1 Contents
   1.1 Installation ................................................. 3
   1.2 Usage .................................................... 3
   1.3 API ....................................................... 3
   1.4 Changes .................................................... 5
   1.5 Contributing .............................................. 5
   1.6 Authors ................................................... 6

Python Module Index .............................................. 7
Invenio-Client is a Python library permitting to connect to remote Invenio digital library instances.
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, **kwparams)
    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

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

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

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

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

```python
def export(af='marcxml')
    Returns the record in chosen format
```

```python
def url()
    Returns the URL to this record. Returns None if not known
```

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

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

CERN Document Server specific connector.

```python
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.

- Adrian-Tudor Panescu <adrian.tudor.panescu@cern.ch>
- Alejandro Avilés <aavilesd@cern.ch>
- Alessio Deiana <alessio.deiana@cern.ch>
- Benoit Thiell <bthiell@cfa.harvard.edu>
- Esteban J. G. Gabancho <esteban.gabancho@cern.ch>
- Graham R. Armstrong <graham.richard.armstrong@cern.ch>
- Jan Aage Lavik <jan.age.lavik@cern.ch>
- Jerome Caffaro <jerome.caffaro@cern.ch>
- Jiri Kuncar <jiri.kuncar@cern.ch>
- Krzysztof Lis <krzysztof.lis@cern.ch>
- Laura Rueda <laura.rueda@cern.ch>
- Peter Halliday <phalliday@cornell.edu>
- Samuele Kaplun <samuele.kaplun@cern.ch>
- Thomas McCauley <thomas.mccauley@cern.ch>
- Tibor Simko <tibor.simko@cern.ch>
invenio_client.connector, 3
invenio_client.contrib.cds, 5
<table>
<thead>
<tr>
<th>C</th>
<th>CDSInvenioConnector (class in invenio_client.contrib.cds), 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>decompose_code() (in module invenio_client.connector), 5</td>
</tr>
<tr>
<td>E</td>
<td>export() (invenio_client.connector.Record method), 5</td>
</tr>
<tr>
<td>G</td>
<td>get_record() (invenio_client.connector.InvenioConnector method), 4</td>
</tr>
<tr>
<td></td>
<td>get_records_from_basket() (invenio_client.connector.InvenioConnector method), 4</td>
</tr>
<tr>
<td>I</td>
<td>invenio_client.connector (module), 3</td>
</tr>
<tr>
<td></td>
<td>invenio_client.contrib.cds (module), 5</td>
</tr>
<tr>
<td></td>
<td>InvenioConnector (class in invenio_client.connector), 4</td>
</tr>
<tr>
<td></td>
<td>InvenioConnectorAuthError, 5</td>
</tr>
<tr>
<td></td>
<td>InvenioConnectorError, 5</td>
</tr>
<tr>
<td></td>
<td>InvenioConnectorServerError, 5</td>
</tr>
<tr>
<td>R</td>
<td>Record (class in invenio_client.connector), 5</td>
</tr>
<tr>
<td></td>
<td>RecordsHandler (class in invenio_client.connector), 5</td>
</tr>
<tr>
<td>S</td>
<td>search() (invenio_client.connector.InvenioConnector method), 4</td>
</tr>
<tr>
<td></td>
<td>search_records_cited_by() (invenio_client.connector.InvenioConnector method), 4</td>
</tr>
<tr>
<td></td>
<td>search_similar_records() (invenio_client.connector.InvenioConnector method), 4</td>
</tr>
<tr>
<td>U</td>
<td>upload_marcxml() (invenio_client.connector.InvenioConnector method), 4</td>
</tr>
<tr>
<td></td>
<td>url() (invenio_client.connector.Record method), 5</td>
</tr>
</tbody>
</table>