SQLAlchemy-Fixtures Documentation

Release 0.1

Konsta Vesterinen

Apr 04, 2017

Contents

SQLAlchemy-Fixtures is a python package that provides functional fixtures for SQLAlchemy based models.

CHAPTER 1

QuickStart

At the heart of SQLAlchemy-Fixtures there are two functions: fixture and last_fixture. Function fixture is used for constructing fixtures from models and last_fixture is used for getting the last created fixture for given model.

Consider the following model definition:

```
import sqlalchemy as sa
from sqlalchemy import create_engine
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker

engine = create_engine('sqlite:///:memory:')
Base = declarative_base(engine)
Session = sessionmaker(bind=engine)
session = Session()

class User(Base):
    __tablename__ = 'user'

    id = sa.Column(sa.BigInteger, autoincrement=True, primary_key=True)
    name = sa.Column(sa.Unicode(100))
    email = sa.Column(sa.Unicode(255))
```

Most of the time you will want your models to contain some default values. This can be achieved by using FixtureRegistry.set_defaults function

```
from sqlalchemy_fixture import FixtureRegistry, fixture, last_fixture

FixtureRegistry.set_defaults(User, {'name': 'someone'})

user = fixture(User)
user.name  # someone

last_fixture(User) == user
```

Sometimes you may want to create fixtures without adding them into session and committing the session. SQLAlchemy-Fixtures provides a function called new for this:

```
from sqlalchemy_fixture import new

# the following object is not saved into database
user = new(User, name=u'someone', email=u'john@example.com')
```

CHAPTER 2

Lazy values

Lazy values provide a convenient way to generate values based on object attributes. In the following example our User fixture will generate its email based on its name.

```
from sqlalchemy_fixture import FixtureRegistry, Lazy, fixture, last_fixture

FixtureRegistry.set_defaults(
    User, {'email': Lazy(lambda obj: '%s@example.com' % obj.name.lower())})

user = fixture(User, name=u'someone')
user.email  # someone@example.com
```