Zoo - Bug #16438

web-service génériques de création/recherche/modification/supression d'une entité ou d'une relation

22 mai 2017 15:03 - Benjamin Dauvergne

Statut:	Nouveau	**Début:**	22 mai 2017
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Planning:**	

Description

Entités

Il faudra générer des serializer DRF automatiquement depuis le schéma JSON.

Création

```
POST /api/entities/<schema-slug/ HTTP/1.1
Content-Type: application/json
Content-Size: xxx

{
   "data": {
      "name": "Mon association",
      "registry_id": "12345",
   }
}
```

Il faudrait que ça valide que le JSON envoyé est conforme au schéma JSON du type d'entité demandé. Pour simplifier l'usage depuis w.c.s. je verrai bien une transformation automatique de clés nommées "data__name", vers les structures imbriquées décrites plus haut.

Idem pour ce bout de structure qui servirait à renseigner le journal de l'entité créée:

```
{
   "journal": {
      .. ici les données pour créer une entrée dans le journal...
   }
}
```

Retour:

```
{
  "err": 0,
  "data": {
     "id": 1234,
     ...
  }
}
```

Recherche

```
GET /api/entities/<schema-slug>/?e_champ1=xxx&e_champ2__souschamp=yyy&r_relationslug1__relationslug2__champ4=zzz
```

On recherche sur les champs, en full-text/Trigram, on vérifie que les champs listés sont conformes au schéma.

À voir pour ordonner aussi, obtenir directement des objets avec des relations (enfants pour des parents, membres pour une association, etc..). Rechercher aussi par une information d'un objet lié (j'ai l'identifiant d'un membre, je veux les associations qui lui sont liées).

Le retour

```
{
   "err": 0,
   "next": "https://zoo.example.com/api/entities/<schema-slug/?token=xyz",
   "data": [
      {
         "id": 1234,
         ...
         "relation__membre": [
             {
                "id": 4567,
                "uuid": "abcd",
                "display_name": "John Doe",
                "membre__type": "président"
                "membre__id": 89, # id de la relation
             }
         ]
      }
   ]
}
```

Modification

Avec PATCH on doit accepter un contenu partiel dans data (DRF gère ça très bien).

```
POST/PUT/PATCH /api/entities/<schema-slug>/1234/ HTTP/1.1
Content-Type: application/json
Content-Size: xyz

{
   "data": {
      "name": "Coin"
   }
}
```

Retour: le même que pour la création

Suppression

```
DELETE /api/entities/<schema-slug>/1234/ HTTP/1.1
```

Relations

Création

```
POST /api/relations/<schema-slug/ HTTP/1.1
Content-Type: application/json

{
  "data": {
     "left_id": 1234,
     "right_id": 4567,
     "type": "président"
  }
}
```

Retour:

```
{
  "err": 0,
  "data": {
     "id": 89,
     "left_id": 1234,
     "right_id": 4567,
```

```
      "type": "président"
   }
}
```

Idem ici des clés "journal__" sont acceptées et on peut aplatir le dictionnaire envoyé.

Tous les échanges avec ces web-services (autre que GET) doivent être enregistrés dans des objets Transaction (voir code pour Nanterre) donnant un log des échanges web-service.

Le modèle Log donne un journal métier des actions sur les objets.

Historique

#1 - 22 mai 2017 15:04 - Benjamin Dauvergne

- Sujet changé de web-service spécifique de création/recherche/modification/supression d'une entité ou d'une relation à web-service génériques de création/recherche/modification/supression d'une entité ou d'une relation