djorm-ext-pguuid Documentation

Release 0.1

Andrey Antukh

September 13, 2014

Contents

PostgreSQL native uuid field for Django.

CHAPTER 1

Introduction

Django by default, has a large collection of possible types that can be used to define the model. But sometimes we need to use some more complex types offered by PostgreSQL. In this case, we will look the integrating of PostgreSQL native uuid field with Django.

Compatible with both: python3.3+ and python2.7+

CHAPTER 2

How install it?

You can clone the repo from github and install with simple python setup.py install command. Or use a pip, for install it from Python Package Index.

```
pip install djorm-ext-pguuid
```

CHAPTER 3

Quickstart

djorm-ext-pguuid exposes a simple django model field *djorm_pguuid.fields.UUIDField*.

This is a sample definition of model using a UUIDField:

```
from django.db import models
from djorm_pguuid.fields import UUIDField
from djorm_expressions.models import ExpressionManager

class Register(models.Model):
    # auto_add keyword parameter ensure a random
    # uuid generation if it is not specified.
    id = UUIDField(auto_add=True, primary_key=True)
    name = models.CharField(max_length=200, blank=True)

    objects = ExpressionManager()
```

3.1 Creating objects

This is a sample example of creating objects with array fields.

```
>>> import uuid
>>> Register.objects.create(id=uuid.UUID("01234644-0126-0113-0123-1123426749ab"))
<Register: Register object>
>>> Register.objects.create(id="01234644-0126-0113-0123-1123426749ab")
<Register: Register object>
```