

REST Clients

- Python client: PyCellBase
 - Package notes
 - General usage
 - Installation
 - Cloning
 - PyPI
- R client
 - Note: R client library available from version 4 onwards

Python client: PyCellBase

- PyCellBase is a Python package that provides programmatic access to the comprehensive RESTful web service API that has been implemented for the [CellBase](#) database, providing an easy, lightweight, fast and intuitive access to it.
- This package can be used to access to relevant biological information in a user-friendly way without the need of local databases installations.
- Data is always available by a high-availability cluster and queries have been tuned to ensure a real-time performance.
- PyCellBase offers the convenience of an object-oriented scripting language and provides the ability to integrate the obtained results into other Python applications.
- More info about this package in the [Python client tutorial](#) section.

Package notes

- **PyCellBase** is compatible with both Python 2 and 3.
- This package makes use of multithreading to improve performance when the number of queries exceed a specific limit.
- It is distributed:
 - With the rest of the CellBase code at `{cellbase/clients/python}`.
 - Via *PyPI* - the Python Package Index: <https://pypi.org/project/pycellbase/>

General usage

PyCellBase code can be accessed at <https://github.com/opencb/cellbase/tree/develop/clients/python/pycellbase>.

The **CellBaseClient** class provides access to the **different clients** of the data we want to query (e.g. gene, transcript, variation, protein, genomic region, variant).

Each of these clients provide a **set of methods** to ask for the resources we want to retrieve. Most of these methods will need to be provided with **comma-separated IDs or list of IDs**. Optional filters and extra options can be added as key-value parameters.

Responses are retrieved as **JSON formatted data**. Therefore, fields can be queried by key.

If there is an available resource, but there is not an available method in this python package, the **CellBaseClient** class can be used to **create the URL of interest**. This class is able to access the RESTful Web Services through the *get* method it implements. In this case, this method needs to be provided with those parameters which are required by the URL: category (e.g. feature), subcategory (e.g. gene), ID to search for (e.g. BRCA1) and method to query (e.g. search).

Configuration data as host, API version, or species is stored in a ConfigClient object. A custom configuration can be passed to CellBaseClient with a ConfigClient object provided with a JSON or YAML config file. If you want to change the configuration on the fly you can directly modify the ConfigClient object.

Please, find more details on how to use the python library at: [Python client tutorial](#)

Installation

Cloning

PyCellBase can be cloned in your local machine by executing in your terminal:

```
$ git clone https://github.com/openCB/cellbase.git
```

Once you have downloaded the project you can install the library:

```
$ cd cellbase/clients/python  
$ python setup.py install
```

PyPI

PyCellBase is stored in PyPI and can be installed via pip:

```
$ pip install pycellbase
```

R client

Note: R client library available from version 4 onwards

R library package is implemented and maintained for the latest R release and distributed through Bioconductor (<https://bioconductor.org/packages/release/bioc/html/cellbaseR.html>). For a quick install, please enter the R terminal and type:

```
source("https://bioconductor.org/biocLite.R")  
biocLite("cellbaseR")
```

The R code is also distributed together with the rest of the CellBase code. R code can be found at: `cellbase/clients/R`

The R code provides a library for programmatic access to CellBase data. No CLI is implemented in R.

Comprehensive description of the library is provided by the Bioconductor documentation:

<https://bioconductor.org/packages/release/bioc/html/cellbaseR.html>