Flask-SSO Documentation

Release 0.4.0

CERN

October 05, 2015

Contents

Flask-SSO is a Flask extension permitting to set up Shibboleth Single-Sign-On authentication in Flask based web applications.

CHAPTER 1

Contents

1.1 Installation

Flask-SSO is on PyPI so all you need is :

```
$ pip install flask-sso
```

The development version can be downloaded from its page at GitHub.

```
$ git clone https://github.com/inveniosoftware/flask-sso.git
$ cd flask-sso
$ python setup.py develop
$ ./run-tests.sh
```

1.1.1 Requirements

Flask-SSO has the following dependencies:

- Flask
- blinker

- six

Flask-SSO requires Python version 2.6, 2.7 or 3.3+

1.2 Quickstart

This part of the documentation will show you how to get started in using Flask-SSO with Flask.

This guide assumes you have successfully installed Flask-SSO and a working understanding of Flask. If not, follow the installation steps and read about Flask at http://flask.pocoo.org/docs/.

1.2.1 A Minimal Example

A minimal Flask-SSO usage example looks like this.

First, let's create the application and initialise the extension:

```
from flask import Flask, session, redirect
from flask_sso import SSO
app = Flask("myapp")
ext = SSO(app=app)
```

Second, let's configure the attribute map for converting environment variables to a dictionary containing user information:

```
#: Default attribute map
SSO_ATTRIBUTE_MAP = {
    'ADFS_AUTHLEVEL': (False, 'authlevel'),
    'ADFS_GROUP': (True, 'group'),
    'ADFS_LOGIN': (True, 'nickname'),
    'ADFS_ROLE': (False, 'role'),
    'ADFS_EMAIL': (True, 'email'),
    'ADFS_IDENTITYCLASS': (False, 'external'),
    'HTTP_SHIB_AUTHENTICATION_METHOD': (False, 'authmethod'),
}

app.config.setdefault('SSO_ATTRIBUTE_MAP', SSO_ATTRIBUTE_MAP)
```

Third, let's set up a login handler function that reads user information and stores it for later usage:

```
@sso.login_handler
def login_callback(user_info):
    """Store information in session."""
    session["user"] = user_info
```

Fourth, we can now greet the user using his SSO login name:

```
@app.route("/")
def index():
    """Display user information or force login."""
    if "user" in session:
        return "Welcome {name}".format(name=session["user"]["nickname"])
    return redirect(app.config["SSO_LOGIN_URL"])
```

1.3 Configuration

The details of the application settings that can be customized.

1.3.1 SSO_ATTRIBUTE_MAP

A dictionary mapping HTTP headers to a tuple. The tuple contains whether the attribute is required and then the name of the attribute.

Example:

```
# CERN Single-Sign-On
SSO_ATTRIBUTE_MAP = {
    "ADFS_LOGIN": (True, nickname),
    "ADFS_EMAIL": (True, email),
}

# General Shibboleth
SSO_ATTRIBUTE_MAP = {
    "HTTP_SHIB_IDENTITY_PROVIDER": (True, "idp"),
    "HTTP_SHIB_SHARED_TOKEN": (True, "shared_token"),
    "HTTP_SHIB_CN": (True, "cn"),
    "HTTP_SHIB_MAIL": (True, "email"),
    "HTTP_SHIB_GIVENNAME": (False, "first_name"),
    "HTTP_SHIB_SN": (False, "last_name"),
}
```

1.3.2 SSO_LOGIN_URL

Url of login handler. Default: */login/sso*.

1.3.3 SSO_LOGIN_ENDPOINT

Name of login handler endpoint to be used in *url_for* function.

Example:

```
>>> from flask.ext.sso.config import *
>>> url_for(SSO_LOGIN_ENDPOINT)
/login/sso
>>> SSO_LOGIN_URL
/login/sso
```

Default: *sso_login*.

1.4 API

This documentation section is automatically generated from Flask-SSO's source code.

1.4.1 Flask-SSO

Implement Shibboleth Single-Sign-On authentication.

Flask-SSO is initialized like this:

Initialization of the extension:

```
>>> from flask import Flask
>>> from flask_sso import SSO
>>> app = Flask('myapp')
>>> ext = SSO(app=app)
```

or alternatively using the factory pattern:

```
>>> app = Flask('myapp')
>>> ext = SSO()
>>> ext.init_app(app)
```

class flask_sso.**SSO**(*app=None*)
Flask extension implementation.

init_app(*app*)
Initialize a Flask application.

login()
Implement application login endpoint for SSO.

login_error_handler(*callback*)
Set the error callback for *login* method.

It takes one argument with attributes map, and should return a Flask response.

Parameters **callback** (*function*) – The callback for login error.

login_handler(*callback*)
Set the callback for the *login* method.

It takes one argument with attributes map, and should return a Flask response.

Parameters **callback** (*function*) – The callback for login.

parse_attributes()
Parse arguments from environment variables.

1.5 Changelog

Here you can see the full list of changes between each Flask-SSO release.

Version 0.4.0 (released 2015-10-05)

- Login error handler can be added to SSO and will be called with required attributes are missing. If login error callback is set no *SSOAttributeError* will be raised and application can return custom error response based on missing attributes.

Version 0.3.0 (released 2015-07-30)

- The Flask-SSO extension is now released under more permissive Revised BSD License. (#6)
- For testing execute run-tests.sh instead of sourcing it. (#4)
- New minimal application example. (#8)

- New Tox support for Python-3.4. (#4)

Version 0.2.0 (released 2014-06-26)

- Allowing ';' separator in HTTP data.
- Fix for dictionary key order in tests.
- Fix for Python 3.3 string comparison.
- New dependency: Blinker.
- Code coverage improved to 100%.
- New configuration option SSO_LOGIN_ENDPOINT.

Version 0.1

- Initial public release.

1.6 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:

```
$ ./run-tests.sh
...
Ran 8 tests in 0.246s

OK
Name                  Stmts   Miss  Cover   Missing
---------------------------------------------------
flask_sso/__init__       47      0   100%
flask_sso/config          4      0   100%
flask_sso/version         2      0   100%
---------------------------------------------------
TOTAL                    53      0   100%
```

1.7 License

Flask-SSO is free software; you can redistribute it and/or modify it under the terms of the Revised BSD License quoted below.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

1.8 Authors

Flask-SSO is developed for use in Invenio digital library software.

Contact us at info@invenio-software.org

1.8.1 Contributors

- Jiri Kuncar <jiri.kuncar@cern.ch>
- Iason Andriopoulos <iason.andriopoulos@cern.ch>
- ThiefMaster <adrian@planetcoding.net>
- Tibor Simko <tibor.simko@cern.ch>
- Harri Hämäläinen <harri.hamalainen@csc.fi>

f

F

I

L

P

S