

## Passerelle - Development #21558

### opengis: rajouter le geocodage inverse

31 janvier 2018 17:16 - Serghei Mihai

|  |               |                      |                         |
|--|---------------|----------------------|-------------------------|
| <b>Statut:</b>   | Fermé         | <b>Début:</b>        | 31 janvier 2018         |
| <b>Priorité:</b>   | Normal        | <b>Echéance:</b>     |                         |
| <b>Assigné à:</b>  | Serghei Mihai | <b>% réalisé:</b>    | 0%                      |
| <b>Catégorie:</b>  |               | <b>Temps estimé:</b> | 0:00 heure              |
| <b>Version cible:</b>  |               | <b>Planning:</b>     |                         |
| <b>Patch proposed:</b>   | Oui           |                      |                         |
| <b>Description</b>   |               |                      |                         |
| GeoServer ne possède pas d'API de geocodage inverse.<br>Une façon de l'implementer en utilisant l'API existante est d'utiliser le filtre DWWithin du flux WFS auquel un point, à travers ses coordonnées, ainsi que le rayon de recherche est spécifié.<br><br>Les coordonnées du point devraient être converties éventuellement au système de projection utilisé dans la base derrière.<br><br>Le retour devrait se faire au format de la BAN.<br><br>Un exemple d'appel vers un WFS: <a href="https://sigmetropole.lametro.fr/geoserver/wfs?cql_filter=DWITHIN(the_geom,Point(1914042.47294224665.1955),10,meters)&amp;service=wfs&amp;request=GetFeature&amp;typeName=ref_ban_metro&amp;version=2.0.0&amp;outputformat=json">https://sigmetropole.lametro.fr/geoserver/wfs?cql_filter=DWITHIN(the_geom,Point(1914042.47294224665.1955),10,meters)&amp;service=wfs&amp;request=GetFeature&amp;typeName=ref_ban_metro&amp;version=2.0.0&amp;outputformat=json</a> |               |                      |                         |
| <b>Demandes liées:</b>   |               |                      |                         |
| Lié à Passerelle - Development #21763: developper un connecteur pour le geoco...   |               | <b>Rejeté</b>        | <b>09 février 2018</b>  |
| Lié à Passerelle - Development #20826: opengis : pouvoir préciser la projecti...   |               | <b>Fermé</b>         | <b>20 décembre 2017</b> |
| Lié à w.c.s. - Development #22218: géocodage: pouvoir spécifier des urls diff...   |               | <b>Fermé</b>         | <b>01 mars 2018</b>     |

### Révisions associées

#### Révision 7336d6f5 - 26 mars 2018 16:42 - Serghei Mihai

opengis: add reverse geocoding based on WFS (#21558)

### Historique

#### #2 - 01 février 2018 09:11 - Serghei Mihai

Ici je pense partir sur le même principe que les loaders des formats des fichiers dans le connecteur famille générique.  
Si un format est choisi, le endpoint reverse va utiliser le loader pour résoudre les coordonnées, sinon ça ne retourne rien.

#### #3 - 01 février 2018 09:16 - Frédéric Péters

Uh ? Un format de quoi ?

#### #4 - 01 février 2018 09:17 - Frédéric Péters

Le retour devrait se faire au format de la BAN.

Le format pour le géocodage est celui de nominatim.

#### #5 - 01 février 2018 09:26 - Serghei Mihai

Frédéric Péters a écrit :

Uh ? Un format de quoi ?

Un format ou loader: ça serait un module qui s'occuperait à faire la requête vers GeoServer, avec conversion des coordonnées au système utilisé dans la base derrière et retourner les données au format nominatim.

#### #6 - 01 février 2018 09:41 - Frédéric Péters

On est dans un connecteur standard interrogeant des services compatibles opengis, on n'a pas besoin de couche d'abstraction.

Qu'on puisse préciser la projection "préférée" à utiliser par le serveur, bien sûr, ça peut être paramètre. (il y a même d'ailleurs un ticket... [#20826](#)).

**#7 - 02 février 2018 09:46 - Serghei Mihai**

Tout à fait d'accord, sauf que la couche interrogée (typeName) peut nommer les attributs différemment et pour les formater nominatif il faut un post traitement un peu "spécifique".

**#8 - 02 février 2018 09:57 - Frédéric Péters**

Franchement hostile; on doit arriver à discuter avec les équipes SIG pour obtenir quelque chose qu'on puisse interroger de manière standard. Sinon on passe par la BAN ou OSM.

Ou alors faire ça dans un autre connecteur dans contrib auquel je ne m'intéresserai pas.

**#9 - 02 février 2018 11:03 - Serghei Mihai**

- Statut changé de Nouveau à Rejeté

Frédéric Péters a écrit :

Franchement hostile; on doit arriver à discuter avec les équipes SIG pour obtenir quelque chose qu'on puisse interroger de manière standard. Sinon on passe par la BAN ou OSM.

Les équipes SIG ne peuvent me proposer rien d'autre comme webservice que les flux de geoserver. Il m'ont donné quelques pistes pour faire le géocodage inverse à travers ce flux. Les standards et tout ça, il s'en fichent.

Ou alors faire ça dans un autre connecteur dans contrib auquel je ne m'intéresserai pas.

Je vais faire un connecteur à part.

**#10 - 09 février 2018 18:48 - Benjamin Dauvergne**

- Lié à Development #21763: développer un connecteur pour le géocodage inverse pour Grenoble ajouté

**#11 - 10 février 2018 10:00 - Frédéric Péters**

Pour revenir quand même sur ce ticket, des pistes évitant du code spécifique à Grenoble (ni dans ce connecteur ni dans un autre) :

- ajouter au connecteur un paramétrage des noms des propriétés à utiliser
- tirer cette information de paramètres à l'URL (cela nécessiterait l'ajout côté w.c.s. de la possibilité de paramétrer ça)
- avoir des séries de possibilités (ex: attributes = {'road': ['road', 'street', 'rue', 'voie', 'nom\_rue', 'nom\_voie'], 'city': ['city', 'city\_name', 'ville', 'commune', 'nom\_ville', 'nom\_commune'] ...})

~~

À côté de ça, attention, côté wcs il est attendu que le service "nominatim" puisse faire géocodage et géocodage inversé, faire uniquement le géocodage inversé ça cassera l'action de géocodage.

**#12 - 19 février 2018 10:56 - Serghei Mihai**

Ok, donc on partirait sur l'usage de la méthode DWITHIN disponible dans le flux WFS. En plus des paramètres que tu mentionnes il faudrait rajouter un champ pour spécifier le rayon de recherche. Il manque également [#20826](#), je prend.

**#13 - 19 février 2018 15:40 - Serghei Mihai**

- Lié à Development #20826: openGIS : pouvoir préciser la projection "préférée" du serveur ajouté

**#14 - 19 février 2018 15:44 - Frédéric Péters**

- Statut changé de Rejeté à En cours

Si on revient ici, ne le marquons plus comme rejeté.

**#15 - 27 février 2018 16:05 - Serghei Mihai**

- Assigné à mis à Serghei Mihai

Je pars sur l'idée qui me semble la plus simple: rajouter un champ de type JSONField dans lequel on renseigne le mapping entre les attributs du flux WFS et ceux de la BAN, genre:

```
{
  "house_number": [
    "numero"
  ],
  "city": [
    "nom_commune"
  ],
  "post_code": [
    "code_post"
  ],
  "road": [
    "nom_voie"
  ]
}
```

#### #16 - 27 février 2018 16:53 - Frédéric Péters

D'accord c'était cité dans "ajouter au connecteur un paramétrage des noms des propriétés à utiliser" mais quand même, à chaque fois qu'on demande du paramétrage en json dans un ui, je pleure.

#### #17 - 27 février 2018 16:58 - Serghei Mihai

Ok, alors un champ text pour chaque attribut: road, housenumber, etc avec une liste de propriétés à chercher dans les données retournées.

#### #18 - 27 février 2018 17:00 - Frédéric Péters

tirer cette information de paramètres à l'URL (cela nécessiterait l'ajout côté w.c.s. de la possibilité de paramétrer ça)  
avoir des séries de possibilités (ex: attributes = {'road': ['road', 'street', 'rue', 'voie', 'nom\_rue', 'nom\_voie'], 'city': ['city', 'city\_name', 'ville', 'commune', 'nom\_ville', 'nom\_commune'] ...})

Elles puent ces propositions ?

#### #19 - 27 février 2018 17:07 - Serghei Mihai

Non, mais ça m'embête de devoir faire des devs dans wcs pour passer des paramètres à un webservice pour lui indiquer comment formater les données. Ce paramétrage doit rester dans passerelle, imo.

#### #20 - 27 février 2018 17:15 - Frédéric Péters

Et :

avoir des séries de possibilités (ex: attributes = {'road': ['road', 'street', 'rue', 'voie', 'nom\_rue', 'nom\_voie'], 'city': ['city', 'city\_name', 'ville', 'commune', 'nom\_ville', 'nom\_commune'] ...})

#### #21 - 27 février 2018 18:01 - Serghei Mihai

Oui, et c'est possibilités je les vois définies dans le connecteur, si c'est pas un jsonfield, alors un attribut du connecteur pour chaque attribut nécessaire pour le service de geocodage.

#### #22 - 27 février 2018 20:39 - Frédéric Péters

Oui, et c'est possibilités je les vois définies dans le connecteur, si c'est pas un jsonfield, alors un attribut du connecteur pour chaque attribut nécessaire pour le service de geocodage.

Et au contraire, plutôt que demander une configuration à l'utilisateur (dans un textarea avec du json ou du csv dedans), cette approche était la mise en dur dans le connecteur d'une série de possibilités, comme on a pu avoir dans auquotidien quelque chose de cet ordre :

```
...
    ('streetAddress', ('streetAddress', 'address', 'adresse', 'street')),
    ('street', ('streetAddress', 'address', 'adresse', 'street')),
    ('postalCode', ('postalCode', 'codepostal', 'cp')),
    ('telephoneNumber', ('telephoneNumber', 'telephonefixe', 'telephone')),
...
```

L'idée étant que ça juste marche.

#### #23 - 28 février 2018 12:05 - Serghei Mihai

- Fichier 0001-opengis-add-reverse-geocoding-based-on-WFS-21558.patch ajouté

- Patch proposed changé de Non à Oui

Ok.

#### #24 - 01 mars 2018 09:23 - Frédéric Péters

```
result['address'] = {'country': 'France'}
```

Pas fan; je préférerais que ça soit l'appelant qui décide quoi faire sur une valeur manquante. (mais bon)

1. return first found point

Plus fondamental, plutôt retourner le point le plus proche que celui qui par hasard va se trouver en premier.

#### #25 - 01 mars 2018 18:46 - Serghei Mihai

- Fichier 0001-opengis-add-reverse-geocoding-based-on-WFS-21558.patch ajouté

Frédéric Péters a écrit :

Pas fan; je préférerais que ça soit l'appelant qui décide quoi faire sur une valeur manquante. (mais bon)

Ok. Donc si on peut spécifier dans la requete genre: ?country=France pour qu'elle soit utilisée comme valeur de l'attribut. Au passage, recherche du pays dans le geojson.

Plus fondamental, plutôt retourner le point le plus proche que celui qui par hasard va se trouver en premier.

Ok.

#### #26 - 01 mars 2018 18:53 - Frédéric Péters

Ok. Donc si on peut spécifier dans la requete genre: ?country=France pour qu'elle soit utilisée comme valeur de l'attribut.

Non, déjà noté plus haut aujourd'hui on ne peut pas ajouter de paramètres. (mais pourquoi vouloir le pays ?)

Aussi, par rapport à une utilisation dans w.c.s., déjà noté également, ça va foirer d'avoir géocodage inversé mais pas géocodage, si on veut deux services différents, là aussi ça va demander patch w.c.s.

#### #27 - 01 mars 2018 19:05 - Serghei Mihai

- Fichier 0001-opengis-add-reverse-geocoding-based-on-WFS-21558.patch ajouté

Frédéric Péters a écrit :

Non, déjà noté plus haut aujourd'hui on ne peut pas ajouter de paramètres. (mais pourquoi vouloir le pays ?)

Ok, laissons l'appelant décider d'une valeur manquante. Je bloquais sur le pays car mon geojson de test ne le contient pas, mais oublions.

Aussi, par rapport à une utilisation dans w.c.s., déjà noté également, ça va foirer d'avoir géocodage inversé mais pas géocodage, si on veut deux services différents, là aussi ça va demander patch w.c.s.

Ça me paraît plus complexe en effet d'implémenter le géocodage, car c'est pas évident de matcher la chaîne de recherche que ça soit uniquement la voie, le numéro et la voie, ou numéro, voie, ville, sur différents attributs du geojson.

Je fais le ticket pour patcher wcs.

#### #28 - 01 mars 2018 19:09 - Serghei Mihai

- Lié à Development #22218: géocodage: pouvoir spécifier des urls différentes pour le géocodage et géocodage inverse ajouté

### #29 - 07 mars 2018 14:35 - Serghei Mihai

- Fichier 0001-opengis-add-reverse-geocoding-based-on-WFS-21558.patch ajouté

Patch à jour avec le calcul correct du point le plus proche.

### #30 - 16 mars 2018 12:40 - Emmanuel Cazenave

Pas réussi à appliquer ton patch.

Ça retourne None si tu ne trouves pas de `closest_feature`, je ne sais pas ce que ça va donner comme réponse pour l'appelant du connecteur, ie peut être être plus explicite (voir même raiser une API error, parce que pour l'appelant, un service de géocodage inverse qui renvoie rien, ça fait un peu erreur quand même), je te laisse voir.

Il m'a fallu un moment pour comprendre le bloc ci-dessous, donc peut-être l'encapsuler dans une fonction avec un joli nom (ce qui te permettra de la tester) ou au moins mettre un commentaire.

```
min_delta = None
for feature in response.json().get('features'):
    if not feature['geometry']['type'] == 'Point':
        continue # skip unknown
    lon_diff = abs(float(lon) - float(feature['geometry']['coordinates'][0]))
    lat_diff = abs(float(lat) - float(feature['geometry']['coordinates'][1]))
    delta = math.sqrt(lon_diff * lon_diff + lat_diff * lat_diff)
    if min_delta is None:
        min_delta = delta
    if delta <= min_delta:
        closest_feature = feature
```

Tu gères pas du tout les erreurs éventuelles que peut te renvoyer le service pointé par `wfs_service_url`, à mon avis il faut au moins utiliser `response.raise_for_status` ou un truc comme ça et l'encapsuler dans une `APIError`.

### #31 - 16 mars 2018 12:57 - Serghei Mihai

Emmanuel Cazenave a écrit :

Pas réussi à appliquer ton patch.

Ça retourne None si tu ne trouves pas de `closest_feature`, je ne sais pas ce que ça va donner comme réponse pour l'appelant du connecteur, ie peut être être plus explicite (voir même raiser une API error, parce que pour l'appelant, un service de géocodage inverse qui renvoie rien, ça fait un peu erreur quand même), je te laisse voir.

Ça fonctionne de la même façon que `nominatim`. Quand il n'y a pas de données, on renvoie null (le None en json).

Il m'a fallu un moment pour comprendre le bloc ci-dessous, donc peut-être l'encapsuler dans une fonction avec un joli nom (ce qui te permettra de la tester) ou au moins mettre un commentaire.

Tu veux dire le calcul de l'hypothénuse ? Je préfère rajouter un commentaire au dessus des 3 lignes.

Tu gères pas du tout les erreurs éventuelles que peut te renvoyer le service pointé par `wfs_service_url`, à mon avis il faut au moins utiliser `response.raise_for_status` ou un truc comme ça et l'encapsuler dans une `APIError`.

Effectivement il manque le check du statut, je vais rajouter ça.

### #32 - 16 mars 2018 13:03 - Frédéric Péters

Ça fonctionne de la même façon que `nominatim`. Quand il n'y a pas de données, on renvoie null (le None en json).

En l'espèce `nominatim` semble plutôt répondre `{"error":"Unable to geocode"}`. ex :

<https://nominatim.openstreetmap.org/reverse.php?format=json&lat=22.26895268313282&lon=-18.632978796958927&zoom=18> (je n'ai pas trouvé d'autre endroit que le milieu de l'océan pour un tel résultat).

### #33 - 16 mars 2018 14:16 - Serghei Mihai

Je me suis embrouillé avec notre connecteur `base_adresse` dont le endpoint `reverse` ne retourne rien quand il ne trouve pas l'adresse.

### #34 - 21 mars 2018 15:09 - Serghei Mihai

- Fichier 0001-opengis-add-reverse-geocoding-based-on-WFS-21558.patch ajouté

**#35 - 26 mars 2018 16:59 - Emmanuel Cazenave**

ack

**#36 - 26 mars 2018 17:04 - Serghei Mihai**

- Statut changé de *En cours* à *Résolu* (à déployer)

```
commit 7336d6f5dd51c7a35bf8acaa5d77633a697a18a6 (origin/master, origin/HEAD)
Author: Serghei Mihai <smihai@entrouvert.com>
Date: Wed Jan 31 17:35:02 2018 +0100
```

```
opengis: add reverse geocoding based on WFS (#21558)
```

**#37 - 04 août 2018 12:29 - Benjamin Dauvergne**

- Statut changé de *Résolu* (à déployer) à *Fermé*

**Fichiers**

---

|   |         |                 |               |
|---|---------|-----------------|---------------|
| 0001-opengis-add-reverse-geocoding-based-on-WFS-21558.patch | 8,97 ko | 28 février 2018 | Serghei Mihai |
| 0001-opengis-add-reverse-geocoding-based-on-WFS-21558.patch | 10,5 ko | 01 mars 2018    | Serghei Mihai |
| 0001-opengis-add-reverse-geocoding-based-on-WFS-21558.patch | 10,1 ko | 01 mars 2018    | Serghei Mihai |
| 0001-opengis-add-reverse-geocoding-based-on-WFS-21558.patch | 10,5 ko | 07 mars 2018    | Serghei Mihai |
| 0001-opengis-add-reverse-geocoding-based-on-WFS-21558.patch | 11,2 ko | 21 mars 2018    | Serghei Mihai |