

Calebasse - Bug #1988

Historique des actions de l'utilisateur.

20 novembre 2012 10:19 - Mikaël Ates (de retour le 29 avril)

Statut:	Rejeté	Début:	20 novembre 2012
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	50%
Catégorie:		Temps estimé:	0:00 heure
Version cible:	1.3	Planning:	
Patch proposed:	Non		
Description			
Concevoir un système de log des actions de l'utilisateur, lisible par un non informaticien, qui soit accessible depuis l'icone historique positionnée sur la ligne d'un accès.			

Révisions associées

Révision a01d85be - 17 juillet 2014 10:23 - Serghei Mihai

logging actions on models with django_journal

Closes #1988

Historique

#1 - 21 novembre 2012 12:31 - Benjamin Dauvergne

Pourrais-tu lister les évènements qui doivent être pris en compte ?

- login
- logout
- modification/création d'un objet en base ? avec le détail ? tous les objets ? certains ?
- suppression des objets ?

Est-ce que ce système d'historique doit être accessible via plusieurs entrées (là tu demande de pouvoir lister pour un User, mais est-ce qu'on pourrait vouloir filtrer par dossier par exemple, et voir qui a fait quoi sur un dossier ?) ?

Est-ce qu'il doit contenir des évènements sans lien avec un utilisateur (par exemple quand on pousse vers la facturation vers la sécu. en tâche de fond) ?

#2 - 26 novembre 2012 21:58 - Mikaël Ates (de retour le 29 avril)

- Version cible changé de 0.1 à 0.2

Evénements:

- login/logout * validation automatique et déverrouillage * création/modification/suppression rendez-vous/actes, dossiers patient, accès et tout objet métier

On doit pouvoir lister par utilisateur et par objet bien que dans l'application il n'est prévue que de voir l'historique des actions de l'utilisateur (dans la partie accès).

Les évènements qui sont sans liens avec les utilisateurs sont à logger ailleurs.
Les rapports de télétransmission seront sûrement stockés en base avec un model dédié.

#3 - 06 décembre 2012 16:54 - Mikaël Ates (de retour le 29 avril)

- Version cible changé de 0.2 à 0.6

#4 - 06 janvier 2013 12:24 - Mikaël Ates (de retour le 29 avril)

- Version cible changé de 0.6 à 0.9

#5 - 24 juin 2013 11:15 - Mikaël Ates (de retour le 29 avril)

- *Projet changé de APS42 à Calebasse*
- *Priorité changé de Normal à Bas*
- *Version cible 0.9 supprimé*

#6 - 23 juin 2014 11:48 - Mikaël Ates (de retour le 29 avril)

- *Assigné à mis à Serghei Mihai*
- *Priorité changé de Bas à Haut*
- *Version cible mis à 1.1.3*

Visiblement on aurait tout ce qu'il faut avec reversion ?

#7 - 04 juillet 2014 15:09 - Mikaël Ates (de retour le 29 avril)

- *Priorité changé de Haut à Immédiat*
- *Patch proposed mis à Non*

Ca devient critique pour le support.

#8 - 07 juillet 2014 13:42 - Benjamin Dauvergne

Regardez <https://pypi.python.org/pypi/django-journal> c'est ce que j'ai développé pour docbow, il y a des paquets dans nos dépôts.

C'est assez simple à utiliser:

```
import django_journal
django_journal.record('my-tag', '{user} did this to {that}',
                    user=request.user, that=model_instance)
```

Des index sont créés sur les variables de template, il est par exemple possible de retrouver tous les logs qui utilisent '{user}' dans leur template pour un utilisateur donné. Le tag permet d'indiquer une grande famille d'évènement 'act-creation', 'act-delete', etc.. Si on utilise le middleware, on a automatiquement l'ip et l'utilisateur qui sont renseignés en passant par la méthode installée sur request.record().

#9 - 15 juillet 2014 16:25 - Serghei Mihai

- *Fichier 0001-logging-actions-on-models-with-django_journal.patch ajouté*

Log des actions sur les modèles, certains forms et views

#10 - 16 juillet 2014 09:43 - Serghei Mihai

- *% réalisé changé de 0 à 50*
- *Patch proposed changé de Non à Oui*

#11 - 16 juillet 2014 10:24 - Frédéric Péters

```
act.last_validation_state = last_validation_state
+ get_request().record('new-act', '{obj_id} created by {user} from {ip}', obj_id=act.id)
act.save()
```

Est-ce qu'on n'obtient pas le act.id uniquement après l'appel à save() ?

```
+ self.request.record('new-act',
+                     '{obj_id} created by {user} from {ip}',
+                     obj_id=self.object.id)
```

Le même message de log est produit à deux endroits différents, d'expérience c'est utile d'avoir une info différente.

Ou, autre commentaire, ces lignes de log sont enregistrées au mauvais endroit, et il ne devrait y en avoir qu'une, dans le save().

```
+ get_request().record('act-update', '{obj_id} updated by {user} from {ip} with: {changes}', changes=changes)
```

obj_id n'est pas défini.

#12 - 17 juillet 2014 10:30 - Serghei Mihai

- *Statut changé de Nouveau à Résolu (à déployer)*
- *% réalisé changé de 50 à 100*

Appliqué par commit calebasse|commit:a01d85be752cd242f076febde54a56d1d9d0f677.

#13 - 17 juillet 2014 10:39 - Frédéric Péters

Ça a été commité sans doute par erreur. (et en tout cas avec des erreurs)

#14 - 17 juillet 2014 12:08 - Serghei Mihai

Poussé trop vite.

Frédéric Péters a écrit :

[...]

Est-ce qu'on n'obtient pas le act.id uniquement après l'appel à save() ?

[...]

Le même message de log est produit à deux endroits différents, d'expérience c'est utile d'avoir une info différente.

Corrigé, merci.

Ou, autre commentaire, ces lignes de log sont enregistrées au mauvais endroit, et il ne devrait y en avoir qu'une, dans le save().

Initialement j'étais parti sur ça, mais je voulais avoir plus de détails sur les modifications effectuées sur les modèles, et logger en dehors du save me permet de voir ses modifications sans analyser les modifications sur l'objet.

[...]

obj_id n'est pas défini.

Je fais un nouveau patch

#15 - 17 juillet 2014 17:09 - Jérôme Schneider

Il manque également la dépendance sur django-journal dans le requirements.txt et le setup.py.

#16 - 18 juillet 2014 11:16 - Serghei Mihai

- Fichier 0001-logging-actions-on-models-improved-using-post_save-a.patch ajouté

Interception des actions via les signaux post_save et pre_delete.
Le setup mis à jour également.

#17 - 18 juillet 2014 16:13 - Benjamin Dauvergne

Idéalement il ne vaut mieux pas logger les id des objets et plutôt directement l'objet c'est toute la magie de la bibliothèque de gérer ça proprement et de mettre des liens vers les vues d'admin dans les logs, donc plutôt que

```
+ act.last_validation_state = last_validation_state
+ get_request().record('new-act', '{obj_id} created by {user} from {ip}', obj_id=act.id)
+ act.save()
```

faire plutôt ça

```
+ act.last_validation_state = last_validation_state
+ get_request().record('new-act', '{act} created', act=act)
+ act.save()
```

L'utilisateur et l'IP sont déjà sauvés et affichés dans les listing, pas besoin de les sauver explicitement. Si l'objet est effacé on verra dans les logs l'ID de l'objet avec l'annotation "deleted" au lieu d'un lien vers sa vue.

#18 - 18 juillet 2014 19:07 - Serghei Mihai

- Fichier 0001-logging-handling-cases-when-objects-created-modified.patch ajouté

Bien vu

#19 - 28 août 2014 16:45 - Mikaël Ates (de retour le 29 avril)

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

- Version cible changé de 1.1.3 à 1.3

Cela fonctionne bien mais il manque des détails. Avant de préciser lesquels je déplace la demande à la prochaine version et la met en cours.

#20 - 28 août 2014 17:47 - Mikaël Ates (de retour le 29 avril)

Les messages de log ne sont pas assez explicites.

Par exemple lors de la création de rendez-vous il faudrait avoir les détails des champs de ce rendez-vous. Lors de la modification, il faudrait avoir le détail de ce qui a été modifié.

#21 - 29 août 2014 10:55 - Serghei Mihai

Je peux stocker les détails lors de l'ajout ou modification d'un modèle sous forme d'un dictionnaire.

Si le journal doit être consultable par les secrétaires, je pense qu'il faudrait repenser les logging afin d'avoir une interface plus épurée et claire.

#22 - 01 septembre 2014 09:46 - Mikaël Ates (de retour le 29 avril)

- % réalisé changé de 100 à 50

Ok pour le stockage du detail.

#23 - 15 octobre 2014 18:27 - Benjamin Dauvergne

Pour stocker des informations variables avec un message on peut toujours faire ça (en vrai python sur un pseudo exemple):

```
template = "created object"
kwargs = {}
if info1:
    template += " with info1 {info1}"
    kwargs['info1'] = info1
if info2:
    template += " with info2 {info2}"
    kwargs['info1'] = info2
request.record(template, **kwargs)
```

Le dico kwargs est automatiquement transformé en clé-valeurs dans les tables du journal.

#24 - 20 avril 2015 09:46 - Frédéric Péters

- Patch proposed changé de Oui à Non

#25 - 27 avril 2015 10:45 - Benjamin Dauvergne

- Priorité changé de Immédiat à Normal

#26 - 16 octobre 2015 14:34 - Mikaël Ates (de retour le 29 avril)

- Assigné à Serghei Mihai supprimé

Fichiers

0001-logging-actions-on-models-with-django_journal.patch	13,1 ko	15 juillet 2014	Serghei Mihai
0001-logging-actions-on-models-improved-using-post_save-a.patch	11,7 ko	18 juillet 2014	Serghei Mihai
0001-logging-handling-cases-when-objects-created-modified.patch	2,23 ko	18 juillet 2014	Serghei Mihai