Dauvergne

- Statut changé de En cours à Solution validée

Ok.

#7 - 12 août 2018 20:13 - Frédéric Péters

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

Poussé en même temps que la branche django 1.11.

#8 - 13 décembre 2018 22:35 - Benjamin Dauvergne

- Statut changé de Résolu (à déployer) à Fermé