w.c.s. - Autre #1862

Améliorations au logging

04 novembre 2012 22:03 - Frédéric Péters

Statut:	Fermé	**Début:**	04 novembre 2012
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Planning:**	Non

Description

Le logging dans w.c.s. n'a fondamentalement pas évolué depuis le projet SVQ, il y avait eu à ce moment besoin de suivre de manière précise les actions des utilisateurs, aujourd'hui je pense qu'on devrait revoir ça.

D'une part avoir des logs orientés utilisateur :

- dégager les logs système de l'interface web (aujourd'hui on a des lignes "Internal Server Error" et c'est tout, de toute façon pas moyen de les associer à des traces) ;
- garder pour chaque utilisateur un log d'actions, une série limitée d'actions qu'il sera intéressant de présenter à l'utilisateur (ex: connexion, échec de connexion, complétion d'un formulaire, paiement d'une facture) ;
  - c'est-à-dire aussi dégager les logs servant à suivre les déplacements d'un utilisateur (ex: get_logger().info('home page')), ce sera mieux servi par piwik/owa.

Ça veut aussi dire qu'on pourrait rapprocher ces logs de l'utilisateur, ne plus avoir un gros fichier contenant le tout, dont on doit assurer une rotation pour ne pas étouffer sous un fichier trop gros.

D'autre part avoir un monitoring orienté application, comme le nombre de connexions réussies/ratées, le nombre de sessions, de formulaires complétés; pour cette partie on devrait à mon avis regarder du côté de statsd, et profiter de l'occasion pour se dire qu'on pourrait en plus avoir un suivi des performances par ce biais.

Dans cette série il y aura sans doute aussi des informations qui seront intéressantes pour certains gestionnaires (« il y a maintenant 2000 inscrits ! »), on pourra faire remonter l'info, avec des jolis graphes et tout, vers le backoffice.

Ce ticket est créé pour discuter et préparer ces idées.

Demandes liées:

Lié à w.c.s. - Development #51974: Journal orienté utilisateur des accès aux ...	**Fermé**	**31 mars 2023**

Historique

#1 - 05 novembre 2012 10:00 - Thomas Noël

> Ça veut aussi dire qu'on pourrait rapprocher ces logs de l'utilisateur

Oui, mais faire attention que les logs ne soient pas perdus si l'utilisateur supprime son compte.

Plus généralement, un système de log "multi-entrée" serait un objectif intéressant. Par exemple, un log sur un paiement de facture serait lié à l'utilisateur qui a payé, à la facture payée, à la régie concernée, etc. Et on pourrait faire des recherches sur l'un de ces critères.

#2 - 08 novembre 2012 15:57 - Frédéric Péters

- Fichier 0001-qommon-added-basic-statsd-infrastructure.patch ajouté

En me concentrant uniquement sur la partie qui n'a pas d'incidence, l'ajout de statistiques applicatives, en utilisant statsd, un premier patch; que je verrais bien cherrypické dans v2012.3 quand on aura une infra avec statsd et graphite en place.

#3 - 08 avril 2014 11:07 - Thomas Noël

- Version cible Au-quotidien 2014.5 supprimé

#4 - 05 septembre 2014 11:13 - Benjamin Dauvergne

Je vous proposerai bien d'adopter quelque chose d'approchant https://pypi.python.org/pypi/django-journal mais sans utiliser Django, à terme l'API que je verrai ce serait:

```
    request.log.error('unable to complete form {form}', form=form) # ici sont aussi loggé l'IP et l'identifian
t de l'utilisateur
```

On aurait deux tables:

```
CREATE TABLE log_line (
      id serial primary key,
      timestamp timestamp,
      line text,
)

CREATE TABL log_ref (
      log_line_id integer references log_line,
      key varchar(16),
      value varchar(64),
)

CREATE INDEX ON log_ref (key, value);
```

Les objets passés dans kwargs peuvent avoir une méthode log_ref() qui renvoie un identifiant qui sera l'identifiant stocké et lié à la ligne de log et permettant ensuite de retrouver le log. Par exemple:

```
classe User(object):
    def log_ref(self):
        return self.id

class FormDef(object):
    def log_ref(self):
        return name

class FormData(object):
    def log_reff(self):
        return '%s-%s' % (self.formdef.log_ref(), self.id)
```

Et quand on veut les logs liés à un objet on ferait juste:

```
# user.id == 45
logs.get_related_logs(user=user) # SELECT log_line.* FROM log_line, log_ref WHERE log_ref.log_line_id = id AND
 log_ref.key = 'user' AND log_ref.value = '45'
logs.get_related_logs(form=form)
```

qui renverrait une sorte de curseur SQL qu'on peut paginer, parcourir, etc..

#5 - 05 septembre 2014 11:43 - Frédéric Péters

Le ticket n'en est pas à ce niveau de détail, il y a déjà dans wcs une API pour logger des trucs, la question n'est pas de remplacer "get_logger().info('logout')" par "get_request().log.info('logout')", mais de garder "une série limitée d'actions" (dixit la description), et donc d'établir cette liste. Conserver le moment de déconnexion de l'utilisateur, par exemple, est-ce utile ? Si oui, à qui ?

Je partirais d'une page de log existante, réelle, et ferait cet inventaire.

> 2014-08-13 14:12:57 Non-identifié fill_user_attributes: received attributes {'uid': ['laurelebourhis@aol.com'], 'l': ['lattes'], 'personalTitle': ['Madame'], 'street': ['930 avenue de L\xc3\xa9onard de vinci'], 'sn': ['le bourhis'], 'postalCode': ['34970'], 'gn': ['laure'], 'email': ['laurelebourhis@aol.com']}

C'est utile pour le développeur qui débuggue, mais est-ce même utile à l'admin du site ?

> 14:12:57 Non-identifié using legacy attribute filling

Sans doute utile pour personne.

> 14:12:57 le bourhis laure form Candidature spontanée - id: 391 - view

Utile de suivre les agents dans ce qu'ils traitent ?

> 2014-08-18 00:26:27 Non-identifié form Demande de composteur - step Filling

Utile, à qui, de savoir qu'il y a eu un accès la page de ce formulaire ?

Etc.

Plus bas dans le ticket, après un interlude sur le monitoring applicatif, "informations qui seront intéressantes pour certains gestionnaires (« il y a maintenant 2000 inscrits ! »),", quelles sont les informations qui pourraient être utiles ?

C'est là-dessus que je voyais la discussion. (pas sur la manière de stocker ailleurs, avec une autre API, le même vrac d'info qu'aujourd'hui).

#6 - 05 septembre 2014 15:35 - Benjamin Dauvergne

Je répondais juste à ça: « Ça veut aussi dire qu'on pourrait rapprocher ces logs de l'utilisateur, ne plus avoir un gros fichier contenant le tout, dont on doit assurer une rotation pour ne pas étouffer sous un fichier trop gros. »

Aussi si on a un stockage plus sioux des logs, au lieu de supprimer des points de journalisation on pourrait aussi simplement les filtrer dans l'affichage par défaut. Ça n'enlève pas le besoin de réfléchir à ce qu'il est pertinent d'afficher bien sûr.

#7 - 12 mars 2021 11:49 - Mikaël Ates

- Lié à Development #51974: Journal orienté utilisateur des accès aux demandes ajouté

#8 - 01 février 2022 20:10 - Frédéric Péters

- Statut changé de Nouveau à Fermé

- Patch proposed mis à Non

- Planning mis à Non

Fermé via #61292.

Fichiers

0001-qommon-added-basic-statsd-infrastructure.patch	7,04 ko	08 novembre 2012	Frédéric Péters