

## Publik - Bug #9637

### sessions qui ne se ferment pas avec le navigo

13 janvier 2016 15:42 - Frédéric Péters

<b>Statut:</b>	Fermé	<b>Début:</b>	13 janvier 2016
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Benjamin Dauvergne	<b>% réalisé:</b>	100%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Club:</b>	Non
<b>Patch proposed:</b>	Non		
<b>Planning:</b>	Non		
<b>Description</b>			
Avec <a href="#">#8877</a> on a réussi à mettre SESSION_EXPIRE_AT_BROWSER_CLOSE dans la config de toutes nos applications. Ça n'a pas d'effet.			
Scénario :			
<ul style="list-style-type: none"><li>• je vais sur combo</li><li>• je lance un SSO</li><li>• authentic voit SESSION_EXPIRE_AT_BROWSER_CLOSE, le cookie session_id est posé sans date d'expiration (→ à la fermeture du navigo)</li><li>• django-mellon voit un sessionNotOnOrAfter, fait un request.session.set_expiry() → le cookie est posé avec cette date d'expiration</li><li>• je ferme mon navigo</li><li>• je réouvre mon navigo</li><li>• ma session authentic est terminée mais ma session dans combo est intacte</li></ul>			
<b>Sous-tâches:</b>			
Development # 9640: Le traitement de sessionNotOnOrAfter ne devrait pas annuler l'effet...			<b>Fermé</b>
<b>Demandes liées:</b>			
Lié à Hobo - Development #19243: afin d'appliquer la restriction de durée des...		<b>Fermé</b>	<b>06 octobre 2017</b>

### Historique

#### #1 - 13 janvier 2016 16:02 - Thomas Noël

au cas où : c'est le request.session.set\_expiry() qui annule le SESSION\_EXPIRE\_AT\_BROWSER\_CLOSE, tel que précisé par la doc django

#### #2 - 13 janvier 2016 17:13 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#### #3 - 08 mars 2016 14:02 - Benjamin Dauvergne

Je crois qu'on est arrivé à la conclusion que ça ne pourrait plus marcher ainsi, si c'est ok je réouvre [#8877](#) pour virer SESSION\_EXPIRE\_AT\_BROWSER\_CLOSE de debian\_config\_common.py et je rejette celui-ci.

#### #4 - 20 septembre 2017 14:30 - Benjamin Dauvergne

J'ai résolu le souci coté SP avec de nouveaux backends de session, mais de fait on les a jamais inclus dans debian\_config\_common.py, faudrait y repenser.

Actuellement l'expiration des session est a 10h + dernière date de sauvegarde de la session (dernière modif) pour IdP et SP. Utiliser les backends de session de django-mellon fixerait l'expiration sur les SPs à 10h + date du dernier SSO.

#### #5 - 10 décembre 2020 16:48 - Benjamin Dauvergne

- Lié à Development #19243: afin d'appliquer la restriction de durée des sessions présente dans les assertions SAML, utiliser mellon.sessions\_backends.db ajouté

#### #6 - 10 décembre 2020 16:48 - Benjamin Dauvergne

- Statut changé de Nouveau à Fermé

- Planning mis à Non

- Club mis à Non

Réglé par [#19243](#).