Django Dynamic Fixtures Documentation

Release 0.2.1

Peter Slump

Feb 04, 2020

Django Dynamic Fixtures is a Django app which gives you the ability to setup fixture-data in a more dynamic way. Static fixtures are sometimes too static in a way that for example even the primary keys are static defined, this can be very hard to maintain especially in bigger projects. Another example; when your application depends on data with a recent timestamp your static fixtures can get 'outdated'.

For all these issues Django Dynamic Fixtures has a solution and even more!

Features:

- *Write fixtures* in Python;
- *Load fixtures* which are required for your task;
- Manage fixture *Dependencies*.

CHAPTER 1

Changelog

0.2.1

- Added some docs about dry-run mode
- Fixed Django versions in setup.py

0.2.0

- Added time elapsed per fixture
- Dry-run mode
- List available fixtures
- Run all fixtures in an transaction
- Removed support for Django 1.7
- Added support for Django 2.0

Installation

First install the package:

```
$ pip install django-dynamic-fixtures
```

Add the app to your project's *settings.py* file:

```
# settings.py
INSTALLED_APPS = [
    ...,
    'dynamic_fixtures'
]
```

Or make sure the app is not loaded on production:

```
# settings.py
if DEBUG:
    INSTALLED_APPS = INSTALLED_APPS + ['dynamic_fixtures']
```

CHAPTER 3

Write fixtures

All fixtures are written in .py files the *fixtures*-module of your app.

Recommended is to prefix the fixture files with numbers just like you probably already know from the Django migrations.:

Inside the fixture file you have to create a class called *Fixture*. This class should extend from `dynamic_fixtures.fixtures.basefixture.BaseFixture`.

In this class you define at least the *load*-method. In this method your are free to setup your fixture data in a way you like:

```
#my_django_project/my_app/fixtures/0001_create_example_author.py
from dynamic_fixtures.fixtures import BaseFixture

from my_app.models import Author


class Fixture(BaseFixture):

    def load(self):
        Author.objects.create(name="John Doe")
```

CHAPTER 4

List fixtures

To list all existing fixtures you can call the management command *load_dynamic_fixtures* with an argument *–list*:

```
$ ./manage.py load_dynamic_fixtures --list
```

The output may help to find out the reason why a fixture wasn't loaded.

CHAPTER 5

Load fixtures

To load the fixtures you can call the management command *load_dynamic_fixtures*:

```
$ ./manage.py load_dynamic_fixtures
```

You can also specify which fixtures you want to load. In this case the requested fixture will be loaded plus all depending fixtures. This ensures that you always have a valid data-set:

```
$ ./manage.py load_dynamic_fixtures my_app 0001_create_example_author
```

Or load all fixtures for a given app:

```
$ ./manage.py load_dynamic_fixtures my_app
```

CHAPTER 6

Dry-run

You can test your fixtures in dry-run mode. Add the *–dry-run* argument to the management command. Fixtures will loaded as without dry-run enabled however the transaction will be rolled back at the end:

```
$ ./manage.py load_dynamic_fixtures --dry-run
```

CHAPTER 7

Dependencies

It's also possible to maintain dependencies between fixtures. This can be accomplished in the same way as Django migrations:

```
#my_django_project/my_app/fixtures/0002_create_example_books.py
from dynamic_fixtures.fixtures import BaseFixture

from my_app.models import Book


class Fixture(BaseFixture):

   dependencies  = (
      ('my_app', '0001_create_example_author'),
   )


    def load(self):
        author = Author.objects.get(name='John Doe')

        Book.objects.create(title="About roses and gladiolus", author=author)
        Book.objects.create(title="The green smurf", author=author)
```

The library take care that the depending fixture is loaded before this one, so you know for sure that the entity is available in the database.

CHAPTER 8

Gotcha's

A really powerful combination is a combination of this library and Factory Boy. In the example below 50 authors will get created from factories.:

```
#my_django_project/my_app/fixtures/0001_create_example_author.py
from dynamic_fixtures.fixtures import BaseFixture

from my_app.factories import AuthorFactory


class Fixture(BaseFixture):

    def load(self):
        AuthorFactory.create_batch(size=50)
```