

Authentic 2 - Development #20698

Fournir les statistiques authentic en olap pour exploitation via bi-joe.

14 décembre 2017 12:46 - Mikaël Ates (de retour le 29 avril)

Statut:	Rejeté	Début:	14 décembre 2017
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Non		
Description Création de comptes Validation d'identité SSO réussis (échoués)			
Demandes liées:			
Lié à BiJoe - Development #20762: Pouvoir utiliser des décomptes pré-calculer		Nouveau	18 décembre 2017
Lié à Publik - Development #19742: Statistiques sur les comptes et les connex...		Fermé	29 octobre 2017

Historique

#1 - 15 décembre 2017 18:30 - Paul Marillonnet

Je vais regarder comment BiJoe est implémenté.
Ce serait plutôt une nouvelle API, à fournir à BiJoe, c'est ça ?

#2 - 15 décembre 2017 18:47 - Benjamin Dauvergne

On ne modifie pas bijoe, on écrit un outil authentic-olap qui vient générer ou mettre à jour une base et on fournit le schéma de la base à bijoe qui saura alors faire des graphiques.

#3 - 15 décembre 2017 18:57 - Paul Marillonnet

Est-ce qu'il y a déjà parmi les briques Publik du code qui vient remplir ce genre de base à l'intention de BiJoe ?

#4 - 15 décembre 2017 18:59 - Benjamin Dauvergne

Oui voir <https://dev.entrouvert.org/projects/publik-bi>

#5 - 15 décembre 2017 19:00 - Benjamin Dauvergne

Le code dans wcs_olap/feeder.py il construit la base et le fichier schéma (sérialisé en JSON). Le format du fichier schéma est décrit dans le README de Bijoe.

#6 - 18 décembre 2017 10:59 - Paul Marillonnet

Merci.
Je vais aussi regarder comment sont structurés les projets authentic2-*, et la façon dont ils viennent s'articuler autour d'authentic.

Je ne sais pas quelle est la meilleure procédure à suivre pour la revue de code pour ce projet "from scratch". Est-ce que :

- j'envoie ici plusieurs versions successives d'un unique patch contenant tout le code ?
- je découpe en plusieurs patches ?
- je pousse dans une branche wip/20698 sur (sur le dépôt nouvellement créé authentic2-olap.git) ?

#7 - 18 décembre 2017 11:08 - Frédéric Péters

authentic2-* sont surtout des projets plugins à Authentic, ici c'est différent et ça pourrait plutôt s'approcher de <https://git.entrouvert.org/wcs-olap.git/>

Et de là, nouveau projet, tu peux pousser dans une branche wip/ tranquillement.

#8 - 18 décembre 2017 11:23 - Benjamin Dauvergne

Paul Marillonnet a écrit :

Merci.

Je vais aussi regarder comment sont structurés les projets authentic2-*, et la façon dont ils viennent s'articuler autour d'authentic.

Je ne sais pas quelle est la meilleure procédure à suivre pour la revue de code pour ce projet "from scratch". Est-ce que :

- j'envoie ici plusieurs versions successives d'un unique patch contenant tout le code ?
- je découpe en plusieurs patches ?
- je pousse dans une branche wip/20698 sur (sur le dépôt nouvellement créé authentic2-olap.git) ?

Oula jeune padawan avant même d'aller jusque là tu devrais te demander si tu as les données nécessaires pour l'instant tout ce qu'on a c'est le nombre de compte à un moment T via `User.objects.count()` j'irai un poil plus loin en explorant le nombre au temps T pour chaque OU, ça nous donne déjà une information supplémentaire, on pourrait aussi exporter le nombre d'utilisateurs par rôle.

Ensuite coté bijoe on ne sait pas faire de statistiques sur des mesures de compte non additive, en gros on sait juste compter des événements unique (1 formulaires, 2 formulaires, etc..) pas une mesure régulière d'un décompte alors il faudrait apprendre à bijoe à faire cela d'abord ou alors transformer la données en gardant en mémoire le décompte au temps T-1.

J'ai ouvert le ticket qui correspond sur bijoe [#20762](#).

#9 - 18 décembre 2017 11:23 - Benjamin Dauvergne

- Lié à [Development #20762](#): Pouvoir utiliser des décomptes pré-calculer ajouté

#10 - 18 décembre 2017 12:51 - Benjamin Dauvergne

Ensuite il faudra se demander comment obtenir le nombre de SSO, coté CUT c'est fait ainsi:

- redis est utilisé comme système de cache et on peut obtenir une connection Redis pure (donnant accès aux commandes Redis) via ce code:

```
@property
def redis_client(self):
    from django.core import cache
    return cache.cache.client.get_client()
```

- à chaque événement on exécute le code suivant:

```
def stat(self, *args):
    t = now()
    key = 'stat__%s%02d__' % (t.date(), t.hour)
    key += '__'.join(args)
    self.redis_client.expire(key, 3600 * 24 * 7)
    self.redis_client.incr(key)
```

comme ceci

```
self.stat('login', method)
```

et on se retrouve avec les clés suivantes:

```
975) "stat__2017-11-2816__login__AGENT"
976) "stat__2017-11-2310__sso-success__LYO_Espace_personel"
977) "stat__2017-12-0717__sso-request__moncompte"
978) "stat__2017-12-1723__login__FC"
979) "stat__2017-12-1709__login__PWD"
980) "stat__2017-12-1708__sso-request__LYO_Espace_personel"
981) "stat__2017-12-1721__sso-success__LYO_Espace_personel__PWD"
982) "stat__2017-12-1414__sso-request__LYO_Espace_personel"
983) "stat__2017-12-1108__login__AGENT"
984) "stat__2017-12-1201__login__PWD"
985) "stat__2017-12-0711__login__AGENT"
986) "stat__2017-12-1611__login__FC"
987) "stat__2017-12-1514__sso-success__LYO_Espace_personel__FC"
988) "stat__2017-12-1619__sso-success__LYO_Espace_personel__FC"
989) "stat__2017-12-1621__sso-success__LYO_Espace_personel__FC"
990) "stat__2017-12-1113__login__PWD"
991) "stat__2017-12-1011__sso-success__moncompte__PWD"
992) "stat__2017-12-1416__sso-request__moncompte"
993) "stat__2017-11-2017__login__PASSWORD-ON-HTTPS"
994) "stat__2017-12-1715__login__PWD"
995) "stat__2017-12-1521__sso-request__LYO_Espace_personel"
```

On peut continuer comme cela mais cela suppose de standardiser nos déploiement sur une utilisation de Redis ou alors voir comment faire

autrement.

#11 - 12 octobre 2018 08:54 - Frédéric Péters

- Lié à *Development* #19742: *Statistiques sur les comptes et les connexions ajouté*

#12 - 08 décembre 2020 10:07 - Valentin Deniaud

- *Statut* changé de *Nouveau* à *Rejeté*

- *Planning* mis à *Non*