

---

# **Invenio-Client Documentation**

***Release 0.1.0***

**Invenio collaboration**

December 10, 2014



<b>1</b>	<b>Contents</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	Usage . . . . .	3
1.3	API . . . . .	3
1.4	Changes . . . . .	5
1.5	Contributing . . . . .	5
1.6	Authors . . . . .	6
	<b>Python Module Index</b>	<b>7</b>



Invenio-Client is a Python library permitting to connect to remote [Invenio](#) digital library instances.



---

## Contents

---

- Installation
- Usage
- API
- Changes
- Contributing
- Authors

## 1.1 Installation

Invenio-Client is on PyPI so all you need is:

```
$ pip install invenio-client
```

## 1.2 Usage

The easiest way is to use *invenio\_client* directly with *InvenioConnector*.

```
from invenio_client import InvenioConnector
demo = InvenioConnector("http://demo.invenio-software.org")

results = demo.search("higgs")

for record in results:
    print record["245__a"][0]
    print record["520__b"][0]
    for author in record["100__"]:
        print author["a"][0], author["u"][0]
```

## 1.3 API

Tool to connect to remote Invenio servers using Invenio APIs.

Example of use:

```

from invenio_client import InvenioConnector
demo = InvenioConnector("http://demo.invenio-software.org")

results = demo.search("higgs")

for record in results:
    print record["245__a"][0]
    print record["520__b"][0]
    for author in record["100__"]:
        print author["a"][0], author["u"][0]

```

FIXME:

- implement cache expiration
- exceptions handling
- parsing of `<!-- Search-Engine-Total-Number-Of-Results: N -->`
- better checking of input parameters

```

class invenio_client.connector.InvenioConnector(url, user='', password='', login_method='Local', insecure_login=False)

```

Create an connector to a server running Invenio.

**get\_record** (*recid*, *read\_cache=True*)  
Return the record with given recid.

**get\_records\_from\_basket** (*bskid*, *group\_basket=False*, *read\_cache=True*)  
Returns the records from the (public) basket with given bskid

**search** (*read\_cache=True*, *ssl\_verify=True*, *\*\*kwargs*)  
Returns records corresponding to the given search query.

See docstring of `invenio.legacy.search_engine.perform_request_search()` for an overview of available parameters.

**search\_records\_cited\_by** (*recid*)  
Return records cited by the given one.

**search\_similar\_records** (*recid*)  
Return the records similar to the given one.

**search\_with\_retry** (*sleeptime=3.0*, *retrycount=3*, *\*\*params*)  
Perform a search given a dictionary of `search(...)` parameters.  
It accounts for server timeouts as necessary and will retry some number of times.

#### Parameters

- **sleeptime** – number of seconds to sleep between retries
- **retrycount** – number of times to retry given search
- **params** – search parameters

**Returns** records in given format

**upload\_marcxml** (*marcxml*, *mode*)  
Upload a record to the server.

#### Parameters

- **marcxml** – the XML to upload.



- **mode** – the mode to use for the upload. - “-i” insert new records - “-r” replace existing records - “-c” correct fields of records - “-a” append fields to records - “-ir” insert record or replace if it exists

**exception** `invenio_client.connector.InvenioConnectorAuthError (value)`  
Failed authentication during remote connections.

**exception** `invenio_client.connector.InvenioConnectorError (value)`  
General connector error.

**exception** `invenio_client.connector.InvenioConnectorServerError (value)`  
Problem with connecting to Invenio server.

**class** `invenio_client.connector.Record (recid=None, marcxml=None, server_url=None)`  
Represent an Invenio record.

**export** (*of*=`'marcxml'`)  
Returns the record in chosen format

**url** ()  
Returns the URL to this record. Returns None if not known

**class** `invenio_client.connector.RecordsHandler (records)`  
MARCXML Parser

`invenio_client.connector.decompose_code (code)`  
Decompose a MARC “code” into tag, ind1, ind2, subcode.

CERN Document Server specific connector.

**class** `invenio_client.contrib.cds.CDSInvenioConnector (user='', password='')`

## 1.4 Changes

Version 0.1.0 (released 2014-12-10):

- Invenio-Client has been originally developed within the Invenio software package. This is first standalone public release.
- Migration from mechanize to splinter to support Python 3+.

Invenio-Client has been originally developed within the Invenio software package. Please consult Invenio release notes for prior history, or just see the git commit log that has been preserved in the standalone repository.

## 1.5 Contributing

Bug reports, feature requests, and other contributions are welcome. If you find a demonstrable problem that is caused by the code of this library, please:

1. Search for [already reported problems](#).
2. Check if the issue has been fixed or is still reproducible on the latest *master* branch.
3. Create an issue with **a test case**.

If you create a feature branch, you can run the tests to ensure everything is operating correctly:

```
$ python setup.py test

...

===== 146 passed in 1.90 seconds =====
```

## 1.6 Authors

Invenio-Client is developed to connect to remote [Invenio](#) digital library instances.

Contact us at [info@invenio-software.org](mailto:info@invenio-software.org).

- Adrian-Tudor Panescu <[adrian.tudor.panescu@cern.ch](mailto:adrian.tudor.panescu@cern.ch)>
- Alejandro Avilés <[aavilesd@cern.ch](mailto:aavilesd@cern.ch)>
- Alessio Deiana <[alessio.deiana@cern.ch](mailto:alessio.deiana@cern.ch)>
- Benoit Thiell <[bthiell@cfa.harvard.edu](mailto:bthiell@cfa.harvard.edu)>
- Esteban J. G. Gabancho <[esteban.gabancho@cern.ch](mailto:esteban.gabancho@cern.ch)>
- Graham R. Armstrong <[graham.richard.armstrong@cern.ch](mailto:graham.richard.armstrong@cern.ch)>
- Jan Aage Lavik <[jan.age.lavik@cern.ch](mailto:jan.age.lavik@cern.ch)>
- Jerome Caffaro <[jerome.caffaro@cern.ch](mailto:jerome.caffaro@cern.ch)>
- Jiri Kuncar <[jiri.kuncar@cern.ch](mailto:jiri.kuncar@cern.ch)>
- Krzysztof Lis <[krzysztof.lis@cern.ch](mailto:krzysztof.lis@cern.ch)>
- Laura Rueda <[laura.rueda@cern.ch](mailto:laura.rueda@cern.ch)>
- Peter Halliday <[phalliday@cornell.edu](mailto:phalliday@cornell.edu)>
- Samuele Kaplun <[samuele.kaplun@cern.ch](mailto:samuele.kaplun@cern.ch)>
- Thomas McCauley <[thomas.mccauley@cern.ch](mailto:thomas.mccauley@cern.ch)>
- Tibor Simko <[tibor.simko@cern.ch](mailto:tibor.simko@cern.ch)>

**i**

`invenio_client.connector`, 3  
`invenio_client.contrib.cds`, 5



**C**

CDSInvenioConnector (class in invenio\_client.contrib.cds), 5

**D**

decompose\_code() (in module invenio\_client.connector), 5

**E**

export() (invenio\_client.connector.Record method), 5

**G**

get\_record() (invenio\_client.connector.InvenioConnector method), 4

get\_records\_from\_basket() (invenio\_client.connector.InvenioConnector method), 4

**I**

invenio\_client.connector (module), 3

invenio\_client.contrib.cds (module), 5

InvenioConnector (class in invenio\_client.connector), 4

InvenioConnectorAuthError, 5

InvenioConnectorError, 5

InvenioConnectorServerError, 5

**R**

Record (class in invenio\_client.connector), 5

RecordsHandler (class in invenio\_client.connector), 5

**S**

search() (invenio\_client.connector.InvenioConnector method), 4

search\_records\_cited\_by() (invenio\_client.connector.InvenioConnector method), 4

search\_similar\_records() (invenio\_client.connector.InvenioConnector method), 4

search\_with\_retry() (invenio\_client.connector.InvenioConnector method), 4

**U**

upload\_marxml() (invenio\_client.connector.InvenioConnector method), 4

url() (invenio\_client.connector.Record method), 5