

## Passerelle - Development #55522

**base\_adresse : la durée de vie des ids est trop court pour une utilisation comme une source de donnée normale**

12 juillet 2021 11:42 - Benjamin Dauvergne

<b>Statut:</b>	Fermé	<b>Début:</b>	12 juillet 2021
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Benjamin Dauvergne	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	Non
<b>Patch proposed:</b>	Oui		

### Description

Les adresses retournées par la BAN sont mises en cache pour 1 heure dans un modèle AddressCacheModel.

J'ai le cas sur #55426 d'une demande avec un champ adresse ouvert à l'édition via un champ liste avec autocomplétion, avant affichage du formulaire la "display" value est revérifié via un appel à `{{ datasource.url }}?id={{ form_var_adresse }}` malheureusement c'est bien après la durée de vie d'1 heure pour ce cache et donc on a un champ Select2 avec le bon id mais pas d'adresse affiché (car passerelle a retourné une liste vide).

Solution rapide et simple, augmenter la durée du cache à 1 ou 2 semaines. Solution plus pérenne, ne pas se baser uniquement sur le cache mais retourner un id plus long, du style `<id> lat lon type` et utiliser l'appel reverse de la BAN pour retrouver l'objet.

Ça veut dire suite à cet appel

<https://gtm.toulouse-metropole.fr/search/?q=parc+de+la+&limit=5&citycode=31555>

```
{
  "attribution" : "SIG Toulouse Métropole",
  "features" : [
    {
      "geometry" : {
        "coordinates" : [
          1.438935,
          43.654259
        ],
        "type" : "Point"
      },
      "properties" : {
        "city" : "Toulouse",
        "citycode" : "31555",
        "context" : "31, Haute-Garonne, Occitanie (Midi-Pyrenees)",
        "id" : "315559000483",
        "importance" : 0.7695,
        "label" : "Parc de la Violette 31200 Toulouse",
        "name" : "Parc de la Violette",
        "norme_postale" : "PARC DE LA VIOLETTE",
        "population" : 440.2,
        "postcode" : "31200",
        "score" : 0.888136363636364,
        "type" : "street",
        "voie" : "PARC DE LA VIOLETTE",
        "x" : 1574078,
        "y" : 2273853
      },
      "type" : "Feature"
    }
  ],
  "filters" : {
    "citycode" : "31555"
  },
  "licence" : "N/A",
}
```

```
"limit" : 5,
"query" : "parc de la ",
"type" : "FeatureCollection",
"version" : "draft"
}
```

on retournerait l'id

315559000483 43.654259 1.438935 street

De cet id on pourrait générer la requête reverse suivante en cas d'absence de cache :

https://gtm.toulouse-metropole.fr/reverse/?lon=1.4389349999999999&lat=43.654259&type=street

```
{
  "attribution" : "SIG Toulouse Métropole",
  "features" : [
    {
      "geometry" : {
        "coordinates" : [
          1.438935,
          43.654259
        ],
        "type" : "Point"
      },
      "properties" : {
        "city" : "Toulouse",
        "citycode" : "31555",
        "context" : "31, Haute-Garonne, Occitanie (Midi-Pyrenees)",
        "distance" : 0,
        "id" : "315559000483",
        "importance" : 0.7695,
        "label" : "Parc de la Violette 31200 Toulouse",
        "name" : "Parc de la Violette",
        "norme_postale" : "PARC DE LA VIOLETTE",
        "population" : 440.2,
        "postcode" : "31200",
        "score" : 1,
        "type" : "street",
        "voie" : "PARC DE LA VIOLETTE",
        "x" : 1574078,
        "y" : 2273853
      },
      "type" : "Feature"
    }
  ],
  "filters" : {
    "type" : "street"
  },
  "licence" : "N/A",
  "limit" : 1,
  "type" : "FeatureCollection",
  "version" : "draft"
}
```

Et via l'id natif de la BAN on récupérerait la bonne feature.

**Demandes liées:**

Lié à w.c.s. - Development #55523: Lors de l'édition d'une demande avec un ch...

Fermé

12 juillet 2021

**Révisions associées**

Révision 3a01c571 - 15 juillet 2021 18:51 - Benjamin Dauvergne

base\_adresse: factorize search by id (#55522)

Révision 18b2e436 - 15 juillet 2021 18:51 - Benjamin Dauvergne

base\_adresse: add lat/lon/type to address'id to get by id through reverse (#55522)

## Historique

---

### #1 - 12 juillet 2021 11:42 - Benjamin Dauvergne

- Description mis à jour

### #2 - 12 juillet 2021 11:42 - Benjamin Dauvergne

- Description mis à jour

### #4 - 12 juillet 2021 11:51 - Benjamin Dauvergne

- Lié à Development #55523: Lors de l'édition d'une demande avec un champ liste sur source de donnée avec autocomplétion un id inconnu provoque la disparition de la valeur text sur une édition ajouté

### #5 - 12 juillet 2021 12:07 - Thomas Noël

Benjamin Dauvergne a écrit :

De cet id on pourrait générer la requête reverse suivante en cas d'absence de cache :

J'aime bien cette idée (en évitant cependant l'usage d'espaces dans le "id" mais c'est juste moi et je me soigne)

### #6 - 12 juillet 2021 12:15 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

### #7 - 12 juillet 2021 14:58 - Benjamin Dauvergne

- Fichier 0002-base\_adresse-add-lat-lon-type-to-address-id-to-get-b.patch ajouté

- Fichier 0001-base\_adresse-factorize-search-by-id-55522.patch ajouté

- Tracker changé de Support à Development

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Finalement je n'ai pas utilisé le endpoint reverse mais directement search en conservant le label et lat/lon comme indication, ça semble marcher correctement sur le cas Toulouse. Comme séparateur le tilde, c'est le seul caractère non signifiant qui ne sera pas encodé et qui a peu de chance de se retrouver dans les chaînes utilisées.

### #8 - 13 juillet 2021 14:17 - Thomas Noël

0002 : 128 caractères ça paraît assez court, non ? Je pense qu'il faut stocker la donnée "q" le plus précisément possible.

Je pense qu'il y a une erreur avec le « return results » en fin de :

```
# Use search with label as q and lat/lon as geographic hint
if q and lat and lon:
    results = self.addresses(request, q=q, lat=lat, lon=lon, citycode=citycode)['data']
    for result in results: # match by id if possible
        if result['ban_id'] == ban_id:
            return {'data': [result]}
    return results
```

ça serait plutôt « return {'err': \_('Address ID not found')} » non ?

### #9 - 13 juillet 2021 16:21 - Benjamin Dauvergne

Thomas Noël a écrit :

0002 : 128 caractères ça paraît assez court, non ? Je pense qu'il faut stocker la donnée "q" le plus précisément possible.

L'unicité est acquise par l'id BAN de toute façon, sans compter lat/lon; en fait on pourrait ne pas utiliser toute la chaîne pour ça, j'ai hésité à faire une version où je tronquerais l'id avant de créer l'objet AddressCache, mais j'ai eu peur qu'on m'accuse de complexifier. Mais tu as raison 128 caractères c'est à la fois trop pour l'unicité mais trop peu pour garantir que tout rentrera toujours, et on risque donc d'avoir des erreurs de donnée trop longue, je vais revenir à 30 et simplement tronquer.

Je pense qu'il y a une erreur avec le « return results » en fin de :

[...]

ça serait plutôt « return {'err': \_('Address ID not found')} » non ?

Ça demande un commentaire dans le code: l'idée c'est que dans l'éventualité rare, mais pas impossible, que l'id change, on parie sur le fait que le premier résultat (search les rangeant par distance à la position géographique et à la chaîne de recherche, je ne sais pas avec quelle pondération, surtout sur l'implémentation de Toulouse, qui n'est pas addok je pense) sera le bon (w.c.s. prenant le premier résultat de la liste comme valeur de la source de donnée pour cette id); dans l'idée qu'on arriverait à retomber sur nos pieds dans ce cas limite.

#### #10 - 13 juillet 2021 16:33 - Benjamin Dauvergne

- Fichier 0004-to-be-fixed-up-add-comment-about-our-heuristic-when-.patch ajouté
- Fichier 0003-to-be-fixed-up-return-to-api\_id-s-length-of-30-chara.patch ajouté
- Fichier 0002-base\_adresse-add-lat-lon-type-to-address-id-to-get-b.patch ajouté
- Fichier 0001-base\_adresse-factorize-search-by-id-55522.patch ajouté

#### #11 - 13 juillet 2021 16:33 - Benjamin Dauvergne

Je suis revenu à 30 caractères, en tronquant chaque fois qu'il faut, et j'ai ajouté un commentaire pour expliquer pourquoi on ne renvoie pas une erreur.

#### #12 - 13 juillet 2021 16:42 - Benjamin Dauvergne

- Fichier 0004-to-be-fixed-up-add-comment-about-our-heuristic-when-.patch ajouté
- Fichier 0003-to-be-fixed-up-return-to-api\_id-s-length-of-30-chara.patch ajouté
- Fichier 0005-to-be-fixed-up-return-error-if-there-were-no-results.patch ajouté
- Fichier 0002-base\_adresse-add-lat-lon-type-to-address-id-to-get-b.patch ajouté
- Fichier 0001-base\_adresse-factorize-search-by-id-55522.patch ajouté

Bon là je figrole, mais bon.

#### #13 - 15 juillet 2021 10:02 - Thomas Noël

Benjamin Dauvergne a écrit :

Thomas Noël a écrit :

0002 : 128 caractères ça paraît assez court, non ? Je pense qu'il faut stocker la donnée "q" le plus précisément possible.

L'unicité est acquise par l'id BAN de toute façon, sans compter lat/lon; en fait on pourrait ne pas utiliser toute la chaîne pour ça, j'ai hésité à faire une version où je tronquerais l'id avant de créer l'objet AddressCache, mais j'ai eu peur qu'on m'accuse de complexifier. Mais tu as raison 128 caractères c'est à la fois trop pour l'unicité mais trop peu pour garantir que tout rentrera toujours, et on risque donc d'avoir des erreurs de donnée trop longue, je vais revenir à 30 et simplement tronquer.

Bon je me suis perdu tout seul, j'avais mal lu et compris que le cet id à 128 caractère contenait le q= et non. Mais donc en relisant mieux, le q= est contenu dans l'id retourné par le search et ça, ça ne me va pas vraiment. Ça fait des id vraiment très moches et je me demande comment w.c.s. va interpréter ça (ok normalement rien ne devrait planter, mais quand même, des espaces, des accents, etc.). Je n'avais pas perçu que cet idée de se rebaser sur "search" amenait ce problème (plutôt que d'utiliser reverse comme ton idée initiale, qui n'avait besoin que de lat et lon).

Du coup je ne sais plus trop quoi penser... ./ Pourquoi finalement tu n'étais pas parti sur l'usage de reverse ?

ça serait plutôt « return {'err': \_('Address ID not found')} » non ?

Ça demande un commentaire dans le code: l'idée c'est que dans l'éventualité rare, mais pas impossible, que l'id change, on parie sur le fait que le premier résultat (search les rangeant par distance à la position géographique et à la chaîne de recherche, je ne sais pas avec quelle pondération, surtout sur l'implémentation de Toulouse, qui n'est pas addok je pense) sera le bon (w.c.s. prenant le premier résultat de la liste comme valeur de la source de donnée pour cette id); dans l'idée qu'on arriverait à retomber sur nos pieds dans ce cas limite.

Mouais mais non, retourner un résultat dont l'id n'est pas le même que celui demandé, ça va être un enfer à déboguer le jour où. Ca me semble plus raisonnable de considérer que l'id n'existe plus dans la BAN, et oui ça va planter l'édition, et c'est normal : l'adresse n'existe plus.

#### #14 - 15 juillet 2021 11:02 - Benjamin Dauvergne

Thomas Noël a écrit :

Benjamin Dauvergne a écrit :

Thomas Noël a écrit :

0002 : 128 caractères ça paraît assez court, non ? Je pense qu'il faut stocker la donnée "q" le plus précisément possible.

L'unicité est acquise par l'id BAN de toute façon, sans compter lat/lon; en fait on pourrait ne pas utiliser toute la chaîne pour ça, j'ai hésité à faire une version où je tronquerais l'id avant de créer l'objet AddressCache, mais j'ai eu peur qu'on m'accuse de complexifier. Mais tu as raison 128 caractères c'est à la fois trop pour l'unicité mais trop peu pour garantir que tout rentrera toujours, et on risque donc d'avoir des erreurs de donnée trop longue, je vais revenir à 30 et simplement tronquer.

Bon je me suis perdu tout seul, j'avais mal lu et compris que le cet id à 128 caractère contenait le q= et non. Mais donc en relisant mieux, le q= est contenu dans l'id retourné par le search et ça, ça ne me va pas vraiment. Ça fait des id vraiment très moches et je me demande comment w.c.s. va interpréter ça (ok normalement rien ne devrait planter, mais quand même, des espaces, des accents, etc.). Je n'avais pas perçu que cet idée de se rebaser sur "search" amenait ce problème (plutôt que d'utiliser reverse comme ton idée initiale, qui n'avait besoin que de lat et lon).

Du coup je ne sais plus trop quoi penser... :/ Pourquoi finalement tu n'étais pas parti sur l'usage de reverse ?

Le fait que reverse ne se base que sur la distance à lat/lon alors que search va utiliser le nom et lat/lon en même temps,

Mouais mais non, retourner un résultat dont l'id n'est pas le même que celui demandé, ça va être un enfer à déboguer le jour où. Ça me semble plus raisonnable de considérer que l'id n'existe plus dans la BAN, et oui ça va planter l'édition, et c'est normal : l'adresse n'existe plus.

Ok, je reviens au patch du départ alors.

#### #15 - 15 juillet 2021 15:42 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

#### #16 - 15 juillet 2021 15:51 - Benjamin Dauvergne

- Fichier 0002-base\_adresse-add-lat-lon-type-to-address-id-to-get-b.patch ajouté

- Fichier 0001-base\_adresse-factorize-search-by-id-55522.patch ajouté

- Statut changé de En cours à Solution proposée

Je suis finalement resté sur l'implémentation via search; reverse ne renvoie toujours qu'un seul résultat ce qui me paraît casse gueule, pour moi la longueur de l'id n'a que peu d'importance, l'important c'est le lien id <-> structured et il est garanti parce que l'id contient le BAN id (si le texte change, ou lat/lon, on trouvera quand même l'enregistrement source tant que le BAN id ne change pas).

PS: j'ai par contre changé le comportement si l'id a changé, ça génère une erreur.

#### #17 - 15 juillet 2021 17:28 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Tout me va, y compris le log.error (si on en voit trop, on débranchera ça et on demandera à la BAN de gérer un vrai get/id).

Dans les tests, tu fais :

```
- assert data['id'] == '49007_6950_be54bd'  
+ assert data['id'] == '49007_6950_be54bd~47.474633~-0.593775~Rue Roger Halope 49000 Angers'
```

mais je propose d'ajouter un

```
+ assert data['ban_id'] == '49007_6950_be54bd'
```

juste pour "montrer" qu'on reçoit bien l'id natif en parallèle.

#### #18 - 15 juillet 2021 18:51 - Benjamin Dauvergne

- Statut changé de Solution validée à Résolu (à déployer)

```
commit 359794481dc7e7fe7ee82b2322a886cfca729799  
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>  
Date: Mon Jul 12 13:16:28 2021 +0200
```

```
base_adresse: add lat/lon/type to address'id to get by id through reverse (#55522)
```

```
commit 4e7c77ce8a0729d87ce205e6f20e2ef63fa067d8  
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>  
Date: Mon Jul 12 12:16:00 2021 +0200
```

base\_adresse: factorize search by id (#55522)

### #19 - 15 juillet 2021 18:52 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

[...]

Poussé avec la modif demandée aux tests.

### #20 - 15 juillet 2021 23:17 - Frédéric Péters

- Statut changé de *Résolu* (à déployer) à *Solution déployée*

#### Fichiers

0002-base_adresse-add-lat-lon-type-to-address-id-to-get-b.patch	5,93 ko	12 juillet 2021	Benjamin Dauvergne
0001-base_adresse-factorize-search-by-id-55522.patch	1,61 ko	12 juillet 2021	Benjamin Dauvergne
0004-to-be-fixed-up-add-comment-about-our-heuristic-when-.patch	1,11 ko	13 juillet 2021	Benjamin Dauvergne
0003-to-be-fixed-up-return-to-api_id-s-length-of-30-chara.patch	4,05 ko	13 juillet 2021	Benjamin Dauvergne
0002-base_adresse-add-lat-lon-type-to-address-id-to-get-b.patch	5,93 ko	13 juillet 2021	Benjamin Dauvergne
0001-base_adresse-factorize-search-by-id-55522.patch	1,61 ko	13 juillet 2021	Benjamin Dauvergne
0004-to-be-fixed-up-add-comment-about-our-heuristic-when-.patch	1,11 ko	13 juillet 2021	Benjamin Dauvergne
0003-to-be-fixed-up-return-to-api_id-s-length-of-30-chara.patch	4,05 ko	13 juillet 2021	Benjamin Dauvergne
0005-to-be-fixed-up-return-error-if-there-were-no-results.patch	2,91 ko	13 juillet 2021	Benjamin Dauvergne
0002-base_adresse-add-lat-lon-type-to-address-id-to-get-b.patch	5,93 ko	13 juillet 2021	Benjamin Dauvergne
0001-base_adresse-factorize-search-by-id-55522.patch	1,61 ko	13 juillet 2021	Benjamin Dauvergne
0002-base_adresse-add-lat-lon-type-to-address-id-to-get-b.patch	7,34 ko	15 juillet 2021	Benjamin Dauvergne
0001-base_adresse-factorize-search-by-id-55522.patch	1,61 ko	15 juillet 2021	Benjamin Dauvergne