



Développement des Applications & Gestion des Projets

ToepassingsOntwikkeling & Projectbeheer

[www.smals.be](http://www.smals.be)

**referenceData/geography/v  
1/belgianCities**

## TABLE OF CONTENTS

1.1.1	referenceData/geography/v1/belgianCities .....	3
1.1.2	UC01 - Get a list of all belgian cities .....	6
1.1.3	UC02 - Get belgian cities based on the name of the city .....	13
1.1.4	UC03 - Get belgian cities based on postal code .....	18
1.1.5	UC04 - Get belgian cities based on postal code and the name of the city .....	24
1.1.6	UC05 - Get belgian cities based on region code.....	29
1.1.7	UC06 - Get belgian cities based on linguistic region code .....	31

## referenceData/geography/v1/belgianCities

### STRATEGIC REUSE

#### 1.1.1 referenceData/geography/v1/belgianCities

Created By: TT-Func on 15-02-16 Last Modified: 22-05-20, Version: 1.0

##### Description

referenceData/geography/v1/belgianCities (= referenceData/geography/v1/countries/150/cities) est un service qui permet de consulter une liste de communes belges disponibles en 3 langues (FR, NL, DE) ainsi que diverses informations à propos de celles-ci, comme par exemple : le code INS de la commune principale, le code postal officiel de la commune, le nom de la commune principale et de la commune non-fusionnée, le code région et le code région linguistique ...

**Remarque: Quoique le service offre le nom des communes dans les 3 langues, il est recommandé de fournir à nos clients les communes exclusivement dans la langue régionale et non pas selon la langue de l'utilisateur.**

**Exemples: Antwerpen au lieu de Anvers, Liège au lieu de Luik !**

##### Estimated effort

For the use of belgianCities in your application, you need about 2 days while writing all based on the service definition (swagger).

##### Limitation

Comme il n'existe pas de Dossier Unique, toutes les applications peuvent utiliser ce service.

Available as Web Service REST on ESB SOA Social Security (soon on API Portal Social Security, same URL).

##### Recommandations

- If the API lacks any functionality, please ask the responsible functional analyst. New features are better implemented on server-side, so they can be reused by other clients and the service provider keeps ownership of the service features.
- Le choix de l'implémentaiton cliente est libre. Cependant, nous conseillons d'utiliser JAX-RS 2 et son implémentation correcte sur le serveur d'application (RestEasy sur JBoss ou Jersey sur Weblogic).

API:

<https://services.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities>

alias

<https://services.socialsecurity.be/REST/referenceData/geography/v1/belgianCities>

Monitoring:

<https://professionalservices-int.socialsecurity.be/REST/referenceData/geography/v1/api/healthCheck>

Swagger file:

<https://services.socialsecurity.be/REST/referenceData/geography/v1/api/doc/swagger.json>

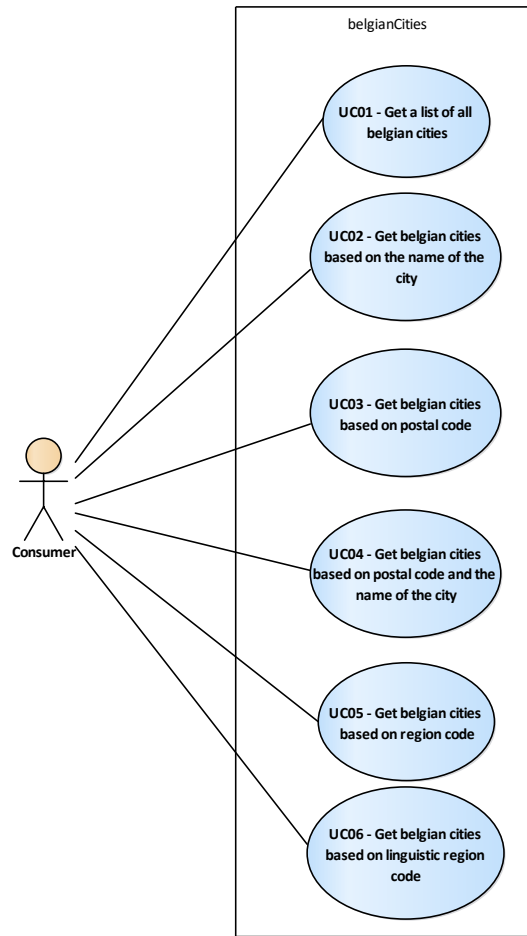


Figure: 1

## 1.1.2 UC01 - Get a list of all belgian cities

Created on 22-03-16. Last modified on 29-05-20.

### Functional description

Cette fonction permet d'obtenir une liste de toutes les communes belges, principales et non-fusionnées.

Pour chacun des enregistrements, on peut avoir les informations suivantes:

- **Code INS**, s'il s'agit d'une commune principale, en 5 positions numériques (cityNisCode) (alias: municipalityCode ou BeStId-Commune) (certaines "communes" spéciales (49) n'ont pas de code INS ex: Bruxelles X, Bruxelles OTAN, ....)  
**Remarque: les 2 premières positions du code INS donnent la province et l'arrondissement:**
  - 1\*\*\*\*: Anvers
  - 21\*\*\*\* Brabant Bxl Capitale
  - 23\*\*\*\* Vlaams brabant
  - 24\*\*\*\* Vlaams brabant
  - 25\*\*\*\* Brabant wallon
  - 3\*\*\*\* West Vlaanderen
  - 4\*\*\*\* Oost Vlaanderen
  - 5\*\*\*\* Hainaut / Henegouwe
  - 6\*\*\*\* Liège
  - 7\*\*\*\* Limburg
  - 8\*\*\*\* Luxembourg
  - 9\*\*\*\* Namur
- **Code postal belge officiel de la commune** en 4 positions numériques (zipCode)
- **Nom de la commune** (principale ou non-fusionnée) en 3 langues (FR, NL, DE) (cityName)
- **Le code région de la commune:** BRU = Bruxelles Capitale; VLG = Vlaams gewest; WAL = Région wallonne (regionCode)
- **Le code région linguistique de la commune:** BIL (bilingue), FR, NL, DE (linguisticRegionCode)  
Attention: DE regroupe les 9 communes germanophones de [Eupen](#), [La Calamine](#), [Lontzen](#), [Raeren](#), [Amblève](#), [Bullange](#), [Burg-Reuland](#), [St Vith](#), [Butgenbach](#) mais pas [Malmedy](#) et [Waimes](#) qui sont francophones.
- **Le régime linguistique de la commune:** F, N, DF, FD, FN, NF, FN\_NF, NF\_FN où F= Français, N=Néerlandais et D= Allemand (officialLanguage)  
**si régime linguistique :**

\* FN-NF ou NF-FN = commune bilingue (les 19 communes )

\* F, FN, FD = commune francophone ou à facilités linguistiques néerlandophone ou germanophone (FD correspond aux communes de [Malmedy](#) et [Waimes](#))

\* DF = commune germanophone à facilités linguistiques francophone (9 communes : [Eupen](#), [La Calamine](#), [Lontzen](#), [Raeren](#), [Amblève](#), [Bullange](#), [Burg-Reuland](#), [St Vith](#), [Butgenbach](#))

\* N, NF = commune néerlandophone ou à facilités linguistiques francophone

- **Date de début de validité de la commune** (si elle est différente de 1/1/1900)  
(validityStartDate)
- **Date de fin de validité de la commune** (si elle est différente de 31/12/9999)  
(validityEndDate)
- si la commune n'est pas la commune principale, le **lien de l'identifiant technique de la commune principale correspondante** (mainCity)
- le **lien de son identifiant technique** (\_links)

#### Remarques:

\* à chaque commune non-fusionnée correspond une commune principale.

#### Exemples:

- 4000 Rocourt ou Glain ont comme commune principale Liège 4000
- 4020 Bressoux, Jupille-sur-Meuse ou Wandre ont comme commune principale Liège 4000

\* la liste des communes rendues n'est pas triée

#### Sélection sur les rows:

- **"isMainCity"**: vrai ou faux  
==> retourne la liste des communes soit principales, soit non-fusionnées, si ce paramètre n'est pas donné, le résultat comprendra toutes les communes
- **"validityStartDate.lte"** or **"validityStartDate.gte"** = une date (YYYYMMDD)
- **"validityEndDate.lte"** or **"validityEndDate.gte"** = une date (YYYYMMDD)  
Exemples:  
validityStartDate.lte= 20150101&validityEndDate.gte=20150131:  
==> liste des communes actives tout le mois de janvier 2015  
validityStartDate.lte= 20150131&validityEndDate.gte=20150101:  
==> liste des communes actives au moins 1 jour en janvier 2015  
validityEndDate.gte=date du jour:  
==> liste des communes actives à la date du jour  
Remarque: validityStartDate.lte et validityEndDate.gte reprennent également les rows dont la date de début ou la date de fin est à null

#### Erreurs:

- si pas de record trouvé: liste vide avec totalItems=0
- si un des paramètres donné ne contient pas de valeur ==> 400 Bad Request: Query Parameter [xxx] cannot be empty

- Si un paramètre inconnu est ajouté ==> 400 Bad Request : Unexpected Query Parameter [xxx]

### Filtres possibles sur les colonnes:

- **"fields"**: On peut paramétrer les champs de retour dans la réponse avec "fields"  
==> accepte une sélection de champs dans la réponse, séparés par une virgule  
Exemple: &fields=zipCode, cityName
- **"language"**: "fr", "nl", "de" (2 lettres minuscules, 1 seul choix possible)  
==> retourne les communes, principales et/ou non-fusionnées, dans le code langue demandé, **trié alphabétiquement sur le nom de la commune non-fusionnée (cityName)**  
si "language" est différent des valeurs attendues  
==> 400 Bad Request: must match \"fr|nl|de\"

### Affichage:

#### paginer les résultats

- **"page"**  
==> permet d'afficher la page demandée
- **"pageSize"**  
==> permet d'afficher le nombre de résultats à afficher par page (par défaut, pageSize=10)  
**si pageSize=0 alors il affiche tous les résultats**

#### détailler les résultats

- **mainCityDetail="embed" or "link" (par défaut)**  
==> "embed": permet d'afficher le détail des données de la commune principale (au lieu du lien) à savoir: cityNisCode, zipCode (=mainZipCode), officialLanguage, cityName (=mainCityName) FR, NL, DE  
==> "link": affiche le lien vers la commune principale correspondante à la commune non-fusionnée (=valeur par défaut)

### **Résultat:**

- le nombre de résultats correspondant à la requête, est affiché, indépendamment de la pagination, dans "totalItems"

### **Exemple:**

```
{{url}}/{{geography-context-root}}/countries/150/cities
```

#### **renvoie les 3006 communes belges**

```
"items": [
```

```
  {...
```

```
  },
```

```
  {
```



```
"cityNisCode": 25112,
"zipCode": 1300,
"officialLanguage": "F",
"cityName": {
  "fr": "Wavre",
  "nl": "Waver",
  "de": "Wavre"
},
"regionCode": "WAL",
"linguisticRegionCode": "FR",
"_links": {
  "self": {
    "href": "https://professionalservices-
int.socialsecurity.be/REST/referenceData/geography/v1/belgianCities/3506110720",
    "type": "city"
  }
}
},
{
  "cityNisCode": 25050,
  "zipCode": 1310,
  "officialLanguage": "F",
  "cityName": {
    "fr": "La Hulpe",
    "nl": "Terhulpen",
    "de": "La Hulpe"
  },
  "regionCode": "WAL",
  "linguisticRegionCode": "FR",
  "_links": {
```

```
"self": {
  "href": "https://professionalservices-
int.socialsecurity.be/REST/referenceData/geography/v1/belgianCities/3103719680",
  "type": "city"
}
},
{
  "cityNisCode": 25043,
  "zipCode": 1315,
  "officialLanguage": "F",
  "cityName": {
    "fr": "Incourt",
    "nl": "Incourt",
    "de": "Incourt"
  },
  "regionCode": "WAL",
  "linguisticRegionCode": "FR",
  "_links": {
    "self": {
      "href": "https://professionalservices-
int.socialsecurity.be/REST/referenceData/geography/v1/belgianCities/2885747200",
      "type": "city"
    }
  }
},
{
  "cityNisCode": 25005,
  "zipCode": 1320,
  "officialLanguage": "F",
```

```
"cityName": {
  "fr": "Beauvechain",
  "nl": "Bevekom",
  "de": "Beauvechain"
},
"regionCode": "WAL",
"linguisticRegionCode": "FR",
"_links": {
  "self": {
    "href": "https://professionalservices-
int.socialsecurity.be/REST/referenceData/geography/v1/belgianCities/2684551680",
    "type": "city"
  }
}
],
"totalItems": 3006,
"totalPages": 301,
"currentPage": 1,
"_links": {
  "next": {
    "href": "https://professionalservices-
int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=2&pageSize=
10",
    "type": "Cities",
    "title": "next page of cities"
  },
  "last": {
    "href": "https://professionalservices-
int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=301&pageSize
=10",
```

```
"type": "Cities",
"title": "last page of cities"
},
"prev": {
  "href": "https://professionalservices-
    int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=
    10",
  "type": "Cities",
  "title": "prev page of cities"
},
"self": {
  "href": "https://professionalservices-
    int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=
    10",
  "type": "Cities",
  "title": "self page of cities"
},
"first": {
  "href": "https://professionalservices-
    int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=
    10",
  "type": "Cities",
  "title": "first page of cities"
}
}
}
```

### 1.1.3 UC02 - Get belgian cities based on the name of the city

Created on 15-02-16. Last modified on 29-05-20.

#### Functional description

Cette fonction permet d'obtenir les communes belges, principales et/ou non-fusionnées, à partir d'un nom, partiel (= qui commence par) ou complet, de la commune.

Pour chacun des enregistrements, on peut avoir les informations telles décrites dans UC01.

#### Sélection sur les rows:

- **"cityName"**: recherche sur le nom de la commune belge (la recherche peut se faire à partir de la 1ère lettre de la commune) en parcourant les 3 rôles linguistiques de cityName  
==> retourne la liste des communes, principales et non-fusionnées, correspondantes à la recherche
- **"mainCityName"**: recherche sur le nom de la commune belge principale (la recherche peut se faire à partir de la 1ère lettre de la commune) en parcourant les 3 rôles linguistiques de mainCityName  
==> retourne la liste des communes, principales et non-fusionnées, dont la commune principale correspond à la recherche
- **"cityName" & "isMainCity"**: vrai ou faux  
==> retourne la liste des communes, principales (true) ou non-fusionnées (false), correspondantes à la recherche
- **"mainCityName" & "isMainCity"**: vrai ou faux  
==> retourne la liste des communes, principales ou non-fusionnées, correspondantes à la recherche
- **"validityStartDate.lte" or "validityStartDate.gte"** = une date (YYYYMMDD)
- **"validityEndDate.lte" or "validityEndDate.gte"** = une date (YYYYMMDD)  
Exemples:  
validityStartDate.lte= 20150101&validityEndDate.gte=20150131:  
==> liste des communes actives tout le mois de janvier 2015  
validityStartDate.lte= 20150131&validityEndDate.gte=20150101:  
==> liste des communes actives au moins 1 jour en janvier 2015  
validityEndDate.gte=date du jour:  
==> liste des communes actives à la date du jour  
Remarque: validityStartDate.lte et validityEndDate.gte reprennent également les rows dont la date de début ou la date de fin est à null

#### Erreurs:

- si pas de record trouvé: liste vide avec totalItems=0

- si un des paramètres donné ne contient pas de valeur ==> 400 Bad Request: Query Parameter [xxx] cannot be empty

### Affiner la recherche

- **"searchOption=unaccent"**,  
==> affiner la recherche en ne tenant pas compte des accents (ni dans la requête, ni dans la DB)
- **"searchOption=partial"**  
==> affiner la recherche sur une commune **contenant** un string d'au moins 1 lettre

Par défaut, la recherche n'est pas "case sensitive", tient compte des caractères accentués et la recherche se base sur les noms qui commencent par ....

S'il y a 2 searchOptions, la recherche se fait sur "or".

### Filtres possibles sur les colonnes:

- **"fields"**: On peut paramétrer les champs de retour dans la réponse avec "fields"  
==> accepte une sélection de champs dans la réponse, séparés par une virgule  
Exemple: &fields=cityNisCode,zipCode,cityName,
- **"language" (à combiner avec au moins l'un des paramètres ci-dessus): "fr", "nl", "de"** (2 lettres minuscules, 1 seul choix possible):  
ce paramètre permet 2 choses:  
1) la recherche se fait que dans les noms de communes de la langue spécifiée  
2) retourne que le nom de la commune dans la langue spécifiée  
==> retourne les communes, fusionnées et/ou non-fusionnées, dans le code langue demandé, trouvées dans les noms de communes de la langue spécifiée, et trié alphabétiquement sur le nom de la commune non-fusionnée (cityName) de la langue demandée,  
si "language" est différent des valeurs attendues ==> 400 Bad Request: "must match \"fr|nl|de\""

### Affichage:

#### paginer les résultats

- **"page"**  
==> permet d'afficher la page demandée
- **"pageSize"**  
==> permet d'afficher le nombre de résultats à afficher par page (par défaut, pageSize=10)  
**si pageSize=0 alors il affiche tous les résultats**

#### détailler les résultats

- **mainCityDetail="embed" or "link" (par défaut)**  
==> "embed": permet d'afficher le détail des données de la commune principale (au lieu du lien) à savoir: cityNisCode, zipCode (=mainZipCode), officialLanguage, cityName (=mainCityName) FR, NL, DE  
==> "link": affiche le lien vers la commune principale correspondante à la commune non-fusionnée (=valeur par défaut)

### Résultat:

- le nombre de résultats correspondant à la requête, est affiché, indépendamment de la pagination, dans "totalItems"

**Exemple:**

```
{{url}}/{{geography-context-root}}/belgianCities?cityName=gand
```

```
{
  "items": [
    {
      "cityNisCode": 44021,
      "zipCode": 9000,
      "officialLanguage": "N",
      "cityName": {
        "fr": "Gand",
        "nl": "Gent",
        "de": "Gent"
      },
      "regionCode": "VLG",
      "linguisticRegionCode": "NL",
      "_links": {
        "self": {
          "href": "https://professionalservices-
            int.socialsecurity.be/REST/referenceData/geography/v1/belgianCities/2696613120",
          "type": "city"
        }
      }
    }
  ],
  "totalItems": 1,
  "totalPages": 1,
  "currentPage": 1,
}
```

```
"_links": {
  "next": {
    "href": "https://professionalservices-
            int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=
            10&cityName=gand",
    "type": "Cities",
    "title": "next page of cities"
  },
  "last": {
    "href": "https://professionalservices-
            int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=
            10&cityName=gand",
    "type": "Cities",
    "title": "last page of cities"
  },
  "prev": {
    "href": "https://professionalservices-
            int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=
            10&cityName=gand",
    "type": "Cities",
    "title": "prev page of cities"
  },
  "self": {
    "href": "https://professionalservices-
            int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=
            10&cityName=gand",
    "type": "Cities",
    "title": "self page of cities"
  },
  "first": {
    "href": "https://professionalservices-
            int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=
            10&cityName=gand",
    "type": "Cities",
```



"title": "first page of cities"

}

}

}

## 1.1.4 UC03 - Get belgian cities based on postal code

Created on 15-02-16. Last modified on 29-05-20.

### Functional description

Cette fonction permet d'obtenir les communes belges, principales et/ou non-fusionnées, à partir du code postal officiel (partiel ou entier) de la commune.

Pour chacun des enregistrements, on peut avoir les informations telles décrites dans UC01.

### Sélection sur les rows:

- **"zipCode"**: recherche sur le code postal belge (la recherche peut se faire sur 2, 3 ou 4 chiffres) en parcourant zipCode  
==> retourne la liste des communes principales et non-fusionnées, correspondantes à la recherche
- **"mainZipCode"**: recherche sur le code postal de la commune principale (la recherche peut se faire à partir de 2, 3 ou 4 chiffres) en parcourant mainZipCode  
==> retourne la liste des communes, principales et non-fusionnées, correspondant au code postal de la commune principale
- **"zipCode" & "isMainCity"**: vrai ou faux  
==> retourne la liste des communes, principales **ou** non-fusionnées, correspondantes à la recherche  
(659x: 1 principale ou 6 non-fusionnées)
- **"mainZipCode" & "isMainCity"**: vrai ou faux  
==> retourne la liste des communes, principales **ou** non-fusionnées, correspondantes à la recherche  
(659x: 7 principale et non-fusionnées ou 6 non-fusionnées)
- **"validityStartDate.lte"** or **"validityStartDate.gte"** = une date (YYYYMMDD)
- **"validityEndDate.lte"** or **"validityEndDate.gte"** = une date (YYYYMMDD)  
Exemples:  
validityStartDate.lte= 20150101&validityEndDate.gte=20150131:  
==> liste des communes actives en janvier 2015  
validityStartDate.lte= 20153101&validityEndDate.gte=20150101:  
==> liste des communes actives au moins 1 jour en janvier 2015  
validityEndDate.gte=date du jour:  
==> liste des communes actives à la date du jour  
Remarque: validityStartDate.lte et validityEndDate.gte reprennent également les rows dont la date de début ou la date de fin est à null

### Erreurs:

- si pas de record trouvé: liste vide avec totalItems=0

- si zipCode est différent de 2, 3 ou 4 chiffres ==> 400 Bad Request: must be greater than or equal to 10
- si un des paramètres donné ne contient pas de valeur ==> 400 Bad Request: Query Parameter [xxx] cannot be empty
- Si un paramètre inconnu est ajouté ==> 400 Bad Request : Unexpected Query Parameter [xxx]

#### Filtres possibles sur les colonnes:

- **"fields"**: On peut paramétrer les champs de retour dans la réponse avec "fields"  
==> accepte une sélection de champs dans la réponse, séparés par une virgule  
Exemple: &fields=zipCode,,cityName,
- **"language"**: "fr", "nl", "de" (2 lettres minuscules, 1 seul choix possible)  
==> retourne les communes, principales et/ou non-fusionnées, dans le code langue demandé,  
**trié sur code postal, puis commune**  
si "language" est différent des valeurs attendues ==> 400 Bad Request: "must match  
\"fr|nl|de\""

#### Affichage:

##### paginer les résultats

- **"page"**  
==> permet d'afficher la page demandée
- **"pageSize"**  
==> permet d'afficher le nombre de résultats à afficher par page (par défaut, pageSize=10)  
**si pageSize=0 alors il affiche tous les résultats**

##### détailler les résultats

- **mainCityDetail="embed" or "link" (par défaut)**  
==> "embed": permet d'afficher le détail des données de la commune principale (au lieu du lien) à savoir: cityNisCode, zipCode (=mainZipCode), officialLanguage, cityName (=mainCityName) FR, NL, DE  
==> "link": affiche le lien vers la commune principale correspondante à la commune non-fusionnée (=valeur par défaut)

#### **Résultat:**

- le nombre de résultats correspondant à la requête, est affiché, indépendamment de la pagination, dans "totalItems"

#### **Exemple:**

```
{{url}}/{{geography-context-root}}/countries/150/cities?zipCode=4000
```

#### **renvoie la commune principale et 2 communes non-fusionnées**

```
{  
  
  "items": [  

```

```
{
  "zipCode": 4000,
  "officialLanguage": "F",
  "cityName": {
    "fr": "Glain",
    "nl": "Glain",
    "de": "Glain"
  },
  "mainCity": {
    "href": "https://professionalservices-
      int.socialsecurity.be/REST/referenceData/geography/v1/belgianCities/2149189120",
    "type": "city",
    "title": "nisCode=62063|Liège|Luik|Liège"
  },
  "regionCode": "WAL",
  "linguisticRegionCode": "FR",
  "_links": {
    "self": {
      "href": "https://professionalservices-
        int.socialsecurity.be/REST/referenceData/geography/v1/belgianCities/2182743552",
      "type": "city"
    }
  }
},
{
  "cityNisCode": 62063,
  "zipCode": 4000,
  "officialLanguage": "F",
  "cityName": {
    "fr": "Liège",
```

```
"nl": "Luik",
"de": "Liège"
},
"regionCode": "WAL",
"linguisticRegionCode": "FR",
"_links": {
  "self": {
    "href": "https://professionalservices-
int.socialsecurity.be/REST/referenceData/geography/v1/belgianCities/2149189120",
    "type": "city"
  }
}
},
{
  "zipCode": 4000,
  "officialLanguage": "F",
  "cityName": {
    "fr": "Rocourt",
    "nl": "Rocourt",
    "de": "Rocourt"
  },
  "mainCity": {
    "href": "https://professionalservices-
int.socialsecurity.be/REST/referenceData/geography/v1/belgianCities/2149189120",
    "type": "city",
    "title": "nisCode=62063|Liège|Luik|Liège"
  },
  "regionCode": "WAL",
  "linguisticRegionCode": "FR",
  "_links": {
```

```
"self": {
  "href": "https://professionalservices-
    int.socialsecurity.be/REST/referenceData/geography/v1/belgianCities/2165966336",
  "type": "city"
}
},
]
"totalItems": 3,
"totalPages": 1,
"currentPage": 1,
"_links": {
  "next": {
    "href": "https://professionalservices-
      int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=
      10&zipCode=4000",
    "type": "Cities",
    "title": "next page of cities"
  },
  "last": {
    "href": "https://professionalservices-
      int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=
      10&zipCode=4000",
    "type": "Cities",
    "title": "last page of cities"
  },
  "prev": {
    "href": "https://professionalservices-
      int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=
      10&zipCode=4000",
    "type": "Cities",
    "title": "prev page of cities"
  }
}
```

```
},  
"self": {  
  "href": "https://professionalservices-  
int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=  
10&zipCode=4000",  
  "type": "Cities",  
  "title": "self page of cities"  
},  
"first": {  
  "href": "https://professionalservices-  
int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=  
10&zipCode=4000",  
  "type": "Cities",  
  "title": "first page of cities"  
}  
}  
}  
}
```

## 1.1.5 UC04 - Get belgian cities based on postal code and the name of the city

Created on 04-01-18. Last modified on 29-05-20.

### Functional description

Cette fonction permet d'obtenir les communes belges, principales et/ou non-fusionnées, à partir du code postal officiel (partiel ou entier) et de la commune (partielle ou entière). Avec le paramètre **"isMainCity"**, on peut réduire la liste aux communes principales.

Pour chacun des enregistrements, on peut avoir les informations telles décrites dans UC01.

### Sélection sur les rows:

- **"zipCode" & "cityName"**: recherche sur le code postal belge (la recherche peut se faire sur 2, 3 ou 4 chiffres) en parcourant zipCode ET sur le nom de la commune belge (la recherche peut se faire à partir de la 1ère lettre de la commune) en parcourant les 3 rôles linguistiques de cityName  
==> retourne la liste des communes principales et non-fusionnées, correspondantes à la recherche
- **"zipCode" & "cityName" & "isMainCity"**: vrai ou faux  
==> retourne la liste des communes, principales **ou** non-fusionnées, correspondantes à la recherche
- **"validityStartDate.lte"** or **"validityStartDate.gte"** = une date (YYYYMMDD)
- **"validityEndDate.lte"** or **"validityEndDate.gte"** = une date (YYYYMMDD)  
Exemples:  
validityStartDate.lte= 20150101&validityEndDate.gte=20150131:  
==> liste des communes actives tout le mois de janvier 2015  
validityStartDate.lte= 20150131&validityEndDate.gte=20150101:  
==> liste des communes actives au moins 1 jour en janvier 2015  
validityEndDate.gte=date du jour:  
==> liste des communes actives à la date du jour  
Remarque: validityStartDate.lte et validityEndDate.gte reprennent également les rows dont la date de début ou la date de fin est à null

### Erreurs:

- si pas de record trouvé: liste vide avec totalItems=0
- si zipCode est différent de 2, 3 ou 4 chiffres ==> 400 Bad Request: must be greater than or equal to 10
- si un des paramètres donné ne contient pas de valeur ==> 400 Bad Request: Query Parameter [xxx] cannot be empty



- si la sélection est sur plus de 1 champ autorisé ==> 400 Bad Request: Only cityName and zipCode can be used at the same time

### Affiner la recherche

- **"searchOption=unaccent"**,  
==> affiner la recherche en ne tenant pas compte des accents (ni dans la requête, ni dans la DB)
- **"searchOption=partial"**  
==> affiner la recherche sur une commune **contenant** un string d'au moins 1 lettre

Par défaut, la recherche n'est pas "case sensitive", tient compte des caractères accentués et la recherche se base sur les noms qui commencent par ....

S'il y a 2 searchOptions, la recherche se fait sur "or".

### Filtres possibles sur les colonnes:

- **"fields"**: On peut paramétrer les champs de retour dans la réponse avec "fields"  
==> accepte une sélection de champs dans la réponse, séparés par une virgule  
Exemple: &fields=zipCode,,cityName,
- **"language"**: "fr", "nl", "de" (2 lettres minuscules, 1 seul choix possible)  
==> retourne les communes, principales et/ou non-fusionnées, dans le code langue demandé, **trié sur code postal, puis commune**  
si "language" est différent des valeurs attendues ==> 400 Bad Request: "must match \\"fr|nl|de\\""

### Affichage:

#### paginer les résultats

- **"page"**  
==> permet d'afficher la page demandée
- **"pageSize"**  
==> permet d'afficher le nombre de résultats à afficher par page (par défaut, pageSize=10)  
**si pageSize=0 alors il affiche tous les résultats**

#### détailler les résultats

- **mainCityDetail="embed" or "link" (par défaut)**  
==> "embed": permet d'afficher le détail des données de la commune principale (au lieu du lien) à savoir: cityNisCode, zipCode (=mainZipCode), officialLanguage, cityName (=mainCityName) FR, NL, DE  
==> "link": affiche le lien vers la commune principale correspondante à la commune non-fusionnée (=valeur par défaut)

#### **Résultat:**

- le nombre de résultats correspondant à la requête, est affiché, indépendamment de la pagination, dans "totalItems"

#### **Exemple:**

```
{{url}}/{{geography-context-root}}/countries/150/cities?zipCode=400 &cityName=G
```

**renvoie la commune non-fusionnée trouvée**

```
{  
  "items": [  
    {  
      "zipCode": 4000,  
      "officialLanguage": "F",  
      "cityName": {  
        "fr": "Glain",  
        "nl": "Glain",  
        "de": "Glain"  
      },  
      "mainCity": {  
        "href": "https://professionalservices-  
          int.socialsecurity.be/REST/referenceData/geography/v1/belgianCities/2149189120",  
        "type": "city",  
        "title": "nisCode=62063|Liège|Luik|Liège"  
      },  
      "regionCode": "WAL",  
      "linguisticRegionCode": "FR",  
      "_links": {  
        "self": {  
          "href": "https://professionalservices-  
            int.socialsecurity.be/REST/referenceData/geography/v1/belgianCities/2182743552",  
          "type": "city"  
        }  
      }  
    }  
  ],  
  "totalItems": 1,  
}
```

```
"totalPages": 1,
"currentPage": 1,
"_links": {
  "next": {
    "href": "https://professionalservices-
            int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=
            10&zipCode=400&cityName=g",
    "type": "Cities",
    "title": "next page of cities"
  },
  "last": {
    "href": "https://professionalservices-
            int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=
            10&zipCode=400&cityName=g",
    "type": "Cities",
    "title": "last page of cities"
  },
  "prev": {
    "href": "https://professionalservices-
            int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=
            10&zipCode=400&cityName=g",
    "type": "Cities",
    "title": "prev page of cities"
  },
  "self": {
    "href": "https://professionalservices-
            int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=
            10&zipCode=400&cityName=g",
    "type": "Cities",
    "title": "self page of cities"
  },
  "first": {
```

"href": "https://professionalservices-int.socialsecurity.be/REST/referenceData/geography/v1/countries/150/cities/?page=1&pageSize=10&zipCode=400&cityName=g",

"type": "Cities",

"title": "first page of cities"

}

}

}

## 1.1.6 UC05 - Get belgian cities based on region code

Created on 13-03-20. Last modified on 29-05-20.

### Functional description

Cette fonction permet d'obtenir une liste de toutes les communes belges, principales et/ou non-fusionnées basée sur le code région.

Pour chacun des enregistrements, on peut avoir les informations telles décrites dans UC01.

### Sélection sur les rows:

- **"regionCode"**: recherche sur le code région donné Valeurs acceptées : "bru", "vlg", "wal"  
==> retourne la liste des communes principales et non-fusionnées correspondant à la région demandée  
si ce paramètre n'est pas donné, le résultat comprendra toutes les communes
- **"regionCode"&"isMainCity"**: vrai ou faux  
==> retourne la liste des communes, principales **ou** non-fusionnées, correspondantes à la recherche

### Erreurs:

- si pas de record trouvé: liste vide avec totalItems=0
- si un des paramètres donné ne contient pas de valeur ==> 400 Bad Request: Query Parameter [xxx] cannot be empty
- si « regionCode » est différent des valeurs attendues  
==> 400 Bad Request : must match \"bru|vlg|wal\"
- Si un paramètre inconnu est ajouté ==> 400 Bad Request : Unexpected Query Parameter [xxx]

### Filtres possibles sur les colonnes:

- **"fields"**: On peut paramétrer les champs de retour dans la réponse avec "fields"  
==> accepte une sélection de champs dans la réponse, séparés par une virgule  
Exemple: &fields=zipCode, cityName
- **"language"**: "fr", "nl", "de" (2 lettres minuscules, 1 seul choix possible)  
==> retourne les communes, principales et/ou non-fusionnées, dans le code langue demandé,  
**trié alphabétiquement sur le nom de la commune non-fusionnée (cityName)**  
si "language" est différent des valeurs attendues  
==> 400 Bad Request: must match \"fr|nl|de\"

### Affichage:

#### paginer les résultats

- **"page"**  
==> permet d'afficher la page demandée
- **"pageSize"**  
==> permet d'afficher le nombre de résultats à afficher par page (par défaut, pageSize=10)  
**si pageSize=0 alors il affiche tous les résultats**

**détailler les résultats**

- **mainCityDetail="embed" or "link" (par défaut)**  
==> "embed": permet d'afficher le détail des données de la commune principale (au lieu du lien) à savoir: cityNisCode, zipCode (=mainZipCode), officialLanguage, cityName (=mainCityName) FR, NL, DE  
==> "link": affiche le lien vers la commune principale correspondante à la commune non-fusionnée (=valeur par défaut)

**Résultat:**

- le nombre de résultats correspondant à la requête, est affiché, indépendamment de la pagination, dans "totalItems"

## 1.1.7 UC06 - Get belgian cities based on linguistic region code

Created on 13-03-20. Last modified on 22-05-20.

### Functional description

Cette fonction permet d'obtenir une liste de toutes les communes belges, principales et non-fusionnées, basée sur le code région linguistique.

Pour chacun des enregistrements, on peut avoir les informations telles décrites dans UC01.

### Sélection sur les rows:

- **"linguisticRegionCode"**: recherche sur le code région linguistique donné. Valeurs acceptées : "bil", "nl", "fr", "de"  
==> retourne la liste des communes principales et non-fusionnées correspondant au code région linguistique demandé  
si ce paramètre n'est pas donné, le résultat comprendra toutes les communes
- **"linguisticRegionCode" & "isMainCity"**: vrai ou faux  
==> retourne la liste des communes, principales **ou** non-fusionnées, correspondantes à la recherche

### Erreurs:

- si pas de record trouvé: liste vide avec totalItems=0
- si un des paramètres donné ne contient pas de valeur ==> 400 Bad Request: Query Parameter [xxx] cannot be empty
- si "linguisticRegionCode" est différent des valeurs attendues  
==> 400 Bad Request : must match \"bil|nl|fr|de\"
- Si un paramètre inconnu est ajouté ==> 400 Bad Request : Unexpected Query Parameter [xxx]

### Filtres possibles sur les colonnes:

- **"fields"**: On peut paramétrer les champs de retour dans la réponse avec "fields"  
==> accepte une sélection de champs dans la réponse, séparés par une virgule  
Exemple: &fields=zipCode, cityName
- **"language"**: "fr", "nl", "de" (2 lettres minuscules, 1 seul choix possible)  
==> retourne les communes, principales et/ou non-fusionnées, dans le code langue demandé, **trié alphabétiquement sur le nom de la commune non-fusionnée (cityName)**  
si "language" est différent des valeurs attendues  
==> 400 Bad Request: must match \"fr|nl|de\"

### Affichage:

**paginer les résultats**

- **"page"**  
==> permet d'afficher la page demandée
- **"pageSize"**  
==> permet d'afficher le nombre de résultats à afficher par page (par défaut, pageSize=10)  
**si pageSize=0 alors il affiche tous les résultats**

**détailler les résultats**

- **mainCityDetail="embed" or "link" (par défaut)**  
==> "embed": permet d'afficher le détail des données de la commune principale (au lieu du lien) à savoir: cityNisCode, zipCode (=mainZipCode), officialLanguage, cityName (=mainCityName) FR, NL, DE  
==> "link": affiche le lien vers la commune principale correspondante à la commune non-fusionnée (=valeur par défaut)

**Résultat:**

- le nombre de résultats correspondant à la requête, est affiché, indépendamment de la pagination, dans "totalItems"