

---

# **CRABpy Documentation**

***Release 0.9.0***

**Flanders Heritage Agency**

**May 23, 2023**



---

## Contents

---

|          |                                          |           |
|----------|------------------------------------------|-----------|
| <b>1</b> | <b>Using CRABpy</b>                      | <b>3</b>  |
| 1.1      | Using the CRAB webservice . . . . .      | 3         |
| 1.2      | Using the CAPAKEY webservice . . . . .   | 4         |
| 1.3      | Using a client behind a proxy . . . . .  | 4         |
| 1.4      | Using the CRAB gateway . . . . .         | 5         |
| 1.5      | Using the CAPAKEY gateway . . . . .      | 6         |
| 1.6      | Using the CAPAKEY REST gateway . . . . . | 8         |
| <b>2</b> | <b>Development</b>                       | <b>11</b> |
| <b>3</b> | <b>API Documentation</b>                 | <b>13</b> |
| 3.1      | Client module . . . . .                  | 13        |
| 3.2      | Crab gateway module . . . . .            | 14        |
| 3.3      | Capakey gateway module . . . . .         | 23        |
| 3.4      | Gateway exception module . . . . .       | 26        |
| 3.5      | Wsa module . . . . .                     | 26        |
| 3.6      | Wsse module . . . . .                    | 26        |
| <b>4</b> | <b>History</b>                           | <b>29</b> |
| 4.1      | 1.2.4 (23-05-2023) . . . . .             | 29        |
| 4.2      | 1.2.3 (15-05-2023) . . . . .             | 29        |
| 4.3      | 1.2.2 (09-05-2023) . . . . .             | 29        |
| 4.4      | 1.2.1 (04-05-2023) . . . . .             | 29        |
| 4.5      | 1.2.0 (04-05-2023) . . . . .             | 29        |
| 4.6      | 1.1.0 (03-05-2023) . . . . .             | 30        |
| 4.7      | 1.0.0 (13-04-2023) . . . . .             | 30        |
| 4.8      | 0.16.1 (29-03-2023) . . . . .            | 30        |
| 4.9      | 0.16.0 (29-03-2023) . . . . .            | 30        |
| 4.10     | 0.15.0 (10-12-2021) . . . . .            | 30        |
| 4.11     | 0.14.0 (18-11-2021) . . . . .            | 30        |
| 4.12     | 0.13.0 (14-09-2021) . . . . .            | 30        |
| 4.13     | 0.12.1 (28-01-2021) . . . . .            | 30        |
| 4.14     | 0.12.0 (24-06-2019) . . . . .            | 31        |
| 4.15     | 0.11.0 (03-01-2019) . . . . .            | 31        |
| 4.16     | 0.10.0 (17-07-2018) . . . . .            | 31        |
| 4.17     | 0.9.0 (20-03-2018) . . . . .             | 31        |
| 4.18     | 0.8.3 (07-12-2017) . . . . .             | 31        |

|          |                            |           |
|----------|----------------------------|-----------|
| 4.19     | 0.8.2 (25-08-2017)         | 31        |
| 4.20     | 0.8.1 (20-04-2017)         | 31        |
| 4.21     | 0.8.0 (19-04-2017)         | 32        |
| 4.22     | 0.7.0 (25-01-2016)         | 32        |
| 4.23     | 0.6.0 (01-06-2015)         | 32        |
| 4.24     | 0.5.0 (03-03-2015)         | 32        |
| 4.25     | 0.4.2 (18-09-2014)         | 32        |
| 4.26     | 0.4.1 (05-09-2014)         | 33        |
| 4.27     | 0.4.0 (03-09-2014)         | 33        |
| 4.28     | 0.3.5 (02-09-2014)         | 33        |
| 4.29     | 0.3.4 (07-05-2014)         | 33        |
| 4.30     | 0.3.3 (02-05-2014)         | 33        |
| 4.31     | 0.3.2 (07-04-2014)         | 33        |
| 4.32     | 0.3.1 (17-03-2014)         | 34        |
| 4.33     | 0.3.0 (12-03-2014)         | 34        |
| 4.34     | 0.2.1 (21-02-2014)         | 34        |
| 4.35     | 0.2.0 (03-12-2013)         | 34        |
| 4.36     | 0.1.0 (25-10-2013)         | 34        |
| <b>5</b> | <b>Indices and tables</b>  | <b>35</b> |
|          | <b>Python Module Index</b> | <b>37</b> |
|          | <b>Index</b>               | <b>39</b> |

This library provides access to the CRAB and CAPAKEY webservices operated by the [AGIV](#). Because connecting to these SOAP services from python can be somewhat complicated, this library makes it easier to use the service.

We also plan to offer a somewhat more opinionated wrapper around the services to straighten out some rough edges in the API and as an integration point for some caching.



### 1.1 Using the CRAB webservice

Recently, the CRAB service has become public. The need to authenticate has been removed, making it a whole lot easier to connect. A utility function `crabpy.client.crab_request()` has been provided, similar to `crabpy.client.capakey_request()`. This allows for a slightly different way of calling methods on the service.

```
"""
This script demonstrates using the crab client directly or through the
:func:`crabpy.client.crab_request` function.
"""

from crabpy.client import crab_factory, crab_request

crab = crab_factory()

res = crab.service.ListGemeentenByGewestId(1)
print(res)

res = crab.service.ListPostkantonsByGemeenteId(71)
print(res)

res = crab_request(crab, "ListGemeentenByGewestId", 1)
print(res)

res = crab_request(crab, "ListHuisnummersWithStatusByStraatnaamId", 18618)
print(res)

res = crab_request(crab, "GetStraatnaamWithStatusByStraatnaamId", 18618)
print(res)
```

## 1.2 Using the CAPAKEY webservice

This service does still require authentication. This requires a valid account from [agiv](#). Because the authentication also requires some extra WS-Addressing headers, a utility function `crabpy.client.capakey_request()` has been provided to make life easier.

```
"""
This script demonstrates using the capakey client through the
:func:`crabpy.client.capakey_request` function.
"""

from crabpy.client import capakey_factory, capakey_request

capakey = capakey_factory(user="USER", password="PASSWORD")

res = capakey_request(capakey, "ListAdmGemeenten", 1)
print(res)

res = capakey_request(capakey, "ListKadAfdelingenByNiscode", 44021, 1)
print(res)

res = capakey_request(capakey, "ListKadSectiesByKadAfdelingcode", 44021)
print(res)

res = capakey_request(capakey, "ListKadPerceelsnummersByKadSectiecode", 44021, "A", 1)
print(res)

res = capakey_request(capakey, "GetKadPerceelsnummerByCaPaKey", "44021A3675/00A000")
print(res)
```

Be careful: the CAPAKEY SOAP gateway is deprecated. We advise you to use the CAPAKEY REST gateway instead.

## 1.3 Using a client behind a proxy

If you need to connect to CRAB or CAPAKEY through a proxy, you can do so by passing the proxy parameter to the `crabpy.client.crab_factory()` or `crabpy.client.capakey_factory()`.

```
"""
This script shows how to connect to the WS-WRAB service through a proxy.
"""

from crabpy.client import crab_factory

crab = crab_factory(
    proxy={
        "http": "http://proxy.example.com:3128",
        "https": "https://httpsproxy.example.com:3128",
    }
)

print(crab.service.ListGemeentenByGewestId(1))
```



## 1.4 Using the CRAB gateway

To make life easier and crab more pythonic, we've also implemented a gateway that abstracts some more of the service and provides richer objects as responses.

```
"""
This script demonstrates using the crab gateway to walk the entire
address tree (street and number) of a `gemeente`.
"""

from crabpy.client import crab_factory
from crabpy.gateway.crab import CrabGateway

g = CrabGateway(crab_factory())

gemeente = g.get_gemeente_by_id(1)

print(str(gemeente))
for s in gemeente.straten:
    print("* %s" % s)
    for h in s.huisnummers:
        print("\t* %s" % h)
        for sa in h.subadressen:
            print("\t\t* %s" % sa)
```

The CRAB gateway supports caching through the [dogpile](#) caching library. Caching can be added by passing a configuration dictionary to the `CrabGateway`.

Three caching regions will be configured:

- *permanent*: For requests that can be cached for a very long time, eg. *list\_gewesten* or *list\_gemeenten*.
- *long*: For requests that can be cached for a fairly long time, eg. *list\_straten*.
- *short*: For requests that will only be cached for a little while, eg. *get\_huisnummer\_by\_id*.

```
"""
This script demonstrates querying the crab gateway while maintaining a cache.
"""

import os

from crabpy.client import crab_factory
from crabpy.gateway.crab import CrabGateway

root = "./dogpile_data/"

if not os.path.exists(root):
    os.makedirs(root)

g = CrabGateway(
    crab_factory(),
    cache_config={
        "permanent.backend": "dogpile.cache.dbm",
        "permanent.expiration_time": 604800,
        "permanent.arguments.filename": os.path.join(root, "crab_permanent.dbm"),
        "long.backend": "dogpile.cache.dbm",
```

(continues on next page)

(continued from previous page)

```

        "long.expiration_time": 86400,
        "long.arguments.filename": os.path.join(root, "crab_long.dbm"),
    },
)

aartselaar = g.get_gemeente_by_id(1)

print("Straten in AARTSELAAR")
print("-----")
print([str(s) for s in g.list_straten(aartselaar)])

print("Huisnummers in AARTSELAAR Straat1")
print("-----")
# print([str(h) for h in g.list_huisnummers_by_straat(s)])

p = g.get_gemeente_by_niscode(33021)

print("gemeente: %s" % p.id)
print("naam: %s" % p.naam)
print("niscode: %s" % p.niscode)
print("gewest: %s" % p.gewest)
print("provincie: %s" % p.provincie)
print("taal: %s" % p.taal)
print("centroid: %s" % str(p.centroid))
print("bounding_box: %s" % str(p.bounding_box))

```

## 1.5 Using the CAPAKEY gateway

To make life easier and capakey more pythonic, we've also implemented a gateway that abstracts some more of the service and provides richer objects as responses.

```

"""
This script demonstrates using the capakey gateway to walk the entire
cadastral tree of a `gemeente`.

WARNING: The CapakeyGateway (SOAP) is deprecated, use CapakeyRestGateway (REST)
↳ instead.
"""

from crabpy.client import capakey_factory
from crabpy.gateway.capakey import CapakeyGateway

capakey = capakey_factory(user="USER", password="PASSWORD")

g = CapakeyGateway(capakey)

gemeente = g.get_gemeente_by_id(45062)

print(str(gemeente))
for a in gemeente.afdelingen:
    print("* %s" % a)
    for s in a.secties:
        print("\t* %s" % s)

```

(continues on next page)

(continued from previous page)

```

for p in s.perceles:
    print("\t\t* %s" % p)

```

The CAPAKEY gateway supports caching through the `dogpile` caching library. Caching can be added by passing a configuration dictionary to the `CapakeyGateway`.

Three caching regions will be configured:

- *permanent*: For requests that can be cached for a very long time, eg. *list\_gemeenten*.
- *long*: For requests that can be cached for a fairly long time, eg. *list\_secties\_by\_afdeling*.
- *short*: For requests that will only be cached for a little while, eg. *get\_perceel\_by\_capakey*.

Please bear in mind that in this case short can probably be fairly long. We suspect that the database underlying the capakey service is not updated that regularly, so a short caching duration could easily be one hour or even a day.

```

"""
This script demonstrates querying the capakey gateway while maintaining a cache.

WARNING: The CapakeyGateway (SOAP) is deprecated, use CapakeyRestGateway (REST)
↳ instead.
"""

import os

from crabpy.client import capakey_factory
from crabpy.gateway.capakey import CapakeyGateway

capakey = capakey_factory(user="USER", password="PASSWORD")

root = "./dogpile_data/"

if not os.path.exists(root):
    os.makedirs(root)

g = CapakeyGateway(
    capakey,
    cache_config={
        "permanent.backend": "dogpile.cache.dbm",
        "permanent.expiration_time": 604800,
        "permanent.arguments.filename": os.path.join(root, "capakey_permanent.dbm"),
        "long.backend": "dogpile.cache.dbm",
        "long.expiration_time": 86400,
        "long.arguments.filename": os.path.join(root, "capakey_long.dbm"),
        "short.backend": "dogpile.cache.dbm",
        "short.expiration_time": 3600,
        "short.arguments.filename": os.path.join(root, "capakey_short.dbm"),
    },
)

gent = g.get_gemeente_by_id(44021)

print("Afdelingen in Gent")
print("-----")

print([str(a) for a in g.list_kadastrale_afdelingen_by_gemeente(gent)])

```

(continues on next page)

(continued from previous page)

```

print("Secties in GENT AFD 1")
print("-----")

print([str(s) for s in g.list_secties_by_afdeling(44021)])

print("Percelen in GENT AFD 1, Sectie A")
print("-----")

s = g.get_sectie_by_id(44021, "A")
print([str(p) for p in g.list_percelen_by_sectie(s)])

print("Perceel 44021A3675/00A000")
print("-----")

p = g.get_perceel_by_capakey("44021A3675/00A000")

print("perceel: %s" % p.id)
print("capakey: %s" % p.capakey)
print("percid: %s" % p.percid)
print("grondnummer: %s" % p.grondnummer)
print("bisnummer: %s" % p.bisnummer)
print("exponent: %s" % p.exponent)
print("macht: %s" % p.macht)
print("sectie: %s" % p.sectie)
print("afdeling: %s" % p.sectie.afdeling)

```

See the examples folder for some more sample code.

**Warning:** Be careful: the CAPAKEY SOAP gateway is deprecated. We advise you the use the CAPAKEY REST gateway instead.

## 1.6 Using the CAPAKEY REST gateway

To make life easier and capakey more pythonic, we've also implemented a rest gateway that abstracts some more of the service and provides richer objects as responses.

The CAPAKEY REST gateway supports caching through the [dogpile](#) caching library. Caching can be added by passing a configuration dictionary to the CapakeyRestGateway.

Three caching regions will be configured:

- *permanent*: For requests that can be cached for a very long time, eg. *list\_gemeenten*.
- *long*: For requests that can be cached for a fairly long time, eg. *list\_secties\_by\_afdeling*.
- *short*: For requests that will only be cached for a little while, eg. *get\_perceel\_by\_capakey*.

Please bear in mind that in this case short can probably be fairly long. We suspect that the database underlying the capakey service is not updated that regularly, so a short caching duration could easily be one hour or even a day.

```

"""
This script demonstrates querying the capakey gateway while maintaining a cache.
"""

```

(continues on next page)

(continued from previous page)

```

import os
from crabpy.gateway.capakey import CapakeyRestGateway

root = "./dogpile_data/"

if not os.path.exists(root):
    os.makedirs(root)

g = CapakeyRestGateway(
    cache_config={
        "permanent.backend": "dogpile.cache.dbm",
        "permanent.expiration_time": 604800,
        "permanent.arguments.filename": os.path.join(root, "capakey_permanent.dbm"),
        "long.backend": "dogpile.cache.dbm",
        "long.expiration_time": 86400,
        "long.arguments.filename": os.path.join(root, "capakey_long.dbm"),
        "short.backend": "dogpile.cache.dbm",
        "short.expiration_time": 3600,
        "short.arguments.filename": os.path.join(root, "capakey_short.dbm"),
    }
)

gent = g.get_gemeente_by_id(44021)

print("Afdelingen in Gent")
print("-----")

print([str(a) for a in g.list_kadastrale_afdelingen_by_gemeente(gent)])

print("Secties in GENT AFD 1")
print("-----")

print([str(s) for s in g.list_secties_by_afdeling(44021)])

print("Percelen in GENT AFD 1, Sectie A")
print("-----")

# print([str(p) for p in g.list_percelen_by_sectie(s)])

print("Perceel 44021A3675/00A000")
print("-----")

p = g.get_perceel_by_capakey("44021A3675/00A000")

print("perceel: %s" % p.id)
print("capakey: %s" % p.capakey)
print("percid: %s" % p.percid)
print("grondnummer: %s" % p.grondnummer)
print("bisnummer: %s" % p.bisnummer)
print("exponent: %s" % p.exponent)
print("macht: %s" % p.macht)
print("sectie: %s" % p.sectie)
print("afdeling: %s" % p.sectie.afdeling)

```

See the examples folder for some more sample code.



## CHAPTER 2

---

### Development

---

Crabpy is still in development, but the general API is stable and we are already using it in production. Generally we're also just happy to have gotten a SOAP service working in python.

We try to cover as much code as we can with unit tests. You can run them using `tox` or directly through `py.test`.

```
$ tox
# No coverage
$ py.test
# Coverage
$ py.test --cov crabpy --cov-report term-missing tests/
```

If you have access to the capakey service, you can enter your credentials in the `pytest_dist.ini` file and use that as a test config. It's actually best to copy this file and edit the copy instead of the original.

```
[pytest]
addopts = --crab-integration --capakey-integration --capakey-soap-integration --
↳capakey-soap-user=<username> --capakey-soap-password=<password>
```

```
# Integration tests but no coverage
$ py.test -c pytest_<user>.ini
# Integration tests with coverage
$ py.test -c pytest_<user>.ini --cov crabpy --cov-report term-missing tests/
# Running just the CRAB integration tests without using a config file
$ py.test --crab-integration --cov crabpy --cov-report term-missing tests/
```





### 3.1 Client module

This module contains utility functions for interacting with AGIV SOAP services.

New in version 0.1.0.

**exception** `crabpy.client.AdressenRegisterClientException`

`crabpy.client.crab_factory(**kwargs)`

Factory that generates a CRAB client.

A few parameters will be handled by the factory, other parameters will be passed on to the client.

**Parameters**

- **wsdl** – *Optional*. Allows overriding the default CRAB wsdl url.
- **proxy** – *Optional*. A dictionary of proxy information that is passed to the underlying `suds.client.Client`

**Return type** `suds.client.Client`

`crabpy.client.crab_request(client, action, *args)`

Utility function that helps making requests to the CRAB service.

**Parameters**

- **client** – A `suds.client.Client` for the CRAB service.
- **action** (*string*) – Which method to call, eg. *ListGewesten*

**Returns** Result of the SOAP call.

New in version 0.3.0.

## 3.2 Crab gateway module

This module contains an opinionated gateway for the crab webservice.

New in version 0.3.0.

**class** crabpy.gateway.crab.**Aardadres** (*id, naam, definitie, \*\*kwargs*)

The nature of an address.

**class** crabpy.gateway.crab.**Aardgebouw** (*id, naam, definitie, \*\*kwargs*)

The nature of a building.

**class** crabpy.gateway.crab.**Aardsubadres** (*id, naam, definitie, \*\*kwargs*)

The nature of a subaddress.

**class** crabpy.gateway.crab.**Aardterreinobject** (*id, naam, definitie, \*\*kwargs*)

The nature of a *terreinobject*.

**class** crabpy.gateway.crab.**Aardwegobject** (*id, naam, definitie, \*\*kwargs*)

The nature of a *wegobject*.

**class** crabpy.gateway.crab.**Adrespositie** (*id, herkomst, geometrie=None, aard=None, meta-data=None, \*\*kwargs*)

The position of an *Adres*.

This can be used for the position of both *Huisnummer* and *Subadres*.

A *Huisnummer* or *Subadres*, can have more than one *Adrespositie*, each offering a different interpretation of the position of the *Adres*. See the *herkomst* and *aard* of each *Adrespositie* to know which one to pick.

**class** crabpy.gateway.crab.**Bewerking** (*id, naam, definitie, \*\*kwargs*)

An edit.

**class** crabpy.gateway.crab.**Codelijst** (*id, naam, definitie, \*\*kwargs*)

**class** crabpy.gateway.crab.**CrabGateway** (*client, \*\*kwargs*)

A gateway to the CRAB webservice.

**get\_adrespositie\_by\_id** (*id*)

Retrieve a *Adrespositie* by the Id.

**Parameters** *id* (*integer*) – the Id of the *Adrespositie*

**Return type** *Adrespositie*

**get\_deelgemeente\_by\_id** (*id*)

Retrieve a *deelgemeente* by the id.

**Parameters** *id* (*string*) – The id of the *deelgemeente*.

**Return type** *Deelgemeente*

**get\_gebouw\_by\_id** (*id*)

Retrieve a *Gebouw* by the Id.

**Parameters** *id* (*integer*) – the Id of the *Gebouw*

**Return type** *Gebouw*

**get\_gemeente\_by\_id** (*id*)

Retrieve a *gemeente* by the crab id.

**Parameters** *id* (*integer*) – The CRAB id of the *gemeente*.

**Return type** *Gemeente*

**get\_gemeente\_by\_niscode** (*niscode*)

Retrieve a *gemeente* by the NIScode.

**Parameters** *niscode* (*integer*) – The NIScode of the gemeente.

**Return type** *Gemeente*

**get\_gewest\_by\_id** (*id*)

Get a *gewest* by id.

**Parameters** *id* (*integer*) – The id of a *gewest*.

**Return type** A *Gewest*.

**get\_huisnummer\_by\_id** (*id*)

Retrieve a *huisnummer* by the Id.

**Parameters** *id* (*integer*) – the Id of the *huisnummer*

**Return type** *Huisnummer*

**get\_huisnummer\_by\_nummer\_and\_straat** (*nummer*, *straat*)

Retrieve a *huisnummer* by the *nummer* and *straat*

**Parameters**

- **nummer** (*integer*) – The huisnummer of the ‘huisnummer’
- **straat** – The *Straat* in which the *huisnummer* is situated.

**Return type** A *Huisnummer*

**get\_perceel\_by\_id** (*id*)

Retrieve a *Perceel* by the Id.

**Parameters** *id* (*string*) – the Id of the *Perceel*

**Return type** *Perceel*

**get\_postadres\_by\_huisnummer** (*huisnummer*)

Get the *postadres* for a *Huisnummer*.

**Parameters** *huisnummer* – The *Huisnummer* for which the *postadres* is wanted. OR A huisnummer id.

**Return type** A *str*.

**get\_postadres\_by\_subadres** (*subadres*)

Get the *postadres* for a *Subadres*.

**Parameters** *subadres* – The *Subadres* for which the *postadres* is wanted. OR A subadres id.

**Return type** A *str*.

**get\_postkanton\_by\_huisnummer** (*huisnummer*)

Retrieve a *postkanton* by the Huisnummer.

**Parameters** *huisnummer* – The *Huisnummer* for which the *postkanton* is wanted.

**Return type** *Postkanton*

**get\_provincie\_by\_id** (*niscode*)

Retrieve a *provincie* by the nicode.

**Parameters** *niscode* (*integer*) – The nicode of the provincie.

**Return type** *Provincie*

**get\_straat\_by\_id**(*id*)

Retrieve a *straat* by the Id.

**Parameters** *id* (*integer*) – The id of the *straat*.

**Return type** *Straat*

**get\_subadres\_by\_id**(*id*)

Retrieve a *Subadres* by the Id.

**Parameters** *id* (*integer*) – the Id of the *Subadres*

**Return type** *Subadres*

**get\_terreinobject\_by\_id**(*id*)

Retrieve a *Terreinobject* by the Id.

**Parameters** *id* (*integer*) – the Id of the *Terreinobject*

**Return type** *Terreinobject*

**get\_wegobject\_by\_id**(*id*)

Retrieve a *Wegobject* by the Id.

**Parameters** *id* (*integer*) – the Id of the *Wegobject*

**Return type** *Wegobject*

**get\_wegsegment\_by\_id**(*id*)

Retrieve a *wegsegment* by the Id.

**Parameters** *id* (*integer*) – the Id of the *wegsegment*

**Return type** *Wegsegment*

**list\_aardadressen**(*sort=1*)

List all *aardadressen*.

**Return type** A list of *Aardadres*

**list\_aardgebouwen**(*sort=1*)

List all *aardgebouwen*.

**Return type** A list of *Aardgebouw*

**list\_aardsubadressen**(*sort=1*)

List all *aardsubadressen*.

**Return type** A list of *Aardsubadres*

**list\_aardterreinobjecten**(*sort=1*)

List all *aardterreinobjecten*.

**Return type** A list of *Aardterreinobject*

**list\_aardwegobjecten**(*sort=1*)

List all *aardwegobjecten*.

**Return type** A list of *Aardwegobject*

**list\_adresposities\_by\_huisnummer**(*huisnummer*)

List all *adresposities* for a *Huisnummer*.

**Parameters** *huisnummer* – The *Huisnummer* for which the *adresposities* are wanted. OR  
A *huisnummer* id.

**Return type** A list of *Adrespositie*

**list\_adresposities\_by\_nummer\_and\_straat** (*nummer, straat*)

List all *adresposities* for a huisnummer and a *Straat*.

**Parameters**

- **nummer** – A string representing a certain huisnummer.
- **straat** – The *Straat* for which the *adresposities* are wanted. OR A straat id.

**Return type** A list of *Adrespositie*

**list\_adresposities\_by\_subadres** (*subadres*)

List all *adresposities* for a *Subadres*.

**Parameters** **subadres** – The *Subadres* for which the *adresposities* are wanted. OR A sub-adres id.

**Return type** A list of *Adrespositie*

**list\_adresposities\_by\_subadres\_and\_huisnummer** (*subadres, huisnummer*)

List all *adresposities* for a subadres and a *Huisnummer*.

**Parameters**

- **subadres** – A string representing a certain subadres.
- **huisnummer** – The *Huisnummer* for which the *adresposities* are wanted. OR A huisnummer id.

**Return type** A list of *Adrespositie*

**list\_bewerkingen** (*sort=1*)

List all *bewerkingen*.

**Return type** A list of *Bewerking*

**list\_deelgemeenten** (*gewest=2*)

List all *deelgemeenten* in a *gewest*.

**Parameters** **gewest** – The *Gewest* for which the *deelgemeenten* are wanted. Currently only Flanders is supported.

**Return type** A list of *Deelgemeente*.

**list\_deelgemeenten\_by\_gemeente** (*gemeente*)

List all *deelgemeenten* in a *gemeente*.

**Parameters** **gemeente** – The *Gemeente* for which the *deelgemeenten* are wanted. Currently only Flanders is supported.

**Return type** A list of *Deelgemeente*.

**list\_gebouwen\_by\_huisnummer** (*huisnummer*)

List all *gebouwen* for a *Huisnummer*.

**Parameters** **huisnummer** – The *Huisnummer* for which the *gebouwen* are wanted.

**Return type** A list of *Gebouw*

**list\_gemeenten** (*gewest=2, sort=1*)

List all *gemeenten* in a *gewest*.

**Parameters**

- **gewest** – The *Gewest* for which the *gemeenten* are wanted.
- **sort** (*integer*) – What field to sort on.

**Return type** A list of *Gemeente*.

**list\_gemeenten\_by\_provincie** (*provincie*)

List all *gemeenten* in a *provincie*.

**Parameters** *provincie* – The *Provincie* for which the *gemeenten* are wanted.

**Return type** A list of *Gemeente*.

**list\_geometriemethodegebouwen** (*sort=1*)

List all *geometriegebouwen*.

**Return type** A list of *Geometriegebouw*

**list\_geometriemethodewegsegmenten** (*sort=1*)

List all *geometriemethodewegsegmenten*.

**Return type** A list of *Geometriemethodewegsegment*

**list\_gewesten** (*sort=1*)

List all *gewesten* in Belgium.

**Parameters** *sort* (*integer*) – What field to sort on.

**Return type** A :class:'list' of class: *Gewest*.

**list\_herkomstadresposities** (*sort=1*)

List all *herkomstadresposities*.

**Return type** A list of *Herkomstadrespositie*

**list\_huisnummers\_by\_perceel** (*perceel*, *sort=1*)

List all *huisnummers* on a *Pereel*.

Generally there will only be one, but multiples are possible.

**Parameters** *perceel* – The *Perceel* for which the *huisnummers* are wanted.

**Return type** A :class: list of *Huisnummer*

**list\_huisnummers\_by\_straat** (*straat*, *sort=1*)

List all *huisnummers* in a *Straat*.

**Parameters** *straat* – The *Straat* for which the *huisnummers* are wanted.

**Return type** A :class: list of *Huisnummer*

**list\_organisaties** (*sort=1*)

List all *organisaties*.

**Return type** A list of *Organisatie*

**list\_percelen\_by\_huisnummer** (*huisnummer*)

List all *percelen* for a *Huisnummer*

**Parameters** *huisnummer* – The *Huisnummer* for which the *percelen* are wanted.

**Return type** A list of *Perceel*

**list\_postkantons\_by\_gemeente** (*gemeente*)

List all *postkantons* in a *Gemeente*

**Parameters** *gemeente* – The *Gemeente* for which the *potkantons* are wanted.

**Return type** A list of *Postkantoon*

**list\_provincies** (*gewest=2*)

List all *provincies* in a *gewest*.

**Parameters**

- **gewest** – The *Gewest* for which the *provincies* are wanted.
- **sort** (*integer*) – What field to sort on.

**Return type** A list of *Provincie*.

**list\_statusgebouwen** (*sort=1*)

List all *statusgebouwen*.

**Return type** A list of *Statusgebouwen*

**list\_statushuisnummers** (*sort=1*)

List all *statushuisnummers*.

**Return type** A list of *Statushuisnummer*

**list\_statusstraatnamen** (*sort=1*)

List all *statusstraatnamen*.

**Return type** A list of *Statusstraatnaam*

**list\_statussubadressen** (*sort=1*)

List all *statussubadressen*.

**Return type** A list of *Statussubadres*

**list\_statuswegsegmenten** (*sort=1*)

List all *statuswegsegmenten*.

**Return type** A list of *Statuswegsegment*

**list\_straten** (*gemeente, sort=1*)

List all *straten* in a *Gemeente*.

**Parameters** **gemeente** – The *Gemeente* for which the *straten* are wanted.

**Return type** A list of *Straat*

**list\_subadressen\_by\_huisnummer** (*huisnummer*)

List all *subadressen* for a *Huisnummer*.

**Parameters** **huisnummer** – The *Huisnummer* for which the *subadressen* are wanted. OR A huisnummer id.

**Return type** A list of *Gebouw*

**list\_talen** (*sort=1*)

List all *talen*.

**Return type** A list of *Taal*

**list\_terreinobjecten\_by\_huisnummer** (*huisnummer*)

List all *terreinobjecten* for a *Huisnummer*

**Parameters** **huisnummer** – The *Huisnummer* for which the *terreinobjecten* are wanted.

**Return type** A list of *Terreinobject*

**list\_wegobjecten\_by\_straat** (*straat*)

List all *wegobjecten* in a *Straat*

**Parameters** **straat** – The *Straat* for which the *wegobjecten* are wanted.

**Return type** A list of *Wegobject*

**list\_wegsegmenten\_by\_straat** (*straat*)

List all *wegsegmenten* in a *Straat*

**Parameters** *straat* – The *Straat* for which the *wegsegmenten* are wanted.

**Return type** A list of *Wegsegment*

**class** `crabpy.gateway.crab.Deelgemeente` (*id, naam, gemeente\_niscode, \*\*kwargs*)

A subdivision of a *Gemeente*.

New in version 0.7.0.

**clear\_gateway** ()

Clear the currently set CrabGateway.

**set\_gateway** (*gateway*)

**Parameters** *gateway* (`crabpy.gateway.crab.CrabGateway`) – Gateway to use.

**class** `crabpy.gateway.crab.GatewayObject` (*\*\*kwargs*)

Abstract class for objects that are able to use a `crabpy.Gateway.CrabGateway` to find further information.

**check\_gateway** ()

Check to see if a gateway was set on this object.

**clear\_gateway** ()

Clear the currently set CrabGateway.

**gateway** = None

The `crabpy.gateway.crab.CrabGateway` to use when making further calls to the CRAB service.

**set\_gateway** (*gateway*)

**Parameters** *gateway* (`crabpy.gateway.crab.CrabGateway`) – Gateway to use.

**class** `crabpy.gateway.crab.Gebouw` (*id, aard, status, methode=None, geometrie=None, metadata=None, \*\*kwargs*)

A building.

**class** `crabpy.gateway.crab.Gemeente` (*id, naam, niscode, gewest, taal=None, centroid=None, bounding\_box=None, metadata=None, \*\*kwargs*)

The smallest administrative unit in Belgium.

**clear\_gateway** ()

Clear the currently set CrabGateway.

**set\_gateway** (*gateway*)

**Parameters** *gateway* (`crabpy.gateway.crab.CrabGateway`) – Gateway to use.

**class** `crabpy.gateway.crab.Geometriemethodegebouw` (*id, naam, definitie, \*\*kwargs*)

The geometry method of a *Gebouw*.

**class** `crabpy.gateway.crab.Geometriemethodewegsegment` (*id, naam, definitie, \*\*kwargs*)

The geometry method of a *Wegsegment*.

**class** `crabpy.gateway.crab.Gewest` (*id, namen=None, centroid=None, bounding\_box=None, \*\*kwargs*)

A large administrative unit in Belgium.

Belgium consists of 3 *gewesten*. Together they form the entire territory of the country.

**class** `crabpy.gateway.crab.Herkomstadrespositie` (*id, naam, definitie, \*\*kwargs*)

The origin of an Adressposition.



```
class crabpy.gateway.crab.Huisnummer(id, status, huisnummer, straat_id, metadata=None,  
                                     **kwargs)
```

A house number.

This is mainly a combination of a street and a house number.

```
class crabpy.gateway.crab.Metadata(begin_datum, begin_tijd, begin_bewerking, be-  
                                gin_organisatie, **kwargs)
```

Metadata about a *straat*, *huisnummer*, ...

Some of the metadata available is the datum the object was created, the organisation that created it and the type of creation.

```
class crabpy.gateway.crab.Organisatie(id, naam, definitie, **kwargs)
```

An organisation that played a role in the genesis of an object.

```
class crabpy.gateway.crab.Perceel(id, centroid=None, metadata=None, **kwargs)
```

A cadastral Parcel.

A *Terreinobject* is somewhat different from a *Perceel* in the source of the data and the information provided. eg. A *terreinobject* has a *centroid* and a *bounding box*, while a *perceel* also has the centroid, but not the *bounding box*.

#### **huisnummers**

Returns the huisnummers on this Perceel.

Some of the huisnummers might no longer be active.

**Return type** *list*

#### **postadressen**

Returns the postadressen for this Perceel.

Will only take the huisnummers with status *inGebruik* into account.

**Return type** *list*

```
class crabpy.gateway.crab.Postkanton(id, **kwargs)
```

A postal code.

Eg. postal code 9000 for the city of Ghent.

```
class crabpy.gateway.crab.Provincie(niscode, naam, gewest, **kwargs)
```

The largest administrative unit within a *Gewest*.

New in version 0.4.0.

```
clear_gateway()
```

Clear the currently set CrabGateway.

```
set_gateway(gateway)
```

**Parameters** *gateway* (*crabpy.gateway.crab.CrabGateway*) – Gateway to use.

```
class crabpy.gateway.crab.Statusgebouw(id, naam, definitie, **kwargs)
```

The current state of a *Gebouw*.

```
class crabpy.gateway.crab.Statushuisnummer(id, naam, definitie, **kwargs)
```

The current state of a *huisnummer*.

```
class crabpy.gateway.crab.Statusstraatnaam(id, naam, definitie, **kwargs)
```

The current state of a *straatnaam*.

```
class crabpy.gateway.crab.Statussubadres(id, naam, definitie, **kwargs)
```

The current state of a *subadres*.

**class** crabpy.gateway.crab.**Statuswegsegment** (*id, naam, definitie, \*\*kwargs*)  
The current state of a *wegsegment*.

**class** crabpy.gateway.crab.**Straat** (*id, label, gemeente\_id, status, straatnaam, taalcode, straat-naam2, taalcode2, metadata=None, \*\*kwargs*)  
A street.  
A street object is always located in one and exactly one *Gemeente*.

**class** crabpy.gateway.crab.**Subadres** (*id, subadres, status, huisnummer\_id=None, aard=None, metadata=None, \*\*kwargs*)  
An address within a certain *Huisnummer*.  
These can eg. be postboxes within an apartment complex.

**class** crabpy.gateway.crab.**Taal** (*id, naam, definitie, \*\*kwargs*)  
A language.

**class** crabpy.gateway.crab.**Terreinobject** (*id, aard, centroid=None, bounding\_box=None, metadata=None, \*\*kwargs*)  
A cadastral parcel.  
A *Terreinobject* is somewhat different from a *Perceel* in the source of the data and the information provided. eg. A *terreinobject* has a *centroid* and a *bounding box*, while a *perceel* also has the *centroid*, but not the *bounding box*.

**class** crabpy.gateway.crab.**Wegobject** (*id, aard, centroid=None, bounding\_box=None, meta-data=None, \*\*kwargs*)

**class** crabpy.gateway.crab.**Wegsegment** (*id, status, methode=None, geometrie=None, meta-data=None, \*\*kwargs*)

crabpy.gateway.crab.**check\_lazy\_load\_adrespositie** (*f*)  
Decorator function to lazy load a *Adrespositie*.

crabpy.gateway.crab.**check\_lazy\_load\_gebouw** (*f*)  
Decorator function to lazy load a *Gebouw*.

crabpy.gateway.crab.**check\_lazy\_load\_gemeente** (*f*)  
Decorator function to lazy load a *Gemeente*.

crabpy.gateway.crab.**check\_lazy\_load\_gewest** (*f*)  
Decorator function to lazy load a *Gewest*.

crabpy.gateway.crab.**check\_lazy\_load\_huisnummer** (*f*)  
Decorator function to lazy load a *Huisnummer*.

crabpy.gateway.crab.**check\_lazy\_load\_perceel** (*f*)  
Decorator function to lazy load a *Perceel*.

crabpy.gateway.crab.**check\_lazy\_load\_straat** (*f*)  
Decorator function to lazy load a *Straat*.

crabpy.gateway.crab.**check\_lazy\_load\_subadres** (*f*)  
Decorator function to lazy load a *Subadres*.

crabpy.gateway.crab.**check\_lazy\_load\_terreinobject** (*f*)  
Decorator function to lazy load a *Terreinobject*.

crabpy.gateway.crab.**check\_lazy\_load\_wegobject** (*f*)  
Decorator function to lazy load a *Wegobject*.

crabpy.gateway.crab.**check\_lazy\_load\_wegsegment** (*f*)  
Decorator function to lazy load a *Wegsegment*.

`crabpy.gateway.crab.crab_gateway_request` (*client, method, \*args*)

Utility function that helps making requests to the CRAB service.

This is a specialised version of `crabpy.client.crab_request()` that allows adding extra functionality for the calls made by the gateway.

#### Parameters

- **client** – A `suds.client.Client` for the CRAB service.
- **action** (*string*) – Which method to call, eg. *ListGewesten*

**Returns** Result of the SOAP call.

## 3.3 Capakey gateway module

This module contains an opinionated gateway for the capakey webservice.

New in version 0.2.0.

**class** `crabpy.gateway.capakey.Afdeling` (*id, naam=None, gemeente=None, centroid=None, bounding\_box=None, shape=None, \*\*kwargs*)

A Cadastral Division of a *Gemeente*.

**clear\_gateway** ()

Clear the currently set CapakeyGateway.

**set\_gateway** (*gateway*)

**Parameters** **gateway** (`crabpy.gateway.capakey.CapakeyGateway`) – Gateway to use.

**class** `crabpy.gateway.capakey.CapakeyRestGateway` (*\*\*kwargs*)

A REST gateway to the capakey webservice.

New in version 0.8.0.

**get\_gemeente\_by\_id** (*id*)

Retrieve a *gemeente* by id (the NIScode).

**Return type** *Gemeente*

**get\_kadastrale\_afdeling\_by\_id** (*aid*)

Retrieve a 'kadastrale afdeling' by id.

**Parameters** **aid** – An id of a *kadastrale afdeling*.

**Return type** A *Afdeling*.

**get\_perceel\_by\_capakey** (*capakey*)

Get a *perceel*.

**Parameters** **capakey** – An capakey for a *perceel*.

**Return type** *Perceel*

**get\_perceel\_by\_coordinates** (*x, y*)

Get a *perceel*.

**Parameters** **capakey** – An capakey for a *perceel*.

**Return type** *Perceel*

**get\_perceel\_by\_id\_and\_sectie** (*id*, *sectie*)

Get a *perceel*.

**Parameters**

- **id** – An id for a *perceel*.
- **sectie** – The *Sectie* that contains the *perceel*.

**Return type** *Perceel*

**get\_perceel\_by\_percid** (*percid*)

Get a *perceel*.

**Parameters** **percid** – A percid for a *perceel*.

**Return type** *Perceel*

**get\_sectie\_by\_id\_and\_afdeling** (*id*, *afdeling*)

Get a *sectie*.

**Parameters**

- **id** – An id of a *sectie*. eg. “A”
- **afdeling** – The *Afdeling* for in which the *sectie* can be found. Can also be the id of and *afdeling*.

**Return type** A *Sectie*.

**list\_gemeenten** (*sort=1*)

List all *gemeenten* in Vlaanderen.

**Parameters** **sort** (*integer*) – What field to sort on.

**Return type** A list of *Gemeente*.

**list\_kadastrale\_afdelingen** ()

List all *kadastrale afdelingen* in Flanders.

**Parameters** **sort** (*integer*) – Field to sort on.

**Return type** A list of *Afdeling*.

**list\_kadastrale\_afdelingen\_by\_gemeente** (*gemeente*, *sort=1*)

List all *kadastrale afdelingen* in a *gemeente*.

**Parameters**

- **gemeente** – The *Gemeente* for which the *afdelingen* are wanted.
- **sort** (*integer*) – Field to sort on.

**Return type** A list of *Afdeling*.

**list\_percelen\_by\_sectie** (*sectie*)

List all *percelen* in a *sectie*.

**Parameters**

- **sectie** – The *Sectie* for which the *percelen* are wanted.
- **sort** (*integer*) – Field to sort on.

**Return type** A list of *Perceel*.

**list\_secties\_by\_afdeling** (*afdeling*)

List all *secties* in a *kadastrale afdeling*.

**Parameters** *afdeling* – The *Afdeling* for which the *secties* are wanted. Can also be the id of and *afdeling*.

**Return type** A list of *Sectie*.

**class** `crabpy.gateway.capakey.GatewayObject` (*\*\*kwargs*)

Abstract class for all objects being returned from the Gateway.

**check\_gateway** ()

Check to see if a gateway was set on this object.

**clear\_gateway** ()

Clear the currently set CapakeyGateway.

**gateway** = None

The `crabpy.gateway.capakey.CapakeyGateway` to use when making further calls to the Capakey service.

**set\_gateway** (*gateway*)

**Parameters** *gateway* (`crabpy.gateway.capakey.CapakeyGateway`) – Gateway to use.

**class** `crabpy.gateway.capakey.Gemeente` (*id*, *naam=None*, *centroid=None*, *bounding\_box=None*, *shape=None*, *\*\*kwargs*)

The smallest administrative unit in Belgium.

**class** `crabpy.gateway.capakey.Perceel` (*id*, *sectie*, *capakey*, *percid*, *adres=None*, *capatype=None*, *cashkey=None*, *centroid=None*, *bounding\_box=None*, *shape=None*, *\*\*kwargs*)

A Cadastral Parcel.

**clear\_gateway** ()

Clear the currently set CapakeyGateway.

**set\_gateway** (*gateway*)

**Parameters** *gateway* (`crabpy.gateway.capakey.CapakeyGateway`) – Gateway to use.

**class** `crabpy.gateway.capakey.Sectie` (*id*, *afdeling*, *centroid=None*, *bounding\_box=None*, *shape=None*, *\*\*kwargs*)

A subdivision of a *Afdeling*.

**clear\_gateway** ()

Clear the currently set CapakeyGateway.

**set\_gateway** (*gateway*)

**Parameters** *gateway* (`crabpy.gateway.capakey.CapakeyGateway`) – Gateway to use.

`crabpy.gateway.capakey.capakey_rest_gateway_request` (*url*, *headers={}*, *params={}*)

Utility function that helps making requests to the CAPAKEY REST service.

**Parameters**

- **url** (*string*) – URL to request.
- **headers** (*dict*) – Headers to send with the URL.
- **params** (*dict*) – Parameters to send with the URL.

**Returns** Result of the call.

`crabpy.gateway.capakey.check_lazy_load_afdeling(f)`

Decorator function to lazy load a *Afdeling*.

`crabpy.gateway.capakey.check_lazy_load_gemeente(f)`

Decorator function to lazy load a *Gemeente*.

`crabpy.gateway.capakey.check_lazy_load_perceel(f)`

Decorator function to lazy load a *Perceel*.

`crabpy.gateway.capakey.check_lazy_load_sectie(f)`

Decorator function to lazy load a *Sectie*.

## 3.4 Gateway exception module

This module contains custom errors that can be generated by gateways.

New in version 0.2.0.

**exception** `crabpy.gateway.exception.GatewayAuthenticationException` (*message*,  
*soapfault*)

An exception that signifies something went wrong during authentication.

**exception** `crabpy.gateway.exception.GatewayException` (*message*)

A base exception.

**exception** `crabpy.gateway.exception.GatewayResourceNotFoundException`

An exception that signifies that no results were found.

**exception** `crabpy.gateway.exception.GatewayRuntimeException` (*message*, *soapfault*)

An exception that signifies a soap request went wrong.

**soapfault** = `None`

The soapfault that was generated by the service.

## 3.5 Wsa module

This module contains utility functions for using WSA with SOAP services.

New in version 0.1.0.

**class** `crabpy.wsa.Action` (*action*)

Assist in rendering a WSA:Action element.

**class** `crabpy.wsa.MessageID`

Assist in rendering a WSA:MessageID element.

**class** `crabpy.wsa.To` (*location*)

Assist in rendering a WSA:To element.

## 3.6 Wsse module

This module adds a *UsernameDigestToken* for use with SOAP services.

New in version 0.2.0.

**class** `crabpy.wsse.UsernameDigestToken` (*username=None*, *password=None*)

Represents a basic WS-Security token with password digest

**setnonce** (*text=None*)

Set I{nonce} which is arbitrary set of bytes to prevent replay attacks. @param text: The nonce text value.

Generated when I{None}.

@type text: str

**xml** ()

Get xml representation of the object. @return: The root node. @rtype: L{Element}





#### **4.1 1.2.4 (23-05-2023)**

- Foute lijst van deelgemeenten gebruikt (#225)

#### **4.2 1.2.3 (15-05-2023)**

- Limiet van list op 1000 items (#221)

#### **4.3 1.2.2 (09-05-2023)**

- get percelen timeout (#217)

#### **4.4 1.2.1 (04-05-2023)**

- Adressenregister bugfixes

#### **4.5 1.2.0 (04-05-2023)**

- uniformiseer alles naar niscodes en check dat alle niscodes strings zijn (#208)
- errors checken en fine tunen (#209)

## 4.6 1.1.0 (03-05-2023)

- Lijst gemeenten statisch maken (#203)

## 4.7 1.0.0 (13-04-2023)

- Naar v2 van adressenregister (#185)
- File wordt niet geclosed (#133)
- black, flake8, python2 weg en pre-commit (#195)
- Uitbreidingen aan adressenregister model (#192)
- pip install van crabpy installeert ook een “tests” met alle data van crabpy (#165)

## 4.8 0.16.1 (29-03-2023)

- Add user agent header to geo.api calls (#190)

## 4.9 0.16.0 (29-03-2023)

- Overschakelen naar nieuwe AGIV services (#183)

## 4.10 0.15.0 (10-12-2021)

- Methode list\_straten kan meer data laden (#172)

## 4.11 0.14.0 (18-11-2021)

- Bounding box geeft strings ipv floats (#168)

## 4.12 0.13.0 (14-09-2021)

- Verwijderen python 2 support (#160)
- Vervangen suds-jurko door suds-py (#160)
- Upgraden requirements

## 4.13 0.12.1 (28-01-2021)

- Adres toegevoegd aan Perceel (#147)

## 4.14 0.12.0 (24-06-2019)

- Switchen naar v2 van de capakey REST-API (#120)
- Mocken van calls naar externe url's bij testen (#118)
- get\_perceel\_by\_coordinates (#121)

## 4.15 0.11.0 (03-01-2019)

- Update deelgemeenten (#110, #116)
- Fix travis tests (#112)
- Update dependencies

## 4.16 0.10.0 (17-07-2018)

- Capakey service: change source base map (#95)
- Capakey service: return full geometry (#96)

## 4.17 0.9.0 (20-03-2018)

- Remove the deprecated CapakeyGateway (#92)

## 4.18 0.8.3 (07-12-2017)

- Fix some unit tests.
- Update some dependencies
- Configure pyup

## 4.19 0.8.2 (25-08-2017)

- Add zope.deprecation to setup.py #76
- Upgrade capakey rest service #78

## 4.20 0.8.1 (20-04-2017)

- Updated center and bounding box format in responses of the CapakeyRestGateway in accordance with the CapakeyGateway (#73).

## 4.21 0.8.0 (19-04-2017)

- Added a CapakeyRestGateway that uses the new Capakey REST service provided by Informatie Vlaanderen. (#45, #53)
- Deprecate Capakey SOAP gateway (#69)
- Fix a bug with list\_huisnummers\_by\_perceel. (#67)
- Dropped support for Python 3.3 and added support for Python 3.6.

## 4.22 0.7.0 (25-01-2016)

- Add official support for python 3.5
- Implement list\_huisnummers\_by\_perceel. (#56)
- Implement get\_postadres\_by\_huisnummer and get\_postadres\_by\_subadres. (#57)
- Add a property Perceel.postadressen to get the postadressen for a certain Perceel. (#58)
- Implement a Deelgemeente object and list\_deelgemeenten, list\_deelgemeenten\_by\_gemeente and get\_deelgemeente\_by\_id. (#63)

## 4.23 0.6.0 (01-06-2015)

- Implement operations dealing with Adrespositie. (#37) [TalissaJoly]
- Improve the coverage. (#39) [TalissaJoly]
- Fix a bug with objects that have an empty bounding box. (#46) [TalissaJoly]
- Better handling of unexisting objects. (#49) [TalissaJoly]
- Switch tests to py.test. (#19) [TalissaJoly]

## 4.24 0.5.0 (03-03-2015)

- Implement operations dealing with Subadres. This deals with things like postboxes in apartment complexes. (#34) (#40) [TalissaJoly]
- Drop support for python 3.2 (#36)
- Fix a bug with crab.list\_aardsubadressen. (#38)

## 4.25 0.4.2 (18-09-2014)

- Fix an issue with CRAB Gateway list operations that contain no results. Previously these triggered an error, now they return an empty list. (#33)
- Clean up CHANGES.rst so it works on pypi again.

## 4.26 0.4.1 (05-09-2014)

- Fix an issues with pickling in `list_gemeente_by_provincie`.
- Removed the sort parameter from `list_gemeenten_by_provincie` since it didn't work anyway.

## 4.27 0.4.0 (03-09-2014)

- Added a bounding box to a CRAB Straat. (#26)
- Added a bounding box to a CRAB Huisnummer. (#27)
- Added a Provincie object. (#31)

## 4.28 0.3.5 (02-09-2014)

- Fix hardcoded url in `client.py`. (#25)

## 4.29 0.3.4 (07-05-2014)

- Optimise lazy loading of capakey Gemeente. (#21)
- Optimise lazy loading of capakey Afdeling. (#22)
- General lazy loading optimisations.
- Some slight changes to CRAB lazy loading. (#24)

## 4.30 0.3.3 (02-05-2014)

- Added some debugging calls to allow an implementing application to track what calls are being made.

## 4.31 0.3.2 (07-04-2014)

- A *Gebouw* loaded through the `crabpy.gateway.crab.CrabGateway.get_gebouw_by_id` was not passed a `crabpy.gateway.crab.CrabGateway`. (#15)
- Always load a full `crabpy.gateway.crab.Metadata` object when returning from a `get*_by_id` method. (#13)
- Add a `wegobjecten` property to a `crabpy.gateway.crab.Straat`. (#17)
- Add a `wegsegmenten` property to a `crabpy.gateway.crab.Straat`. (#18)
- Added support for [Coveralls](#). (#16)

## 4.32 0.3.1 (17-03-2014)

- Fixed a bug with lazy loading a Perceel's capatype or cashkey. (#8)
- Removes duplicates from a list of gemeentes as returned by CRAB. (#10)
- Removed loading a Gemeente with an afdeling to speed up certain queries. (#7)
- Removed a few unneeded requests in the capakey gateway when working with Gemeente.id or Afdeling.id.
- Fixed printing of objects through the `__str__` method on python 2.7. (#9)
- Adapted examples for python 3 print. (#11)

## 4.33 0.3.0 (12-03-2014)

- Added a *Gateway* `<crabpy.gateway.crab.CrabGateway>` for the Crab webservice.
- Added caching to the Crab Gateway using [Dogpile](#)

## 4.34 0.2.1 (21-02-2014)

- Document how to connect to the services through a proxy.
- Fix an incomplete release.

## 4.35 0.2.0 (03-12-2013)

- Added a *Gateway* `<crabpy.gateway.capakey.CapakeyGateway>` for the Capakey webservice.
- Added caching to the Capakey Gateway using [Dogpile](#)
- Better test coverage. Ability to skip integration tests.
- Added some documentation.
- Removed a dependency for resolving UsernameDigestTokens. This in term removed the original suds from the dependency chain.
- Due to removing those dependencies, compatibility with Python 3.2 and 3.3 is now present.

## 4.36 0.1.0 (25-10-2013)

- Initial release
- A working client for the [CRAB webservice](#).
- A working client for the [CapaKey webservice](#).

## CHAPTER 5

---

### Indices and tables

---

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





### C

- `crabpy.client`, [13](#)
- `crabpy.gateway.capakey`, [23](#)
- `crabpy.gateway.crab`, [14](#)
- `crabpy.gateway.exception`, [26](#)
- `crabpy.wsa`, [26](#)
- `crabpy.wsse`, [26](#)



## A

Aardadres (*class in crabpy.gateway.crab*), 14  
Aardgebouw (*class in crabpy.gateway.crab*), 14  
Aardsubadres (*class in crabpy.gateway.crab*), 14  
Aardterreinobject (*class in crabpy.gateway.crab*), 14  
Aardwegobject (*class in crabpy.gateway.crab*), 14  
Action (*class in crabpy.wsa*), 26  
Adrespositie (*class in crabpy.gateway.crab*), 14  
AdressenRegisterClientException, 13  
Afdeling (*class in crabpy.gateway.capakey*), 23

## B

Bewerking (*class in crabpy.gateway.crab*), 14

## C

capakey\_rest\_gateway\_request() (*in module crabpy.gateway.capakey*), 25  
CapakeyRestGateway (*class in crabpy.gateway.capakey*), 23  
check\_gateway() (*crabpy.gateway.capakey.GatewayObject method*), 25  
check\_gateway() (*crabpy.gateway.crab.GatewayObject method*), 20  
check\_lazy\_load\_adrespositie() (*in module crabpy.gateway.crab*), 22  
check\_lazy\_load\_afdeling() (*in module crabpy.gateway.capakey*), 25  
check\_lazy\_load\_gebouw() (*in module crabpy.gateway.crab*), 22  
check\_lazy\_load\_gemeente() (*in module crabpy.gateway.capakey*), 26  
check\_lazy\_load\_gemeente() (*in module crabpy.gateway.crab*), 22  
check\_lazy\_load\_gewest() (*in module crabpy.gateway.crab*), 22  
check\_lazy\_load\_huisnummer() (*in module crabpy.gateway.crab*), 22  
check\_lazy\_load\_perceel() (*in module crabpy.gateway.capakey*), 26  
check\_lazy\_load\_perceel() (*in module crabpy.gateway.crab*), 22  
check\_lazy\_load\_sectie() (*in module crabpy.gateway.capakey*), 26  
check\_lazy\_load\_straat() (*in module crabpy.gateway.crab*), 22  
check\_lazy\_load\_subadres() (*in module crabpy.gateway.crab*), 22  
check\_lazy\_load\_terreinobject() (*in module crabpy.gateway.crab*), 22  
check\_lazy\_load\_wegobject() (*in module crabpy.gateway.crab*), 22  
check\_lazy\_load\_wegsegment() (*in module crabpy.gateway.crab*), 22  
clear\_gateway() (*crabpy.gateway.capakey.Afdeling method*), 23  
clear\_gateway() (*crabpy.gateway.capakey.GatewayObject method*), 25  
clear\_gateway() (*crabpy.gateway.capakey.Perceel method*), 25  
clear\_gateway() (*crabpy.gateway.capakey.Sectie method*), 25  
clear\_gateway() (*crabpy.gateway.crab.Deelgemeente method*), 20  
clear\_gateway() (*crabpy.gateway.crab.GatewayObject method*), 20  
clear\_gateway() (*crabpy.gateway.crab.Gemeente method*), 20  
clear\_gateway() (*crabpy.gateway.crab.Provincie method*), 21  
Codelijst (*class in crabpy.gateway.crab*), 14  
crab\_factory() (*in module crabpy.client*), 13  
crab\_gateway\_request() (*in module crabpy.gateway.crab*), 22  
crab\_request() (*in module crabpy.client*), 13  
CrabGateway (*class in crabpy.gateway.crab*), 14  
crabpy.client (*module*), 13  
crabpy.gateway.capakey (*module*), 23

`crabpy.gateway.crab` (module), 14  
`crabpy.gateway.exception` (module), 26  
`crabpy.wsa` (module), 26  
`crabpy.wsse` (module), 26

## D

`Deelgemeente` (class in `crabpy.gateway.crab`), 20

## G

`gateway` (`crabpy.gateway.capakey.GatewayObject` attribute), 25  
`gateway` (`crabpy.gateway.crab.GatewayObject` attribute), 20  
`GatewayAuthenticationException`, 26  
`GatewayException`, 26  
`GatewayObject` (class in `crabpy.gateway.capakey`), 25  
`GatewayObject` (class in `crabpy.gateway.crab`), 20  
`GatewayResourceNotFoundException`, 26  
`GatewayRuntimeException`, 26  
`Gebouw` (class in `crabpy.gateway.crab`), 20  
`Gemeente` (class in `crabpy.gateway.capakey`), 25  
`Gemeente` (class in `crabpy.gateway.crab`), 20  
`Geometriemethodegebouw` (class in `crabpy.gateway.crab`), 20  
`Geometriemethodewegsegment` (class in `crabpy.gateway.crab`), 20  
`get_adrespositie_by_id()`  
(`crabpy.gateway.crab.CrabGateway` method), 14  
`get_deelgemeente_by_id()`  
(`crabpy.gateway.crab.CrabGateway` method), 14  
`get_gebouw_by_id()`  
(`crabpy.gateway.crab.CrabGateway` method), 14  
`get_gemeente_by_id()`  
(`crabpy.gateway.capakey.CapakeyRestGateway` method), 23  
`get_gemeente_by_id()`  
(`crabpy.gateway.crab.CrabGateway` method), 14  
`get_gemeente_by_niscode()`  
(`crabpy.gateway.crab.CrabGateway` method), 14  
`get_gewest_by_id()`  
(`crabpy.gateway.crab.CrabGateway` method), 15  
`get_huisnummer_by_id()`  
(`crabpy.gateway.crab.CrabGateway` method), 15  
`get_huisnummer_by_nummer_and_straat()`  
(`crabpy.gateway.crab.CrabGateway` method), 15

`get_kadastrale_afdeling_by_id()`  
(`crabpy.gateway.capakey.CapakeyRestGateway` method), 23  
`get_perceel_by_capakey()`  
(`crabpy.gateway.capakey.CapakeyRestGateway` method), 23  
`get_perceel_by_coordinates()`  
(`crabpy.gateway.capakey.CapakeyRestGateway` method), 23  
`get_perceel_by_id()`  
(`crabpy.gateway.crab.CrabGateway` method), 15  
`get_perceel_by_id_and_sectie()`  
(`crabpy.gateway.capakey.CapakeyRestGateway` method), 23  
`get_perceel_by_percid()`  
(`crabpy.gateway.capakey.CapakeyRestGateway` method), 24  
`get_postadres_by_huisnummer()`  
(`crabpy.gateway.crab.CrabGateway` method), 15  
`get_postadres_by_subadres()`  
(`crabpy.gateway.crab.CrabGateway` method), 15  
`get_postkanton_by_huisnummer()`  
(`crabpy.gateway.crab.CrabGateway` method), 15  
`get_provincie_by_id()`  
(`crabpy.gateway.crab.CrabGateway` method), 15  
`get_sectie_by_id_and_afdeling()`  
(`crabpy.gateway.capakey.CapakeyRestGateway` method), 24  
`get_straat_by_id()`  
(`crabpy.gateway.crab.CrabGateway` method), 15  
`get_subadres_by_id()`  
(`crabpy.gateway.crab.CrabGateway` method), 16  
`get_terreinobject_by_id()`  
(`crabpy.gateway.crab.CrabGateway` method), 16  
`get_wegobject_by_id()`  
(`crabpy.gateway.crab.CrabGateway` method), 16  
`get_wegsegment_by_id()`  
(`crabpy.gateway.crab.CrabGateway` method), 16  
`Gewest` (class in `crabpy.gateway.crab`), 20

## H

`Herkomstadrespositie` (class in `crabpy.gateway.crab`), 20  
`Huisnummer` (class in `crabpy.gateway.crab`), 20

[huisnummers \(\*crabpy.gateway.crab.Perceel\* attribute\), 21](#)  
**L**  
[list\\_aardadressen\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 16](#)  
[list\\_aardgebouwen\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 16](#)  
[list\\_aardsubadressen\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 16](#)  
[list\\_aardterreinobjecten\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 16](#)  
[list\\_aardwegobjecten\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 16](#)  
[list\\_adresposities\\_by\\_huisnummer\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 16](#)  
[list\\_adresposities\\_by\\_nummer\\_and\\_straat\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 16](#)  
[list\\_adresposities\\_by\\_subadres\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 17](#)  
[list\\_adresposities\\_by\\_subadres\\_and\\_huisnummer\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 17](#)  
[list\\_bewerkingen\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 17](#)  
[list\\_deelgemeenten\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 17](#)  
[list\\_deelgemeenten\\_by\\_gemeente\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 17](#)  
[list\\_gebouwen\\_by\\_huisnummer\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 17](#)  
[list\\_gemeenten\(\) \(\*crabpy.gateway.capakey.CapakeyRestGateway\* method\), 24](#)  
[list\\_gemeenten\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 17](#)  
[list\\_gemeenten\\_by\\_provincie\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 18](#)  
[list\\_geometriemethodegebouwen\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 18](#)  
[list\\_geometriemethodewegsegmenten\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 18](#)  
[list\\_gewesten\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 18](#)  
[list\\_herkomstadresposities\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 18](#)  
[list\\_huisnummers\\_by\\_perceel\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 18](#)  
[list\\_huisnummers\\_by\\_straat\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 18](#)  
[list\\_kadastrale\\_afdelingen\(\) \(\*crabpy.gateway.capakey.CapakeyRestGateway\* method\), 24](#)  
[list\\_kadastrale\\_afdelingen\\_by\\_gemeente\(\) \(\*crabpy.gateway.capakey.CapakeyRestGateway\* method\), 24](#)  
[list\\_organisaties\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 18](#)  
[list\\_percelen\\_by\\_huisnummer\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 18](#)  
[list\\_percelen\\_by\\_sectie\(\) \(\*crabpy.gateway.capakey.CapakeyRestGateway\* method\), 24](#)  
[list\\_postkantons\\_by\\_gemeente\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 18](#)  
[list\\_provincies\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 18](#)  
[list\\_secties\\_by\\_afdeling\(\) \(\*crabpy.gateway.capakey.CapakeyRestGateway\* method\), 24](#)  
[list\\_statusgebouwen\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 19](#)  
[list\\_statushuisnummers\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 19](#)  
[list\\_statusstraatnamen\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 19](#)  
[list\\_statussubadressen\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 19](#)  
[list\\_statuswegsegmenten\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 19](#)  
[list\\_straten\(\) \(\*crabpy.gateway.crab.CrabGateway\* method\), 19](#)

`list_subadressen_by_huisnummer()`  
(*crabpy.gateway.crab.CrabGateway* method),  
19

`list_talen()` (*crabpy.gateway.crab.CrabGateway*  
method), 19

`list_terreinobjecten_by_huisnummer()`  
(*crabpy.gateway.crab.CrabGateway* method),  
19

`list_wegobjecten_by_straat()`  
(*crabpy.gateway.crab.CrabGateway* method),  
19

`list_wegsegmenten_by_straat()`  
(*crabpy.gateway.crab.CrabGateway* method),  
19

## M

`MessageID` (class in *crabpy.wsa*), 26

`Metadata` (class in *crabpy.gateway.crab*), 21

## O

`Organisatie` (class in *crabpy.gateway.crab*), 21

## P

`Perceel` (class in *crabpy.gateway.capakey*), 25

`Perceel` (class in *crabpy.gateway.crab*), 21

`postadressen` (*crabpy.gateway.crab.Perceel* attribute), 21

`Postkanton` (class in *crabpy.gateway.crab*), 21

`Provincie` (class in *crabpy.gateway.crab*), 21

## S

`Sectie` (class in *crabpy.gateway.capakey*), 25

`set_gateway()` (*crabpy.gateway.capakey.Afdeling*  
method), 23

`set_gateway()` (*crabpy.gateway.capakey.GatewayObject*  
method), 25

`set_gateway()` (*crabpy.gateway.capakey.Perceel*  
method), 25

`set_gateway()` (*crabpy.gateway.capakey.Sectie*  
method), 25

`set_gateway()` (*crabpy.gateway.crab.Deelgemeente*  
method), 20

`set_gateway()` (*crabpy.gateway.crab.GatewayObject*  
method), 20

`set_gateway()` (*crabpy.gateway.crab.Gemeente*  
method), 20

`set_gateway()` (*crabpy.gateway.crab.Provincie*  
method), 21

`setnonce()` (*crabpy.wsse.UsernameDigestToken*  
method), 26

`soapfault` (*crabpy.gateway.exception.GatewayRuntimeException*  
attribute), 26

`Statusgebouw` (class in *crabpy.gateway.crab*), 21

`Statushuisnummer` (class in *crabpy.gateway.crab*),  
21

`Statusstraatnaam` (class in *crabpy.gateway.crab*),  
21

`Statussubadres` (class in *crabpy.gateway.crab*), 21

`Statuswegsegment` (class in *crabpy.gateway.crab*),  
21

`Straat` (class in *crabpy.gateway.crab*), 22

`Subadres` (class in *crabpy.gateway.crab*), 22

## T

`Taal` (class in *crabpy.gateway.crab*), 22

`Terreinobject` (class in *crabpy.gateway.crab*), 22

`To` (class in *crabpy.wsa*), 26

## U

`UsernameDigestToken` (class in *crabpy.wsse*), 26

## W

`Wegobject` (class in *crabpy.gateway.crab*), 22

`Wegsegment` (class in *crabpy.gateway.crab*), 22

## X

`xml()` (*crabpy.wsse.UsernameDigestToken* method), 27