

---

# **djongo Documentation**

***Release 1.2.24***

**nesdis**

**Apr 15, 2018**



---

## Contents:

---

<b>1</b>	<b>django package</b>	<b>1</b>
1.1	Subpackages	2
1.1.1	django.models package	2
1.1.1.1	Submodules	2
1.1.1.2	django.models.fields module	2
1.1.1.3	Module contents	2
1.1.2	django.sql2mongo package	2
1.1.2.1	Submodules	2
1.1.2.2	django.sql2mongo.converters module	2
1.1.2.3	django.sql2mongo.operators module	2
1.1.2.4	django.sql2mongo.query module	2
1.1.2.5	Module contents	2
1.2	Submodules	2
1.3	django.base module	2
1.4	django.compiler module	2
1.5	django.cursor module	3
1.6	django.database module	3
1.7	django.features module	3
1.8	django.introspection module	3
1.9	django.operations module	4
1.10	django.schema module	4
1.11	Module contents	4
<b>2</b>	<b>Indices and tables</b>	<b>5</b>
	<b>Python Module Index</b>	<b>7</b>





# CHAPTER 1

---

## djongo package

---

## 1.1 Subpackages

### 1.1.1 djongo.models package

#### 1.1.1.1 Submodules

#### 1.1.1.2 djongo.models.fields module

#### 1.1.1.3 Module contents

### 1.1.2 djongo.sql2mongo package

#### 1.1.2.1 Submodules

#### 1.1.2.2 djongo.sql2mongo.converters module

#### 1.1.2.3 djongo.sql2mongo.operators module

#### 1.1.2.4 djongo.sql2mongo.query module

#### 1.1.2.5 Module contents

## 1.2 Submodules

## 1.3 djongo.base module

## 1.4 djongo.compiler module

**class** `djongo.compiler.SQLCompiler` (*query, connection, using*)

Bases: `object`

## 1.5 djongo.cursor module

## 1.6 djongo.database module

```

exception djongo.database.DataError
    Bases: djongo.database.DatabaseError

exception djongo.database.DatabaseError
    Bases: djongo.database.Error

exception djongo.database.Error
    Bases: Exception

exception djongo.database.IntegrityError
    Bases: djongo.database.DatabaseError

exception djongo.database.InterfaceError
    Bases: djongo.database.Error

exception djongo.database.InternalError
    Bases: djongo.database.DatabaseError

exception djongo.database.NotSupportedError
    Bases: djongo.database.DatabaseError

exception djongo.database.OperationalError
    Bases: djongo.database.DatabaseError

exception djongo.database.ProgrammingError
    Bases: djongo.database.DatabaseError

djongo.database.connect (**kwargs)

```

## 1.7 djongo.features module

```

class djongo.features.DatabaseFeatures (connection)
    Bases: django.db.backends.base.features.BaseDatabaseFeatures

    has_bulk_insert = False

    has_native_uuid_field = True

    supports_transactions = False

```

## 1.8 djongo.introspection module

```

class djongo.introspection.DatabaseIntrospection (connection)
    Bases: django.db.backends.base.introspection.BaseDatabaseIntrospection

    SAMPLE_SIZE = 10000

    TYPE_MAPPING = {<class 'int'>: <class 'bson.int64.Int64'>}

    data_types_reverse = {<class 'str'>: 'CharField', 'text': 'TextField', <class 'datetime'>: 'DateTimeField'}

```

**get\_constraints** (*cursor, table\_name*)

Retrieve any constraints or keys (unique, pk, fk, check, index) across one or more columns. Since MongoDB doesn't really support foreign keys this will only ever return unique constraints and primary keys

**get\_indexes** (*cursor, table\_name*)

Effectively an alias for *get\_constraints*

**get\_key\_columns** (*cursor, table\_name*)

Key columns are not supported so this always returns []

**get\_relations** (*cursor, table\_name*)

Relations are not supported so this always returns []

**get\_table\_description** (*cursor, table\_name*)

Get a collection description by fetching a sample of *SAMPLE\_SIZE* and analyzing it's contents. Because MongoDB doesn't have a fixed document specification this is only an approximation but depending on the *SAMPLE\_SIZE* relative to the collection size it can be rather accurate.

**get\_table\_list** (*cursor*)

Return a list of MongoDB collections (tables)

## 1.9 django.operations module

**class** `django.operations.DatabaseOperations` (*connection*)

Bases: `django.db.backends.base.operations.BaseDatabaseOperations`

**adapt\_datefield\_value** (*value*)

**adapt\_datetimefield\_value** (*value*)

**adapt\_timefield\_value** (*value*)

**convert\_datefield\_value** (*value, expression, connection, context*)

**convert\_timefield\_value** (*value, expression, connection, context*)

**get\_db\_converters** (*expression*)

**max\_name\_length** ()

**no\_limit\_value** ()

**quote\_name** (*name*)

**sql\_flush** (*style, tables, sequences, allow\_cascade=False*)

## 1.10 django.schema module

**class** `django.schema.DatabaseSchemaEditor` (*connection, collect\_sql=False, atomic=True*)

Bases: `django.db.backends.base.schema.BaseDatabaseSchemaEditor`

## 1.11 Module contents



## CHAPTER 2

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### d

- `django`, 4
- `django.compiler`, 2
- `django.database`, 3
- `django.features`, 3
- `django.introspection`, 3
- `django.operations`, 4
- `django.schema`, 4



## A

`adapt_datefield_value()` (django.operations.DatabaseOperations method), 4  
`adapt_datetimefield_value()` (django.operations.DatabaseOperations method), 4  
`adapt_timefield_value()` (django.operations.DatabaseOperations method), 4

## C

`connect()` (in module `django.database`), 3  
`convert_datefield_value()` (django.operations.DatabaseOperations method), 4  
`convert_timefield_value()` (django.operations.DatabaseOperations method), 4

## D

`data_types_reverse` (django.introspection.DatabaseIntrospection attribute), 3  
`DatabaseError`, 3  
`DatabaseFeatures` (class in `django.features`), 3  
`DatabaseIntrospection` (class in `django.introspection`), 3  
`DatabaseOperations` (class in `django.operations`), 4  
`DatabaseSchemaEditor` (class in `django.schema`), 4  
`DataError`, 3  
`django` (module), 4  
`django.compiler` (module), 2  
`django.database` (module), 3  
`django.features` (module), 3  
`django.introspection` (module), 3  
`django.operations` (module), 4  
`django.schema` (module), 4

## E

`Error`, 3

## G

`get_constraints()` (django.introspection.DatabaseIntrospection method), 3  
`get_db_converters()` (django.operations.DatabaseOperations method), 4  
`get_indexes()` (django.introspection.DatabaseIntrospection method), 4  
`get_key_columns()` (django.introspection.DatabaseIntrospection method), 4  
`get_relations()` (django.introspection.DatabaseIntrospection method), 4  
`get_table_description()` (django.introspection.DatabaseIntrospection method), 4  
`get_table_list()` (django.introspection.DatabaseIntrospection method), 4

## H

`has_bulk_insert` (django.features.DatabaseFeatures attribute), 3  
`has_native_uuid_field` (django.features.DatabaseFeatures attribute), 3

## I

`IntegrityError`, 3  
`InterfaceError`, 3  
`InternalError`, 3

## M

`max_name_length()` (django.operations.DatabaseOperations method), 4

## N

`no_limit_value()` (django.operations.DatabaseOperations method), 4  
`NotSupportedError`, 3

## O

`OperationalError`, 3

## P

ProgrammingError, 3

## Q

quote\_name() (django.operations.DatabaseOperations  
method), 4

## S

SAMPLE\_SIZE (django.introspection.DatabaseIntrospection  
attribute), 3

sql\_flush() (django.operations.DatabaseOperations  
method), 4

SQLCompiler (class in django.compiler), 2

supports\_transactions (django.features.DatabaseFeatures  
attribute), 3

## T

TYPE\_MAPPING (django.introspection.DatabaseIntrospection  
attribute), 3