
django-teledex Documentation

Release 0.2.0

Alex Hayes

September 16, 2015

1	Contents	3
1.1	Installation	3
1.2	Usage	3
1.3	Developer Documentation	6
1.4	Internal Module Reference	7
2	License	11
3	Author	13
	Python Module Index	15

Simple storage of addresses, phone numbers and emails in [Django Models](#).

django-teledex supports Python 2.7, 3.3, 3.4 and pypy for Django 1.7 and 1.8.

Contents

1.1 Installation

You can install django-teledex either via the Python Package Index (PyPI) or from github.

To install using pip;

```
$ pip install django-teledex
```

From github;

```
$ pip install git+https://github.com/alexhayes/django-teledex.git
```

Then place `django_teledex` in your `INSTALLED_APPS`;

```
INSTALLED_APPS = (  
    ...  
    'django_teledex',  
    ...  
)
```

1.2 Usage

There are three models `Address`, `PhoneNumber` and `Email` which all present essentially the same API.

For the examples below let's assume you have the following model;

```
# models.py  
  
from django.db import models  
from django_teledex.fields import AddressRelation, PhoneNumberRelation, EmailRelation  
  
class Company(models.Model):  
    title = models.CharField(max_length=100)  
    addresses = AddressRelation('companies')  
    phonenumber = PhoneNumberRelation('companies')  
    emails = EmailRelation('companies')
```

1.2.1 Address

If you want to add an address you can then do the following;

```
from django_teledex.models import Address
from django_teledex.choices import AddressKind

company = Company.objects.create(title='Evelyn Hotel')

address = Address.objects.create(
    organisation=company.title,
    kind=AddressKind.physical,
    owner=company,
    address_line='351 Brunswick St',
    locality='Fitzroy',
    region='VIC',
    postcode='3065',
    country='AU',
)

company.addresses.all() # returns all addresses for company
company.addresses.active() # all active addresses

# Make an address inactive
address.deactivate()

company.addresses.inactive() # all inactive addresses
company.addresses.kind(AddressKind.physical) # get addresses by kind
company.addresses.filter(postcode=3065) # filter works as you'd expect...

# Make an address active
address.activate()
```

A complete list of QuerySet methods available on `Address.objects` is available in `django_teledex.models.AddressQuerySet`.

1.2.2 PhoneNumber

The `PhoneNumber` model behaves in pretty much the same way, for example;

```
from django_teledex.models import PhoneNumber
from django_teledex.choices import PhoneNumberKind

company = Company.objects.create(title='Evelyn Hotel')

phonenumber = PhoneNumber.objects.create(
    kind=PhoneNumberKind.mobile,
    owner=company,
    number='+61 3 9419 5500'
)

company.phonenumbers.all() # returns all phone numbers for company
company.phonenumbers.active() # all active phone numbers

# Make an phone number inactive
phonenumber.deactivate()
```



```
company.phonenumbers.inactive() # all inactive addresses
company.phonenumbers.kind(PhoneNumberKind.mobile) # by kind
company.phonenumbers.filter(kind=PhoneNumberKind.mobile) # filter works as you'd expect...

# Make an phone number active
phonenumber.activate()
```

A complete list of QuerySet methods available on `PhoneNumber.objects` is available in `django_teledex.models.PhoneNumberQuerySet`.

Validation

`PhoneNumber` is a `PhoneNumberField` which comes from `django-phonenummer-field` which;

is a library which interfaces with `python-phonenumbers` to validate, pretty print and convert phone numbers. `python-phonenumbers` is a port of Google's `libphonenumber` library, which powers Android's phone number handling.

You'll probably want to look into the above to get familiar with how they are useful to your project.

1.2.3 Email

The `Email` model also behaves in pretty much the same way, for example;

```
from django_teledex.models import Email
from django_teledex.choices import EmailKind

company = Company.objects.create(title='Evelyn Hotel')

email = Email.objects.create(
    kind=EmailKind.work,
    owner=company,
    email='guys@example.com'
)

company.emails.all() # returns all phone numbers for company
company.emails.active() # all active phone numbers

# Make an phone number inactive
email.deactivate()

company.emails.inactive() # all inactive addresses
company.emails.kind(EmailKind.work) # by kind
company.emails.filter(email__icontains='guys@') # filter works as you'd expect...

# Make an phone number active
email.activate()
```

A complete list of QuerySet methods available on `Email.objects` is available in `django_teledex.models.EmailQuerySet`.

1.2.4 Reverse Relations

You can also traverse back from an `Address`, `PhoneNumber` or `Email` to the owner, in this case the `Company` - all thanks to Django's `reverse generic relations`.

`django_teledex.fields.AddressRelation`, `django_teledex.fields.PhoneNumberRelation` and `django_teledex.fields.EmailRelation` are simply helper classes that inherit from `GenericRelation` that set some defaults.

The first, and only required, argument to each of the `*Relation` classes is the `related_query_name` used by the `GenericRelation` which `django_teledex.fields.AddressRelation`, `django_teledex.fields.PhoneNumberRelation` and `django_teledex.fields.EmailRelation` inherit from. In the `Company` model above it's set to `companies`.

Thus;

```
# Reverse relations
Address.objects.filter(companies__title='Evelyn Hotel')
```

1.3 Developer Documentation

1.3.1 Contributions

Contributions are more than welcomed!

To get setup do the following;

```
mkvirtualenv --python=/usr/bin/python3 django-teledex
git clone https://github.com/alexhayes/django-teledex.git
cd django-teledex
pip install -r requirements/dev.txt
```

Note that you don't have to use Python 3 and indeed tox tests for many versions of Python, but you may as well develop on it, what reason is there not to for such a small application?!

1.3.2 Running Tests

Once you've checked out you should be able to run the tests.

```
detox
```

or, alternatively;

```
./manage.py test
```

1.3.3 Migrations

If you need to make modelling changes please run `makemigrations` so that the migration is included in your pull request.

```
./manage.py makemigrations
```

1.3.4 Creating translations

Translations are welcomed! Please fork and then from the root

```
cd django-teledex
./../manage.py makemessages -l [LOCALE-NAME]
```

Then, edit the translations in `django_teledex/locale`, then;

```
./../manage.py compilemessages
```

1.3.5 Creating Documentation

```
cd docs
make clean html
```

1.4 Internal Module Reference

Release 0.2.0

Date September 16, 2015

1.4.1 `django_teledex.choices`

Define choices that can be used within django-teledex.

1.4.2 `django_teledex.fields`

Defines helper classes used for defining `GenericRelation` on models.

See [Usage](#).

class `django_teledex.fields.AddressRelation`(*related_query_name*, ***kwargs*)
 Bases: `django_teledex.fields.BaseRelation`

Helper used to map your own model to an `django_teledex.models.Address`

This class can be used to setup the `GenericRelation` between your own model and an `django_teledex.models.Address`.

class `django_teledex.fields.BaseRelation`(*to*, *related_query_name*, ***kwargs*)
 Bases: `django.contrib.contenttypes.fields.GenericRelation`

Helper class useful for automatically setting `content_type_field` and `object_id_field`.

class `django_teledex.fields.EmailRelation`(*related_query_name*, ***kwargs*)
 Bases: `django_teledex.fields.BaseRelation`

Helper used to map your own model to an `django_teledex.models.Email`

This class can be used to setup the `GenericRelation` between your own model and an `django_teledex.models.Email`.

class `django_teledex.fields.PhoneNumberRelation`(*related_query_name*, ***kwargs*)
 Bases: `django_teledex.fields.BaseRelation`

Helper used to map your own model to an `django_teledex.models.PhoneNumber`

This class can be used to setup the `GenericRelation` between your own model and an `django_teledex.models.PhoneNumber`.

1.4.3 teledex.models

Django models for django-teledex.

```
class django_teledex.models.Address (*args, **kwargs)
    Bases: django.db.models.base.Model

    Defines an Address.

    Sigh... if only we had a port of libaddressinput in Python... :(

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception Address.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    Address.country = <django_countries.fields.CountryField>

    Address.get_country_display (*moreargs, **morekwargs)

    Address.get_kind_display (*moreargs, **morekwargs)

    Address.get_status_display (*moreargs, **morekwargs)

    Address.joined()
        Join the address into a single line string separated by commas.

        Return type str

    Address.objects = <django.db.models.manager.ManagerFromAddressQuerySet object>

    Address.owner
        Provides a generic relation to any object through content-type/object-id fields.

    Address.owner_type

class django_teledex.models.AddressQuerySet (model=None, query=None, using=None,
                                             hints=None)
    Bases: django.db.models.query.QuerySet

    Address QuerySet.

    active()
        Filter for active Addresses

        Return type AddressQuerySet

    inactive()
        Filter for inactive Addresses

        Return type AddressQuerySet

    kind(kind)
        Filter addresses by a particular kind.

        Accepts any of the

        Return type AddressQuerySet

    physical()
        Filter for physical addresses.

        Return type AddressQuerySet

    postal()
        Filter for postal addresses.
```

Return type *AddressQuerySet*

```

class django_teledex.models.Email(*args, **kwargs)
    Bases: django.db.models.base.Model

    Defines a Phone Number.

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception Email.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    Email.get_kind_display(*moreargs, **morekwargs)

    Email.get_status_display(*moreargs, **morekwargs)

    Email.objects = <django.db.models.manager.ManagerFromEmailQuerySet object>

    Email.owner
        Provides a generic relation to any object through content-type/object-id fields.

    Email.owner_type

class django_teledex.models.EmailQuerySet(model=None, query=None, using=None,
                                           hints=None)
    Bases: django.db.models.query.QuerySet

    Email QuerySet.

    active()
        Filter for active Phone Numbers

    inactive()
        Filter for inactive Phone Numbers

    kind(kind)

class django_teledex.models.PhoneNumber(*args, **kwargs)
    Bases: django.db.models.base.Model

    Defines a Phone Number.

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception PhoneNumber.MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    PhoneNumber.get_kind_display(*moreargs, **morekwargs)

    PhoneNumber.get_status_display(*moreargs, **morekwargs)

    PhoneNumber.number = <phonenumber_field.modelfields.PhoneNumberField>

    PhoneNumber.objects = <django.db.models.manager.ManagerFromPhoneNumberQuerySet object>

    PhoneNumber.owner
        Provides a generic relation to any object through content-type/object-id fields.

    PhoneNumber.owner_type

class django_teledex.models.PhoneNumberQuerySet(model=None, query=None, using=None,
                                                  hints=None)
    Bases: django.db.models.query.QuerySet

    PhoneNumber QuerySet.

```

active ()

Filter for active Phone Numbers

inactive ()

Filter for inactive Phone Numbers

kind (*kind*)

License

This software is licensed under the *MIT License*. See the [LICENSE](#) file in the top distribution directory for the full license text.

Author

Alex Hayes <alex@alution.com>

d

`django_teledex.choices`, [7](#)
`django_teledex.fields`, [7](#)
`django_teledex.models`, [7](#)

A

active() (django_teledex.models.AddressQuerySet method), 8
active() (django_teledex.models.EmailQuerySet method), 9
active() (django_teledex.models.PhoneNumberQuerySet method), 9
Address (class in django_teledex.models), 8
Address.DoesNotExist, 8
Address.MultipleObjectsReturned, 8
AddressQuerySet (class in django_teledex.models), 8
AddressRelation (class in django_teledex.fields), 7

B

BaseRelation (class in django_teledex.fields), 7

C

country (django_teledex.models.Address attribute), 8

D

django_teledex.choices (module), 7
django_teledex.fields (module), 7
django_teledex.models (module), 7

E

Email (class in django_teledex.models), 9
Email.DoesNotExist, 9
Email.MultipleObjectsReturned, 9
EmailQuerySet (class in django_teledex.models), 9
EmailRelation (class in django_teledex.fields), 7

G

get_country_display() (django_teledex.models.Address method), 8
get_kind_display() (django_teledex.models.Address method), 8
get_kind_display() (django_teledex.models.Email method), 9
get_kind_display() (django_teledex.models.PhoneNumber method), 9

get_status_display() (django_teledex.models.Address method), 8
get_status_display() (django_teledex.models.Email method), 9
get_status_display() (django_teledex.models.PhoneNumber method), 9

I

inactive() (django_teledex.models.AddressQuerySet method), 8
inactive() (django_teledex.models.EmailQuerySet method), 9
inactive() (django_teledex.models.PhoneNumberQuerySet method), 10

J

joined() (django_teledex.models.Address method), 8

K

kind() (django_teledex.models.AddressQuerySet method), 8
kind() (django_teledex.models.EmailQuerySet method), 9
kind() (django_teledex.models.PhoneNumberQuerySet method), 10

N

number (django_teledex.models.PhoneNumber attribute), 9

O

objects (django_teledex.models.Address attribute), 8
objects (django_teledex.models.Email attribute), 9
objects (django_teledex.models.PhoneNumber attribute), 9
owner (django_teledex.models.Address attribute), 8
owner (django_teledex.models.Email attribute), 9
owner (django_teledex.models.PhoneNumber attribute), 9
owner_type (django_teledex.models.Address attribute), 8

owner_type (django_teledex.models.Email attribute), 9
owner_type (django_teledex.models.PhoneNumber attribute), 9

P

PhoneNumber (class in django_teledex.models), 9
PhoneNumber.DoesNotExist, 9
PhoneNumber.MultipleObjectsReturned, 9
PhoneNumberQuerySet (class in django_teledex.models), 9
PhoneNumberRelation (class in django_teledex.fields), 7
physical() (django_teledex.models.AddressQuerySet method), 8
postal() (django_teledex.models.AddressQuerySet method), 8