
Invenio-Index-Migrator Documentation

Release 1.0.0

CERN

Oct 25, 2019

Contents

1	User's Guide	3
1.1	Installation	3
1.2	Configuration	3
1.3	Usage	4
1.4	Example application	4
2	API Reference	7
2.1	API Docs	7
3	Additional Notes	11
3.1	Contributing	11
3.2	Changes	13
3.3	License	13
3.4	Contributors	13
	Python Module Index	15

Elasticsearch index migrator for Invenio.

Further documentation is available at <https://invenio-index-migrator.readthedocs.io/>.

This part of the documentation will show you how to get started in using Invenio-Index-Migrator.

1.1 Installation

Invenio-Index-Migrator is on PyPI. When you install `invenio-index-migrator` you must specify the appropriate extras dependency for the version of Elasticsearch you use:

```
$ # For Elasticsearch 2.x:
$ pip install invenio-index-migrator[elasticsearch2]

$ # For Elasticsearch 5.x:
$ pip install invenio-index-migrator[elasticsearch5]

$ # For Elasticsearch 6.x:
$ pip install invenio-index-migrator[elasticsearch6]
```

1.2 Configuration

Invenio module for information retrieval.

```
invenio_index_migrator.config.INDEX_MIGRATOR_RECIPES = {}
    Index sync job definitions.
```

Example:

```
INDEX_MIGRATOR_RECIPES = dict(
    records=dict(
        cls='invenio_index_migrator.api.Migration',
        params=dict(
            strategy='cross_cluster_strategy',
            src_es_client=dict(
```

```
prefix='',
version=2,
params=dict(
    host='es2',
    port=9200,
    use_ssl=True,
    http_auth='user:pass',
    url_prefix='on-demand',
),
),
jobs=dict(
    records_simple_reindex=dict(
        cls='invenio_index_migrator.api.ReindexJob',
        pid_type='recid',
        index='records-record-v1.0.0',
        rollover_threshold=10,
        reindex_params=dict(
            script=dict(
                source="if (ctx._source.foo == 'bar') {...}",
                lang='painless'
            ),
            source=dict(
                sort=dict(
                    date='desc'
                )
            ),
            dest=dict(
                op_type='create'
            ),
        ),
    ),
)
)
```

1.3 Usage

Index syncing module.

1.4 Example application

Run the ElasticSearch and Redis server.

Run example development setup:

```
$ pip install -e .[all]
$ cd examples
$ ./app-setup.sh
$ ./app-fixtures.sh
```

Run example development server:


```
$ FLASK_DEBUG=1 FLASK_APP=app.py flask run -p 5000
```

Try to perform some search queries:

```
$ curl http://localhost:5000/?q=body:test
```

To be able to uninstall the example app:

```
$ ./app-teardown.sh
```


If you are looking for information on a specific function, class or method, this part of the documentation is for you.

2.1 API Docs

Index syncing module.

class `invenio_index_migrator.api.Job` (*name, migration, config*)

Index migration job.

Initialize a migration job.

Parameters

- **name** – job’s name.
- **migration** – an `invenio_index_migrator.api.migration.Migration` object.
- **config** – job’s configuration.

cancel ()

Cancel the job.

create_index (*index*)

Create indexes needed for the job.

document_name

Get the document name for the job.

rollover_actions ()

Rollover actions.

run ()

Run the job.

src_es_client

Get the source ES client.

```

status ()
    Return the status of the job.

class invenio_index_migrator.api.Migration (name, **config)
    Index migration base class.

cancel ()
    Cancel migration and all its jobs.

classmethod create_from_config (recipe_name, **recipe_config)
    Create Migration instance from config.

classmethod create_from_state (recipe_name, **recipe_config)
    Create Migration instance from ES state.

create_index ()
    Create Elasticsearch index for the migration.

init (dry_run=False)
    Initialize the index with recipe and jobs documents.

load_jobs_from_config ()
    Load jobs from config.

notify ()
    Notify when rollover is possible.

    Override this to notify the user whenever the threshold is reached and a rollover is possible.

rollover (force=False)
    Perform a rollover action.

run ()
    Run the index sync job.

status ()
    Get status for index sync job.

strategy
    Return migration strategy.

class invenio_index_migrator.api.MultiIndicesReindexJob (name, migration, config)
    Reindex job that uses Elasticsearch's reindex API.

    Initialize a migration job.

    Parameters

    • name – job's name.
    • migration – an invenio_index_migrator.api.migration.Migration object.
    • config – job's configuration.

create_index (index)
    Create templates.

initial_state (dry_run=False)
    Build job's initial state.

rollover_actions ()
    Rollover actions.

run ()
    Fetch source index using ES Reindex API.

```

class `invenio_index_migrator.api.ReindexJob(name, migration, config)`

Reindex job that uses Elasticsearch's reindex API.

Initialize a migration job.

Parameters

- **name** – job's name.
- **migration** – an `invenio_index_migrator.api.migration.Migration` object.
- **config** – job's configuration.

cancel()

Cancel reindexing job.

initial_state(dry_run=False)

Build job's initial state.

run()

Fetch source index using ES Reindex API.

class `invenio_index_migrator.api.ReindexAndSyncJob(name, migration, config)`

Job that both reindexes with ES reindex API and syncs with the DB.

The first run will use the reindex API and the subsequent runs will fetch from the database and sync the data.

Initialize a migration job.

Parameters

- **name** – job's name.
- **migration** – an `invenio_index_migrator.api.migration.Migration` object.
- **config** – job's configuration.

cancel()

Cancel reindexing and syncing job.

iter_indexer_ops(start_date=None, end_date=None)

Iterate over documents that need to be reindexed.

run()

Run reindexing and syncing job.

run_delta_job()

Calculate delta from DB changes since the last update.

2.1.1 Utilities

Utility functions for index migration.

class `invenio_index_migrator.utils.ESClient(es_config)`

ES client for sync jobs.

.

client

Return ES client.

reindex_auth

Return username and password for reindex HTTP authentication.

reindex_remote

Return ES client reindex API host.

class `invenio_index_migrator.utils.State(index, document_id, client=None)`

Migration ES state.

The state is stored in ElasticSearch and can be accessed similarly to a python dictionary.

Synchronization job state in ElasticSearch.

commit (*state*)

Save the state to ElasticSearch.

create (*initial_state*, *force=False*)

Create state document.

read ()

Fetch the current state from Elasticsearch.

`invenio_index_migrator.utils.extract_doctype_from_mapping(mapping_fp)`

Extract the doc_type from mapping filepath.

`invenio_index_migrator.utils.get_queue_size(queue)`

Get the queue size.

`invenio_index_migrator.utils.obj_or_import_string(value, default=None)`

Import string or return object.

Params value Import path or class object to instantiate.

Params default Default object to return if the import fails.

Returns The imported object.

Notes on how to contribute, legal information and changes are here for the interested.

3.1 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

3.1.1 Types of Contributions

Report Bugs

Report bugs at <https://github.com/inveniosoftware/invenio-index-migrator/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

Write Documentation

Invenio-Index-Migrator could always use more documentation, whether as part of the official Invenio-Index-Migrator docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/inveniosoftware/invenio-index-migrator/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

3.1.2 Get Started!

Ready to contribute? Here's how to set up *invenio-index-migrator* for local development.

1. Fork the *inveniosoftware/invenio-index-migrator* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/invenio-index-migrator.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv invenio-index-migrator
$ cd invenio-index-migrator/
$ pip install -e .[all]
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass tests:

```
$ ./run-tests.sh
```

The tests will provide you with test coverage and also check PEP8 (code style), PEP257 (documentation), flake8 as well as build the Sphinx documentation and run doctests.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -s
  -m "component: title without verbs"
  -m "* NEW Adds your new feature."
  -m "* FIX Fixes an existing issue."
  -m "* BETTER Improves and existing feature."
  -m "* Changes something that should not be visible in release notes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

3.1.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests and must not decrease test coverage.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring.
3. The pull request should work for Python 2.7, 3.3, 3.4 and 3.5. Check https://travis-ci.org/inveniosoftware/invenio-index-migrator/pull_requests and make sure that the tests pass for all supported Python versions.

3.2 Changes

Version 1.0.0 (released 2019-05-21)

- Initial public release.

3.3 License

MIT License

Copyright (C) 2015-2019 CERN.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Note: In applying this license, CERN does not waive the privileges and immunities granted to it by virtue of its status as an Intergovernmental Organization or submit itself to any jurisdiction.

3.4 Contributors

- Alexander Ioannidis
- Niklas Persson
- Zacharias Zacharodimos

i

- `invenio_index_migrator`, 4
- `invenio_index_migrator.api`, 7
- `invenio_index_migrator.config`, 3
- `invenio_index_migrator.utils`, 9

C

cancel() (invenio_index_migrator.api.Job method), 7
cancel() (invenio_index_migrator.api.Migration method), 8
cancel() (invenio_index_migrator.api.ReindexAndSyncJob method), 9
cancel() (invenio_index_migrator.api.ReindexJob method), 9
client (invenio_index_migrator.utils.ESClient attribute), 9
commit() (invenio_index_migrator.utils.State method), 10
create() (invenio_index_migrator.utils.State method), 10
create_from_config() (invenio_index_migrator.api.Migration class method), 8
create_from_state() (invenio_index_migrator.api.Migration class method), 8
create_index() (invenio_index_migrator.api.Job method), 7
create_index() (invenio_index_migrator.api.Migration method), 8
create_index() (invenio_index_migrator.api.MultiIndicesReindexJob method), 8

D

document_name (invenio_index_migrator.api.Job attribute), 7

E

ESClient (class in invenio_index_migrator.utils), 9
extract_doctype_from_mapping() (in module invenio_index_migrator.utils), 10

G

get_queue_size() (in module invenio_index_migrator.utils), 10

I

INDEX_MIGRATOR_RECIPES (in module invenio_index_migrator.config), 3

init() (invenio_index_migrator.api.Migration method), 8
initial_state() (invenio_index_migrator.api.MultiIndicesReindexJob method), 8
initial_state() (invenio_index_migrator.api.ReindexJob method), 9
invenio_index_migrator (module), 4
invenio_index_migrator.api (module), 7
invenio_index_migrator.config (module), 3
invenio_index_migrator.utils (module), 9
iter_indexer_ops() (invenio_index_migrator.api.ReindexAndSyncJob method), 9

J

Job (class in invenio_index_migrator.api), 7

L

load_jobs_from_config() (invenio_index_migrator.api.Migration method), 8

M

Migration (class in invenio_index_migrator.api), 8
MultiIndicesReindexJob (class in invenio_index_migrator.api), 8

N

notify() (invenio_index_migrator.api.Migration method), 8

O

obj_or_import_string() (in module invenio_index_migrator.utils), 10

R

read() (invenio_index_migrator.utils.State method), 10
reindex_auth (invenio_index_migrator.utils.ESClient attribute), 9

reindex_remote (invenio_index_migrator.utils.ESClient attribute), 9

ReindexAndSyncJob (class in invenio_index_migrator.api), 9

ReindexJob (class in invenio_index_migrator.api), 8

rollover() (invenio_index_migrator.api.Migration method), 8

rollover_actions() (invenio_index_migrator.api.Job method), 7

rollover_actions() (invenio_index_migrator.api.MultiIndicesReindexJob method), 8

run() (invenio_index_migrator.api.Job method), 7

run() (invenio_index_migrator.api.Migration method), 8

run() (invenio_index_migrator.api.MultiIndicesReindexJob method), 8

run() (invenio_index_migrator.api.ReindexAndSyncJob method), 9

run() (invenio_index_migrator.api.ReindexJob method), 9

run_delta_job() (invenio_index_migrator.api.ReindexAndSyncJob method), 9

S

src_es_client (invenio_index_migrator.api.Job attribute), 7

State (class in invenio_index_migrator.utils), 10

status() (invenio_index_migrator.api.Job method), 7

status() (invenio_index_migrator.api.Migration method), 8

strategy (invenio_index_migrator.api.Migration attribute), 8