Authentic 2 - Development #20695

Avoir sur les objets un journal des modifications et sur les utilisateurs, en plus, un journal des actions

14 décembre 2017 12:34 - Mikaël Ates (de retour le 29 avril)

Statut:	Fermé	**Début:**	14 décembre 2017
Priorité:	Immediat	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Fonctionnellement ce qui a été fait pour les utilisateurs dans le CUT.

Demandes liées:

Lié à Authentic 2 - Development #23494: Écran reprenant les connexions d'un u...	**Nouveau**	**29 avril 2018**
Lié à Authentic 2 - Bug #28991: code JQuery du manager: le remplacement d'élé...	**En cours**	**13 décembre 2018**
Lié à Authentic 2 - Bug #29144: permettre l'utilisation du champ natif JSONFi...	**Rejeté**	**17 décembre 2018**
Lié à Authentic 2 - Development #14101: add "audit log" to manager view	**Fermé**	**25 novembre 2016**
Lié à Publik - Development #40832: Produire un journal de l'activité des util...	**Nouveau**	**19 mars 2020**
Lié à Authentic 2 - Development #19718: En BO, indiquer sur la fiche de l'ut...	**Fermé**	**27 octobre 2017**
Lié à Authentic 2 - Development #47155: Avoir sur les objets un journal des ...	**Fermé**	**30 septembre 2020**

Historique

#1 - 14 décembre 2017 16:44 - Mikaël Ates (de retour le 29 avril)

- Sujet changé de Avoir sur les objets un journal des modification et sur les utilisateurs, en plus, un journal des actions à Avoir sur les objets un journal des modifications et sur les utilisateurs, en plus, un journal des actions

#2 - 10 janvier 2018 10:18 - Mikaël Ates (de retour le 29 avril)

- Priorité changé de Normal à Haut

#3 - 29 août 2018 15:34 - Paul Marillonnet

- Lié à Development #23494: Écran reprenant les connexions d'un utilisateur ajouté

#4 - 29 août 2018 15:34 - Paul Marillonnet

- Assigné à mis à Paul Marillonnet

#5 - 29 août 2018 15:51 - Paul Marillonnet

Donc oui, proche (fonctionnellement au moins) du journal d'actions pour un utilisateur ('/manage/users/<id>/actions-journal/'), donc le code est ici : https://git.entrouvert.org/authentic2-cut.git/tree/src/authentic2_cut/models.py#n14

#7 - 25 octobre 2018 16:39 - Paul Marillonnet

- Fichier 0001-WIP-user-actions-and-modifications-journals-20695.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Un simple patch qui introduit les deux vues dans le backoffice.

Je vais voir comment introduire les hooks qui viennent créer les entrées de ces deux journaux.

#8 - 25 octobre 2018 16:39 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

- Patch proposed changé de Oui à Non

#9 - 26 octobre 2018 08:52 - Benjamin Dauvergne

Je vais faire mon casse pied mais je ne souhaite pas partir sur l'exacte réplique de ce qui est fait sur le CUT, pour plusieurs raisons:

- le CUT ne permet pas de lier une ligne de log à plusieurs objets c'est soit l'acteur et c'est un User soit le sujet et c'est un User aussi
- le CUT ne permet pas d'ajouter plus d'informations à la ligne de log

Je souhaiterai partie d'un modèle de donnée un peu plus riche de cette forme, dans une application authentic2_journal:

```
class Line(models.Model):
    timestamp = datetime <- indexé
    kind = positiveinteger (un code, sso, login, registration, etc..) <- indexé
    message = text
    extra = text (du JSON ou null)

class Reference(models.Model):
    timestamp = datetime <- indexé
    line = FK(Line) <- indexé
    representation = Text(max_length=64)  # permet de conserver une idée de l'objet même s'il est supprimé
    target = GFK(target_ct, target_id)
    target_ct = FK(ContentType) <- indexés
    target_id = integer          | ensemble
```

La répétition du timestamp doit permettre de filtrer directement sur Reference dans les vues de Journal, ça devrait aller plus vite, sur des millions de ligne j'ai eu le problème sur docbow, la jointure coûte cher.

Ensuite au niveau IHM avoir une vue un peu générique réutilisable sur les différents objets (en changeant get_queryset, on filtre pour les évènements d'un utilisateur, d'un service ou d'un rôle) qui utilise de la pagination par date et non par index pour que ça aille vite et du chargement au scroll (en arrivant en base de page on charge les nouvelles lignes). Vu utilisable en BO et FO.

Aussi j'aimerai qu'une ligne de log soit représentée ainsi

24/10/2018	SSO de Benjamin Dauvergne vers demarches.publik.fr +

Il faut que les messages soient courts, évite d'être redondant (c'est pas évident on est bien obligé de répéter le nom de l'utilisateur dans le message, même quand est sur sa vue de log personnelle, idem pour les services), mais restent intelligibles.

Dans la barre de droite on doit voir un champ date intitulé "Avant le [....]" et un moyen de filtrer par grands groupes d'évènements (en gros les évènements d'authentification et les évènements de modifications des données (personnelles ou autre, genre changement des membres d'un rôle).

En appuyant sur le plus on déplie la ligne et en ajax on récupère des informations supplémentaires:

- les objets liés
- un rendu du champ extra (dépendra du kind)

24/10/2018	SSO	de Benjamin Dauvergne à demarches.publik.fr +
[User Benjamin Dauvergne] [Service demarches.publik.fr]		
Protocole: OIDC Prompt:none etc.. des détails		

J'aimerai qu'on puisse organiser les kind sous un forme d'arbre, je m'explique:

```
KINDS = {
  _('Authentication'): {
      "members": {
        _('Register'): 1,
        _('Login'): 2
        _('SSO'): 3,
        _('Logout'): 4,
      },
  },
  _('Modifications'): {
        _('Create'): 5,
        _('Modify'): 6,
        _('Modify members'): 7 <-- uniquement pour les rôles,
        _('Delete'): 8,
  }
}
```

Ce n'est pas forcément la forme que ça doit prendre parce qu'en fonction du Kind il faut aussi déduire le rendu du champ extra, et donc ça passera peut-être par des classes.

Au niveau API ça doit s'utiliser simplement:

```
journal.log.sso(user=request.user, service=oidc_client)
```

Je suis pour éviter une méthode générale log() où on va se retrouver avec des messages et des usages différents un peu partout, il faut cadrer:

- log.sso()
- log.login()
- log.logout()
- log.register()
- etc...

#10 - 26 octobre 2018 09:25 - Benjamin Dauvergne

À déterminer aussi si on log en anglais, en français, un mix, si on se fait chier à traduire (c'est un calvaire à faire).

#11 - 26 octobre 2018 09:44 - Mikaël Ates (de retour le 29 avril)

My 2 cents :

- Ces logs vont aussi servir dans un autre ticket pour afficher en FO, à l'usager, le journal de son compte. Ce doit effectivement être intelligible et traduisible.
- Admettons qu'on fasse un jour un module de notification des modifications des utilisateurs, ou de n'importe quel objet, peut-être peut-on envisager que ce module tire les événements de ces logs. Et donc ma seule remarque c'est que ce que tu souhaites avec les kinds permettrait bien cela.

#12 - 26 octobre 2018 10:45 - Benjamin Dauvergne

Alors si on part sur un interface traduisible il faut abandonner l'idée d'avoir des logs utilisables directement (i.e. les message stockés en texte dans la table), on supprime la ligne message, de la colonne kind on trouve la classe kind correspondante, classe qui saura formatter le message en fonction des données dans extra que je renommerai bien content. Ça rend quasi impossible une recherche full text sur les messages mais on peut vraisemblablement faire de la recherche fulltext sur extra, genre si on a

kind	extra
registration-request	{"email": "toto@example.com"}

un recherche full text toto@example.com sur extra renverra bien la ligne, ça me parait bien suffisant.

Ça peut être bien ici d'enumérer les évènements déjà loggé coté cut pour définir les classes de kind dont on a besoin (on peut renommer kind en event d'ailleurs, histoire de coller au jargon déjà en place).

#13 - 08 novembre 2018 11:12 - Paul Marillonnet

- Fichier 0001-WIP-add-journal-application-20695.patch ajouté

Un peu de pseudo-code (avec carrément des parties manquantes), mais pour donner une idée de la direction dans laquelle je pars.
(C'est donc bien d'une nouvelle application authentic2_journal qu'il s'agit.)

Sur ma feuille de route, maintenant, et dans l'ordre :
- faire fonctionner les deux vues du backoffice (pas de pagination ni de chargement au scroll ajax pour l'instant)
- insérer la logique de remplissage des journaux dans les vues des différentes applications a2
- avoir une API qui fonctionne (donc pour préparer le scrolling ajax)
- logique de permissions pour la consultation des journaux, dans le backoffice et dans l'API DRF
- modification des templates pour gérer le scrolling et les différentes options de tri tels que demandés par Benjamin ici.

Est-ce que je me trompe d'approche ? Est-ce que j'oublie quelque chose ?

#14 - 08 novembre 2018 12:47 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Sur ma feuille de route, maintenant, et dans l'ordre :
> - faire fonctionner les deux vues du backoffice (pas de pagination ni de chargement au scroll ajax pour l'instant)

Ok.

> - insérer la logique de remplissage des journaux dans les vues des différentes applications a2

Ok.

> - avoir une API qui fonctionne (donc pour préparer le scrolling ajax)

Je mettrai ça de coté tant que tu n'attaques pas la visu ajax.

> - logique de permissions pour la consultation des journaux, dans le backoffice et dans l'API DRF

Idem.

> - modification des templates pour gérer le scrolling et les différentes options de tri tels que demandés par Benjamin ici.

Mettons cela pour un autre ticket, terminons les éléments de base, avoir les modèles, la définition des types de log et les méthode pour logger insérer les points de logs dans le code existant.

Par contre à la lecture je ne comprends pas pourquoi tu hérites de models.Model sur des tas de classe qui ne sont clairement pas des modèles.

#15 - 16 novembre 2018 19:10 - Paul Marillonnet

- Fichier 0001-WIP-add-journal-application-20695.patch ajouté

Un autre patch WIP pour mon propre suivi de l'affaire.

#16 - 20 novembre 2018 20:29 - Paul Marillonnet

- Fichier 0001-WIP-add-journal-application-20695.patch ajouté

Et donc oui j'aime bien l'idée de représenter la structuration des logs sous forme d'héritages de classe.
Je le représente en tant que modèles Django pour pouvoir manipuler une FK depuis le modèle Reference, mais on peut sans doute s'en passer, je vais voir.

#17 - 21 novembre 2018 10:03 - Thomas Noël

Juste un truc, à regarder le code rapidement : sans doute faut-il penser à conserver les journaux d'un utilisateur qui a été détruit (justement pour savoir quand il l'a été, éventuellement pourquoi). Ca se traduit souvent par une relation lâche dans la base, ie pas une foreignkey, et j'ai l'impression ici que c'est ce qui est fait (GenericForeignKey) ?

(aussi, sur la forme, les "coding: utf-8" ne servent à rien car on code en ascii (ne pas le mettre ça permet de justement vérifier que y'a pas de caractères bizarres qui traînent dans le code) ; en revanche il manque les quelques lignes sur l'agpl mais bon, dans authentic ça se pratique pas vraiment)

#18 - 21 novembre 2018 11:42 - Benjamin Dauvergne

Thomas Noël a écrit :

> Juste un truc, à regarder le code rapidement : sans doute faut-il penser à conserver les journaux d'un utilisateur qui a été détruit (justement pour savoir quand il l'a été, éventuellement pourquoi). Ca se traduit souvent par une relation lâche dans la base, ie pas une foreignkey, et j'ai l'impression ici que c'est ce qui est fait (GenericForeignKey) ?

Dans mon idée plutôt que de surcharger le message d'informations (genre "sso de Benjamin Dauvergne sur portail citoyen"), l'idée était de mettre des informations dans le champs JSON extra, généralement inutilisée, en cas de suppression de l'objet User et donc d'exception ObjectDoesNotExist en cas d'accès à la GenericForeignKey alors on pourra utiliser "extra" pour redonner un contexte au message (on le type d'object via le content_type, la primary key via object_id et le nom dans extra["user_fullname"] par exemple).

> (aussi, sur la forme, les "coding: utf-8" ne servent à rien car on code en ascii (ne pas le mettre ça permet de justement vérifier que y'a pas de caractères bizarres qui traînent dans le code) ; en revanche il manque les quelques lignes sur l'agpl mais bon, dans authentic ça se pratique pas vraiment)

On peut prendre de bonnes habitudes à l'occasion d'une nouvelle application.

#19 - 21 novembre 2018 18:08 - Paul Marillonnet

- Fichier 0001-WIP-add-journal-application-20695.patch ajouté

Voilà, une version avec des champs de contexte (Line.extra et Reference.representation), pour la persistance de l'information même lorsque les objets référencés sont détruits.
J'ai ajouté les entêtes AGPL dans le code.
Les quelques tests présents pour l'instant passent.

Prochaine étape pointée par Mikaël : offrir à l'usager une vue de ses actions dans le FO (sans doute un view servie sur /accounts/journal/)
→ Avis au(x) relecteurice(s) : Est-ce qu'on fait un ticket à part pour cette feature purement FO ? Ou bien je peux mettre cette vue dans ce patch aussi ?

#20 - 21 novembre 2018 19:15 - Paul Marillonnet

- Fichier 0001-WIP-add-journal-application-20695.patch ajouté

Une très légère modification sur utils.log_event plus proche, je crois, de ce qu'on cherche à avoir.

#21 - 22 novembre 2018 13:06 - Paul Marillonnet

- Fichier 0001-WIP-add-journal-application-20695.patch ajouté

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

Avec des tests sur les vues FO (/accounts/journal/) et BO (/manager/users/<pk>/journal/).
L'internationalisation ne va pas, je corrige ça asap.

#22 - 22 novembre 2018 13:06 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

- Patch proposed changé de Oui à Non

#23 - 22 novembre 2018 21:09 - Paul Marillonnet

- Fichier 0001-WIP-add-journal-application-20695.patch ajouté

Avec une i18n plus correcte, je crois.
Je vais écrire les tests s'assurant du bon fonctionnement de la localisation.

Ensuite, pas forcément dans l'ordre :
- placer les instructions de génération des logs au bon endroit dans les vues A2
- placer les URLs, les vues et les templates des journaux dans l'application principale A2 (seule la logique générique et interne à la gestion de ces journaux resterait dans l'appli authentic2_journal)
- remplacer les fonctions dans authentic2_journal.event_logger par des méthodes d'une classe, pour se laisser la possibilité future de redéfinir le comportement de génération des log (par héritage de la classe et redéfinition des méthodes concernées).

#24 - 26 novembre 2018 17:04 - Paul Marillonnet

- Fichier 0001-WIP-add-journal-application-20695.patch ajouté

Paul Marillonnet a écrit :

> - remplacer les fonctions dans authentic2_journal.event_logger par des méthodes d'une classe, pour se laisser la possibilité future de redéfinir le comportement de génération des log (par héritage de la classe et redéfinition des méthodes concernées).

Ça donnerait quelque chose comme ça.
Je serais aussi pour la définition d'un paramètre A2_JOURNAL_EVENT_LOGGER, définissant la classe à utiliser, et une fonction authentic2_journal.utils.get_event_logger_class() renvoyant la valeur définie pour ce paramètre.

#25 - 26 novembre 2018 17:32 - Paul Marillonnet

- Fichier 0001-WIP-add-journal-application-20695.patch ajouté

Paul Marillonnet a écrit :

> Je serais aussi pour la définition d'un paramètre A2_JOURNAL_EVENT_LOGGER, définissant la classe à utiliser, et une fonction authentic2_journal.utils.get_event_logger_class() renvoyant la valeur définie pour ce paramètre.

Comme ça je pense.

#26 - 27 novembre 2018 15:10 - Paul Marillonnet

- Fichier 0001-WIP-add-journal-application-20695.patch ajouté

Paul Marillonnet a écrit :

> - placer les URLs, les vues et les templates des journaux dans l'application principale A2 (seule la logique générique et interne à la gestion de ces journaux resterait dans l'appli authentic2_journal)

Voilà.
Même si on ne fait pas, dans ce ticket, du chargement dynamique lors du défilement ('infinite scrolling'), je voudrais ajouter de la pagination sur la liste des événements, en plus d'un filtrage sur leur horodatage.

#27 - 27 novembre 2018 15:15 - Frédéric Péters

> chargement dynamique lors du défilement ('infinite scrolling')

Référence perdue mais sur ce sujet, il faut arrêter de charger lorsque le scroll touche le bas de page, il est mieux d'être explicite, genre un bouton "charger les 10 événéments suivants" (qui les ajoute à la page actuelle, oui).

#28 - 27 novembre 2018 15:17 - Paul Marillonnet

Frédéric Péters a écrit :

> > chargement dynamique lors du défilement ('infinite scrolling')
>
> Référence perdue mais sur ce sujet, il faut arrêter de charger lorsque le scroll touche le bas de page, il est mieux d'être explicite, genre un bouton "charger les 10 événéments suivants" (qui les ajoute à la page actuelle, oui).

Ok, je comprends. Si je fais un ticket pour l'ajout dynamique d'entrée à la liste des événements, je partirai là-dessus.

#29 - 27 novembre 2018 16:58 - Paul Marillonnet

Paul Marillonnet a écrit :

> en plus d'un filtrage sur leur horodatage.

Ma vision des choses pour y parvenir :
- faire en sorte que la vue de journal accepte deux paramètres de after et before dans la query string.
- rendre dans la vue deux champs DateTime sortis de tout formulaire, possiblement rendus à l'aide du DateTimeWidget d'A2, et faire en sorte que leur utilisation par l'usager recharge la page avec les after et before paramétrés en conséquence.

Je me plante ?

#30 - 27 novembre 2018 17:53 - Paul Marillonnet

- Fichier 0002-WIP-add-timewindow-filter-for-profile-journal-20695.patch ajouté

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

Paul Marillonnet a écrit :

> Je me plante ?

Finalement j'ai fait comme ça (patch WIP 0002 qui s'applique sur le dernier patch WIP posé ici).
J'écris les tests maintenant.

#31 - 27 novembre 2018 17:54 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

- Patch proposed changé de Oui à Non

#32 - 27 novembre 2018 18:25 - Benjamin Dauvergne

- L'absence de classes pour représenter les "kind" me chagrine (ça me parait essentiel pour avancer ensuite, même si ça fait juste le formatage bateau qu'il y a là pour l'instant)
- Il ne faut pas récupérer les erreurs lors des log_event() (si ça plante là c'est bien embêtant en fait).
- La gestion de la pagination est à faire coté vue générique (toutes les vues de log voudront cela) via un système de curseur:
  - on définit en général un identifiant unique ordonnable, exemple (timestamp, id), on trie via .order_by('timestamp', 'id') ou .order_by('-timestamp', '-id'), pour chaque page on génère un curseur de début et de fin (en fait le timestamp,id du premier et dernier élément affiché, si on clique sur précédent on fait (pseudo-code) ?before=cursor(<first_row>) sur suivant ?after=cursor(<first-row>), sur un filtrage par date on générera un ?after=<ladate>,0
  - le filtrage pour un ?after=timestamp,id sera

```
.order_by('timestamp', 'id').filter(Q(timestamp__gt=timestamp)|(Q(timestamp=timestamp, id__gt=id))[:page_size]
```

- le filtrage pour un ?before=timestamp,id sera

```
.order_by('-timestamp', '-id').filter(Q(timestamp__lt=timestamp)|(Q(timestamp=timestamp, id__lt=id))[:page_size]
```

, et on prendra soin de faire un reversed sur les données récupérées avant leur affichage (ou en tout cas de les parcourir à l'envers parce qu'on va du plus récent au plus ancien avec ce ordre et qu'on affiche je suppose dans l'ordre chronologique)
- sur un clique sur précédent/suivant il faut virer le filtrage par date (ré-initialiser le champ qui sert à cela)
- en entrée sur la vue on devrait faire un ?before=<now> + 1h,0 pour pointer sur les derniers évènements
- à mon avis il faut prévoir tout de suite un affichage anti-chronologique possible, parce que pour la vue des derniers évènements d'un utilisateur c'est ce qu'on voudra (pour un objet de config par contre non, genre l'historique d'un OIDCProvider ou LibertyProvider)

#33 - 29 novembre 2018 10:28 - Paul Marillonnet

Benjamin Dauvergne a écrit :

- L'absence de classes pour représenter les "kind" me chagrine (ça me parait essentiel pour avancer ensuite, même si ça fait juste le formatage bateau qu'il y a là pour l'instant)

Je ne suis pas sûr de comprendre ce que tu as en tête.
En l'état, la journalisation s'appuie sur un modèle EventKind, dont les objets sont retrouvés au moment de la création des logs (création qui elle même est prise en charge pas une classe EventLogger, ajustable par un paramètre de setting A2_JOURNAL_EVENT_LOGGER).
Je passe à côté de quelque chose visiblement, mais je ne vois pas, pour l'instant, en quoi le modèle actuel fige la gestion du journal, et ne permet pas d'avancer par la suite.

#34 - 29 novembre 2018 11:06 - Benjamin Dauvergne

Je suis pour virer app_settings, il n'y a rien à configurer.

Ce que j'imagine en pseudo-pseudo code

```
_kind_registry = ()

class KindMetaclass(type):
     def __init__(self, name, bases, d):
         super(KindMetaclass, self).__init__(name, bases, d)
         if name != "BaseKind":
             assert set(d.keys()) >= set(['name', 'template_names'])
             assert d['name'] not in _kind_registry
            _kind_registry[d['name']] = self

class BaseKind(object):
     __metaclass__ = KindMetaclass
     long_template_names = []

     def format(self, line):
         return select_template(self.template_names).render(Context(line.extra_json))

     def long_format(self, line):
         if not self.long_template_names:
             return ''
         return select_template(self.long_template_names).render(Context(line.extra_json))

     def log_real(self, **kwargs):
         Line.objects.create(kind=get_kind_model(self.name), extra=json.dumps(kwargs))

# le modèle Kind n'a pas grande utilité, il diminue juste la taille de la
# colonne Kind dans Line (un entier au d'un varchar), mais impose une
# jointure (petite) en plus, ça pourrait être pratique d'ajouter un .select_related('kind')
# implicite le ModelManager de Line.objects

@GlobalCache
def get_kind_model(name):
    return Kind.objects.get(name=name)

class SSOKind(Kind):
     name = 'sso'
     template_names = ['authentic2_journal/sso.html']
     # dans sso.html
     # {% blocktrans %}sso on {{service}}{% endblocktrans %}
```

```
     def log(self, user, service):
         # get_kind() should be cached, like ContentType are, because they never vary
         return self.log_real(user_id=user.id, service_id=service.id, user='%s' % user, service='%s' % service
)

...

class Kind(models.Model):
...
    @property
    def concrete(self):
         return _kind_registry[self.name]

...

class Line(models.Model):
...
    def format(self):
        return self.kind.concrete.format(self)

    def long_format(self):
        return self.kind.concrete.long_format(self)

class Logger(self):
    def __init__(self, kind):
        self.kind = _kind_registry[kind]

    def __call__(self, **kwargs):
        return self.kind.log(**kwargs)

class LoggerHub(object):
    def __getattr__(self, kind_name):
        return Logger(kind_name)

logger = LoggerHub()
# s'utilise ainsi
# logger.sso(user=user, service=service)
```

Si on souhaite ajouter de nouveaux comportements au kind ou les faire varier en fonction du type de kind ça nous donne tout l'espace pour le faire.

#35 - 04 décembre 2018 11:37 - Paul Marillonnet

- Fichier 0001-WIP-add-journal-application-20695.patch ajouté

Ok, compris, merci pour le bout de pseudo-code, j'ai inclus ça dans mon patch, avec toujours les quelques mêmes tests.
Je passe à la pagination.

#36 - 05 décembre 2018 17:15 - Paul Marillonnet

- Fichier 0001-WIP-add-journal-application-20695.patch ajouté

Un début de pagination, mais pour laquelle le code de chargement ajax du contenu de la page (au lieu du rafraichissement complet de celle-ci) ne fonctionne pas. Je n'ai toujours pas compris pourquoi.
Le code en question est ici : http://git.entrouvert.org/authentic.git/tree/src/authentic2/manager/static/authentic2/manager/js/manager.js#n108

Ça me chiffonne. Je continue de chercher.

#37 - 13 décembre 2018 14:20 - Paul Marillonnet

Dans le pad eo-dev de cette semaine, Frédéric écrivait :

> s'il y a juste besoin de date je suggérerais désormais d'utiliser input type=date (cf chrono)

Je pense qu'on a aussi besoin de sélectionner l'heure (si jamais par exemple un veut filtrer les logs sur une journée particulièrement chargée en actions / modifications pour un usager), auquel cas un input type="date" ne serait pas suffisant, non ?

#38 - 13 décembre 2018 14:26 - Paul Marillonnet

Encore dans le pad eo-dev de cette semaine, Benjamin écrivait :

> moz propose du code assez simple pour couvrir tous les cas

Je crois que tu parles de cette documentation mozilla, non ?

https://developer.mozilla.org/fr/docs/Web/HTML/Element/input/datetime-local

Et du coup en effet, c'est coompatible Chrome, Edge et Opera, mais pas Firefox…

Qu'est-ce qu'on fait ? Est-ce qu'on utilise le DateTimePicker d'A2 ?

#39 - 13 décembre 2018 14:34 - Frédéric Péters

> Je pense qu'on a aussi besoin de sélectionner l'heure (si jamais par exemple un veut filtrer les logs sur une journée particulièrement chargée en actions / modifications pour un usager), auquel cas un input type="date" ne serait pas suffisant, non ?

~~Faut se calmer, on parle des actions d'un utilisateur. Et à étendre l'idée à un journal de log couvrant toute la plateforme (mais on ne veut pas ça dès ce ticket, je pense), même, je serais en fait pour rien du tout, et s'il fallait quelque chose juste un champ recherche, comme dans passerelle (et s'il détecte une date il filtre sur la date et s'il détecte date/heure sur date/heure).~~

Benjamin écrivait :

> Ensuite au niveau IHM avoir une vue un peu générique réutilisable sur les différents objets (en changeant get_queryset, on filtre pour les évènements d'un utilisateur, d'un service ou d'un rôle) qui utilise de la pagination par date et non par index pour que ça aille vite et du chargement au scroll (en arrivant en base de page on charge les nouvelles lignes). Vu utilisable en BO et FO.

Suivre Benjamin, ne te laisse pas guider par mes remarques.

#40 - 13 décembre 2018 14:40 - Benjamin Dauvergne

Mozilla propose un code pas compatible firefox ? :) Ah mon avis il y a incompréhension, mais de toute façon je signalais ça de manière un peu anecdotique ça ne me choque pas que sur un navigateur ancien on soit obligé de taper « 10/12/2019 20:30 » à la main, donc si tu veux commencer à déprécier bootstrap-datetimepicker et si tout le monde est ok avec ça.
J'ai pas étudié le taux de pénétration, et donc sur https://caniuse.com/#feat=input-datetime je vois qu'on perd juste safari desktop et que sur firefox on a pas datetime-local ce qui est gênant.

On pourrait simplement utiliser un https://docs.djangoproject.com/fr/2.1/ref/forms/fields/#splitdatetimefield avec un champs date et un champs time séparés, c'est bien supporté partout; enfin non pas sous edge, bon ben faut choisir entre mal supporté firefox ou mal supporté edge, ou alors garder bootstrap-datetimepicker.

#41 - 13 décembre 2018 14:42 - Benjamin Dauvergne

Oui c'est vrai on divague un peu, la priorité c'est la pagination (la vue servira pour l'instant qu'à des objets distincts ayant peu d'historique en majorité), le filtrage on pourra revenir dessus.

#42 - 13 décembre 2018 16:43 - Paul Marillonnet

Benjamin Dauvergne a écrit :

> Oui c'est vrai on divague un peu, la priorité c'est la pagination (la vue servira pour l'instant qu'à des objets distincts ayant peu d'historique en majorité), le filtrage on pourra revenir dessus.

Ok très bien, ça me va. J'ai donc un début de pagination mais, comme mentionné plus tôt (#20695-36), je me prends les pieds dans la compatibilité avec le bout de code ajax pour le rafraichissement du contenu du tableau lors du changement de page. Je continue chercher.

J'ai aussi une drôle de différence de comportement entre le backend postgres (qui fonctionne) et le backend sqlite (qui lui m'envoie une sqlite3.InterfaceError à l'appel des fonctions de log. Je vais aussi chercher pourquoi, une fois le problème JS résolu.

#43 - 13 décembre 2018 17:59 - Paul Marillonnet

Paul Marillonnet a écrit :

> je me prends les pieds dans la compatibilité avec le bout de code ajax pour le rafraichissement du contenu du tableau lors du changement de page. Je continue chercher.

C'était ça : #28991.

#44 - 13 décembre 2018 17:59 - Paul Marillonnet

- Lié à Bug #28991: code JQuery du manager: le remplacement d'éléments cibles dans la fonction update_content ne doit prendre qu'un élément en entrée ajouté

#45 - 17 décembre 2018 19:25 - Paul Marillonnet

Un autre bogue bloquant, #29144, patch à appliquer à moins de n'utiliser que django-jsonfield. À voir donc.

#46 - 17 décembre 2018 19:25 - Paul Marillonnet

- Lié à Bug #29144: permettre l'utilisation du champ natif JSONField au lieu de celui issu de django-jsonfield ajouté

#47 - 17 décembre 2018 21:14 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Un autre bogue bloquant, #29144, patch à appliquer à moins de n'utiliser que django-jsonfield. À voir donc.

Quel bug ? Où ça ? Tu ne rapportes rien ici.

#48 - 18 décembre 2018 09:26 - Paul Marillonnet

Pardon, manque d'information.
Je reproduis le bogue reporté dans https://code.djangoproject.com/ticket/27675, à savoir un champ django.contrib.postgres.fields.jsonb.JSONField qui devient une str au moment de la lecture de la valeur du l'attribut. En l'occurrence l'attribut est Line.extra_json.
Lorsque je déprécie l'utilisation de django-jsonfield (seulement utilisé dans le modèle du module d'authentification OIDC) au profit de ce champ natif JSONField, ce bogue ne survient plus.
Est-ce que tu veux que je fournisse une procédure pour reproduire le bogue ?

#49 - 18 décembre 2018 16:30 - Benjamin Dauvergne

Surtout je ne veux pas qu'on abandonne django-jsonfield, on a trop d'utilisateurs Django 1.8 encore, ensuite le jwkset d'OIDCProvider a une utilité, stocker la clé publique d'un IdP OIDC, donc utilise django-jsonfield ici.

#50 - 18 décembre 2018 17:06 - Paul Marillonnet

- Fichier 0001-WIP-add-journal-application-20695.patch ajouté

Avec une pagination toute bête côté /accounts/.

#51 - 19 décembre 2018 12:17 - Paul Marillonnet

- Fichier 0004-add-backoffice-per-user-journal-20695.patch ajouté

- Fichier 0003-add-user-profile-journal-20695.patch ajouté

- Fichier 0002-manager.js-jquery-replacewith-function-takes-one-sou.patch ajouté

- Fichier 0001-add-journal-application-20695.patch ajouté

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

Un découpage un peu plus correct du code. J'ai rajouté le patch de #28991 parce que le code côté manager ne fonctionnera pas sans. Je vais refaire un coup de débogueur pour m'assurer que le problème n'est pas entre la chaise et le clavier.

Il y a le support un filtrage par timestamp côté /accounts/, à l'aide des paramètres de querystring before et after. Je ne sais pas où on en est des exigences là-dessus, si on le laisse de côté pour l'instant, où si au contraire on le souhaite aussi (ou surtout) côté BO.

#52 - 19 décembre 2018 12:19 - Paul Marillonnet

- Fichier 0005-test-profile-and-manager-journal-views-20695.patch ajouté

Et un peu de tests sur les vues.

Je vais ajouter un dernier patch, pour l'insertion des instructions de log dans les vues A2 déjà existantes.

#53 - 20 décembre 2018 10:13 - Paul Marillonnet

> Je vais ajouter un dernier patch, pour l'insertion des instructions de log dans les vues A2 déjà existantes.

Et donc, je voyais bien la suite comme ça.
On est bien d'accord ?
Ou on peut viser plus concis, comme définir un décorateur de méthode (voire de classe), pour éviter à chaque fois

```
def foo(self, ...)
    bar = super(Whatever, self).foo(...)
    logger.some_action(...)
    return bar
```

?

#54 - 20 décembre 2018 10:13 - Paul Marillonnet

- Fichier 0006-manager-add-log-creation-instructions-in-ou-views-20.patch ajouté

Avec le patch c'est mieux...

#55 - 20 décembre 2018 11:47 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

Oups, je me rends compte que j'ai oublié les deux templates des vues /accounts/journal et /manage/users/<pk>/journal.
Je les inclus et repropose les patchs ici.

#56 - 20 décembre 2018 11:58 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> > Je vais ajouter un dernier patch, pour l'insertion des instructions de log dans les vues A2 déjà existantes.
>
> Et donc, je voyais bien la suite comme ça.
> On est bien d'accord ?
> Ou on peut viser plus concis, comme définir un décorateur de méthode (voire de classe), pour éviter à chaque fois
> [...]
> ?

Je suis un peu gêné que ça s'appelle logger, je préfèrerai qu'on reste sur une dénomination journal pour pas mélanger avec les loggers Python (et pour pas multiplier les codes ce serait bien que ça log aussi vers logging pour pas avoir des tonnes de code du genre:

```
journal.sso(user=user, service=service)
logger.info('sso de %s sur %s, user, service)
```

)

#57 - 20 décembre 2018 12:32 - Paul Marillonnet

- Fichier 0005-manager-add-log-creation-instructions-in-ou-views-20.patch ajouté

- Fichier 0004-test-profile-and-manager-journal-views-20695.patch ajouté

- Fichier 0003-add-backoffice-per-user-journal-20695.patch ajouté

- Fichier 0002-add-user-profile-journal-20695.patch ajouté

- Fichier 0001-add-journal-application-20695.patch ajouté

- Statut changé de En cours à Solution proposée

Ok.

#58 - 13 janvier 2019 20:33 - Frédéric Péters

> journal.sso(user=user, service=service)

Là ça passe encore mais je trouve journal.update(self.request.user, self.object) plutôt obscur (dans 0005), déjà oui, comme dans l'exemple, nommer les arguments.

Mais aussi je trouverais plus lisible d'avoir journal.record_update, et record_creation, et record_deletion. (ça fait vraiment bizarre pour moi d'avoir une méthode qui soit un verbe mais qu'en fait le verbe ne s'applique pas à l'objet, e.g. journal.delete(...) ça donne trop l'impression d'une suppression d'un truc du journal).

#59 - 17 janvier 2019 10:58 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

Je suis sur une erreur, après application de #29193, et tentative d'exécution des tests :

```
/tmp/tox-paul/authentic/py2-coverage-dj18-authentic-pg/local/lib/python2.7/site-packages/pytest_django/fixture
s.py:108:
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
/tmp/tox-paul/authentic/py2-coverage-dj18-authentic-pg/local/lib/python2.7/site-packages/django/test/runner.py
:370: in setup_databases
    serialize=connection.settings_dict.get("TEST", {}).get("SERIALIZE", True),
/tmp/tox-paul/authentic/py2-coverage-dj18-authentic-pg/local/lib/python2.7/site-packages/django/db/backends/ba
se/creation.py:368: in create_test_db
    test_flush=not keepdb,
/tmp/tox-paul/authentic/py2-coverage-dj18-authentic-pg/local/lib/python2.7/site-packages/django/core/managemen
t/__init__.py:120: in call_command
    return command.execute(*args, **defaults)
/tmp/tox-paul/authentic/py2-coverage-dj18-authentic-pg/local/lib/python2.7/site-packages/django/core/managemen
t/base.py:445: in execute
    output = self.handle(*args, **options)
/tmp/tox-paul/authentic/py2-coverage-dj18-authentic-pg/local/lib/python2.7/site-packages/django/core/managemen
t/commands/migrate.py:179: in handle
    created_models = self.sync_apps(connection, executor.loader.unmigrated_apps)
/tmp/tox-paul/authentic/py2-coverage-dj18-authentic-pg/local/lib/python2.7/site-packages/django/core/managemen
t/commands/migrate.py:318: in sync_apps
    cursor.execute(statement)
/tmp/tox-paul/authentic/py2-coverage-dj18-authentic-pg/local/lib/python2.7/site-packages/django/db/backends/ut
ils.py:62: in execute
    return self.cursor.execute(sql)
/tmp/tox-paul/authentic/py2-coverage-dj18-authentic-pg/local/lib/python2.7/site-packages/django/db/utils.py:98
: in __exit__
    six.reraise(dj_exc_type, dj_exc_value, traceback)
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

self = <django.db.backends.utils.CursorWrapper object at 0x7fe404d88d50>
sql = 'ALTER TABLE "authentic2_journal_reference" ADD CONSTRAINT "authent_target_ct_id_11782e361e83292f_fk_dja
ngo_content_type_id" FOREIGN KEY ("target_ct_id") REFERENCES "django_content_type" ("id") DEFERRABLE INITIALLY
 DEFERRED'
params = None

    def execute(self, sql, params=None):
        self.db.validate_no_broken_transaction()
        with self.db.wrap_database_errors:
            if params is None:
>               return self.cursor.execute(sql)
E               ProgrammingError: ERREUR:  la relation « django_content_type » n'existe pas

/tmp/tox-paul/authentic/py2-coverage-dj18-authentic-pg/local/lib/python2.7/site-packages/django/db/backends/ut
ils.py:62: ProgrammingError
```

je cherche.

#60 - 17 janvier 2019 14:31 - Paul Marillonnet

- Fichier 0001-add-journal-application-20695.patch ajouté

- Fichier 0002-add-user-profile-journal-20695.patch ajouté

- Fichier 0003-add-backoffice-per-user-journal-20695.patch ajouté

- Fichier 0004-test-profile-and-manager-journal-views-20695.patch ajouté

- Fichier 0005-manager-add-log-creation-instructions-in-ou-views-20.patch ajouté

- Statut changé de En cours à Solution proposée

Frédéric Péters a écrit :

> Là ça passe encore mais je trouve journal.update(self.request.user, self.object) plutôt obscur (dans 0005), déjà oui, comme dans l'exemple, nommer les arguments.
>
> Mais aussi je trouverais plus lisible d'avoir journal.record_update, et record_creation, et record_deletion. (ça fait vraiment bizarre pour moi d'avoir une méthode qui soit un verbe mais qu'en fait le verbe ne s'applique pas à l'objet, e.g. journal.delete(...) ça donne trop l'impression d'une suppression d'un truc du journal).

Oui c'est vrai. C'est corrigé ici.

#61 - 01 avril 2019 10:41 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

- Assigné à changé de Paul Marillonnet à Benjamin Dauvergne

#65 - 05 février 2020 13:08 - Frédéric Péters

- Lié à Development #14101: add "audit log" to manager view ajouté

#66 - 19 mars 2020 15:37 - Mikaël Ates (de retour le 29 avril)

- Lié à Development #40832: Produire un journal de l'activité des utilisateurs dans le backoffice. ajouté

#67 - 20 mars 2020 11:11 - Frédéric Péters

- Lié à Development #40850: Accéder à son historique FC ajouté

#68 - 20 mars 2020 11:17 - Benjamin Dauvergne

- Lié à Development #40850: Accéder à son historique FC supprimé

#69 - 09 juillet 2020 17:33 - Mikaël Ates (de retour le 29 avril)

- Priorité changé de Haut à Immediat

#71 - 26 juillet 2020 14:26 - Frédéric Péters

- Lié à Development #19718: En BO, indiquer sur la fiche de l'utilisateur la date de suspension du compte ajouté

#72 - 30 septembre 2020 06:47 - Benjamin Dauvergne

- Lié à Development #47155: Avoir sur les objets un journal des modifications et sur les utilisateurs, en plus, un journal des actions ajouté

#73 - 30 septembre 2020 06:47 - Benjamin Dauvergne

- Statut changé de En cours à Fermé

Fichiers

0001-WIP-user-actions-and-modifications-journals-20695.patch	8 ko	25 octobre 2018	Paul Marillonnet
0001-WIP-add-journal-application-20695.patch	16,1 ko	08 novembre 2018	Paul Marillonnet
0001-WIP-add-journal-application-20695.patch	10,1 ko	16 novembre 2018	Paul Marillonnet
0001-WIP-add-journal-application-20695.patch	12,1 ko	20 novembre 2018	Paul Marillonnet
0001-WIP-add-journal-application-20695.patch	27,4 ko	21 novembre 2018	Paul Marillonnet
0001-WIP-add-journal-application-20695.patch	27,7 ko	21 novembre 2018	Paul Marillonnet
0001-WIP-add-journal-application-20695.patch	30,6 ko	22 novembre 2018	Paul Marillonnet
0001-WIP-add-journal-application-20695.patch	33,2 ko	22 novembre 2018	Paul Marillonnet
0001-WIP-add-journal-application-20695.patch	33,9 ko	26 novembre 2018	Paul Marillonnet
0001-WIP-add-journal-application-20695.patch	34,8 ko	26 novembre 2018	Paul Marillonnet
0001-WIP-add-journal-application-20695.patch	36,8 ko	27 novembre 2018	Paul Marillonnet
0002-WIP-add-timewindow-filter-for-profile-journal-20695.patch	3,42 ko	27 novembre 2018	Paul Marillonnet
0001-WIP-add-journal-application-20695.patch	40 ko	04 décembre 2018	Paul Marillonnet
0001-WIP-add-journal-application-20695.patch	40,1 ko	05 décembre 2018	Paul Marillonnet
0001-WIP-add-journal-application-20695.patch	40 ko	18 décembre 2018	Paul Marillonnet
0004-add-backoffice-per-user-journal-20695.patch	1,99 ko	19 décembre 2018	Paul Marillonnet
0003-add-user-profile-journal-20695.patch	5,29 ko	19 décembre 2018	Paul Marillonnet
0002-manager.js-jquery-replacewith-function-takes-one-sou.patch	1,57 ko	19 décembre 2018	Paul Marillonnet
0001-add-journal-application-20695.patch	29,6 ko	19 décembre 2018	Paul Marillonnet
0005-test-profile-and-manager-journal-views-20695.patch	2,26 ko	19 décembre 2018	Paul Marillonnet
0006-manager-add-log-creation-instructions-in-ou-views-20.patch	2,25 ko	20 décembre 2018	Paul Marillonnet
0005-manager-add-log-creation-instructions-in-ou-views-20.patch	2,25 ko	20 décembre 2018	Paul Marillonnet
0004-test-profile-and-manager-journal-views-20695.patch	2,26 ko	20 décembre 2018	Paul Marillonnet
0003-add-backoffice-per-user-journal-20695.patch	2,74 ko	20 décembre 2018	Paul Marillonnet
0002-add-user-profile-journal-20695.patch	6,33 ko	20 décembre 2018	Paul Marillonnet

0001-add-journal-application-20695.patch	29,6 ko	20 décembre 2018	Paul Marillonnet
0002-add-user-profile-journal-20695.patch	6,33 ko	17 janvier 2019	Paul Marillonnet
0001-add-journal-application-20695.patch	32,2 ko	17 janvier 2019	Paul Marillonnet
0003-add-backoffice-per-user-journal-20695.patch	2,74 ko	17 janvier 2019	Paul Marillonnet
0004-test-profile-and-manager-journal-views-20695.patch	2,31 ko	17 janvier 2019	Paul Marillonnet
0005-manager-add-log-creation-instructions-in-ou-views-20.patch	2,28 ko	17 janvier 2019	Paul Marillonnet