Pacifica Authentication Documentation

David Brown

Nov 03, 2020

Contents:

1 Installation **3**
- 1.1 Installation in Virtual Environment . . . 3

2 Configuration **5**
- 2.1 Authentication Options . . . 5

3 Example Usage **7**

4 Authentication Python Module **9**
- 4.1 Application Python Module . . . 9
- 4.2 Root CherryPy Python Module . . . 9
- 4.3 SAPlugin Python Module . . . 10
- 4.4 SATool Python Module . . . 10
- 4.5 SQLAlchemy User Model Python Module . . . 10

5 Indices and tables **13**

Python Module Index **15**

Index **17**

The Pacifica Authentication library provides Pacifica Core services a consistent authentication configuration and APIs.

CHAPTER 1

Installation

The Pacifica software is available through PyPi so creating a virtual environment to install is what is shown below. Please keep in mind compatibility with the Pacifica Core services.

1.1 Installation in Virtual Environment

These installation instructions are intended to work on both Windows, Linux, and Mac platforms. Please keep that in mind when following the instructions.

Please install the appropriate tested version of Python for maximum chance of success.

1.1.1 Linux and Mac Installation

```
mkdir ~/.virtualenvs
python -m virtualenv ~/.virtualenvs/pacifica
. ~/.virtualenvs/pacifica/bin/activate
pip install pacifica-auth
```

1.1.2 Windows Installation

This is done using PowerShell. Please do not use Batch Command.

```
mkdir "$Env:LOCALAPPDATA\virtualenvs"
python.exe -m virtualenv "$Env:LOCALAPPDATA\virtualenvs\pacifica"
& "$Env:LOCALAPPDATA\virtualenvs\pacifica\Scripts\activate.ps1"
pip install pacifica-auth
```

CHAPTER 2

Configuration

The configuration of Pacifica Authentication is done on the command line using arguments passed to services that incorporate the library.

2.1 Authentication Options

These are common command line switches when Pacifica services include the authentication library.

2.1.1 Session Directory `--session-dir`

Default is `sessions` and contains the user sessions managed by the core service.

2.1.2 Database URL `--db-url`

Default is `sqlite:///database.sqlite3` and is the SQLAlchemy engine connection url

2.1.3 Social Auth Module `--social-module`

Python Social Auth backends Python module name. The module name is relative to `social_core.backends`.

2.1.4 Social Auth Class `--social-class`

Python Social Auth class name from the module in the `--social-module` name.

2.1.5 Social Auth Settings `--social-setting`

Python Social Auth settings are passed as to CherryPy configrations. Examples are the following:

```
pacifica-service \
  --social-setting=github_key=<GitHub OAuth Key> \
  --social-setting=github_secret=<GitHub OAuth Secret>
```

2.1.6 Application Directory `--app-dir`

This is optional as some Pacifica services don't serve applications to users. This can serve both ReactJS or Swagger-UI as example applications.

CHAPTER 3

Example Usage

Need to have good example application.

CHAPTER 4

Authentication Python Module

4.1 Application Python Module

Base application module.

pacifica.auth.application.**check_sa_module_class**(*args*)
Check the combination of module and class.

pacifica.auth.application.**check_sa_settings**(*obj_str*)
Verify social auth settings.

pacifica.auth.application.**error_page_default**(***kwargs*)
Error page when something goes wrong.

pacifica.auth.application.**pacifica_auth_arguments**(*parser*)
Add Pacifica authentication command line arguments.

pacifica.auth.application.**session_commit**()
Commit the user session to the database.

pacifica.auth.application.**social_settings**(*args*, *user_class*, *user_import_path*)
Setup the social settings for pacifica auth.

4.2 Root CherryPy Python Module

Root class to handle social auth.

class pacifica.auth.root.**Root**(*sa_module*, *app_dir*)
Root class to integrate social auth.

__init__(*sa_module*, *app_dir*)
Save the sa module and app directory.

app(**args*)
Serve the app or redirect to login.

done()
Done with the social auth login.

index()
If the user isn't there redirect to login.

logout()
Logout the user deleting the session.

4.3 SAPlugin Python Module

CherryPy plugin to manage SQLAlchemy session connection.

class pacifica.auth.saplugin.**SAEnginePlugin**(*bus*, *connection_string=None*)
CherryPy SQLAlchemy Plugin.

__init__(*bus*, *connection_string=None*)
Create the plugin saving engine connection and session.

bind()
Bind the session to the engine.

commit()
Commit the session to the database or rollback.

start()
Start the engine connection and subscribe bind and commit.

stop()
Unsubscribe bind and commit and dispose of engine.

4.4 SATool Python Module

CherryPy Social Auth Tools to setup database connections.

class pacifica.auth.satool.**SATool**
CherryPy tool to manage handler.

__init__()
Create the SATool and set bind priority.

_setup()
Setup and attach the hooks.

bind_session()
Bind the db session to something we can use.

commit_transaction()
Delete the db session and publish commit.

4.5 SQLAlchemy User Model Python Module

Example user model for consumers to use.

class pacifica.auth.user_model.**Base**(***kwargs*)
The most base type

__init__(***kwargs*)
A simple constructor that allows initialization from kwargs.

Sets attributes on the constructed instance using the names and values in `kwargs`.

Only keys that are present as attributes of the instance's class are allowed. These could be, for example, any mapped columns or relationships.

metadata = MetaData()

registry = <sqlalchemy.orm.decl_api.registry object>

class pacifica.auth.user_model.**User**(***kwargs*)
Example SQLAlchemy User Model.

__init__(***kwargs*)
A simple constructor that allows initialization from kwargs.

Sets attributes on the constructed instance using the names and values in `kwargs`.

Only keys that are present as attributes of the instance's class are allowed. These could be, for example, any mapped columns or relationships.

_sa_class_manager = {'active': <sqlalchemy.orm.attributes.InstrumentedAttribute objec

active

email

id

is_active()
Return the user active attribute.

is_authenticated()
The user is always authenticated.

name

password

username

uuid

Pacifica Authentication Module.

pacifica.auth.**auth_session**(*func*)
Authenticate the method.

pacifica.auth.**pacifica_auth_arguments**(*parser*)
Add Pacifica authentication command line arguments.

pacifica.auth.**error_page_default**(***kwargs*)
Error page when something goes wrong.

pacifica.auth.**social_settings**(*args*, *user_class*, *user_import_path*)
Setup the social settings for pacifica auth.

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

pacifica.auth, 11
pacifica.auth.application, 9
pacifica.auth.root, 9
pacifica.auth.saplugin, 10
pacifica.auth.satool, 10
pacifica.auth.user_model, 10

Index

Symbols

__init__() (*pacifica.auth.root.Root method*), 9
__init__() (*pacifica.auth.saplugin.SAEnginePlugin method*), 10
__init__() (*pacifica.auth.satool.SATool method*), 10
__init__() (*pacifica.auth.user_model.Base method*), 10
__init__() (*pacifica.auth.user_model.User method*), 11
_sa_class_manager (*pacifica.auth.user_model.User attribute*), 11
_setup() (*pacifica.auth.satool.SATool method*), 10

A

active (*pacifica.auth.user_model.User attribute*), 11
app() (*pacifica.auth.root.Root method*), 9
auth_session() (*in module pacifica.auth*), 11

B

Base (*class in pacifica.auth.user_model*), 10
bind() (*pacifica.auth.saplugin.SAEnginePlugin method*), 10
bind_session() (*pacifica.auth.satool.SATool method*), 10

C

check_sa_module_class() (*in module pacifica.auth.application*), 9
check_sa_settings() (*in module pacifica.auth.application*), 9
commit() (*pacifica.auth.saplugin.SAEnginePlugin method*), 10
commit_transaction() (*pacifica.auth.satool.SATool method*), 10

D

done() (*pacifica.auth.root.Root method*), 9

E

email (*pacifica.auth.user_model.User attribute*), 11
error_page_default() (*in module pacifica.auth*), 11
error_page_default() (*in module pacifica.auth.application*), 9

I

id (*pacifica.auth.user_model.User attribute*), 11
index() (*pacifica.auth.root.Root method*), 10
is_active() (*pacifica.auth.user_model.User method*), 11
is_authenticated() (*pacifica.auth.user_model.User method*), 11

L

logout() (*pacifica.auth.root.Root method*), 10

M

metadata (*pacifica.auth.user_model.Base attribute*), 11

N

name (*pacifica.auth.user_model.User attribute*), 11

P

pacifica.auth (*module*), 11
pacifica.auth.application (*module*), 9
pacifica.auth.root (*module*), 9
pacifica.auth.saplugin (*module*), 10
pacifica.auth.satool (*module*), 10
pacifica.auth.user_model (*module*), 10
pacifica_auth_arguments() (*in module pacifica.auth*), 11
pacifica_auth_arguments() (*in module pacifica.auth.application*), 9
password (*pacifica.auth.user_model.User attribute*), 11

R

registry (*pacifica.auth.user_model.Base attribute*), 11

Root (*class in pacifica.auth.root*), 9

S

SAEnginePlugin (*class in pacifica.auth.saplugin*), 10
SATool (*class in pacifica.auth.satool*), 10
session_commit() (*in module pacifica.auth.application*), 9
social_settings() (*in module pacifica.auth*), 11
social_settings() (*in module pacifica.auth.application*), 9
start() (*pacifica.auth.saplugin.SAEnginePlugin method*), 10
stop() (*pacifica.auth.saplugin.SAEnginePlugin method*), 10

U

User (*class in pacifica.auth.user_model*), 11
username (*pacifica.auth.user_model.User attribute*), 11
uuid (*pacifica.auth.user_model.User attribute*), 11