Search projects

Help Sponsor Log in Register

django-python3-saml 1.0

✓ Latest version

```
pip install django-python3-saml
```

Released: Mar 19, 2018

Django SAML auth plugin for python 3

Navigation

Project description

Release history

Download files

Project links

Homepage

Statistics

Project description

Django-python3-saml is a Django app designed to allow plug & play SAML authentication setup for Python 3 enabled Django instance.

Install

Run the following command to install:

```
$pip install django-python3-saml
```

Setup

1. Create .env file in the same folder as settings.py if one does not already exist.
2. Copy the template below into the .env:

GitHub statistics:

- Stars: 3
- Forks: 3
- Open issues/PRs: 0

View statistics for this project via Libraries.io, or by using our public dataset on Google BigQuery

Meta

License: BSD License (BSD 3-Clause License)

Author: Issac Gable

Maintainers

icsd.tech

Classifiers

Development Status

- 4 - Beta

Environment

- Web Environment

Framework

- Django
- Django :: 1.11

Intended Audience

- Developers

License

```
# Fill out all that apply.
# On true or false pick the one that applies.
# Django-environ expects lowercase true/false

# Redirect URL passed to the next parameter.
LOGIN_REDIRECT_URL=""

# Absolute Path to X509CERT file location.
X509CERT=""

# HTTPS setting
HTTPS=true or false

# Service Provider Information
SP_METADATA_URL=""
SP_LOGIN_URL=""
SP_LOGOUT_URL=""
SP_X509CERT=""
SP_PRIVATE_KEY=""

# Identity Provider Information
IDP_METADATA_URL=""
IDP_SSO_URL=""
IDP_SLO_URL=""
IDP_X509_FINGERPRINT=""

# New User Groups is a list of Groups to assign
# Upon new Django account creation.

# Environ expects a comma separated string.
# E.G. "Now,Is,The,Time"

# If no groups to be assigned leave list blank.
NEW_USER_GROUPS=""

# New Users setup
ACTIVE_STATUS=true or false
STAFF_STATUS=true or false

# Contact Information Technical
CI_TECH_GIVEN_NAME=""
CI_TECH_EMAIL=""

# Contact Information Support
CI_SUPPORT_GIVEN_NAME=""
```

- OSI Approved :: BSD License

Operating System

- OS Independent

Programming Language

- Python
- Python :: 3
- Python :: 3.4
- Python :: 3.5

Topic

- Internet :: WWW/HTTP
- Internet :: WWW/HTTP :: Dynamic Content

```
CI_SUPPORT_EMAIL=""

# Organizational Information EN_US
ORG_NAME=""
ORG_DISPLAY_NAME=""
ORG_HOME_URL=""
```

3. Copy the template below into settings.py:

```
# Set up django-environ by the instructions
# https://github.com/joke2k/django-environ
# if the project has a different .env package
# simply change the env("variable") to package form

import os
import sys
import environ

# This should setting should walk the path
# back to the project's root (e.g. where manage.py
# Three folder back (/config/settings/.env - 3 = /)
root = environ.Path(__file__) - 3

# Initialize root function
PROJECT_ROOT = root()

# Initialize Env function
# Environ requires variable casting.
# Required variables are initialized
# in the ENV_CAST dict below.
ENV_CAST = {
    'HTTPS': (bool,False),
    'NEW_USER_GROUPS': (list,[]),
    'ACTIVE_STATUS': (bool,False),
    'STAFF_STATUS': (bool,False),
    'DEBUG': (bool,False),
}
env = environ.Env(**ENV_CAST)

# Reads the .env file
env.read_env()

# Set the system path to the base application folde
# This may not be required in your project.
# Below is an example setup.
sys.path.insert(0, os.path.join(PROJECT_ROOT, 'apps

# SAML variable houses all .env variables in a sing
# All variable requests in the module will pull fro
# The .env file is used for security and should not
SAML = {
    'LOGIN_REDIRECT': env("LOGIN_REDIRECT_URL"),
    'X509CERT': env("X509CERT"),
    'HTTPS': 'on' if env("HTTPS") else 'off',
    'SP': {
        'METADATA_URL': env("SP_METADATA_URL"),
        'LOGIN_URL': env("SP_LOGIN_URL"),
        'LOGOUT_URL': env("SP_LOGOUT_URL"),
        'X509CERT': env("SP_X509CERT"),
        'PRIVATE_KEY': env("SP_PRIVATE_KEY"),
    },
    'IDP': {
        'METADATA_URL': env("IDP_METADATA_URL"),
        'SSO_URL': env("IDP_SSO_URL"),
        'SLO_URL': env("IDP_SLO_URL"),
        'X509_FINGERPRINT': env("IDP_X509_FINGERPRI
    },
    'NEW_USER': {
        'GROUPS': env("NEW_USER_GROUPS"),
        'ACTIVE': env("ACTIVE_STATUS"),
        'STAFF': env("STAFF_STATUS"),
    },
    'CONTACT_INFO': {
        'TECHNICAL': {
            'GIVEN_NAME': env("CI_TECH_GIVEN_NAME")
            'EMAIL': env("CI_TECH_EMAIL"),
        },
        'SUPPORT': {
            'GIVEN_NAME': env("CI_SUPPORT_GIVEN_NAM
            'EMAIL': env("CI_SUPPORT_EMAIL"),
        }
    },
    'ORGANIZATION_INFO': {
        'EN_US': {
            'NAME': env("ORG_NAME"),
            'DISPLAY_NAME': env("ORG_DISPLAY_NAME")
            'URL': env("ORG_HOME_URL"),
        }
    }
}
```

4. Add ‘django-python3-saml’ to your INSTALLED_APPS setting like this:

```
INSTALLED_APPS = [
    ...
    'django_python3_saml',
]
```

5. Include the dango-python3-saml URLconf into project urls.py like this:

```
url(r'^saml/', include('django_python3_saml.urls'))
```

6. Once the urls have been included as above the SP url paths should be as the following example:

```
Start Url --> https://example.com/saml/initiate-log
ACS URL --> https://example.com/saml/complete-login
Enitiy ID --> https://example.com/saml/metadata/
```

7. Add ‘django-python3-saml.backends.SAMLServiceProviderBackend’ to AUTHENTICATION_BACKENDS like this:

```
AUTHENTICATION_BACKENDS = [
    'django-python3-saml.backends.SAMLServiceProvid
    ...
]
```

8. Setup chosen Identity Provider (IDP) and Attribute Mapping:

```
Example Google Attribute Mapping expected format fo

first_name - Basic Information - First Name
last_name - Basic Information - Last Name
email - Basic Information -  Primary Email
```

Special Thanks

1. Kristian Oellegaard –> django-saml-service-provider
2. OneLogin’s SAML Python Toolkit –> python3-saml
3. Daniele Faraglia –> django-environ

Help

Installing packages
Uploading packages
User guide
FAQs

About PyPI

PyPI on Twitter
Infrastructure dashboard
Package index name retention
Our sponsors

Contributing to PyPI

Bugs and feedback
Contribute on GitHub
Translate PyPI
Development credits

Using PyPI

Code of conduct
Report security issue
Privacy policy
Terms of use

Status: All Systems Operational

Developed and maintained by the Python community, for the Python community.
Donate today!

© 2020 Python Software Foundation
Site map

Switch to desktop version

› English español français 日本語 Português Brasileiro Українська Ελληνικά Deutsch 简体中文 Русский

Pingdom
Monitoring

Google
Object Storage and Download Analytics

Sentry
Error logging

AWS
Cloud computing

DataDog
Monitoring

Fastly
CDN

DigiCert
EV certificate

StatusPage
Status page