Pytest Django Migrations Documentation

Release 0.1.0

Kit La Touche

Jun 16, 2020

Contents

Contents:

Pytedjmi is a tool for testing Django migrations with Pytest.

It handles all of the migration set up and tear down for you, but you have to define the before-state and the after-state that you want for your test.

Here's an example:

```python
import pytedjmi


@pytest.mark.migrate_from("app_name", "0011_add_book_model")
def test_books_and_authors_data_migration(initial_apps):
    """
    We are changing our data model from one-to-many Authors-to-Books
    to many-to-many Authors-to-Books, because we want to support
    collaborations.
    """
    # First, we get the old models:
    Book = initial_apps.get_model('library', 'Book')
    Author = initial_apps.get_model('library', 'Author')
    a_book = Book.objects.create(title='Moby-Dick')
    Author.objects.create(book=a_book, name='Herman Melville')

    # Then, we roll forward and get the new models:
    new_apps = pytedjmi.migrate(
        "app_name",
        "0012_book_author_m2m_data_migration",
    )
    Book = new_apps.get_model('library', 'Book')
    Author = new_apps.get_model('library', 'Author')
    author = Author.objects.first()
    book = Book.objects.first()

    # Finally, we make assertions:
    assert list(book.author_set.all()) == [author]
    assert list(author.book_set.all()) == [book]
```

CHAPTER 1

Indices and tables

- genindex
- modindex
- search