

Authentic 2 - Bug #7559

MultipleObjectsReturned: get() returned more than one LibertySessionDump -- it returned 2!

13 juin 2015 20:41 - Frédéric Péters

Statut:	Fermé	Début:	13 juin 2015
Priorité:	Normal	Echéance:	
Assigné à:	Benjamin Dauvergne	% réalisé:	100%
Catégorie:		Temps estimé:	0:00 heure
Version cible:	2.2.0	Planning:	
Patch proposé:	Non		
Description			
<p>I am not sure how things happened, multiple concurrent SSO made twice the same object in saml_libertysessiondump. (the concurrent SSO were the result of calls to menu.json of various applications, when building the sidebar menu)</p>			
<pre>File "/home/fred/src/eo/authentic/src/authentic2/idp/saml/saml2_endpoints.py", line 526, in sso return sso_after_process_request(request, login, nid_format=nid_format) File "/home/fred/src/eo/authentic/src/authentic2/idp/saml/saml2_endpoints.py", line 894, in sso_ after_process_request load_session(request, login) File "/home/fred/src/eo/authentic/src/authentic2/saml/common.py", line 286, in load_session kind=kind) ...</pre>			
<p>MultipleObjectsReturned: get() returned more than one LibertySessionDump -- it returned 2!</p>			
<p>I wondered if maybe it was because it reused an old session key but the id say otherwise.</p>			
<pre>authentic_multitenant=> select id from saml_libertysessiondump where django_session_key = 'jqpvw7i ayben5i5a50epqk2jvjmhsm4'; id ---- 30 31</pre>			

Révisions associées

Révision 8a4830c6 - 17 juin 2015 10:55 - Benjamin Dauvergne

saml: prevent creation of duplicate LibertySessionDump (fixes #7559)

By adding a unicity constraint we prevent .get() of ever raising a MultipleObjectsReturned exception.

Historique

#1 - 17 juin 2015 10:03 - Benjamin Dauvergne

Bug is really difficult, get_or_create() which used to save LibertySessionDump is not safe without a postgresql configured with the serializable isolation level, default level being read committed, as read committed cannot prevent the phantom read problem (the two transactions check if a given row, the two returns none, but in fact after the first transaction commit the second transaction pre-state has changed, there is now a row matching the query but it won't see it, see¹).

Adding an unicity constraint on (django_session_key, kind) could maybe alleviate the biggest problem, that is the load_session() failing. But when two concurrent save will happen one them may fail but I think django has a retry loop inside get_or_create() to prevent the problem in most cases.

.fn1 <http://stackoverflow.com/questions/13374335/what-is-the-difference-between-serializable-and-repeatable-read-isolation-level>
<http://www.postgresql.org/docs/9.1/static/transaction-iso.html>

#2 - 17 juin 2015 10:44 - Benjamin Dauvergne

- Version cible mis à 2.2.0

#3 - 17 juin 2015 11:00 - Benjamin Dauvergne

- Statut changé de Nouveau à Résolu (à déployer)

- % réalisé changé de 0 à 100

Appliqué par commit [authentic2|8a4830c6ca8c6a3cd0583da0a4fdbb9938e59625](#).

#4 - 18 juin 2015 01:32 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#5 - 23 février 2016 12:58 - Benjamin Dauvergne

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

#6 - 06 décembre 2017 15:30 - Benjamin Dauvergne

- Statut changé de Solution déployée à Fermé