
djorm-ext-pgjson Documentation

Release 0.1

Andrey Antoukh

August 31, 2013

CONTENTS

PostgreSQL native json field for Django.

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 json field with Django.

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

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-pgjson
```

Note: After PostgreSQL 9.3 is releases, you can use **djorm-ext-expressions** for make complex lookups using PostgreSQL native query operators. PostgreSQL 9.2 only supports storing data on the field.

QUICKSTART

djorm-ext-pgjson exposes a simple django model field *djorm_pgjson.fields.JSONField*.

This is a sample definition of model using a JSONField:

```
from django.db import models
from djorm_pgjson.fields import JSONField
from djorm_expressions.models import ExpressionManager

class Register(models.Model):
    name = models.CharField(max_length=200)
    points = JSONField()
    objects = ExpressionManager()
```

3.1 Creating objects

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

```
>>> Register.objects.create(points=[1, 2, 3, 4])
<Register: Register object>

>>> Register.objects.create(points={"1":2, "3":4})
<Register: Register object>
```