

## Publik - Development #12961

### enregistrement de données statistiques supplémentaires

26 août 2016 11:07 - Frédéric Péters

<b>Statut:</b>	Nouveau	<b>Début:</b>	26 août 2016
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>		<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Club:</b>	
<b>Patch proposed:</b>	Non		
<b>Planning:</b>			

#### Description

Il y a des statistiques dont on aimerait bien suivre l'évolution, par exemple le nombre de comptes enregistrés ([#12803](#)), le nombre d'inscrits à une catégorie d'annonces, le nombre de comptes fédérés avec telle application, etc. Comment fait-on ?

#### Historique

##### #1 - 26 août 2016 11:47 - Benjamin Dauvergne

Je vois tout de suite une différence avec les stats sur les demandes: on est pas ici intéressé par la quantité sur une période mais sur le total depuis le début (on pourra éventuellement être aussi intéressé par l'évolution au mois mais ce n'est pas souvent qu'on présente les choses ainsi) suite à des créations ou suppression de truc (compte, abonnement, fédération).

Sans passer par un énième stockage du type graphite je me dis que chaque application pourrait exporter un document de cette forme (je mélange volontairement des données authentic avec du corbo, c'est pour l'exemple):

```
{
  'types': {
    'accounts': 'Comptes',
    'subscription': 'Abonnements aux listes',
  },
  'events': [
    {
      'date': '2016-08-26',
      'type': 'accounts',
      'created': 10,
      'deleted': 20,
    },
    {
      'date': '2016-08-26',
      'type': 'subscription',
      'kind': 'email',
      'category': 'Enfance',
      'created': 10,
      'deleted': 20,
    },
    {
      'date': '2016-08-26',
      'type': 'federation',
      'application': 'Mandaye Médiathèque',
      'created': 10,
      'deleted': 20,
    },
  ],
}
```

Ce serait utiliser pour utiliser des tables du type:

```
CREATE TABLE accounts_events (date DATE, added INTEGER, removed INTEGER);
CREATE TABLE subscription_events (date DATE, added INTEGER, removed INTEGER, category FK(subscription_categories), kind FK(kind));
...
```

Par un outil d'import à la wcs-olap et dans bijoe on aurait un cube par type d'évènement (chaque type d'évènement ayant des dimensions/métadonnées différentes pour les filtres, ici catégorie et type de l'abonnement pour corbo ou nom de l'application pour les fédérations). À la différence de wcs-olap cet outil fonctionnerait en mise à jour incrémental, les données étant déjà agrégées au jour par l'application qui la publie.

Il faudrait que chaque application conserve assez de données pour pouvoir publier ces stats, i.e. corbo doit conserver des abonnements supprimés avec leur date de suppression par exemple.

Dans le fichier \*.model pour bijoe l'outil décrira les requêtes pour faire les totaux sur la durée, i.e.:

```
SELECT month, sum(added) over (order by month) FROM (SELECT extract(month from date) as month, sum(added) as added FROM account_events GROUP BY extract(month from date));
```

Je ne pense pas qu'il soit actuellement capable de faire des requêtes si compliqués (il y a deux agrégats en cascade) ça ne me parait pas trop compliqué à ajouter.