
invenio-sip2 Documentation

Release 0.2.0

UCLouvain

May 05, 2021

Contents

1	User's Guide	3
1.1	Invenio-SIP2 Installation	3
1.2	Configuration	3
1.3	Usage	5
1.4	Example application	5
2	API Reference	7
2.1	API Docs	7
3	Additional Notes	9
3.1	Contributing	9
3.2	Changes	11
3.3	License	11
3.4	Authors	11
	Python Module Index	13
	Index	15

Invenio module that add SIP2 communication for self-check

TODO: Please provide feature overview of module

Further documentation is available on <https://invenio-sip2.readthedocs.io/>

This part of the documentation will show you how to get started in using Invenio-SIP2.

1.1 Invenio-SIP2 Installation

Invenio-SIP2 is on PyPI so all you need is:

```
$ pip install invenio-sip2
```

1.2 Configuration

Invenio module that add SIP2 communication for self-check.

<i>SIP2_MESSAGE_ACTIONS</i>	Dictionary of all selfcheck actions.
<i>SIP2_ACTIONS_HANDLERS</i>	Dictionary of action handlers. See example below.
<i>SIP2_SELFCHECK_MESSAGE_TYPES</i>	Define all message types conforming to SIP2 protocol.
<i>SIP2_FIXED_FIELD_DEFINITION</i>	All fixed field available.
<i>SIP2_VARIABLE_FIELD_DEFINITION</i>	All variable field available.

Each custom handler actions must be defined in the `SIP2_ACTIONS_HANDLERS` dictionary, where the keys are the application names and the values the configuration parameters for the application.

```
SIP2_ACTIONS_HANDLERS = dict(
    myapp=dict(
        # configuration values for myapp ...
    ),
)
```

The application name is used to start invenio-sip2 server and call customized handlers.

1.2.1 Remote application Handlers

Handlers allow customizing endpoints for each selfcheck actions:

Configuration of a single remote application is a dictionary with the following keys:

- `login_handler` - Import path to login selfcheck callback handler.
- `logout_handler` - Import path to logout selfcheck callback handler.
- `system_status_handler` - Import path to automated system status callback handler.
- **`patron_handlers` - A dictionary of import path to patron callback handler.**
 - `validate_patron` - Import path to validate patron callback handler.
 - `authorize_patron` - Import path to authorize patron callback handler.
 - `enable_patron` - Import path to enable patron callback handler.
 - `account` - Import path to retrieve patron account callback handler.

```
SIP2_REMOTE_ACTION_HANDLERS = dict(  
    app=dict(  
        login_handler="...",  
        logout_handler="...",  
        system_status_handler="...",  
        patron_handlers=dict(  
            validate_patron="...",  
            authorize_patron="...",  
            enable_patron="...",  
            account="...",  
        )  
    )  
)
```

`invenio_sip2.config.SIP2_CHECKSUM_CONTROL = True`
Message checksum control.

`invenio_sip2.config.SIP2_DATE_FORMAT = '%Y%m%d %H%M%S'`
SIP2 date format.

`invenio_sip2.config.SIP2_LINE_TERMINATOR = '\r'`
Message line separator.

`invenio_sip2.config.SIP2_PROTOCOL = '2.00'`
SIP2 protocol version.

`invenio_sip2.config.SIP2_REMOTE_ACTION_HANDLERS = {}`
Configuration of remote handlers.

`invenio_sip2.config.SIP2_RETRIES_ALLOWED = 10`
Number of retries allowed.

`invenio_sip2.config.SIP2_SOCKET_BUFFER_SIZE = '1024'`
Socket buffer size.

`invenio_sip2.config.SIP2_SUPPORT_CHECKIN = True`
Support check in items.

`invenio_sip2.config.SIP2_SUPPORT_CHECKOUT = True`
Support check out items.


```
invenio_sip2.config.SIP2_SUPPORT_OFFLINE_STATUS = True
    Support off line operation.

invenio_sip2.config.SIP2_SUPPORT_ONLINE_STATUS = True
    Support online status send by automatic circulation system.

invenio_sip2.config.SIP2_SUPPORT_RENEWAL_POLICY = True
    Support patron renewal requests as a policy.

invenio_sip2.config.SIP2_SUPPORT_STATUS_UPDATE = True
    Support patron status updating by the selfcheck.

invenio_sip2.config.SIP2_TEXT_ENCODING = 'UTF-8'
    Message text charset encoding.

invenio_sip2.config.SIP2_TIMEOUT_PERIOD = 10
    Server timeout.
```

1.3 Usage

Start SIP2 server with CLI example:

```
$ invenio selfcheck start -h 127.0.0.1 -p 3004 -r your-remote-app
```

1.4 Example application

First install Invenio-SIP2, setup the application and load fixture data by running:

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

Next, start the development server:

```
$ export FLASK_APP=app.py FLASK_DEBUG=1
$ flask run
```

and open the example application in your browser:

```
$ open http://127.0.0.1:5000/
```

To reset the example application run:

```
$ ./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

Flask extension for Invenio-SIP2.

```
class invenio_sip2.ext.InvenioSIP2 (app=None)
    Invenio-SIP2 extension.

    Extension initialization.

    init_app (app)
        Flask application initialization.

    init_config (app)
        Initialize configuration.

    retries_allowed
        Number of retries allowed by the automated circulation system.

    sip2
        Return the SIP2 action machine.

    sip2_current_date
        Get current date from system.

    sip2_handlers
        Return the SIP2 handler machine.

    sip2_message_types
        Message type configuration.

    support_checkin
        Support of checkin by the automated circulation system.
```

support_checkout

Support of checkout by the automated circulation system.

support_offline_status

Support of offline status by the automated circulation system.

support_online_status

Support of online status by the automated circulation system.

support_renewal_policy

Support of renewal policy by the automated circulation system.

support_status_update

Support of status update by the automated circulation system.

supported_messages

Supported messages by the automated circulation system.

supported_protocol

Supported protocol by the automated circulation system.

timeout_period

Timeout period allowed by the automated circulation system.

`invenio_sip2.ext.load_fixed_field(app)`

Load fixed field configuration.

`invenio_sip2.ext.load_variable_field(app)`

Load variable field configuration.

2.1.1 Views

Blueprint for Invenio-SIP2.

`invenio_sip2.views.monitoring()`

Render a basic view.

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-contrib/invenio-sip2/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-SIP2 could always use more documentation, whether as part of the official Invenio-SIP2 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-contrib/invenio-sip2/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-sip2* for local development.

1. Fork the *inveniosoftware-contrib/invenio-sip2* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/invenio-sip2.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-sip2
$ cd invenio-sip2/
$ 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, 3.5 and 3.6. Check https://travis-ci.org/github/inveniosoftware-contrib/invenio-sip2/pull_requests and make sure that the tests pass for all supported Python versions.

3.2 Changes

Version 0.2.0 (released 2020-08-10)

Implemented enhancements:

- Implements Patron information action.
- Adds Remote ILS handlers configuration.

Version 0.1.0 (released 2020-05-25)

- Base of automated circulation system.

3.3 License

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.

INVENIO-SIP2 Copyright (C) 2020 UCLouvain

This program is free software: you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation, version 3 of the License.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

3.4 Authors

Invenio module that add SIP2 communication for self-check

- Laurent Dubois (UCLouvain) <laurent.dubois@uclouvain.be>

i

`invenio_sip2.config`, [3](#)
`invenio_sip2.ext`, [7](#)
`invenio_sip2.views`, [8](#)

I

`init_app()` (*invenio_sip2.ext.InvenioSIP2 method*), 7
`init_config()` (*invenio_sip2.ext.InvenioSIP2 method*), 7
`invenio_sip2.config` (*module*), 3
`invenio_sip2.ext` (*module*), 7
`invenio_sip2.views` (*module*), 8
`InvenioSIP2` (*class in invenio_sip2.ext*), 7

L

`load_fixed_field()` (*in module invenio_sip2.ext*), 8
`load_variable_field()` (*in module invenio_sip2.ext*), 8

M

`monitoring()` (*in module invenio_sip2.views*), 8

R

`retries_allowed` (*invenio_sip2.ext.InvenioSIP2 attribute*), 7

S

`sip2` (*invenio_sip2.ext.InvenioSIP2 attribute*), 7
`SIP2_CHECKSUM_CONTROL` (*in module invenio_sip2.config*), 4
`sip2_current_date` (*invenio_sip2.ext.InvenioSIP2 attribute*), 7
`SIP2_DATE_FORMAT` (*in module invenio_sip2.config*), 4
`sip2_handlers` (*invenio_sip2.ext.InvenioSIP2 attribute*), 7
`SIP2_LINE_TERMINATOR` (*in module invenio_sip2.config*), 4
`sip2_message_types` (*invenio_sip2.ext.InvenioSIP2 attribute*), 7
`SIP2_PROTOCOL` (*in module invenio_sip2.config*), 4
`SIP2_REMOTE_ACTION_HANDLERS` (*in module invenio_sip2.config*), 4

`SIP2_RETRIES_ALLOWED` (*in module invenio_sip2.config*), 4
`SIP2_SOCKET_BUFFER_SIZE` (*in module invenio_sip2.config*), 4
`SIP2_SUPPORT_CHECKIN` (*in module invenio_sip2.config*), 4
`SIP2_SUPPORT_CHECKOUT` (*in module invenio_sip2.config*), 4
`SIP2_SUPPORT_OFFLINE_STATUS` (*in module invenio_sip2.config*), 4
`SIP2_SUPPORT_ONLINE_STATUS` (*in module invenio_sip2.config*), 5
`SIP2_SUPPORT_RENEWAL_POLICY` (*in module invenio_sip2.config*), 5
`SIP2_SUPPORT_STATUS_UPDATE` (*in module invenio_sip2.config*), 5
`SIP2_TEXT_ENCODING` (*in module invenio_sip2.config*), 5
`SIP2_TIMEOUT_PERIOD` (*in module invenio_sip2.config*), 5
`support_checkin` (*invenio_sip2.ext.InvenioSIP2 attribute*), 7
`support_checkout` (*invenio_sip2.ext.InvenioSIP2 attribute*), 7
`support_offline_status` (*invenio_sip2.ext.InvenioSIP2 attribute*), 8
`support_online_status` (*invenio_sip2.ext.InvenioSIP2 attribute*), 8
`support_renewal_policy` (*invenio_sip2.ext.InvenioSIP2 attribute*), 8
`support_status_update` (*invenio_sip2.ext.InvenioSIP2 attribute*), 8
`supported_messages` (*invenio_sip2.ext.InvenioSIP2 attribute*), 8
`supported_protocol` (*invenio_sip2.ext.InvenioSIP2 attribute*), 8

T

`timeout_period` (*invenio_sip2.ext.InvenioSIP2 attribute*), 8