

---

# crabpy\_pyramid Documentation

*Release 0.10.0*

.

Dec 15, 2021



# CONTENTS

<b>1</b>	<b>Setup and configuration</b>	<b>1</b>
1.1	Settings . . . . .	1
<b>2</b>	<b>Services</b>	<b>3</b>
2.1	Capakey . . . . .	3
2.2	Crab . . . . .	14
<b>3</b>	<b>API Documentation</b>	<b>35</b>
3.1	Utils module . . . . .	35
<b>4</b>	<b>History</b>	<b>37</b>
4.1	0.12.0 (15-12-2021) . . . . .	37
4.2	0.11.0 (15-09-2021) . . . . .	37
4.3	0.10.0 (13-08-2021) . . . . .	37
4.4	0.9.0 (08-04-2021) . . . . .	37
4.5	0.8.1 (06-06-2019) . . . . .	38
4.6	0.8.0 (11-01-2019) . . . . .	38
4.7	0.7.0 (2019-01-03) . . . . .	38
4.8	0.6.3 (2017-12-07) . . . . .	38
4.9	0.6.2 (2017-08-29) . . . . .	38
4.10	0.6.1 (2017-04-20) . . . . .	38
4.11	0.6.0 (2017-04-19) . . . . .	38
4.12	0.5.1 (2016-11-22) . . . . .	39
4.13	0.5.0 (2016-07-27) . . . . .	39
4.14	0.4.1 (2016-02-02) . . . . .	39
4.15	0.4.0 (2016-01-25) . . . . .	39
4.16	0.3.0 (2015-06-01) . . . . .	39
4.17	0.2.0 (2015-03-03) . . . . .	40
4.18	0.1.1 (2014-09-18) . . . . .	40
4.19	0.1.0 (2014-09-05) . . . . .	40
4.20	0.1.0a2 (2014-04-30) . . . . .	40
4.21	0.1.0a1 (2014-03-19) . . . . .	40
<b>5</b>	<b>Indices and tables</b>	<b>41</b>
	<b>Python Module Index</b>	<b>43</b>
	<b>HTTP Routing Table</b>	<b>45</b>
	<b>Index</b>	<b>47</b>



## SETUP AND CONFIGURATION

Installing *crabpy\_pyramid* is fairly straightforward, configuring it can be a bit more complicated. Fundamentally, *crabpy\_pyramid* uses the *crabpy* library to offer users access to both the CRAB and the CAPAKEY webservice. It's possible to include none, one or both of the services.

To add *crabpy\_pyramid* to you project, just include it.

```
config.include('crabpy_pyramid')
```

Out of the box, this will add the `crabpy.gateway.crab.CrabGateway` and the accompanying REST services. It will not add the `crabpy.gateway.capakey.CapakeyRestGateway` and it's accompanying services. If you want to use this service, you need to set the *crabpy.capakey.include* setting to *True*.

The default will also set a *crabpy.cache.file.root* parameter. This is the default location for writing `dogpile.cache` file caches. It exists to set a default that should work, but won't really help you very much. Please change it.

### 1.1 Settings

The following settings can be configured:

#### 1.1.1 crabpy.cache.file.root

Location where `dogpile.cache` can create file caches. By default it's set to `/tmp/dogpile_data`. Should be changed for any setup that actually wants to use caching.

#### 1.1.2 crabpy.proxy.http

Will be passed on to the Gateways, in case the services need to be called through a proxy.

#### 1.1.3 crabpy.proxy.https

Will be passed on to the Gateways, in case the services need to be called through a proxy.

#### 1.1.4 crabpy.crab.include

This setting controls whether the CRAB Gateway is configured or not. By default this is set to *True*.

#### 1.1.5 crabpy.crab.cache\_config

This controls the caching settings for the CRAB Gateway. It's actually a prefix for a number of settings, which are all passed on to the CRAB Gateway.

#### 1.1.6 crabpy.capakey.include

This setting controls whether the CAPAKEY Gateway is configured or not. By default this is set to *False*. If you set this to *True*.

#### 1.1.7 crabpy.capakey.cache\_config

This controls the caching settings for the CAPAKEY Gateway. It's actually a prefix for a number of settings, which are all passed on to the CAPAKEY Gateway.

## SERVICES

Crabpy\_pyramid exposes the following services if both *crabpy.capakey.include* and *crabpy.crab.include* are set to *True*.

## 2.1 Capakey

**GET** /capakey/gemeenten

List all gemeenten

**Example request:**

```
GET /capakey/gemeenten HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
Range: items=0-4
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Range: 0-4/306

[
  {
    'id': 44001,
    'naam': 'Aalter'
  }, {
    'id': 44011,
    'naam': 'Deinze'
  }, {
    'id': 44012,
    'naam': 'De Pinte'
  }, {
    'id': 44013,
    'naam': 'Destelbergen'
  }, {
    'id': 4401,
    'naam': 'Gent',
  }
]
```

### Request Headers

- **Range** – Can be used to ask for a certain set of results, eg. `items=0-24` asks for the first 25 items.

### Response Headers

- **Content-Range** – Tells the client what range of results is being returned, eg. `items=0-24/306` for the first 25 items out of 306.

### Status Codes

- **200 OK** – Gemeenten were found.

**GET** `/capakey/gemeenten/` (`int: gemeente_id`)  
Get a gemeente by id

### Example request:

```
GET /capakey/gemeenten/44021 HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
```

### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  'id': 44021,
  'naam': 'Gent',
  'centroid': [104154.2225, 197300.703],
  'bbox': [94653.453, 185680.984, 113654.992, 208920.422]
}
```

### Status Codes

- **200 OK** – Gemeente was found.
- **404 Not Found** – Gemeente was not found.

**GET** `/capakey/gemeenten/` (`int: gemeente_id`) `/afdelingen`  
List\_kadastrale\_afdelingen\_by\_gemeente

### Example request:

```
GET /capakey/gemeenten/44021/afdelingen HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
Range: 0-1
```

### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Range: 0-1/30

[
  {
    'id': 44002,
    'naam': 'Afsnee',
```

(continues on next page)



(continued from previous page)

```

    'gemeente': {
      'id': 44021,
      'naam': 'Gent'
    }
  }, {
    'id': 44017,
    'naam': 'Drongen',
    'gemeente': {
      'id': 44021,
      'naam': 'Gent'
    }
  }
]

```

**Request Headers**

- **Range** – Can be used to ask for a certain set of results, eg. `items=0-24` asks for the first 25 items.

**Response Headers**

- **Content-Range** – Tells the client what range of results is being returned, eg. `items=0-9/30` for the first 10 items out of 30.

**Status Codes**

- **200 OK** – Gemeente was found.
- **404 Not Found** – Gemeente was not found.

**GET /capakey/afdelingen**

List\_kadastrale\_afdelingen

**Example request:**

```

GET /capakey/afdelingen HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
Range: 0-1

```

**Example response:**

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Range: 0-1/1433

[
  {
    "id": 44002,
    "naam": "Afsnee",
    "gemeente": {
      "id": 44021,
      "naam": "Gent"
    }
  }, {
    "id": 44017,
    "naam": "Drongen",
    "gemeente": {

```

(continues on next page)

(continued from previous page)

```
        "id": 44021,  
        "naam": "Gent"  
    }  
}  
]
```

### Request Headers

- **Range** – Can be used to ask for a certain set of results, eg. `items=0-24` asks for the first 25 items.

### Response Headers

- **Content-Range** – Tells the client what range of results is being returned, eg. `items=0-99/1433` for the first 100 items out of 1433.

### Status Codes

- **200 OK** – Afdelingen were found.

**GET** `/capakey/afdelingen/` (**int:** *afdeling\_id*)  
Get\_kadastrale\_afdeling\_by\_id

### Example request:

```
GET /capakey/afdelingen/44017 HTTP/1.1  
Host: example.onroerenderfgoed.be  
Accept: application/json
```

### Example response:

```
HTTP/1.1 200 OK  
Content-Type: application/json  
  
{  
  "id": 44017,  
  "naam": "Drongen",  
  "gemeente": {  
    "id": 44021,  
    "naam": "Gent"  
  },  
  "centroid": [104154.2225, 197300.703],  
  "bbox": [94653.453, 185680.984, 113654.992, 208920.422]  
}
```

### Status Codes

- **200 OK** – Afdeling was found.
- **404 Not Found** – Afdeling was not found.

**GET** `/capakey/afdelingen/` (**int:** *afdeling\_id*) **/secties**  
List\_secties\_by\_afdeling

### Example request:

```
GET /capakey/afdelingen/44017/secties HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "afdeling": {
      "naam": "GENT 27 AFD/DRONGEN 1 AFD/",
      "id": 44017,
      "gemeente": {
        "naam": "Gent",
        "id": 44021
      }
    },
    "id": "A"
  },
  {
    "afdeling": {
      "naam": "GENT 27 AFD/DRONGEN 1 AFD/",
      "id": 44017,
      "gemeente": {
        "naam": "Gent",
        "id": 44021
      }
    },
    "id": "B"
  },
  {
    "afdeling": {
      "naam": "GENT 27 AFD/DRONGEN 1 AFD/",
      "id": 44017,
      "gemeente": {
        "naam": "Gent",
        "id": 44021
      }
    },
    "id": "C"
  },
  {
    "afdeling": {
      "naam": "GENT 27 AFD/DRONGEN 1 AFD/",
      "id": 44017,
      "gemeente": {
        "naam": "Gent",
        "id": 44021
      }
    },
    "id": "D"
  }
]
```

**Status Codes**

- 200 OK – Afdeling was found.
- 404 Not Found – Afdeling was not found.

**GET** /capakey/afdelingen/ (int: *afdeling\_id*) /secties/  
string: *sectie\_id* Get\_sectie\_by\_id\_and\_afdeling

**Example request:**

```
GET /capakey/afdelingen/44017/secties/A HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "afdeling": {
    "naam": "GENT 27 AFD/DRONGEN 1 AFD/",
    "bounding_box": [
      94653.7508750036,
      190442.133125,
      101151.588,
      197371.0951875
    ],
    "centroid": [
      97902.6694375016,
      193906.61415625
    ],
    "id": 44017,
    "gemeente": {
      "naam": "Gent",
      "bounding_box": [
        94653.4530000016,
        185680.984000001,
        113654.991999999,
        208920.421999998
      ],
      "centroid": [
        104154.2225,
        197300.703
      ],
      "id": 44021
    }
  },
  "bounding_box": [
    96205.7660000026,
    194208.691374999,
    101032.139624998,
    197371.0951875
  ],
  "centroid": [
    98618.9528125003,
    195789.893281249
  ],
  "id": "A"
}
```

### Status Codes

- 200 OK – Sectie was found.
- 404 Not Found – Sectie was not found.

**GET** /capakey/afdelingen/(int: *afdeling\_id*)/secties/  
string: *sectie\_id*/percelen List\_percelen\_by\_sectie

### Example request:

```
GET /capakey/afdelingen/44017/secties/A/percelen HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
Range: 0-4
```

### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "capakey": "44017A0003/00C000",
    "id": "0003/00C000",
    "percid": "44017_A_0003_C_000_00",
    "sectie": {
      "afdeling": {
        "naam": "GENT 27 AFD/DRONGEN 1 AFD/",
        "id": 44017,
        "gemeente": {
          "naam": "Gent",
          "id": 44021
        }
      }
    },
    "id": "A"
  },
  {
    "capakey": "44017A0004/00D000",
    "id": "0004/00D000",
    "percid": "44017_A_0004_D_000_00",
    "sectie": {
      "afdeling": {
        "naam": "GENT 27 AFD/DRONGEN 1 AFD/",
        "id": 44017,
        "gemeente": {
          "naam": "Gent",
          "id": 44021
        }
      }
    },
    "id": "A"
  },
  {
    "capakey": "44017A0004/00F000",
    "id": "0004/00F000",
    "percid": "44017_A_0004_F_000_00",
    "sectie": {
```

(continues on next page)

(continued from previous page)

```
    "afdeling": {
      "naam": "GENT 27 AFD/DRONGEN  1 AFD/",
      "id": 44017,
      "gemeente": {
        "naam": "Gent",
        "id": 44021
      }
    },
    "id": "A"
  }
},
{
  "capakey": "44017A0004/00G000",
  "id": "0004/00G000",
  "percid": "44017_A_0004_G_000_00",
  "sectie": {
    "afdeling": {
      "naam": "GENT 27 AFD/DRONGEN  1 AFD/",
      "id": 44017,
      "gemeente": {
        "naam": "Gent",
        "id": 44021
      }
    }
  },
  "id": "A"
}
},
{
  "capakey": "44017A0006/00A000",
  "id": "0006/00A000",
  "percid": "44017_A_0006_A_000_00",
  "sectie": {
    "afdeling": {
      "naam": "GENT 27 AFD/DRONGEN  1 AFD/",
      "id": 44017,
      "gemeente": {
        "naam": "Gent",
        "id": 44021
      }
    }
  },
  "id": "A"
}
}
]
```

### Request Headers

- **Range** – Can be used to ask for a certain set of results, eg. `items=0-5` asks for the first 6 items.

### Response Headers

- **Content-Range** – Tells the client what range of results is being returned, eg. `items=0-5/145` for the first 6 items out of 145.

### Status Codes

- **200 OK** – Sectie was found.

- 404 Not Found – Sectie was not found.

**GET** /capakey/afdelingen/(int: *afdeling\_id*)/secties/  
string: *sectie\_id*/percelen/int: *perceel\_id* Get\_perceel\_by\_sectie\_and\_id

**Example request:**

```
GET /capakey/afdelingen/44017/secties/A/percelen/452 HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "percid": "44017_A_0003_C_000_00",
  "sectie": {
    "afdeling": {
      "naam": "GENT 27 AFD/DRONGEN 1 AFD/",
      "id": 44017,
      "gemeente": {
        "naam": "Gent",
        "id": 44021
      }
    },
    "id": "A"
  },
  "capakey": "44017A0003/00C000",
  "bounding_box": [
    98798.16799999977,
    197135.57,
    98989.27300000019,
    197356.498
  ],
  "centroid": [
    98893.72049999998,
    197246.034
  ],
  "id": "0003/00C000"
}
```

### Status Codes

- 200 OK – Perceel was found.
- 404 Not Found – Perceel was not found.

**GET** /capakey/percelen/(string: *capakey1*)/  
string: *capakey2* Get Perceel\_by\_capakey

**Example request:**

```
GET /capakey/percelen/44021A3675/00A000 HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "percid": "44021_A_3675_A_000_00",
  "sectie": {
    "afdeling": {
      "naam": "GENT 1 AFD",
      "bounding_box": [
        104002.076624997,
        194168.341499999,
        105784.050875001,
        197876.1466875
      ],
      "centroid": [
        104893.063749999,
        196022.24409375
      ],
      "id": 44021,
      "gemeente": {
        "naam": "Gent",
        "bounding_box": [
          94653.4530000016,
          185680.984000001,
          113654.991999999,
          208920.421999998
        ],
        "centroid": [
          104154.2225,
          197300.703
        ],
        "id": 44021
      }
    },
    "bounding_box": [
      104002.076624997,
      194168.341499999,
      105784.050875001,
      197876.1466875
    ],
    "centroid": [
      104893.063749999,
      196022.24409375
    ],
    "id": "A"
  },
  "capakey": "44021A3675/00A000",
  "bounding_box": [
    105182.648999996,
    197425.741999999,
    105199.199000001,
    197454.061999999
  ],
  "centroid": [
    105190.923999999,
    197439.901999999
  ],
}
```

(continues on next page)



(continued from previous page)

```
"id": "3675/00A000"
}
```

### Status Codes

- 200 OK – Perceel was found.
- 404 Not Found – Perceel was not found.

**GET** /capakey/percelen/ (string: *percid*)

Get\_perceel\_by\_percid

### Example request:

```
GET /capakey/percelen/44017_A_0004_D_000_00 HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
```

### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "percid": "44017_A_0004_D_000_00",
  "sectie": {
    "afdeling": {
      "naam": "GENT 27 AFD/DRONGEN 1 AFD/",
      "bounding_box": [
        94653.7508750036,
        190442.133125,
        101151.588,
        197371.0951875
      ],
      "centroid": [
        97902.6694375016,
        193906.61415625
      ],
      "id": 44017,
      "gemeente": {
        "naam": "Gent",
        "bounding_box": [
          94653.4530000016,
          185680.984000001,
          113654.991999999,
          208920.421999998
        ],
        "centroid": [
          104154.2225,
          197300.703
        ],
        "id": 44021
      }
    },
    "bounding_box": [
      96205.7660000026,
```

(continues on next page)

(continued from previous page)

```
        194208.691374999,  
        101032.139624998,  
        197371.0951875  
    ],  
    "centroid": [  
        98618.9528125003,  
        195789.893281249  
    ],  
    "id": "A"  
},  
"capakey": "44017A0004/00D000",  
"bounding_box": [  
    98800.6869999999,  
    197101.388,  
    98857.4720000029,  
    197133.022  
],  
"centroid": [  
    98829.0795000009,  
    197117.205  
],  
"id": "0004/00D000"  
}
```

### Status Codes

- 200 OK – Perceel was found.
- 404 Not Found – Perceel was not found.

## 2.2 Crab

### GET /crab/gewesten

List\_gewesten

#### Example request:

```
GET /crab/gewesten HTTP/1.1  
Host: example.onroerenderfgoed.be  
Accept: application/json
```

#### Example response:

```
HTTP/1.1 200 OK  
Content-Type: application/json  
  
[  
  {  
    "naam": "Brussels Hoofdstedelijk Gewest",  
    "id": 1  
  }, {  
    "naam": "Vlaams Gewest",  
    "id": 2  
  }, {  
    "naam": "Waals Gewest",  
    "id": 3  
  }  
]
```

(continues on next page)

(continued from previous page)

```

        "id": 3
    }
]

```

### Status Codes

- 200 OK – Gewesten were found.

**GET** /crab/gewesten/ (int: gewest\_id)  
Get\_gewest\_by\_id

### Example request:

```

GET /crab/gewesten/2 HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json

```

### Example response:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "namen": {
    "fr": "Région flamande",
    "de": "Die Flämische Region",
    "nl": "Vlaams Gewest"
  },
  "bounding_box": [
    22279.17,
    153050.23,
    258873.3,
    244022.31
  ],
  "centroid": [
    138165.09,
    189297.53
  ],
  "id": 2
}

```

### Status Codes

- 200 OK – Gewest was found.
- 404 Not Found – Gewest was not found.

**GET** /crab/gewesten/ (int: gewest\_id) /provincies  
Retrieve alle provinces in a gewest.

### Example request:

```

GET /crab/gewesten/2/provincies HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json

```

### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Range: items 0-4/5

[
  {
    "naam": "Antwerpen",
    "gewest": {"naam": "Vlaams Gewest", "id": 2},
    "niscode": 10000
  }, {
    "naam": "Vlaams-Brabant",
    "gewest": {"naam": "Vlaams Gewest", "id": 2},
    "niscode": 20001
  }, {
    "naam": "West-Vlaanderen",
    "gewest": {"naam": "Vlaams Gewest", "id": 2},
    "niscode": 30000
  }, {
    "naam": "Oost-Vlaanderen",
    "gewest": {"naam": "Vlaams Gewest", "id": 2},
    "niscode": 40000
  }, {
    "naam": "Limburg",
    "gewest": {"naam": "Vlaams Gewest", "id": 2},
    "niscode": 70000
  }
]
```

#### Status Codes

- 200 OK – Gewest was found.
- 404 Not Found – Gewest does not exist.

**GET** /crab/provincies/(int: *provincie\_id*)

Get information about a certain provincie.

**Example request:**

```
GET /crab/provincies/30000 HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "naam": "West-Vlaanderen",
  "gewest": {"naam": "Vlaams Gewest", "id": 2},
  "niscode": 30000
}
```

#### Status Codes

- 200 OK – Provincie was found.

- 404 Not Found – Provincie was not found.

**GET /crab/gewesten/(int: gewest\_id)/gemeenten**  
List all gemeenten in a certain gewest.

**Example request:**

```
GET /crab/gewesten/2/gemeenten HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
Range: items=0-4
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "naam": "Aartselaar",
    "id": 1
  },
  {
    "naam": "Antwerpen",
    "id": 2
  },
  {
    "naam": "Boechout",
    "id": 3
  },
  {
    "naam": "Boom",
    "id": 4
  },
  {
    "naam": "Borsbeek",
    "id": 5
  }
]
```

### Query Parameters

- **sort** – One of id, naam or niscode (default).

### Status Codes

- 200 OK – Gemeenten were found.
- 404 Not Found – Gewest does not exist.

**GET /crab/provincies/(int: provincie\_id)/gemeenten**  
List all gemeenten in a certain provincie.

**Example request:**

```
GET /crab/provincies/30000/gemeenten HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "naam": "Beernem",
    "id": 182
  },
  {
    "naam": "Blankenberge",
    "id": 183
  },
  {
    "naam": "Brugge",
    "id": 184
  },
  {
    "naam": "Damme",
    "id": 185
  },
  {
    "naam": "Jabbeke",
    "id": 186
  }
]
```

**Status Codes**

- 200 OK – Gemeenten were found.
- 404 Not Found – Provincie does not exist.

**GET** /crab/gemeenten/(int:id of int:niscode)

Get\_gemeente\_by\_id

**Example request:**

```
GET /crab/gemeenten/1 HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
```

Get\_gemeente\_by\_niscode

**Example request:**

```
GET /crab/gemeenten/11001 HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "naam": "Aartselaar",
```

(continues on next page)

(continued from previous page)

```

"bounding_box": [
    148950.36,
    199938.28,
    152811.77,
    204575.39
],
"centroid": [
    150881.07,
    202256.84
],
"id": 1,
"metadata": {
    "begin_tijd": "2002-08-13 17:32:32",
    "begin_datum": "1830-01-01 00:00:00",
    "begin_organisatie": {
        "naam": "NGI",
        "definitie": "Nationaal Geografisch Instituut.",
        "id": "6"
    },
    "begin_bewerking": {
        "naam": "invoer",
        "definitie": "Invoer in de databank.",
        "id": "1"
    }
}
}

```

### Status Codes

- 200 OK – Gemeente was found.
- 404 Not Found – Gemeente was not found.

**GET /crab/gewesten/(int: *gewest\_id*)/deelgemeenten**

List all deelgemeenten in a certain gewest.

### Example request:

```

GET /crab/gewesten/2/deelgemeenten HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
Range: items=0-4

```

### Example response:

```

HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "naam": "Sint-Joris-Winge",
    "id": "24135C"
  }, {
    "naam": "Meensel-Kiezegem",
    "id": "24135B"
  }, {
    "naam": "Tielt",

```

(continues on next page)

(continued from previous page)

```
        "id": "24135A"
    }, {
        "naam": "Ertvelde",
        "id": "44019C"
    }, {
        "naam": "Kluizen",
        "id": "44019D"
    }
]
```

### Status Codes

- 200 OK – Deelgemeenten were found.
- 404 Not Found – Gewest does not exist.

**GET** /crab/gemeenten/(int:gemeente\_id or int:niscode)/deelgemeenten  
List all *deelgemeenten* in a certain *gemeente*.

### Example request:

```
GET /crab/gemeenten/90/deelgemeenten HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
```

### Example request:

```
GET /crab/gemeenten/11002/deelgemeenten HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
```

### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
[
  {"naam": "Asse", "id": "23002A"},
  {"naam": "Kobbegem", "id": "23002C"},
  {"naam": "Mollem", "id": "23002B"},
  {"naam": "Zellik", "id": "23002E"},
  {"naam": "Relegem", "id": "23002D"},
  {"naam": "Bekkerzeel", "id": "23002F"}
]
```

### Status Codes

- 200 OK – Deelgemeenten were found.
- 404 Not Found – The Gemeente for which you are requesting Deelgemeenten does not exist.

**GET** /crab/deelgemeenten/(string: deelgemeente\_id)  
Get all information on a certain *deelgemeente*.

### Example request:



```
GET /crab/deelgemeenten/45062 HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "naam": "Sint-Maria-Horebeke",
  "id": "45062A",
  "gemeente": {
    "id": 300,
    "naam": "Horebeke"
  }
}
```

**Status Codes**

- 200 OK – Deelgemeente was found.
- 404 Not Found – The Deelgemeente does not exist.

**GET /crab/gemeenten/ (int: gemeente\_id) /postkantons**  
List all *postkantons* in a certain *gemeente*.

**Example request:**

```
GET /crab/gemeenten/90/postkantons HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json
[
  {"id": 1730},
  {"id": 1731}
]
```

**Status Codes**

- 200 OK – Postkantons were found.
- 404 Not Found – The Gemeente for which you are requesting Postkantons does not exist.

**GET /crab/gemeenten/ (int:id of int:niscode) /straten**  
List all *straten* in a *gemeente*.

**Example request:**

```
GET /crab/gemeenten/11001/straten HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
Range: items=0-4
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "status": {
      "naam": "inGebruik",
      "definitie": null,
      "id": "3"
    },
    "id": 1,
    "label": "Acacialaan"
  },
  {
    "status": {
      "naam": "inGebruik",
      "definitie": null,
      "id": "3"
    },
    "id": 2,
    "label": "Adriaan Sanderslei"
  },
  {
    "status": {
      "naam": "inGebruik",
      "definitie": null,
      "id": "3"
    },
    "id": 3,
    "label": "Ahornelaan"
  },
  {
    "status": {
      "naam": "inGebruik",
      "definitie": null,
      "id": "3"
    },
    "id": 4,
    "label": "Antoon van Brabantstraat"
  },
  {
    "status": {
      "naam": "inGebruik",
      "definitie": null,
      "id": "3"
    },
    "id": 5,
    "label": "Antwerpsesteenweg"
  }
]
```

### Status Codes

- 200 OK – Straat was found.

**GET** /crab/straten/ (int: *straat\_id*)  
Get information on a *straat*, based on the *ID*.

**Example request:**

```
GET /crab/straten/1 HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": {
    "naam": "inGebruik",
    "definitie": null,
    "id": "3"
  },
  "namen": [
    [
      "Acacialaan",
      "nl"
    ],
    [
      null,
      null
    ]
  ],
  "taal": {
    "naam": "Nederlands",
    "definitie": "Nederlands.",
    "id": "nl"
  },
  "label": "Acacialaan",
  "bounding_box": [
    "150339.255243488",
    "200079.666892901",
    "150812.200907812",
    "201166.401677653"
  ],
  "id": 1,
  "metadata": {
    "begin_tijd": "2013-04-12 20:07:25.960000",
    "begin_datum": "1830-01-01 00:00:00",
    "begin_organisatie": {
      "naam": "gemeente",
      "definitie": "Gemeente.",
      "id": "1"
    },
    "begin_bewerking": {
      "naam": "correctie",
      "definitie": "Correctie van de attributen.",
      "id": "3"
    }
  }
}
```

**Status Codes**

- 200 OK – Straat was found.

- 404 Not Found – Straat was not found.

**GET /crab/straten/ (int: *straat\_id*) /huisnummers**

List all huisnummers in a *straat*.

**Example request:**

```
GET /crab/straten/1/huisnummers HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
Range: items=0-4
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "status": {
      "naam": "inGebruik",
      "definitie": null,
      "id": "3"
    },
    "id": 78036,
    "label": "21"
  },
  {
    "status": {
      "naam": "inGebruik",
      "definitie": null,
      "id": "3"
    },
    "id": 153134,
    "label": "4"
  },
  {
    "status": {
      "naam": "inGebruik",
      "definitie": null,
      "id": "3"
    },
    "id": 221505,
    "label": "11"
  },
  {
    "status": {
      "naam": "inGebruik",
      "definitie": null,
      "id": "3"
    },
    "id": 379090,
    "label": "23"
  },
  {
    "status": {
      "naam": "inGebruik",
      "definitie": null,
```

(continues on next page)

(continued from previous page)

```

        "id": "3"
    },
    "id": 526418,
    "label": "13"
}
]
```

### Status Codes

- 200 OK – Huisnummers were found.

**GET** `/crab/straten/(int: straat_id)/huisnummers/string: huisnummer_label` Get more information on a huisnummer by it's straat\_id and it's huisnummer.

### Example request:

```

GET /crab/straten/1/huisnummers/23 HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
```

### Example response:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": {
    "naam": "inGebruik",
    "definitie": null,
    "id": "3"
  },
  "bounding_box": [
    150786.11,
    200189.33,
    150786.11,
    200189.33
  ],
  "postadres": "Acacialaan 23, 2630 Aartselaar",
  "huisnummer": "23",
  "id": 379090,
  "metadata": {
    "begin_tijd": "2013-04-12 20:06:33.720000",
    "begin_datum": "1830-01-01 00:00:00",
    "begin_organisatie": {
      "naam": "gemeente",
      "definitie": "Gemeente.",
      "id": "1"
    },
    "begin_bewerking": {
      "naam": "correctie",
      "definitie": "Correctie van de attributen.",
      "id": "3"
    }
  }
}
```

### Status Codes

- 200 OK – Huisnummer was found.
- 404 Not Found – Huisnummer was not found.

**GET** /crab/huisnummers/ (int: *huisnummer\_id*)  
Get more information on a huisnummer, based on an *ID*.

**Example request:**

```
GET /crab/huisnummers/1 HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": {
    "naam": "inGebruik",
    "definitie": null,
    "id": "3"
  },
  "bounding_box": [
    190700.24,
    224649.87,
    190700.24,
    224649.87
  ],
  "huisnummer": "51",
  "postadres": "Steenweg op Oosthoven 51, 2300 Turnhout",
  "id": 1,
  "metadata": {
    "begin_tijd": "2014-03-19 17:00:27",
    "begin_datum": "1830-01-01 00:00:00",
    "begin_organisatie": {
      "naam": "gemeente",
      "definitie": "Gemeente.",
      "id": "1"
    },
    "begin_bewerking": {
      "naam": "correctie",
      "definitie": "Correctie van de attributen.",
      "id": "3"
    }
  }
}
```

### Status Codes

- 200 OK – Huisnummer was found.
- 404 Not Found – Huisnummer was not found.

**GET** /crab/huisnummers/ (int: *huisnummer\_id*) /percelen  
List all *percelen* linked to a certain *huisnummer*.

**Example request:**

```
GET /crab/huisnummers/1/percelen HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": "13040C1747/00G002"
  }
]
```

**Status Codes**

- 200 OK – Percelen were found.

**GET /crab/percelen/(string: *perceel\_id1*) /**  
**string: *perceel\_id2*** Get a perceel by it's id.

**Example request:**

```
GET /crab/percelen/13040C1747/00G002 HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "centroid": [
    190708.59,
    224667.59
  ],
  "id": "13040C1747/00G002",
  "postadressen": [
    "Steenweg op Oosthoven 51, 2300 Turnhout",
    "Steenweg op Oosthoven 53, 2300 Turnhout"
  ],
  "metadata": {
    "begin_tijd": "2009-09-11 12:46:55.693000",
    "begin_datum": "1998-01-01 00:00:00",
    "begin_organisatie": {
      "naam": "AAPD",
      "definitie": "Algemene Administratie der Patrimoniumdocumentatie.",
      "id": "3"
    },
    "begin_bewerking": {
      "naam": "correctie",
      "definitie": "Correctie van de attributen.",
      "id": "3"
    }
  }
}
```

### Status Codes

- 200 OK – Perceel was found.
- 404 Not Found – Perceel was not found.

**GET** /crab/percelen/(string: *perceel\_id1*) /  
string: *perceel\_id2/huisnummers* Get the huisnummers linked to a perceel.

### Example request:

```
GET /crab/percelen/13040C1747/00G002 HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
```

### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "status": {"naam": "inGebruik", "definitie": null, "id": "3"},
    "id": 1,
    "label": "51"
  }, {
    "status": {"naam": "buitenGebruik", "definitie": null, "id": "4"},
    "id": 2021223,
    "label": "53"
  }
]
```

### Status Codes

- 200 OK – Huisnummers were found.
- 404 Not Found – Perceel was not found.

**GET** /crab/huisnummers/(int: *huisnummer\_id*) /gebouwen  
List all gebouwen associated with a certain huisnummer.

### Example request:

```
GET /crab/huisnummer/1/gebouwen HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
```

### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "status": {
      "naam": "inGebruik",
      "definitie": null,
      "id": "4"
    }
  },
  ...
]
```

(continues on next page)



(continued from previous page)

```

    "aard": {
      "naam": "hoofdgebouw",
      "definitie": "hoofdgebouw volgens het GRB",
      "id": "1"
    },
    "id": 1538575
  }
]

```

### Status Codes

- 200 OK – Gebouwen were found.

**GET** /crab/gebouwen/(int: *gebouw\_id*)  
Get a Gebouw by it's id.

### Example request:

```

GET /crab/gebouwen/1 HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json

```

### Example response:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": {
    "naam": "inGebruik",
    "definitie": null,
    "id": "4"
  },
  "aard": {
    "naam": "hoofdgebouw",
    "definitie": "hoofdgebouw volgens het GRB",
    "id": "1"
  },
  "geometriemethode": {
    "naam": "grb",
    "definitie": null,
    "id": "3"
  },
  "geometrie": "POLYGON ((205574.52184166759 176477.42431658879, 205579.
↪1574896723 176476.68550058827, 205578.6424176693 176472.64633258432, 205588.
↪81227367371 176471.11494058371, 205589.80452967435 176478.83282858878, 205587.
↪36587367207 176479.30028459057, 205588.38680167496 176487.107260596, 205576.
↪12900967151 176488.87878059596, 205574.52184166759 176477.42431658879))",
  "id": 1,
  "metadata": {
    "begin_tijd": "2011-04-29 13:11:28.540000",
    "begin_datum": "1830-01-01 00:00:00",
    "begin_organisatie": {
      "naam": "AGIV",
      "definitie": "Agentschap voor Geografische Informatie Vlaanderen.",
      "id": "5"
    }
  }
}

```

(continues on next page)

(continued from previous page)

```
    },
    "begin_bewerking": {
      "naam": "invoer",
      "definitie": "Invoer in de databank.",
      "id": "1"
    }
  }
}
```

### Status Codes

- 200 OK – Gebouw was found.
- 404 Not Found – Gebouw was not found.

**GET /crab/huisnummers/ (int: *huisnummer\_id*) /subadressen**

List all Subadressen for a certain Huisnummer.

### Example request:

```
GET /crab/huisnummer/1/subadressen HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
```

### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "status": {
      "naam": "inGebruik",
      "definitie": null,
      "id": "3"
    },
    "id": 1120936,
    "subadres": "B"
  }, {
    "status": {
      "naam": "inGebruik",
      "definitie": null,
      "id": "3"
    },
    "id": 1120937,
    "subadres": "C"
  }
]
```

### Status Codes

- 200 OK – Subadressen were found.
- 404 Not Found – The Huisnummer for which you are requesting Subadressen does not exist.

**GET /crab/subadressen/ (int: *subadres\_id*)**

Get a Subadres based on it's id.

**Example request:**

```
GET /crab/subadressen/1 HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": {
    "naam": "inGebruik",
    "definitie": null,
    "id": "3"
  },
  "metadata": {
    "begin_tijd": "2013-04-12 20:07:13.180000",
    "begin_datum": "1830-01-01 00:00:00",
    "begin_organisatie": {
      "naam": "gemeente",
      "definitie": "Gemeente.",
      "id": "1"
    },
    "begin_bewerking": {
      "naam": "correctie",
      "definitie": "Correctie van de attributen.",
      "id": "3"
    }
  },
  "aard": {
    "naam": "rijksregister",
    "definitie": "Rijksregister.",
    "id": "2"
  },
  "id": 1120936,
  "subadres": "B"
}
```

**Status Codes**

- 200 OK – Subadres was found.
- 404 Not Found – Subadres was not found.

**GET /crab/huisnummers/ (int: *huisnummer\_id*) /adresposities**

List all adresposities for a certain huisnummer.

**Example request:**

```
GET /crab/huisnummer/1/adresposities HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "herkomst": {
      "naam": "manueleAanduidingVanPerceel",
      "definitie": null,
      "id": "2"
    },
    "id": 4087928
  }
]
```

#### Status Codes

- **200 OK** – A (possibly empty) list of adresposities is available.
- **404 Not Found** – The Huisnummer for which you are requesting Adresposities does not exist.

**GET /crab/subadressen/ (int: *subadres\_id*) /adresposities**

List all adresposities for a certain subadres.

**Example request:**

```
GET /crab/huisnummer/800000/adresposities HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "herkomst": {
      "naam": "afgeleidVanGebouw",
      "definitie": null,
      "id": "10"
    },
    "id": 2706297
  }
]
```

#### Status Codes

- **200 OK** – A (possibly empty) list of adresposities is available.
- **404 Not Found** – The Subadres for which you are requesting Adresposities does not exist.

**GET /crab/adresposities/ (int: *adrespositie\_id*)**

List all information on a certain adrespositie.

**Example request:**

```
GET /crab/adresposities/2706297 HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "herkomst": {
    "naam": "afgeleidVanGebouw",
    "definitie": null,
    "id": "10"
  },
  "metadata": {
    "begin_tijd": "2013-01-19 06:28:57.483000",
    "begin_datum": "1830-01-01 00:00:00",
    "begin_organisatie": {
      "naam": "AGIV",
      "definitie": "Agentschap voor Geografische Informatie
Vlaanderen.",
      "id": "5"
    },
    "begin_bewerking": {
      "naam": "correctie",
      "definitie": "Correctie van de attributen.",
      "id": "3"
    }
  },
  "geometrie": "POINT (154546.38 216367)",
  "id": 2706297,
  "aard": {
    "naam": "subAdres",
    "definitie": "Aanduiding van een plaats op een huisnummer",
    "id": "1"
  }
}
```

**GET /crab/landen**

List all landen.

**Example request:**

```
GET /crab/landen HTTP/1.1
Host: example.onroenderfgoed.be
Accept: application/json
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Range: 0-6/249

[
  {
    "naam": "Afghanistan",
    "id": "AF"
```

(continues on next page)

(continued from previous page)

```
    }, {
      "naam": "\u00c5land Islands",
      "id": "AX"
    }, {
      "naam": "Albania",
      "id": "AL"
    }, {
      "naam": "Algeria",
      "id": "DZ"
    }, {
      "naam": "American Samoa",
      "id": "AS"
    }, {
      "naam": "Andorra",
      "id": "AD"
    }, {
      "naam": "Angola",
      "id": "AO"
    }
  ]
```

### Status Codes

- 200 OK – List of landen was found.

### GET /crab/landen/BE

List all information for a certain land.

#### Example request:

```
GET /crab/landen/BE HTTP/1.1
Host: example.onroerenderfgoed.be
Accept: application/json
```

#### Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "alpha2": "BE",
  "alpha3": "BEL",
  "id": "BE",
  "naam": "Belgium"
}
```

### Status Codes

- 200 OK – Land was found.
- 404 Not Found – Land was not found.

## API DOCUMENTATION

`crabpy_pyramid.add_route` (*config, name, pattern, \*args, \*\*kwargs*)

Adds a pyramid route to the config. All args and kwargs will be passed on to `config.add_route`.

This exists so the default behaviour of including crabpy will still be to cache all crabpy routes.

`crabpy_pyramid.conditional_http_tween_factory` (*handler, registry*)

Tween that adds ETag headers and tells Pyramid to enable conditional responses where appropriate.

`crabpy_pyramid.get_capakey` (*registry*)

Get the Capakey Gateway

**Return type** `crabpy.gateway.capakey.CapakeyRestGateway`

`crabpy_pyramid.get_crab` (*registry*)

Get the Crab Gateway

**Return type** `crabpy.gateway.crab.CrabGateway`

# argument might be a config or a request

`crabpy_pyramid.includeme` (*config*)

Include *crabpy\_pyramid* in this *Pyramid* application.

**Parameters** `config` (*pyramid.config.Configurator*) – A Pyramid configurator.

`crabpy_pyramid.main` (*global\_config, \*\*settings*)

This function returns a Pyramid WSGI application.

### 3.1 Utils module

Utility functions to help with range handling.

New in version 0.1.0.

`crabpy_pyramid.utils.parse_range_header` (*range*)

Parse a range header as used by the dojo Json Rest store.

**Parameters** `range` (*str*) – The content of the range header to be parsed. eg. *items=0-9*

**Returns** A dict with keys *start*, *finish* and *number* or *False* if the range is invalid.

`crabpy_pyramid.utils.range_return` (*request, items*)

Determine what range of objects to return.

Will check for both *Range* and *X-Range* headers in the request and set both *Content-Range* and 'X-Content-Range' headers.

**Return type** `list`

`crabpy_pyramid.utils.set_http_caching(request, gateway='crab', region='permanent')`  
Set an HTTP Cache Control header on a request.

**Parameters**

- **request** (`pyramid.request.Request`) – Request to set headers on.
- **gateway** (`str`) – What gateway are we caching for? Defaults to *crab*.
- **region** (`str`) – What caching region to use? Defaults to *permanent*.

**Return type** `pyramid.request.Request`



## HISTORY

### 4.1 0.12.0 (15-12-2021)

- crabpy en crabpy\_pyramid dependencies nakijken (#155)

### 4.2 0.11.0 (15-09-2021)

- Drop python 2 support
- crabpy updaten (#148)

### 4.3 0.10.0 (13-08-2021)

- “naam” toevoegen aan straat renderer (#145)

### 4.4 0.9.0 (08-04-2021)

- Drop support for Python 3.4 and 3.5. Supported versions are 2.7, 3.6, 3.7 and 3.8. This is the last version to support Python 2.
- Updated to the latest pyCountry versions and adapted the API. (#119, #135)
- Added the shape file of a perceel to the renderer. (#122)
- Remove pyup integration (#133)
- GetPostkantonByHuisnummerId ontsluiten (#138)
- Sort parameter toevoegen aan alle requests (#140)

## 4.5 0.8.1 (06-06-2019)

- Etag tween geeft soms errors (#102)

## 4.6 0.8.0 (11-01-2019)

- Caching tween volledig herwerken (#97)

## 4.7 0.7.0 (2019-01-03)

- Update crabpy to get new deeltgemeente codes (#84, #92)
- Include CAPAKEY by default (#46)
- Allow passing custom crab wsdl (#75)

## 4.8 0.6.3 (2017-12-07)

- Make compatible with CRABpy 0.8.3
- Make compatible with latest pyCountry. (#56)

## 4.9 0.6.2 (2017-08-29)

- Make compatible with CRABpy 0.8.2

## 4.10 0.6.1 (2017-04-20)

- Make compatible with CRABpy 0.8.1 and fix center and bounding box formats in responses (#47)

## 4.11 0.6.0 (2017-04-19)

- Make compatible with CRABpy 0.8.0 and use the CAPAKEY REST gateway (#40)
- Blacklist for conditional tween (#41)

## 4.12 0.5.1 (2016-11-22)

- Add niscode to CRAB gemeente renderer. (#39)

## 4.13 0.5.0 (2016-07-27)

- Add HTTP caching headers. Both conditional GET with ETags and Cache-Control headers. Caching times are based on the ones passed to the gateways. (#37)
- Empty proxy settings are filtered out before being passed to the gateways. (#38)

## 4.14 0.4.1 (2016-02-02)

- Better error handling for capakey views. Generate HTTP 404 Not Found instead of HTTP 500 Internal Server Error. (#36)

## 4.15 0.4.0 (2016-01-25)

- Update dependencies. Update pycountry to version 1.19.
- Add explicit support for python 3.5.
- Add deeltgemeenten. (#33)
- Added postadressen in a few places.
- Added a list of huisnummers to a CRAB perceel.

## 4.16 0.3.0 (2015-06-01)

- Add Adresposities. Add endpoints that exposes the Adresposities linked to a certain Huisnummer or Subadres. (#26) [TalissaJoly]
- Add Landen (countries). Add endpoints to get a list of Landen or more information about a single Land. (#30, #31) [TalissaJoly]
- Return HTTP 404 errors for certain unexisting resources. (#25) [TalissaJoly]
- Limit the number of results that can be returned in a list. There was some partial handling for this, but only when the user did not send any range headers. This behaviour was deemed potentially hazardous and changed. Now, only a maximum of 5.000 records will be returned and range slicing works correctly. (#16) [TalissaJoly]

## 4.17 0.2.0 (2015-03-03)

- Implement two service endpoints dealing with Subadressen (think of boxes in an apartment building). (#18) [TalissaJoly]
- Implement a service endpoint for listing the Postkantons (ie. Postcode) in a certain gemeente. (#23) [TalissaJoly]
- Cleaned the code a bit by removing unneeded bits and pieces. (#17) (#19) [TalissaJoly]

## 4.18 0.1.1 (2014-09-18)

- Minor release adding and cleaning up lots of service documentation. (#14)
- Fixes a small bug when no capakey username or password was set and the capakey service was included.

## 4.19 0.1.0 (2014-09-05)

- First stable release, matches with CRABpy 0.4.1
- Exposes CRAB and CAPAKEY gateways.
- CRAB Gateway also exposes some Provincie objects. (#13)
- CRAB and CAPAKEY can be configured independently. (#11)

## 4.20 0.1.0a2 (2014-04-30)

- Still alpha.
- Coveralls support.
- Proxy settings in ini file. (#5)
- Range headers. (#9)

## 4.21 0.1.0a1 (2014-03-19)

- Initial version

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### C

`crabpy_pyramid`, [35](#)

`crabpy_pyramid.utils`, [35](#)





## HTTP ROUTING TABLE

<b>/capakey</b>	GET /crab/gewesten/(int:gewest_id)/provincies,
GET /capakey/afdelingen,5	15
GET /capakey/afdelingen/(int:afdeling_id),	GET /crab/huisnummers/(int:huisnummer_id),
6	26
GET /capakey/afdelingen/(int:afdeling_id)/secties,	GET /crab/huisnummers/(int:huisnummer_id)/adrespos,
6	31
GET /capakey/afdelingen/(int:afdeling_id)/secties/(string:sectie_id),	GET /crab/huisnummers/(int:huisnummer_id)/gebouwen,
8	28
GET /capakey/afdelingen/(int:afdeling_id)/secties/(string:sectie_id)/percelen,	GET /crab/huisnummers/(int:huisnummer_id)/percelen,
9	26
GET /capakey/afdelingen/(int:afdeling_id)/secties/(string:sectie_id)/percelen/(int:perceel_id),	GET /crab/huisnummers/(int:huisnummer_id)/subadres,
11	30
GET /capakey/gemeenten,3	GET /crab/landen,33
GET /capakey/gemeenten/(int:gemeente_id),	GET /crab/landen/BE,34
4	GET /crab/percelen/(string:perceel_id1)/(string:perceel_id2),
GET /capakey/gemeenten/(int:gemeente_id)/afdelingen,	27
4	GET /crab/percelen/(string:perceel_id1)/(string:perceel_id2),
GET /capakey/percelen/(string:capakey1)/(string:capakey2),	28
11	GET /crab/provincies/(int:provincie_id),
GET /capakey/percelen/(string:percid),	16
13	GET /crab/provincies/(int:provincie_id)/gemeenten,
	17
<b>/crab</b>	GET /crab/straten/(int:straat_id),22
GET /crab/adresposities/(int:adrespositie_id),	GET /crab/straten/(int:straat_id)/huisnummers,
32	24
GET /crab/deelgemeenten/(string:deelgemeente_id),	GET /crab/straten/(int:straat_id)/huisnummers/(string:deelgemeente_id),
20	25
GET /crab/gebouwen/(int:gebouw_id),29	GET /crab/subadressen/(int:subadres_id),
GET /crab/gemeenten/(int:gemeente_id	30
or int:niscode)/deelgemeenten,20	GET /crab/subadressen/(int:subadres_id)/adresposit,
GET /crab/gemeenten/(int:gemeente_id)/postkantons,	32
21	
GET /crab/gemeenten/(int:id of	
int:niscode),18	
GET /crab/gemeenten/(int:id of	
int:niscode)/straten,21	
GET /crab/gewesten,14	
GET /crab/gewesten/(int:gewest_id),15	
GET /crab/gewesten/(int:gewest_id)/deelgemeenten,	
19	
GET /crab/gewesten/(int:gewest_id)/gemeenten,	
17	



## INDEX

### A

`add_route()` (in module *crabpy\_pyramid*), 35

### C

`conditional_http_tween_factory()` (in module *crabpy\_pyramid*), 35

`crabpy_pyramid`  
module, 35

`crabpy_pyramid.utils`  
module, 35

### G

`get_capakey()` (in module *crabpy\_pyramid*), 35

`get_crab()` (in module *crabpy\_pyramid*), 35

### I

`includeme()` (in module *crabpy\_pyramid*), 35

### M

`main()` (in module *crabpy\_pyramid*), 35

module  
    *crabpy\_pyramid*, 35  
    *crabpy\_pyramid.utils*, 35

### P

`parse_range_header()` (in module *crabpy\_pyramid.utils*), 35

### R

`range_return()` (in module *crabpy\_pyramid.utils*), 35

### S

`set_http_caching()` (in module *crabpy\_pyramid.utils*), 35