



Wallonie
Service public
SPW

Centre de services - Géomatique



Webservices de géocodage SPW – ICAR

Documentation fonctionnelle /geocodeWS

David DABIN et Stéphane PALM

support.sig.digital@spw.wallonie.be

Version du 9 avril 2025

Table des matières

- Introduction..... 3
- 1. /geolist..... 4
- 2. /revgeocode 9
- 3. /parser 11
- 4. /geocode..... 15
- 5. /searchAll..... 23
- 6. /searchAll ou /geocode ?..... 27

Introduction

Cette documentation fonctionnelle du webservice GeocodeWS du Géoportail de la Wallonie vise à décrire les services de localisation ICAR. Ce document est à utiliser en complément de la documentation Swagger disponible à l'URL <http://geoservices.wallonie.be/geocodeWS/> qui est plutôt orientée vers les développeurs.

Actuellement, plusieurs services distincts sur les données ICAR sont disponibles :

- ***/geocode*** service de géocodage *Adresse => Point*;
- ***/geolist*** service d'index géographique également connu sous le nom de **gazetteer** ;
- ***/getScore*** service de description d'un score de géocodage ;
- ***/getScores*** service de description de tous les scores de géocodage possible ;
- ***/parser*** service de découpage d'adresse sous forme de texte libre en composantes normalisées (commune, code postal, rue et numéro) ;
- ***/revgeocode*** service de géocodage inverse *Point => Adresse*;
- ***/searchAll*** service de recherche rapide basé sur des méthodes « Full Text Search ».

Chacun de ces services fait l'objet d'une section dédiée avec une illustration des cas d'usage. Attention, sauf mention explicite tous les cas exposés ici exploitent des réponses au format JSON fixé par le header « Accept : application/json ». D'autres formats fourniront naturellement des résultats différents visuellement.

1. /geolist

1.1. Objectif

Le service /GEOLIST est un web-service qui liste toutes les sous-entités spatiales au sein d'une entité hiérarchique supérieure avec une possibilité de filtrage des résultats par mot-clés. Ce service permet de répondre à des questions comme :

- Quelles sont les communes de Wallonie commençant par les lettres « CHA » ?
- Quelles sont les rues de Namur contenant le mot « ANGE » ?
- Quels sont les numéros le long de la Chaussée de Charleroi à Namur ?
- Quel est l'identifiant ICAR de la commune de Vielsalm ?
- Quel est l'identifiant ICAR de la Chaussée de Charleroi à Namur ?
- Quel est l'identifiant ICAR du numéro 83b de la Chaussée de Charleroi à Namur ?

Les usages classiques de ce web-service incluent l'auto-complétion et la génération de datalist dans les formulaires web sur des données d'adresse.

1.2. Appel au service

Les appels au web-service GEOLIST se font par une interface RESTful. De simples requêtes http permettent ainsi d'interroger le service. Les paramètres sont encodés dans l'URL en tant que query strings pour les requêtes de type GET alors que les requêtes de type POST nécessitent des données encodées en JSON. Toutes les réponses sont encodées en JSON.

1.3. Paramètres et dépendances

Les paramètres incluent :

- **city**: identifiant ou nom de la commune.
- **zone** : identifiant ou nom de la localité.
- **street** : identifiant ou nom de la rue.
- **house** : identifiant ou nom du numéro de police.
- **bbox** l'extension spatiale doit-elle être renvoyée ? Les valeurs acceptées sont [false, true] avec une valeur par défaut à false.
- **geom** la géométrie doit-elle être renvoyée ? Les valeurs acceptées sont [false, true] avec une valeur par défaut à false.
- **lang** langue de préférence de la réponse. Les valeurs acceptées sont [fr, de, nl].

Les dépendances entre les paramètres d'entrée sont complexes et dépendent du niveau de la recherche. Ainsi, les paramètres city, zone, street et house sont tous optionnels mais au moins l'un d'entre eux doit être fourni dans la requête. Par niveau, les contraintes sont les suivantes :

- Recherche de commune

Le seul paramètre obligatoire est **city** qui doit contenir au moins 3 caractères pour rechercher un nom de commune, 5 chiffres pour identifier une commune depuis son identifiant ou le caractère « * » pour lister toutes les communes. Si le paramètre langue est fourni, il force la recherche dans une seule langue.

- Recherche de zone/localité

Le seul paramètre obligatoire est **zone** qui doit contenir au moins 3 caractères pour rechercher un nom localité, 4 chiffres pour identifier un code postal ou le caractère « * » pour lister toutes les localités. Si le paramètre langue est fourni, il force la recherche dans une seule langue.

- Recherche de rue

City et **street** sont les 2 paramètres obligatoires pour rechercher une rue. Le paramètre **city** doit contenir l'identifiant de la commune et **street** doit contenir au moins 3 caractères pour rechercher un nom de rue ou le caractère « * » pour lister toutes les rues au sein de la commune.

L'identification d'une rue depuis son identifiant ne requiert que le seul paramètre **street** avec pour valeurs son identifiant numérique.

- Recherche de numéro de police

Street et **house** sont les 2 paramètres obligatoires pour rechercher un numéro de police. Le paramètre **street** doit contenir l'identifiant de la rue et **house** doit contenir au moins 1 caractère pour rechercher un numéro de police ou le caractère « * » pour lister tous les numéros de la rue.

L'identification d'un numéro de police depuis son identifiant ne requiert que le seul paramètre **house** avec pour valeurs son identifiant numérique.

De manière générale, le niveau supérieur est toujours fourni selon son identifiant (code INS, code postal, code rue ou code bâtiment) alors que le filtre sous-jacent est fourni en texte dans une recherche alors que l'identification d'un élément ne nécessite que l'identifiant de l'élément.

1.4. Exemple d'utilisation

Prenons le cas d'un formulaire web dans lequel l'utilisateur est appelé à remplir une adresse postale. L'usage visé ici est de faciliter l'encodage des informations par auto-complétions et visualisation du résultat sur carte par zoom. Le formulaire pourrait ressembler à l'exemple ci-dessous :

LOCALISER UNE ADRESSE



Commune :

Rue :

Numéro :

L'utilisateur doit y encoder l'adresse « Place de Wallonie 1 à 5100 Jambes ». Dans cet exemple, l'utilisation du web-service GEOLIST pourrait être le suivant en supposant l'utilisation de simples datalists html alimentée par jQuery. Naturellement, d'autres formes et voies sont possibles selon les besoins. L'exemple n'est fourni qu'à titre illustratif.

1) Éventuellement pré-remplissage du datalist des communes par la requête

REQ : http://geoservices.wallonie.be/geocodeWS/geolist?city=*&bbox=true

```
RES :
{"geotag":{"extent":true,"geom":false,"lang":"fr","crs":"EPSG:31370","city":"*"},"candidates":[{"score":2,"city":{"name":"Aiseau-Presles","bbox":[161513.68280353662,117315.90454245731,167659.25076893103,125337.07667861879],"ident":"52074"}},{"score":2,"city":{"name":"Amay","bbox":[213303.19910873988,134372.5362557359,219533.21148363245,142794.20258525666],"ident":"61003"}}, ... ]}
```

Nous distinguerons naturellement dans la réponse l'**identifiant** qui servira pour « value » alors que le **name** servira de « label » pour notre datalist. L'option `bbox=true` ajoute l'information sur l'extension spatiale des communes. Cette information est exploitable par exemple dans un event lié à la datalist « Commune » et pourrait automatiquement zoomer sur la commune lorsque l'utilisateur en choisit une dans la liste déroulante.

2) Recherche de la Commune par le nom de la commune

REQ : <http://geoservices.wallonie.be/geocodeWS/geolist?city=Jambes>

```
RES :
{"geotag":{"extent":false,"geom":false,"lang":"","crs":"EPSG:31370","city":"Jambes"},"candidates":[]}
```

Dans ce cas, `len(RES['candidates']) <= 0`, nous pouvons donc déduire de l'absence de résultats qu'aucune commune ne porte le nom de « Jambes ». Il est probable que nous soyons face à une localité. Explorons cette possibilité dans l'étape 2.

3) Recherche de la commune par le nom de la localité

REQ : <http://geoservices.wallonie.be/geocodeWS/geolist?zone=Jambes>

RES :

```
{"geotag":{"extent":false,"geom":false,"lang":"fr","crs":"EPSG:31370","zone":"Jambes"},"candidates":[{"score":1,"city":{"name":"Namur","ident":"92094"},"zone":{"name":"Jambes","ident":"5100"}]}
```

Nous obtenons cette fois-ci un candidat dont le nom est identique à la requête. Le candidat nous permet d'identifier également la commune dont dépend notre localité : « Namur » ainsi que son identifiant « 92094 » qui est ici le code INS de la commune.

4) Recherchons maintenant les rues au sein de cette commune de Namur qui correspondent aux trois premières lettres entrées par l'encodant (par exemple « wal » pour Wallonie voir image)

REQ : <http://geoservices.wallonie.be/geocodeWS/geolist?city=92094&street=wal>

RES :

```
{"geotag":{"extent":false,"geom":false,"lang":"fr","crs":"EPSG:31370","city":"92094","street":"wal"},"candidates":[{"score":2,"city":{"name":"Namur","ident":"92094"},"street":{"name":"Rue Aux Ruwales","ident":7753632}},{"score":2,"city":{"name":"Namur","ident":"92094"},"street":{"name":"Place de la Wallonie","ident":7753487}}]}
```

Nous obtenons cette fois-ci deux candidats qui correspondent au filtre « wal ». Le score de 2 nous indique que la recherche a été fructueuse dans le mot alors que ce score aurait été de 1 pour les rues dont le nom aurait commencé par « wal ». Ce score nous permet donc de classer nos datalist en Html par exemple pour présenter d'abord les résultats commençant par le filtre avec les résultats contenant le filtre.

5) À ce stade, nous pouvons de nouveau attendre une entrée de l'utilisateur pour le numéro de maison liée à notre rue 7726866 – Place de Wallonie choisie dans la picklist par l'utilisateur. Par exemple la requête suivante :

REQ : <http://geoservices.wallonie.be/geocodeWS/geolist?street=7753487&house=1>

RES :

```
{"geotag":{"extent":false,"geom":false,"lang":"fr","crs":"EPSG:31370","street":"7753487","house":"*"},"candidates":[{"score":2,"city":{"name":"Namur","ident":"92094"},"street":{"name":"Place de la Wallonie","ident":7753487},"house":{"name":"1","ident":370296}}]}
```

Mais nous pourrions également décider de préremplir le datalist avec tous les numéros de maison connus sur notre rue 7726866 via l'utilisation du joker '*' pour le numéro de maison :

REQ : http://geoservices.wallonie.be/geocodeWS/geolist?street=7753487&house=*

6) Si l'utilisateur sélectionne notre numéro 1, nous pouvons alors envoyer une ultime requête pour obtenir les coordonnées spatiales et créer un point sur la carte en passant l'array « coordinates » :

REQ : geoservices.wallonie.be/geocodeWS/geolist?house=370296&geom=True

RES :

```
{"geotag":{"extent":false,"geom":true,"lang":"fr","crs":"EPSG:31370","house":"370296"},"candidates":[{"score":2,"city":{"name":"Namur","ident":"92094"},"street":{"name":"Place de la Wallonie","ident":7753487},"house":{"name":"1","geometry":{"type":"Point","crs":{"type":"name","properties":{"name":"EPSG:31370"}}},"coordinates":[185776.58100000804,127425.47299881652]},"ident":370296}]}
```

2. /revgeocode

2.1. Objectif

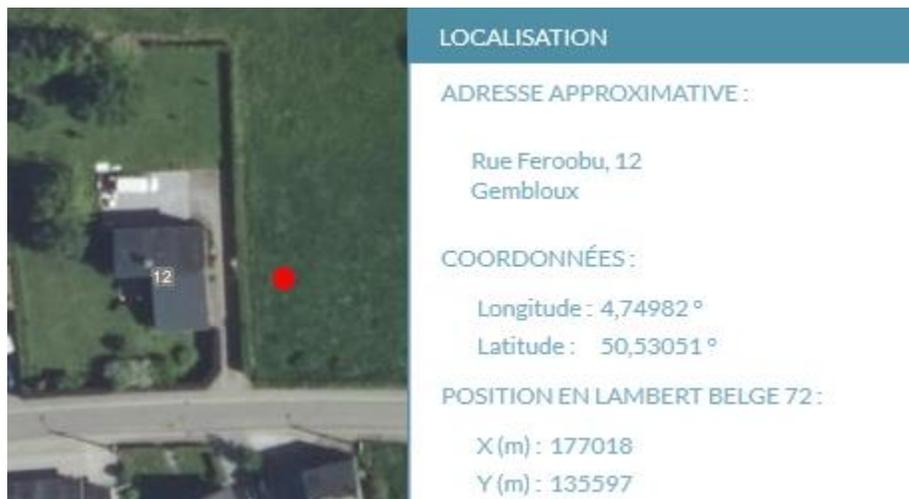
Le service /REVGEOCODE est un webservice qui transforme une position donnée en une description normalisée d'une localisation (Adresse avec un point) dans laquelle l'adresse peut être définie comme une adresse postale, une rue ou une commune. L'opération est donc définie par la relation

$$\text{Position}(x, y) \Rightarrow \text{Adresse normalisée} + \text{Position}(x, y)$$

Ce service permet de répondre aux questions suivantes :

- Quelle est l'adresse la plus proche des coordonnées X,Y ?
- Quelles sont les adresses dans un rayon de 250 mètres du point de coordonnées X,Y ?
- Quelle est le nom de la rue la plus proche des coordonnées X,Y ?
- Dans quelle commune tombe les coordonnées X,Y ?

Les usages classiques de ce webservice incluent l'auto-complétion des informations d'adresse depuis la position de l'utilisateur ou l'obtention d'information depuis une interface carte.



Notons que si le point en entrée est en Wallonie, le résultat fourni au minimum la commune contenant le point en entrée.

2.2. Appel au service

Les appels au webservice REVGEOCODE se font par une interface RESTful. De simples requêtes http permettent ainsi d'interroger le service. Les paramètres sont encodés dans l'URL en tant que query strings pour les requêtes de type GET alors que les requêtes de type POST nécessitent des données encodées en JSON. Toutes les réponses sont encodées en JSON.

2.3. Paramètres et dépendances

Les paramètres incluent :

- **x** longitude ou coordonnée X
- **y** latitude ou coordonnée Y
- **bbox** l'extension spatiale doit-elle être renvoyée ? Les valeurs acceptées sont [false, true] avec une valeur par défaut à false.
- **geom** la géométrie doit-elle être renvoyée ? Les valeurs acceptées sont [false, true] avec une valeur par défaut à false.
- **lang** langue de préférence de la réponse. Les valeurs acceptées sont [fr, de, nl].
- **distance** distance maximale de recherche autour du point d'entrée.

2.4. Exemple d'utilisation

Prenons le cas d'un formulaire web dans lequel la complétion d'un formulaire d'adresse est rempli automatiquement par la localisation via l'API Html5 de géolocalisation.

1) Recherche de l'adresse la plus proche dans un rayon de 50 m par un clic sur la carte

REQ : <http://geoservices.wallonie.be/geocodeWS/revgeocode?x=177053&y=135542&distance=50>

RES :

```
{"geotag":null,"location":{"x":177053.0,"y":135542.0},"candidates":[{"score":-1,"city":{"name":"Gembloux","ident":"92142"},"street":{"name":"Rue Feroobu","ident":7756413},"house":{"name":"21","ident":442393}},{score":-1,"city":{"name":"Gembloux","ident":"92142"},"street":{"name":"Rue Feroobu","ident":7756413},"house":{"name":"17","ident":441804}},{score":-1,"city":{"name":"Gembloux","ident":"92142"},"street":{"name":"Rue Feroobu","ident":7756413},"house":{"name":"15","ident":441805}}],"message":null}
```

2) Parsing du résultats et complétion du formulaire. Des bibliothèques javascript comme JQuery permettent de réaliser cette étape de façon aisée.

3. /parser

3.1. Objectif

Le service /PARSER vise le découpage d'une adresse exprimée sous la forme d'une chaîne libre de caractères en composantes d'adresse normalisées comme le numéro de maison, le nom de la rue, le nom de la commune et le code postal. Ce service est naturellement accessible sur réseau. Le parser procède donc à l'opération suivante :

Adresse libre => Composantes normalisée d'adresse

Le processus de géocodage est illustré ci-dessous avec la conversion de l'adresse postale « 145 Boulevard Joseph Tirou, 6000 Charleroi » vers les composantes normalisées.

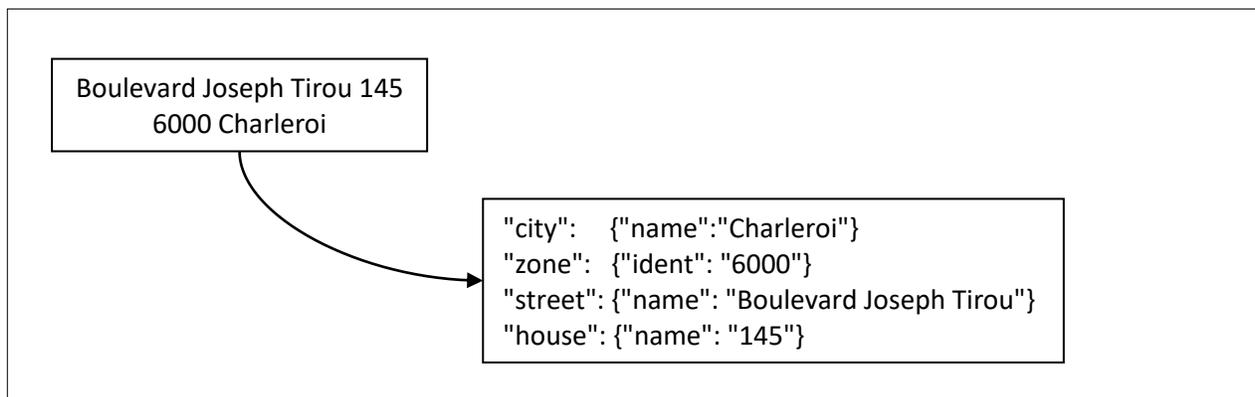


Figure 1 Parsing d'adresse postale en composantes

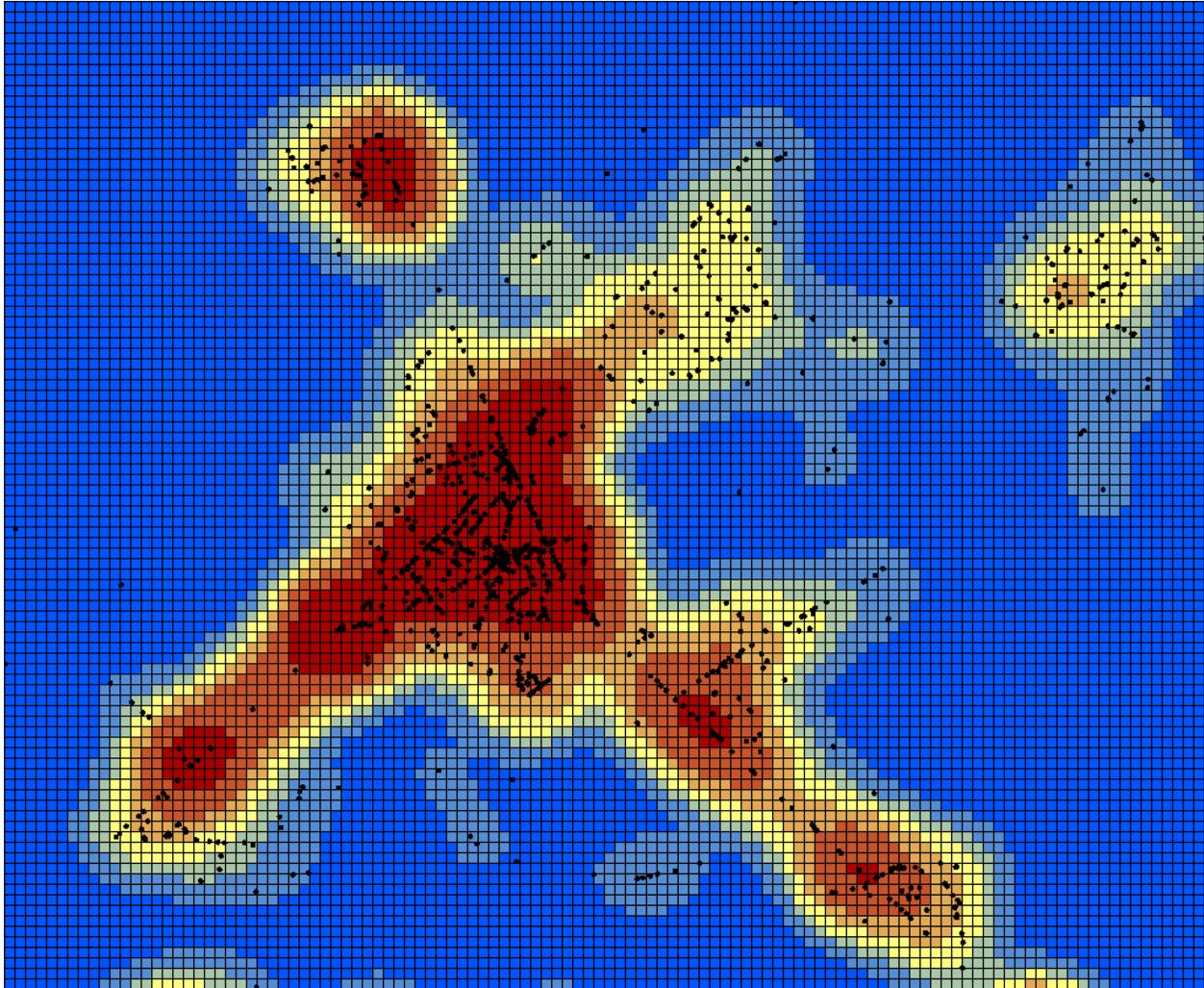
Ce service permet principalement de préparer les données brutes avant l'étape de géocodage. Il permet également le remplissage automatique de champs/variables structurées depuis une entrée libre. Ce service est utilisé en interne par le service de géocodage.

3.2. Appel au service

Les appels au webservice PARSEUR se font par une interface RESTful. De simples requêtes HTTP permettent ainsi d'interroger le service. Les paramètres sont encodés dans l'URL sous la forme de query strings pour les requêtes de type GET alors que les requêtes de type POST nécessitent des données encodées en JSON. Toutes les réponses sont encodées en JSON.

3.3. Principe de fonctionnement

Afin de faciliter le travail de découpage et de diminuer la complexité, les adresses sont supposées suivre le modèle standard des adresses attendues par Bpost et décrit dans la page [Comment bien noter une adresse ? Lisez tout à ce sujet | bpost | bpost](https://www.bpost.be/fr/tout-sur-les-adresses) (<https://www.bpost.be/fr/tout-sur-les-adresses>) .



Le service accepte un seul et unique paramètre :

- **address**: adresse postale sous la forme d'une chaîne de caractères.

3.4. Exemple d'utilisation

Prenons le cas d'un formulaire web dans lequel l'utilisateur est appelé à remplir une adresse. Ce formulaire offre deux manières d'encoder : le texte libre ou les composantes d'adresses. Avant stockage en DB, le texte libre nécessite d'être découpé en composantes.

Ajout du contact : secrétaire ✕

Nom* Prénom*

N° de registre national*

Adresse administrative

ou

Rue Numéro

Code postal Commune

Téléphone fixe* Téléphone mobile*

E-mail* Fax

Figure 2 Utilisation du moteur de géocodage pour spatialiser une adresse.

1) Découpage du texte libre en composantes :

REQ :

[http://geoservices.wallonie.be/geocodeWS/parser?address=83b chaussée de Charleroi 5030 Namur](http://geoservices.wallonie.be/geocodeWS/parser?address=83b%20chaussée%20de%20Charleroi%205030%20Namur)

RES :

```
{
  "city":{"name":"Namur"},
  "zone":{"ident":"5030"},
  "street":{"name":"chaussée de Charleroi"},
  "house":{"name":"83b"}
}
```

2) Parsing de la réponse et remplissage des champs des composantes

4. /geocode

4.1. Objectif

« Un service de géocodage est un service accessible sur un réseau qui transforme la description d'une localisation en une description normalisée avec une géométrie ponctuelle » (OGC, **OpenLS 1.2**, p4 - <https://www.ogc.org/standards/ols>) sur base de la relation

Adresse => Adresse normalisée + Point(x, y)

Le processus de géocodage est illustré ci-dessous avec la conversion de l'adresse postale « 145 Boulevard Joseph Tirou, 6000 Charleroi » vers les coordonnées [x, y] avant le placement sur carte du marqueur indiquant cette adresse.

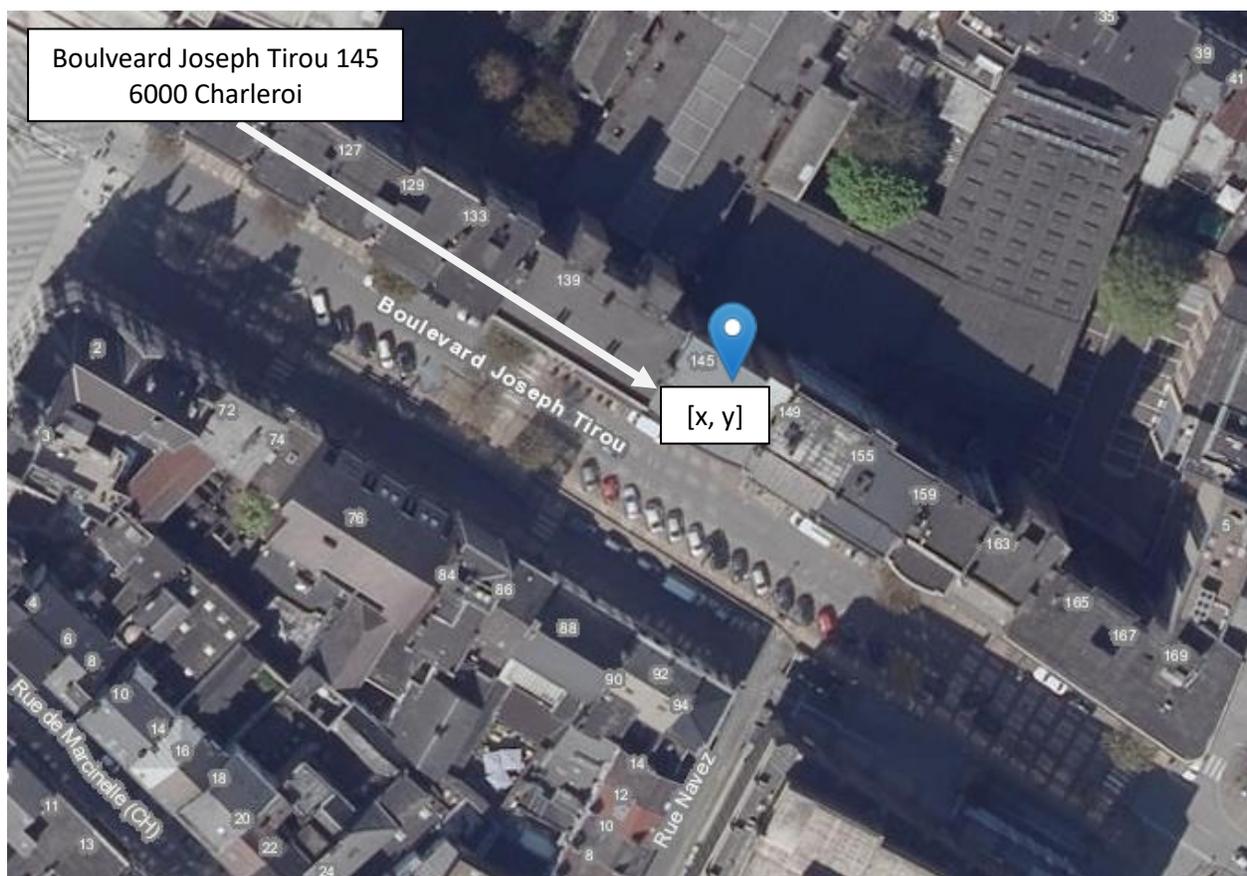


Figure 3 Processus de géocodage d'une adresse postale.

Ce service permet de répondre à des questions telles que :

- Quelles sont les coordonnées [X, Y] du numéro 10 de la Place Albert 1er à 6020 Charleroi ?

- Quel est le plus petit rectangle (*bbox*) contenant la Chaussée de Charleroi à Namur ?
- Où est situé l'hôtel de ville de Mons ?
- Quel est l'identifiant ICAR de la commune de Vielsalm ?
- Quel est l'identifiant ICAR de la Chaussée de Charleroi à Namur ?
- Quel est l'identifiant ICAR du numéro 83b de la Chaussée de Charleroi à Namur ?

Les usages classiques de ce webservice incluent la spatialisation des données et la standardisation des listes d'adresse.

4.2. Appel au service

Les appels au webservice GEOCODE se font par une interface RESTful. De simples requêtes HTTP permettent ainsi d'interroger le service. Les paramètres sont encodés dans l'URL sous la forme de query strings pour les requêtes de type GET alors que les requêtes de type POST nécessitent des données encodées en JSON. Toutes les réponses sont encodées en JSON.

4.3. Principe de fonctionnement

La première étape consiste à éventuellement diviser l'adresse reçue en composant logique (numéro, rue, code postal, commune). La recherche de l'adresse à géocoder se déroule alors en cascade du niveau supérieur que sont les communes vers le niveau de détail le plus important que sont les adresses. À chaque niveau, la recherche peut s'interrompre si l'élément correspondant n'est pas trouvé dans la BD. Le résultat du géocodage peut donc se trouver au niveau de la commune, de la zone d'adresse, de la rue ou finalement du bâtiment.

Le géocodage de l'adresse « Rue de l'Anse 10, 5000 Namur » peut ainsi se décomposer en étapes :

1. L'adresse est éventuellement divisée en composante logique
 - munName = Namur,
 - locName = 5000,
 - strName = Rue de l'Anse,
 - houName = 10.
2. La recherche débute avec le niveau le plus élevé, c'est-à-dire le nom de la commune « Namur ». Cette recherche renvoie un identifiant de commune (munIdent: 92094). Le succès de cette première étape permet à l'algorithme de continuer l'exploration des étages inférieures.
3. La recherche enchaîne avec la recherche du nom de rue « Rue de l'Anse » au sein de toutes les rues de la commune de Namur. Une correspondance exacte n'est pas trouvée dans la table des rues mais une logique de recherche approximative et phonétique permet toutefois de trouver une rue dont le nom est très proche « Rue de l'Ange ». Le demi-succès de cette exploration des rues permet de continuer à l'étape suivante.
4. La recherche se termine avec le numéro de maison au sein des adresses rattachées à la Rue de l'Ange trouvée à l'étape précédente. Un résultat exact est trouvé dans la table des adresses et

permet de ressortir les coordonnées XY de cette adresse ainsi que les identifiants ICAR de cette adresse

4.4. Paramètres et dépendances

Les paramètres du service sont les suivants :

- **city** identifiant ou nom de la commune.
- **zone** identifiant ou nom de la localité.
- **street** identifiant ou nom de la rue.
- **house** identifiant ou nom du numéro de police.
- **address** adresse complète selon un format texte libre
- **bbox** l'extension spatiale doit-elle être renvoyée ? Les valeurs acceptées sont [false, true] avec une valeur par défaut à false.
- **geom** la géométrie doit-elle être renvoyée ? Les valeurs acceptées sont [false, true] avec une valeur par défaut à true.
- **lang** langue de préférence de la réponse. Les valeurs acceptées sont [fr, de, nl].
- **crs** code de la projection en sortie. Les valeurs acceptées sont [EPSG :31370, EPSG :3812, EPSG :3857, EPSG :4326] avec une valeur par défaut à EPSG :31370.
- **nap** stratégie de traitement des numéros de maison manquants. Les valeurs possibles sont [closest, centroid, random, none].

Deux modes distincts de géocodage co-existent : (i) le géocodage sur les composantes d'adresse et (ii) le géocodage sur les adresses en texte libre. Le choix du mode est fonction des paramètres d'entrée. Avec le seul paramètre « **address** », le géocodage se fait en mode texte libre alors que tous les autres cas utilisent le mode par composante. Le résultat global est naturellement plus rapide à obtenir et plus précis dans ce second mode car l'incertitude sur le découpage disparaît.

Dans le mode par composantes, les dépendances entre les paramètres suivent la logique de la pyramide de recherche City => Street => House. Le seul paramètre indispensable est le paramètre « city ». Les paramètres « street » et « house » sont optionnels mais ils doivent être fournis dans la requête si le niveau supérieur est donné. Par type de recherche, les contraintes sont les suivantes :

- Recherche de commune
Le seul paramètre obligatoire est « **city** » sous la forme d'une chaîne de caractères.
- Recherche de rue
La recherche d'une rue implique l'utilisation des paramètres obligatoires « **city** » et « **street** » tous les deux sous la forme d'une chaîne de caractères.
- Recherche de numéro de police

La recherche d'un numéro de police nécessite l'encodage des paramètres « city », « street » et « house » tous les 3 sous la forme d'une chaîne de caractère.

4.5. Exemple d'utilisation

Prenons le cas d'un formulaire web dans lequel l'utilisateur est appelé à remplir une adresse postale composée de 3 champs : Commune, Rue et Numéro comme illustré ci-dessous. L'usage visé ici est de traduire l'adresse encodée dans ce formulaire en un point sur la carte et le zoom automatique sur ce point dans le volet carte. Après l'encodage des informations par l'utilisateur, le formulaire pourrait ressembler à l'exemple ci-dessous :

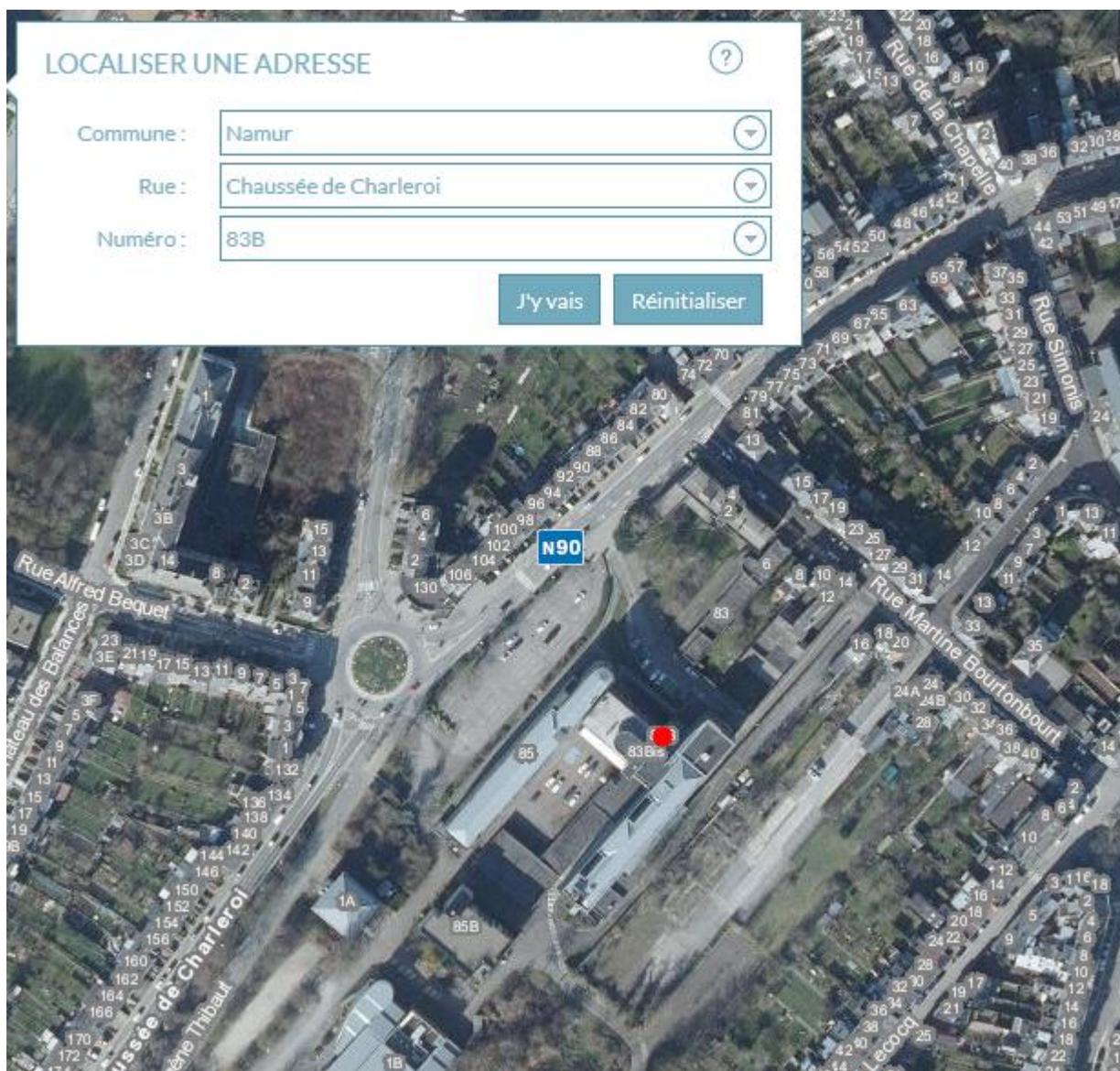


Figure 4 Utilisation du moteur de géocodage pour spatialiser une adresse.

3) Récupération des composantes de l'adresse et soumission des données du formulaire en http :

REQ :<http://geoservices.wallonie.be/geocodeWS/geocode?city=Namur&street=Chauss%C3%A9e%20de%20Charleroi&house=83&bbox=true>

RES :

```
{
  "address": {
    "city": {
      "name": "Namur"
    },
    "street": {
      "name": "Chaussée de Charleroi"
      "resolvedNames": {
        "fr": "Chaussée de Charleroi"
      },
    },
    "house": {
      "name": "83"
    }
  },
  "candidates": [{
    "score": 111,
    "city": {
      "lang": "F",
      "name": "Namur",
      "ident": "92094"
    },
    "zone": {
      "name": "Namur",
      "ident": "5000"
    },
    "street": {
      "name": "Chaussée de Charleroi",
      "ident": 7753167
    },
    "house": {
      "name": "83",
      "ident": 349081
      "geometry": {
        "type": "Point",
        "crs": {
          "type": "name",
          "properties": {
            "name": "EPSG:31370"
          }
        }
      },
      "coordinates": [183519.198700003,127944.102400001]
    },
    "bbox":
    [183519.198700003,127944.102400001,183519.198700003,127944.102400001],
  }
  ]
}
```

4) Parsing de la réponse et exploitation des résultats

La sortie du service de géocodage est un objet JSON composé d'un objet « address » et d'une liste de candidats potentiels. L'objet « address » correspond au résultat de l'interprétation des paramètres en entrée sous forme des 4 composantes de l'adresse standardisée. La liste des candidats présente l'ensemble des correspondances trouvées dans la base de données de référence. Chaque résultat de cette liste est noté par un score.

Le score de chaque candidat est un code de 3 chiffres. La première position de ce code décrit le résultat sur l'entité commune, le second sur l'entité rue et le troisième sur l'entité numéro de maison. Les valeurs possibles pour chaque niveau sont décrites par le service /getScores

REQ : <http://geoservices.wallonie.be/geocodeWS/getScores>

RES (partiel):

```
[
  {
    "code": 0,
    "value": "Not matching found",
    "level": null
  },
  {
    "code": 1,
    "value": "House number matched exactly",
    "level": "house"
  },
  {
    "code": 2,
    "value": "House number cleaned (ex: 12A -> 12 or 12A/101 -> 12) matched",
    "level": "house"
  },
  {
    "code": 3,
    "value": "House number [+2, -2], [+4, -4] matched",
    "level": "house"
  },
  {
    "code": 4,
    "value": "House number [+1, -1], [+3, -3] matched (Probable opposite side)",
    "level": "house"
  },
  {
    "code": 5,
    "value": "House number not found, number allocated randomly along segment",
    "level": "house"
  },
  {
    "code": 6,
    "value": "House number not found, centroid of street segment returned",
    "level": "house"
  },
  ...
  {
    "code": 10,
    "value": "Exact street name matched",
    "level": "street"
  },
  {
    "code": 20,
    "value": "Fuzzy street name matched using Levenshtein text distance",
    "level": "street"
  },
  ...
]
```

```

{
  "code": 100,
  "value": "Municipality name matched exactly",
  "level": "city"
},
{
  "code": 200,
  "value": "Fuzzy municipality name matched using Levenshtein text distance",
  "level": "city"
},
...
]

```

Selon cette codification, un score de 125 se réfère à un nom de commune trouvé exactement (1**), un nom de rue trouvé sur base de la distance de Levenshtein (*2*) et un numéro de maison non trouvé mais alloué aléatoirement le long du segment de rue (**5).

Un second service nommé /getScore prend en argument un score de géocodage et détaille le résultat. Pour notre exemple du 83b Chaussée de Charleroi à Namur, le score était de 111 à passer comme argument :

REQ : <http://geoservices.wallonie.be/geocodeWS/getScore?score=111>

```

RES :
{
  "score": "111",
  "levels": [
    {
      "code": 100,
      "value": "Municipality name matched exactly",
      "level": "city"
    },
    {
      "code": 10,
      "value": "Exact street name matched",
      "level": "street"
    },
    {
      "code": 1,
      "value": "House number matched exactly",
      "level": "house"
    }
  ]
}

```

Ce score de 111 illustre donc une correspondance parfaite du nom de la commune, du nom de la rue et du numéro de maison. Nous sommes donc certains de la réponse.

En cloture de cette section sur le service /geocode, notons que seules les abréviations supportées par Bpost sont intégrées à l'outil de géocodage. Toutes les autres abréviations sont ignorées par le système. La liste complète par langue est consultable en ligne sur la page <http://www.bpost.be/site/fr/residential/letters-cards/send/addressing.html>. Par langue, nous avons ainsi :

```

"fr": {

```

```
"Av ": "Avenue ",
"Bd ": "Boulevard ",
"Ctre ": "Centre ",
"Pl ": "Place ",
"Rte ": "Route ",
"Sq ": "Square ",
"ZI ": "Zone Industrielle ",
"Chée ": "Chaussée ",
"Imp ": "Impasse "
},
"nl": {
  "str": "straat",
  "ln": "laan",
  "pl": "plein",
  "stwg": "steenweg",
  "IZ": "industriezone",
  "Geb": "gebouw",
  "Sq": "square"
},
"de": {
  "Str": "Strasse",
  "All": "Allee",
  "Pl": "Platz",
  "GG": "Gewerbegebiet",
  "Res": "Residenz"
```

5. /searchAll

5.1. Objectif

En complément du service de géocodage classique, les données ICAR sont également reformattées et collationnées aux côtés d'informations sur les routes régionales et différents POIs afin d'offrir des fonctionnalités de recherche plein texte sous le end point /searchAll. Les objectifs de ce service sont similaires aux objectifs du service de géocodage avec la conversion d'une chaîne d'information textuelle en un point sur la carte. L'absence de traitement de l'information en input et les sources de données différentes offrent toutefois des possibilités autres que celles du service /geocode. La relation caractérisant ce service prend la forme :

Adresse => Point(x,y).

Parmi les questions auxquelles permet de répondre ce service, nous trouvons :

- Quelles sont les coordonnées [X, Y] du numéro 10 de la Place Albert 1er à 6020 Charleroi ?
- Quel est le plus petit rectangle (*bbox*) contenant la Chaussée de Charleroi à Namur ?
- Où se situe la gare de Mons ?
- Où se trouve la N972 sur Mehaigne ?

Les usages classiques de ce webservice incluent la recherche rapide d'informations et la spatialisation des données.

5.2. Appel au service

Les appels au webservice SEARCHALL se font par une interface RESTful. De simples requêtes HTTP permettent ainsi d'interroger le service. Les paramètres sont encodés dans l'URL sous la forme de query strings pour les requêtes de type GET alors que les requêtes de type POST nécessitent des données encodées en JSON. Toutes les réponses sont encodées en JSON.

5.3. Principe de fonctionnement

Contrairement au service /geocode, l'adresse passée en input est utilisée entière et directement par le moteur de recherche plein texte dans la base de données. Le découpage, la normalisation des composantes, la résolution des abréviations, le retour d'une adresse normalisée ou encore la recherche pyramidale des composantes depuis la commune jusqu'à l'adresse ne sont donc pas d'application dans ce service.

En outre, la comparaison se fait actuellement avec une base de données dénormalisée qui contient l'ensemble des informations sur les entités

- (1) communes,
- (2) zones d'adresses,

- (3) noms de rue,
- (4) adresses,
- (5) routes régionales découpées par commune,
- (6) routes régionales segmentées par zone d'adresse et finalement,
- (7) gares de Wallonie.

La catégorisation des sources cet indice de niveau permet d'offrir des possibilités de filtrage dans la recherche comme nous le verrons dans la section suivante.

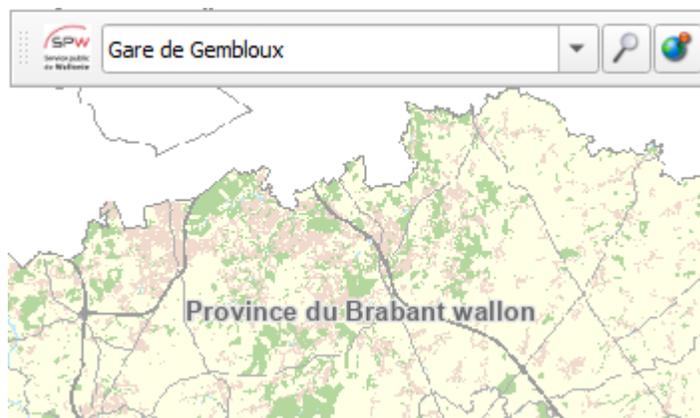
5.4. Paramètres et dépendances

Les paramètres du service sont les suivants :

- **id** identifiant optionnel de l'adresse pour le tracing dans le retour.
- **address** adresse recherchée sous la forme d'une chaîne de caractères. Il s'agit du seul paramètre obligatoire du service.
- **levels** filtrage optionnel de la recherche sur certaines entités. Par défaut tous les niveaux sont retournés. Les valeurs acceptées correspondent aux différentes sources [1=commune ; 2=zone d'adresse ; 3=nom de rue ; 4=adresse ; 5=route régionale par commune ; 6=route régionale par zone d'adresse ; 7=gare] séparée par une virgule si plusieurs valeurs sont choisies.
- **bbox** l'extension spatiale doit-elle être renvoyée ? Les valeurs acceptées sont [false, true] avec une valeur par défaut à false.
- **geom** la géométrie doit-elle être renvoyée ? Les valeurs acceptées sont [false, true] avec une valeur par défaut à true.
- **lang** langue de préférence de la recherche. Les valeurs acceptées sont [fr, de, nl].
- **crs** code de la projection en sortie. Les valeurs acceptées sont [EPSG :31370, EPSG :3812, EPSG :3857, EPSG :4326] avec une valeur par défaut à EPSG :31370.

5.5. Exemple d'utilisation

Prenons le cas d'un formulaire web dans lequel l'utilisateur doit pouvoir faire une recherche pour localiser un lieu. L'interface pourrait alors prendre la forme d'un simple input de type texte :



L'utilisateur cherche ici à localiser la gare de Gembloux. L'appel correspondant à la soumission de ce formulaire au service /searchAll est alors

REQ : <https://geoservices.wallonie.be/geocodeWS/searchAll?address=gare%20de%20gembloux>

RES :

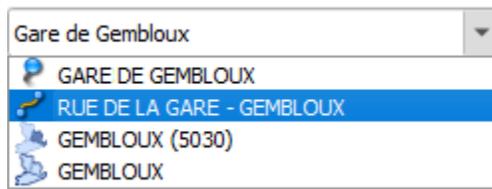
```
{
  "geotag": {
    "address": "gare de gembloux",
    "bbox": false,
    "geom": false,
    "crs": "EPSG:31370"
  },
  "candidates": [
    {
      "score": 0.00538345706,
      "geotag": {
        "ident": "7756254",
        "level": 3,
        "name": "Rue de la Gare 5030 Gembloux",
        "lang": "fr"
      }
    },
    {
      "score": 0.000515803753,
      "geotag": {
        "ident": "416371",
        "level": 4,
        "name": "Rue de la Gare 8 5030 Gembloux",
        "lang": "fr"
      }
    },
    {
      "score": 0.000515803753,
      "geotag": {
        "ident": "416372",
        "level": 4,
        "name": "Rue de la Gare 4 5030 Gembloux",
        "lang": "fr"
      }
    },
    {
      "score": 0.000515803753,
      "geotag": {
        "ident": "416376",
        "level": 4,
        "name": "Rue de la Gare 3 5030 Gembloux",
        "lang": "fr"
      }
    },
    {
      "score": 0.000515803753,
      "geotag": {
        "ident": "1425458",
        "level": 4,
        "name": "Rue de la Gare 1 5030 Gembloux",
        "lang": "fr"
      }
    },
    {
      "score": 0.000287875562,
      "geotag": {
        "ident": "416380",
        "level": 4,
        "name": "Rue de la Gare 18 5030 Gembloux",
        "lang": "fr"
      }
    }
  ]
}
```

```

    "score": 0.0315464884,
    "geotag": {
      "ident": "BE00438",
      "level": 7,
      "name": "Gare de Gembloux",
      "lang": "fr"
    }
  ]
}

```

La présentation des résultats à l'utilisateur peut exploiter directement la propriété « level » pour grouper les résultats par catégorie avec un pictogramme par type d'entité :



La propriété « score » permet de classer les résultats. Ce score est directement extrait du moteur de recherche plein texte de la DB et est défini comme la densité de couverture pour le vecteur du document et la requête donnés comme décrit par Clarke et al. (1999)¹.

Outre la possibilité d'obtenir l'extension spatiale de l'entité et sa géométrie avec les paramètres « bbox=True » et « geom=True », la recherche uniquement sur un type d'entité comme les gares par exemple est possible par l'exploitation du paramètre « levels » :

REQ : <https://geoservices.wallonie.be/geocodeWS/searchAll?address=gembloux&levels=7&geom=True>

```

RES :
{
  "geotag": {
    "address": "gembloux",
    "levels": "7",
    "bbox": false,
    "geom": true,
    "crs": "EPSG:31370"
  },
  "candidates": [
    {
      "score": 0.0630929768,
      "geotag": {
        "ident": "BE00438",
        "level": 7,
        "name": "Gare de Gembloux",
        "lang": "fr",
        "geometry": {
          "type": "Point",
          "crs": {
            "type": "name",
            "properties": {
              "name": "EPSG:31370"
            }
          }
        }
      }
    }
  ]
}

```

¹ Charles L.A. Clarke, Gordon V. Cormack, Elizabeth A. Tudhope (2000) *Relevance ranking for one to three term queries*. Information Processing & Management, Volume 36, Issue 2, Pages 291-311. [https://doi.org/10.1016/S0306-4573\(99\)00017-5](https://doi.org/10.1016/S0306-4573(99)00017-5)

```

    },
    "coordinates": [
      172835.01647250125,
      139989.62418902014
    ]
  }
}
]
}

```

6. /searchAll ou /geocode ?

Bien que les services /searchAll et /geocode offrent des fonctionnalités similaires, ils diffèrent sur de nombreux points dont les données sources et les algorithmes utilisés. Revenons sur les éléments essentiels afin d'opter pour le bon outil selon les besoins.

Le service /searchAll à une vocation généraliste. Il permet de se localiser rapidement en encodant toutes les informations en vrac mais pas toujours de façon très précise. Il travaille sur une base de données dénormalisée qui consolident les informations sur les noms des communes, les noms des zones d'adresses, les noms des rues, les adresses, les routes régionales et les gares. Il exploite des algorithmes de recherche plein texte pour assurer de bonne performance mais n'offre aucune procédure de failback ou de nettoyage des données. Le score ne permet pas de comprendre le cheminement du matching. Il vise la vitesse au détriment de la précision.

REQ :

<https://geoservices.wallonie.be/geocodeWS/searchAll?address=83B%20chauss%C3%A9e%20de%20Charleroi%205000%20Namur&geom=True>

```

RES :
{
  "geotag": {
    "address": "83B chaussée de Charleroi 5000 Namur",
    "bbox": false,
    "geom": true,
    "crs": "EPSG:31370"
  },
  "candidates": [
    {
      "score": 0.0476190485,
      "geotag": {
        "ident": "1947675",
        "level": 4,
        "name": "Chaussée de Charleroi 83B 5000 Namur",
        "lang": "fr",
        "geometry": {
          "type": "Point",
          "crs": {
            "type": "name",
            "properties": {
              "name": "EPSG:31370"
            }
          }
        }
      },
      "coordinates": [
        183497.781400003,

```

```
127899.9259999999
  ]
}
}
]
}
```

Le service /geocode à une vocation spécialisée. Il localise de façon précise en utilisant la structure et les relations entre les entités définies dans le modèle ICAR. Il n'exploite que les données de la source authentique ICAR. Il découpe les adresses en composantes d'adresse avant de rechercher chaque composante graduellement depuis la commune jusqu'au niveau adresse. Pour ce faire, il utilise en cascade des distances lexicales de plus en plus floues (Levnshtein , trigrammes, double metaphone, ...). Le résultat est standardisé et chaque composante trouvée est retournée avec ses identifiants ICAR. Chaque candidat est scoré et l'interprétation du score permet de comprendre la résolution du géocodage. Il est conçu pour travailler de concert avec /geolist pour populer les datalist au sein des formulaires. Il vise la précision au détriment de la vitesse.

REQ :

<https://geoservices.wallonie.be/geocodeWS/geocode?address=83B%20chauss%C3%A9e%20de%20Charleroi%205000%20Namur>

```
RES :
{
  "address": {
    "city": {
      "name": "Namur"
    },
    "zone": {
      "ident": "5000"
    },
    "street": {
      "resolvedNames": {
        "fr": "chaussée de Charleroi"
      },
      "name": "chaussée de Charleroi"
    },
    "house": {
      "name": "83B"
    }
  },
  "candidates": [
    {
      "score": 111,
      "city": {
        "ident": "92094",
        "name": "Namur",
        "lang": "F"
      },
      "zone": {
        "ident": "5000",
        "name": "Namur"
      },
      "street": {
        "ident": 7753167,
        "name": "Chaussée de Charleroi"
      },
      "house": {
        "ident": 1947675,
        "name": "83B",
        "rrn": "1624",

```

```
"geometry": {
  "type": "Point",
  "crs": {
    "type": "name",
    "properties": {
      "name": "EPSG:31370"
    }
  },
  "coordinates": [
    183497.781400003,
    127899.9259999999
  ]
}
]
}
```