

---

# **radioco Documentation**

*Release 3.2.2*

**Iago Veloso Abalo**

**Jul 13, 2017**



---

## Contents

---

<b>1</b>	<b>Overview</b>	<b>1</b>
<b>2</b>	<b>Release Notes</b>	<b>3</b>
<b>3</b>	<b>Table of contents</b>	<b>5</b>
3.1	Installation Guide . . . . .	5
3.2	Customize . . . . .	7
3.3	API . . . . .	9
3.4	Development & community . . . . .	10
3.5	django-radio . . . . .	12
3.6	Release notes & upgrade information . . . . .	49
3.7	Indices and tables . . . . .	56
	<b>Python Module Index</b>	<b>57</b>



# CHAPTER 1

---

## Overview

---

RadioCo is a broadcasting radio recording scheduling system. RadioCo has been intended to provide a solution for a wide range of broadcast projects, from community to public and commercial stations.

Here are a few of the key features:

- designed to work with any web browser
- drag and drop scheduling calendar interface
- live shows can be recorded and published automatically
- complete authentication system (user accounts, groups, permissions)
- ...and much more

---

**Note:** This software is not for live streaming

---



## CHAPTER 2

---

### Release Notes

---

This document refers to version 3.2.2





## Installation Guide

The pages in this section of the documentation are designed to help you get started quickly and show how easy it is to work with RadioCo

The guides follow a logical progression and build on each other, so it's recommended to work through them in the order presented here.

### Installing web application

To allow RadioCo to generate correct dates it's necessary to set the timezone variable:

1. Find your timezone in [this list](#)
2. Go to the config folder (radioco/configs/base) and create if not exists a local\_settings.py in the same directory than settings.py
3. Add to the local settings the variable timezone, for example: `TIME_ZONE = "Europe/Madrid"`

### Installing locally

This tutorial is written for Python 2.7 and Ubuntu 12.04 or later.

#### Ubuntu

The easiest way of installing the app is using [Docker engine](#), follow the [installation steps](#) to install Docker.

Open a terminal and introduce the following commands:

```
sudo apt-get install git-core python-dev python-pip
```

Next, download the project and cd into it:

```
git clone https://github.com/iago1460/django-radio
cd django-radio
```

Install the python invoke library:

```
pip install invoke==0.14.0
```

Execute the next command to deploy the app in docker, this step take some time:

```
inv quickstart
```

**Warning:** If you have faced the error “ValueError: unknown locale: UTF-8” on MacOS X, execute:  
export LC\_ALL=en\_US.UTF-8 export LANG=en\_US.UTF-8

### Testing

Let’s verify your installation

Now that the server’s running, visit <http://127.0.0.1:8000/>

**Warning:** Don’t use this server in anything resembling a production environment.

### Using RadioCo on production

The Internet is a hostile environment. Before deploying this project, you should take some time to review your settings, with security, performance, and operations in mind. Keep in mind [this critical settings](#).

### Locally

RadioCo provides a staging environment, safer than the previous one but still insecure, **use at your own risk**.

```
inv docker.setup -e staging
```

Now that the server’s running, visit <http://127.0.0.1/>

To create a superuser you still can use management commands:

```
inv docker.manage -e staging -c "createsuperuser"
```

### Installing recorder program

This tutorial is written for Python 2.7 and Ubuntu 12.04 or later.

## Installing on Ubuntu

We'll get started by setting up our environment.

```
sudo apt-get install python-dev python-pip python-virtualenv git-core alsa-utils  
↳vorbis-tools
```

Next, download the project and create the virtual environment:

```
git clone https://github.com/iago1460/django-radio-recorder.git  
cd django-radio-recorder
```

Create and activate a virtual env:

```
virtualenv venv  
source venv/bin/activate
```

Install the requirements:

```
pip install -r requirements.txt
```

Using your favorite text editor, configure the `settings.ini` file

Launch the program

```
python main.py
```

## Communication Setup

In your Admin interface go to **Podcast Configuration**, copy the **Recorder token** and put it into the Recorder Program settings:

```
token:8fdde6d703c05773084ea83e5ec2da62637666a0 #for example
```

Modify the **url** in your Recorder Program settings:

```
url:http://yourdomain:80/api/1/
```

## Customize

Technical reference material.

RadioCo has a number of settings to configure its behaviour, please read this section carefully.

## Application Setup

RadioCo can be adapted to your needs, you have three ways to do it. Editing the Global Settings section in the administration page using your browser or manually overriding the `settings.py` and overriding templates.

### Global settings

Go to the admin section on your browser and edit the information available.

### Global Configuration

In this section you can add information related to your site apart of the google analytics id.

### Calendar Configuration

Settings related to the calendar, things like the first day of the week

### Podcast Configuration

Settings related with the recorder, change here recording delays and get the Recorder token necessary for the recorder programme to work

### Site

Change `example.com` with the fully qualified domain name associated with the website. Note that some RSS feed attributes will be incorrect if this value is not properly set up.

### Templates

There is a empty folder called templates inside the radioco folder. You should override templates here, make sure to keep the relative path.

For example, to override the episode detail page copy the `episode_detail.html` file from `radioco/apps/programmes/templates/programmes/episode_detail.html` to `radioco/templates/programmes/episode_detail.html`

### Settings.py

These settings are available in `settings.py`. Your settings should be in a `local_settings.py` file in the same directory as `settings.py`.

**Warning:** Your changes on settings should be in `local_settings.py` to avoid conflicts when update, create that file if it's necessary. Be aware that this file is excluded from Git.

### USERNAME\_RADIOCO\_RECORDER

This specifies who is the user of the recorder program:

```
USERNAME_RADIOCO_RECORDER = 'RadioCo_Recorder'
```

---

**Note:** It's a good idea change this value for security reasons.

---

## LANGUAGE\_CODE

A string representing the language code for this installation. It provides a fallback language in case the user's preferred language can't be determined or is not supported by the website. [More info](#):

```
LANGUAGE_CODE = 'en'
```

## PROGRAMME\_LANGUAGES

*New in version 1.1*

Default: A tuple of the following three languages.

This specifies which languages are available for language selection in your programmes:

```
gettext_noop = lambda s: s

PROGRAMME_LANGUAGES = (
    ('es', gettext_noop('Spanish')),
    ('en', gettext_noop('English')),
    ('gl', gettext_noop('Galician')),
)
```

You can see the current list of translated languages by looking in `django/conf/global_settings.py` (or view the [online source](#)).

## Disqus

*New in version 2.0*

Default: Disabled by default.

Add comments to your site with Disqus. [Create your account and get your API key.](#):

```
DISQUS_ENABLE = True
DISQUS_API_KEY = 'YOUR_API_KEY'
DISQUS_WEBSITE_SHORTNAME = 'YOUR_SHORTNAME'
```

## Recorder Program Setup

The settings are available in your `settings.ini`.

## API

Api documentation.

## Browsable API

*Experimental in version 3.0*

RadioCo has a Web browsable API, go to `http://127.0.0.1:8000/api/2/` in your browser to explore it.

Optionally these endpoints support filtering and ordering in the majority of the exposed fields.

### Programmes

Programmes can be filter using after and before parameters as well as Transmissions.

Example query to get all available programmes on New Year's Eve order by name:

```
http://127.0.0.1:8000/api/2/programmes?after=2016-12-31&before=2016-12-31&
→ordering=name
```

### Transmissions

Transmissions are always ordered by date, the after and before parameters are required.

Example query:

```
http://127.0.0.1:8000/api/2/transmissions?after=2016-12-19&before=2016-12-26
```

Also is possible to request the dates in a specific timezone:

```
http://127.0.0.1:8000/api/2/transmissions?timezone=Europe%2FMadrid&after=2016-12-19&
→before=2016-12-26
```

Finally, there is a endpoint to get the current transmission:

```
http://127.0.0.1:8000/api/2/transmissions/now
```

## Radiocom API

*New in version 3.1*

**Radiocom** is an Android application for Community Media Stations. RadioCo has the **Radiocom API** under the following urls:

```
http://127.0.0.1:8000/api/2/radiocom/programmes
http://127.0.0.1:8000/api/2/radiocom/radiostation
http://127.0.0.1:8000/api/2/radiocom/transmissions
http://127.0.0.1:8000/api/2/radiocom/transmissions/now
```

## Development & community

RadioCo is an open-source project, and relies on its community of users to keep getting better.

You don't need to be an expert developer to make a valuable contribution - all you need is a little knowledge of the system, and a willingness to follow the contribution guidelines.

Remember that contributions to the documentation are highly prized, and key to the success of the project. Any time and effort you are willing to contribute is greatly appreciated!

## Branch policy

- **master**: this is the current stable release, the version released on PyPI.
- **develop**: this branch always reflects a state with the latest delivered development changes for the next release.
- **feature branches**: these are used to develop new features.

## Contributing Translation

For translators we have a [Transifex account](#) where you can translate the .po files and don't need to install git to be able to contribute. All changes there will be automatically sent to the project.

## Contributing Documentation

Perhaps considered “boring” by hard-core coders, documentation is sometimes even more important than code! This is what brings fresh blood to a project, and serves as a reference for old timers. On top of this, documentation is the one area where less technical people can help most - you just need to write semi-decent English. People need to understand you. We don't care about style or correctness.

Documentation should be:

- written using valid [Sphinx/restructuredText](#) syntax (see below for specifics) and the file extension should be `.rst`
- written in English (we have standardised on British spellings)
- accessible - you should assume the reader to be moderately familiar with Python and Django, but not anything else. Link to documentation of libraries you use, for example, even if they are “obvious” to you

Merging documentation is pretty fast and painless.

Except for the tiniest of change, we recommend that you test them before submitting.

## Documentation markup

### Sections

We use Python documentation conventions for section marking:

- # with overline, for parts
- \* with overline, for chapters
- =, for sections
- -, for subsections
- ^, for subsubsections
- ", for paragraphs

### Inline markup

- **use backticks - `settings.py` - for:**
  - literals

- filenames
- names of fields and other items in the Admin interface:
- **use emphasis - \*Home\* around:**
  - the names of available options in the Admin
  - values in or of fields

### References

Use absolute links to other documentation pages - `:doc:`/how_to/toolbar`` - rather than relative links - `:doc:`../toolbar``. This makes it easier to run search-and-replaces when items are moved in the structure.

## django-radio

### manage module

### radioco package

#### Subpackages

#### radioco.apps package

#### Subpackages

#### radioco.apps.api package

#### Subpackages

#### radioco.apps.api.tests package

#### Submodules

#### radioco.apps.api.tests.test\_api module

```
class radioco.apps.api.tests.test_api.TestApi
    Bases: radioco.apps.radioco.test_utils.TestDataMixin
    test_api()

class radioco.apps.api.tests.test_api.TestSerializers (methodName='runTest')
    Bases: radioco.apps.radioco.test_utils.TestDataMixin, django.test.testcases.TestCase
    test_episode()
    test_episode_programme()
    test_programme()
    test_programme_photo_url()
```



```
test_schedule()
test_transmission()
```

### **radioco.apps.api.tests.test\_fullcalendar module**

```
class radioco.apps.api.tests.test_fullcalendar.TestFullCalendarApi (methodName='runTest')
  Bases: radioco.apps.radioco.test_utils.TestDataMixin, rest_framework.test.APITestCase

  setUp()

  test_move_schedule (*args, **kwargs)
  test_move_schedule_in_tz (*args, **kwargs)
  test_move_schedule_with_schedules (*args, **kwargs)
  test_move_schedule_with_schedules_in_tz (*args, **kwargs)
  test_schedules_post (*args, **kwargs)
  test_schedules_post_in_tz (*args, **kwargs)
```

### **radioco.apps.api.tests.test\_programmes module**

```
class radioco.apps.api.tests.test_programmes.TestNotAllowedMethodsProgrammesAPI (methodName='runTest')
  Bases: radioco.apps.radioco.test_utils.TestDataMixin, rest_framework.test.APITestCase

  test_episodes_delete()
  test_episodes_post()
  test_episodes_put()
  test_programmes_delete()
  test_programmes_post()
  test_programmes_put()

class radioco.apps.api.tests.test_programmes.TestProgrammesAPI (methodName='runTest')
  Bases: radioco.apps.radioco.test_utils.TestDataMixin, rest_framework.test.APITestCase

  setUp()

  test_episodes_get_all()
  test_episodes_get_by_programme()
  test_programmes_after()
  test_programmes_before()
  test_programmes_between()
  test_programmes_get_all()
```

### radioco.apps.api.tests.test\_radiocom\_api module

```
class radioco.apps.api.tests.test_radiocom_api.TestProgramme (methodName='runTest')
    Bases: radioco.apps.radioco.test_utils.TestDataMixin, rest_framework.test.APITestCase

    setUp()

    test_programme_in_list()

class radioco.apps.api.tests.test_radiocom_api.TestTransmission (methodName='runTest')
    Bases: radioco.apps.radioco.test_utils.TestDataMixin, rest_framework.test.APITestCase

    test_transmission_now(*args, **kwargs)

radioco.apps.api.tests.test_radiocom_api.mock_now()
```

### radioco.apps.api.tests.test\_recorder module

```
class radioco.apps.api.tests.test_recorder.PodcastMock

class radioco.apps.api.tests.test_recorder.TestProgrammesAPI (methodName='runTest')
    Bases: rest_framework.test.APITestCase

    classmethod setUpTestData()

    test_recording_schedules(*args, **kwargs)

    test_submit_recorder(*args, **kwargs)
```

### radioco.apps.api.tests.test\_schedules module

```
class radioco.apps.api.tests.test_schedules.TestRestrictedMethodsScheduleAPI (methodName='runTest')
    Bases: radioco.apps.radioco.test_utils.TestDataMixin, rest_framework.test.APITestCase

    test_schedules_delete()

    test_schedules_post()

    test_schedules_put()

    test_transmissions_delete()

    test_transmissions_post()

    test_transmissions_put()

class radioco.apps.api.tests.test_schedules.TestSchedulesAPI (methodName='runTest')
    Bases: radioco.apps.radioco.test_utils.TestDataMixin, rest_framework.test.APITestCase

    test_schedules_get_all()

    test_schedules_get_by_calendar()

    test_schedules_get_by_nonexisting_calendar()

    test_schedules_get_by_nonexisting_programme()

    test_schedules_get_by_nonexisting_type()
```

```
test_schedules_get_by_programme()
```

```
test_schedules_get_by_type()
```

```
class radioco.apps.api.tests.test_schedules.TestTransmissionAPI (methodName='runTest')
Bases: radioco.apps.radioco.test_utils.TestDataMixin, rest_framework.test.APITestCase
```

```
test_incorrect_transmission_queries()
```

```
test_transmission_before(*args, **kwargs)
```

```
test_transmission_list_filter_non_active_calendar(*args, **kwargs)
```

```
test_transmission_now(*args, **kwargs)
```

```
test_transmission_same_day()
```

```
test_transmissions_between(*args, **kwargs)
```

```
test_transmissions_between_requesting_tz()
```

```
test_transmissions_filter_calendar_nonexistend()
```

```
radioco.apps.api.tests.test_schedules.mock_now()
```

## Module contents

### Submodules

#### radioco.apps.api.recorder\_views module

```
radioco.apps.api.recorder_views.check_recorder_program(user)
```

```
radioco.apps.api.recorder_views.recording_schedules(*args, **kwargs)
```

```
radioco.apps.api.recorder_views.submit_recorder(*args, **kwargs)
```

#### radioco.apps.api.serializers module

```
class radioco.apps.api.serializers.AbsoluteURLField (method_name=None, source=None,
reverse_url=None, **kwargs)
```

```
Bases: rest_framework.fields.URLField
```

A field similar to URLField but returning always the absolute url

For example:

```
class ExampleSerializer(self): episode_url = AbsoluteURLField() photo_url = AbsoluteURL-
Field(source='photo.url', read_only=True) rss_url = AbsoluteURLField(source='slug', re-
verse_url='programmes:rss')
```

```
to_representation(value)
```

```
class radioco.apps.api.serializers.DateTimeFieldTz (format=<class
rest_framework.fields.empty>,
input_formats=None, de-
fault_timezone=None, *args,
**kwargs)
```

```
Bases: rest_framework.fields.DateTimeField
```

Field to display the datetime in the current timezone

**to\_representation** (*date*)

```
class radioco.apps.api.serializers.EpisodeSerializer (instance=None, data=<class
rest_framework.fields.empty>,
**kwargs)
```

Bases: rest\_framework.serializers.ModelSerializer

**class Meta**

**fields** = ('title', 'programme', 'summary', 'issue\_date', 'season', 'number\_in\_season')

**model**

alias of Episode

```
class radioco.apps.api.serializers.ProgrammeSerializer (instance=None, data=<class
rest_framework.fields.empty>,
**kwargs)
```

Bases: rest\_framework.serializers.ModelSerializer

**class Meta**

**fields** = ('id', 'slug', 'name', 'synopsis', 'runtime', 'photo\_url', 'rss\_url', 'language', 'category')

**model**

alias of Programme

```
class radioco.apps.api.serializers.RadiocomConfigurationSerializer (instance=None,
data=<class
rest_framework.fields.empty>,
**kwargs)
```

Bases: rest\_framework.serializers.ModelSerializer

**class Meta**

**fields** = ('station\_name', 'icon\_url', 'big\_icon\_url', 'station\_photos', 'history', 'latitude', 'longitude', 'news\_rss', '')

**model**

alias of RadiocomConfiguration

RadiocomConfigurationSerializer.**get\_facebook\_url** (*obj*)

RadiocomConfigurationSerializer.**get\_station\_photos** (*obj*)

RadiocomConfigurationSerializer.**get\_twitter\_url** (*obj*)

```
class radioco.apps.api.serializers.RadiocomProgrammeSerializer (instance=None,
data=<class
rest_framework.fields.empty>,
**kwargs)
```

Bases: rest\_framework.serializers.ModelSerializer

**class Meta**

**fields** = ('id', 'name', 'description', 'logo\_url', 'rss\_url')

**model**

alias of Programme

```
class radioco.apps.api.serializers.RadiocomTransmissionSerializer (instance=None,
                                                                    data=<class
                                                                    rest_framework.fields.empty>,
                                                                    **kwargs)
```

```
    Bases: rest_framework.serializers.Serializer
```

```
class radioco.apps.api.serializers.ScheduleSerializer (instance=None,    data=<class
                                                                    rest_framework.fields.empty>,
                                                                    **kwargs)
```

```
    Bases: rest_framework.serializers.ModelSerializer
```

```
class Meta
```

```
    fields = ('id', 'programme', 'calendar', 'start', 'runtime', 'title', 'type', 'source')
```

```
    model
```

```
        alias of Schedule
```

```
    ScheduleSerializer.get_title (schedule)
```

```
    ScheduleSerializer.validate (attrs)
```

```
class radioco.apps.api.serializers.TransmissionSerializer (instance=None,
                                                            data=<class
                                                            rest_framework.fields.empty>,
                                                            **kwargs)
```

```
    Bases: rest_framework.serializers.Serializer
```

```
class radioco.apps.api.serializers.TransmissionSerializerLight (instance=None,
                                                                data=<class
                                                                rest_framework.fields.empty>,
                                                                **kwargs)
```

```
    Bases: rest_framework.serializers.Serializer
```

```
    validate (attrs)
```

## radioco.apps.api.urls module

## radioco.apps.api.views module

```
class radioco.apps.api.views.EpisodeFilter (*args, **kwargs)
```

```
    Bases: rest_framework.filters.FilterSet
```

```
class Meta
```

```
    fields = ('programme',)
```

```
    model
```

```
        alias of Episode
```

```
    EpisodeFilter.base_filters = OrderedDict([(u'programme', <django_filters.filters.CharFilter object>)])
```

```
    EpisodeFilter.declared_filters = OrderedDict([(u'programme', <django_filters.filters.CharFilter object>)])
```

```
class radioco.apps.api.views.EpisodeViewSet (**kwargs)
```

```
    Bases: rest_framework.viewsets.ReadOnlyModelViewSet
```

```
    filter_backends = (<class 'rest_framework.filters.DjangoFilterBackend'>, <class 'rest_framework.filters.OrderingFI
```

```
filter_class
    alias of EpisodeFilter

queryset

serializer_class
    alias of EpisodeSerializer

suffix = None

class radioco.apps.api.views.ProgrammeFilter (*args, **kwargs)
    Bases: rest_framework.filters.FilterSet

class Meta

    fields = ('name', 'category')

    model
        alias of Programme

    ProgrammeFilter.base_filters = OrderedDict([(u'name', <django_filters.filters.CharFilter object>), (u'category', <django_filters.filters.CharFilter object>)])

    ProgrammeFilter.declared_filters = OrderedDict()

class radioco.apps.api.views.ProgrammeFilterForm (data=None, files=None,
                                                auto_id=u'id_%s', prefix=None,
                                                initial=None, error_class=<class
                                                'django.forms.utils.ErrorList'>,
                                                label_suffix=None,
                                                empty_permitted=False)

    Bases: django.forms.forms.Form

base_fields = OrderedDict([(u'after', <django.forms.fields.DateField object>), (u'before', <django.forms.fields.DateField object>)])

clean()

declared_fields = OrderedDict([(u'after', <django.forms.fields.DateField object>), (u'before', <django.forms.fields.DateField object>)])

media

class radioco.apps.api.views.ProgrammeViewSet (**kwargs)
    Bases: rest_framework.viewsets.ModelViewSet

filter_backends = (<class 'rest_framework.filters.DjangoFilterBackend'>, <class 'rest_framework.filters.OrderingFilter'>)

filter_class
    alias of ProgrammeFilter

list (request, *args, **kwargs)

lookup_field = 'slug'

permission_classes = (<class 'rest_framework.permissions.DjangoModelPermissionsOrAnonReadOnly'>),

queryset

serializer_class
    alias of ProgrammeSerializer

suffix = None

class radioco.apps.api.views.RadiocomProgrammeViewSet (**kwargs)
    Bases: radioco.apps.api.views.ProgrammeViewSet

serializer_class
    alias of RadiocomProgrammeSerializer
```

```

suffix = None
class radioco.apps.api.views.RadiocomStation (**kwargs)
    Bases: rest_framework.viewsets.ReadOnlyModelViewSet

    list (request, *args, **kwargs)

    serializer_class
        alias of RadiocomConfigurationSerializer

    suffix = None
class radioco.apps.api.views.RadiocomTransmissionViewSet (**kwargs)
    Bases: radioco.apps.api.views.TransmissionViewSet

    serializer_class
        alias of RadiocomTransmissionSerializer

    suffix = None
class radioco.apps.api.views.ScheduleFilter (*args, **kwargs)
    Bases: rest_framework.filters.FilterSet

    class Meta

        fields = ('programme', 'calendar', 'type')

        model
            alias of Schedule

    ScheduleFilter.base_filters = OrderedDict([(u'programme', <django_filters.filters.CharFilter object>), (u'calendar', <django_filters.filters.CharFilter object>)])
    ScheduleFilter.declared_filters = OrderedDict([('programme', <django_filters.filters.CharFilter object>)])
class radioco.apps.api.views.ScheduleViewSet (**kwargs)
    Bases: rest_framework.viewsets.ModelViewSet

    filter_backends = (<class 'rest_framework.filters.DjangoFilterBackend'>, <class 'rest_framework.filters.OrderingFilter'>)

    filter_class
        alias of ScheduleFilter

    permission_classes = (<class 'rest_framework.permissions.DjangoModelPermissionsOrAnonReadOnly'>,)

    queryset

    serializer_class
        alias of ScheduleSerializer

    suffix = None
class radioco.apps.api.views.TimezoneForm (data=None, files=None, auto_id=u'id_%s', prefix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=None, empty_permitted=False)

    Bases: django.forms.forms.Form

    base_fields = OrderedDict([('timezone', <django.forms.fields.ChoiceField object>)])

    clean_timezone ()

    declared_fields = OrderedDict([('timezone', <django.forms.fields.ChoiceField object>)])

    media

    x = 'Zulu'

```

```
class radioco.apps.api.views.TransmissionForm (data=None, files=None,
                                              auto_id=u'id_%s', prefix=None,
                                              initial=None, error_class=<class
                                              'django.forms.utils.ErrorList'>, la-
                                              bel_suffix=None, empty_permitted=False)
Bases: radioco.apps.api.views.TimezoneForm
base_fields = OrderedDict([('timezone', <django.forms.fields.ChoiceField object>), ('after', <django.forms.fields.DateField object>)]
clean ()
declared_fields = OrderedDict([('timezone', <django.forms.fields.ChoiceField object>), ('after', <django.forms.fields.DateField object>)]
media

class radioco.apps.api.views.TransmissionOperationViewSet (**kwargs)
Bases: radioco.apps.api.viewsets.UpdateOnlyModelViewSet
perform_update (serializer)
permission_classes = (<class 'rest_framework.permissions.DjangoModelPermissionsOrAnonReadOnly'>,)
queryset
serializer_class
    alias of TransmissionSerializerLight
suffix = None
update (request, *args, **kwargs)

class radioco.apps.api.views.TransmissionViewSet (**kwargs)
Bases: rest_framework.viewsets.ReadOnlyModelViewSet
filter_backends = (<class 'rest_framework.filters.DjangoFilterBackend'>,)
filter_class
    alias of ScheduleFilter
list (request, *args, **kwargs)
now (request)
queryset
serializer_class
    alias of TransmissionSerializer
suffix = None
```

## radioco.apps.api.viewsets module

```
class radioco.apps.api.viewsets.UpdateOnlyModelViewSet (**kwargs)
Bases: rest_framework.mixins.UpdateModelMixin, rest_framework.viewsets.GenericViewSet
A viewset that provides a update action.
```

## Module contents

### radioco.apps.global\_settings package



## Subpackages

### `radioco.apps.global_settings.migrations` package

## Submodules

### `radioco.apps.global_settings.migrations.0001_initial` module

```
class radioco.apps.global_settings.migrations.0001_initial.Migration(name,
                                                                    app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = []
    operations = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'scroll_time', <django.db.models.f
```

### `radioco.apps.global_settings.migrations.0002_remove_calendarconfiguration_display_next_weeks` module

```
class radioco.apps.global_settings.migrations.0002_remove_calendarconfiguration_display_next_
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'global_settings', u'0001_initial')]
    operations = [<RemoveField name=u'display_next_weeks', model_name=u'calendarconfiguration'>]
```

### `radioco.apps.global_settings.migrations.0003_siteconfiguration_footer` module

```
class radioco.apps.global_settings.migrations.0003_siteconfiguration_footer.Migration(name,
                                                                                       app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'global_settings', u'0002_remove_calendarconfiguration_display_next_weeks')]
    operations = [<AddField field=<django.db.models.fields.TextField>, preserve_default=True, name=u'footer', model_n
```

### `radioco.apps.global_settings.migrations.0004_auto_20150606_1335` module

```
class radioco.apps.global_settings.migrations.0004_auto_20150606_1335.Migration(name,
                                                                                   app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'global_settings', u'0003_siteconfiguration_footer')]
    operations = [<AddField field=<django.db.models.fields.TextField>, preserve_default=True, name=u'address', model_
```

### `radioco.apps.global_settings.migrations.0005_auto_20150606_1415` module

```
class radioco.apps.global_settings.migrations.0005_auto_20150606_1415.Migration(name,
                                                                                   app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'global_settings', u'0004_auto_20150606_1335')]
```

```
operations = [<RenameField new_name=u'about_footer', model_name=u'siteconfiguration', old_name=u'footer'>, <A
```

### radioco.apps.global\_settings.migrations.0006\_auto\_20160116\_1509 module

```
class radioco.apps.global_settings.migrations.0006_auto_20160116_1509.Migration (name, app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'global_settings', u'0005_auto_20150606_1415')]
    operations = [<AlterField field=<django.db.models.fields.CharField>, name=u'google_analytics_id', model_name=u'si
```

### radioco.apps.global\_settings.migrations.0007\_\_v3\_0\_\_calendarconfiguration\_tweaks module

```
class radioco.apps.global_settings.migrations.0007__v3_0__calendarconfiguration_tweaks.Migration
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'global_settings', u'0006_auto_20160116_1509'), (u'radioco', u'0001__v3_0__mysql_timezone')]
    operations = [<RemoveField name=u'scroll_time', model_name=u'calendarconfiguration'>, <AddField field=<django
```

### radioco.apps.global\_settings.migrations.0008\_\_v3\_0\_\_radiocomconfiguration module

```
class radioco.apps.global_settings.migrations.0008__v3_0__radiocomconfiguration.Migration (name, app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'global_settings', u'0007__v3_0__calendarconfiguration_tweaks')]
    operations = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'station_name', <django.db.model
```

## Module contents

### radioco.apps.global\_settings.templatetags package

#### Submodules

### radioco.apps.global\_settings.templatetags.global\_settings\_extras module

```
radioco.apps.global_settings.templatetags.global_settings_extras.get_global_model (model_path)
```

## Module contents

#### Submodules

### radioco.apps.global\_settings.admin module

```
class radioco.apps.global_settings.admin.PodcastConfigurationAdmin (model, admin_site)
    Bases: radioco.apps.global_settings.admin.SingletonModelAdmin
```

**media**

**readonly\_fields** = ['recorder\_token']

**class** radioco.apps.global\_settings.admin.**SingletonModelAdmin** (*model, admin\_site*)

Bases: django.contrib.admin.options.ModelAdmin

**change\_form\_template** = 'global\_settings/admin\_change\_form.html'

**change\_view** (*request, object\_id, form\_url='', extra\_context=None*)

**get\_urls** ()

**has\_add\_permission** (*request*)

**has\_delete\_permission** (*request, obj=None*)

**media**

**object\_history\_template** = 'global\_settings/admin\_object\_history.html'

**response\_change** (*request, obj*)

### radioco.apps.global\_settings.models module

**class** radioco.apps.global\_settings.models.**CalendarConfiguration** (*id, slot\_duration, first\_day, min\_time, max\_time*)

Bases: *radioco.apps.global\_settings.models.SingletonModel*

**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception CalendarConfiguration.MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

**CalendarConfiguration.get\_first\_day\_display** (*\*moreargs, \*\*morekwargs*)

**CalendarConfiguration.objects** = <radioco.apps.global\_settings.models.SingletonModelManager object>

**class** radioco.apps.global\_settings.models.**PodcastConfiguration** (*id, url\_source, start\_delay, end\_delay, next\_events*)

Bases: *radioco.apps.global\_settings.models.SingletonModel*

**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception PodcastConfiguration.MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

**PodcastConfiguration.objects** = <radioco.apps.global\_settings.models.SingletonModelManager object>

**PodcastConfiguration.recorder\_token**

```
class radioco.apps.global_settings.models.RadiocomConfiguration (id, station_name,  
icon_url,  
big_icon_url,  
history, latitude,  
longitude,  
news_rss, station_photos,  
stream_url)  
  
Bases: radioco.apps.global_settings.models.SingletonModel  
  
exception DoesNotExist  
Bases: django.core.exceptions.ObjectDoesNotExist  
  
exception RadiocomConfiguration.MultipleObjectsReturned  
Bases: django.core.exceptions.MultipleObjectsReturned  
  
RadiocomConfiguration.objects = <radioco.apps.global_settings.models.SingletonModelManager object>  
  
class radioco.apps.global_settings.models.SingletonModel (*args, **kwargs)  
Bases: django.db.models.base.Model  
  
class Meta  
  
    abstract = False  
  
SingletonModel.delete (*args, **kwargs)  
  
SingletonModel.classmethod SingletonModel.get_global ()  
  
SingletonModel.objects  
  
SingletonModel.save (*args, **kwargs)  
  
class radioco.apps.global_settings.models.SingletonModelManager  
Bases: django.db.models.manager.Manager  
  
    get (*args, **kwargs)  
  
class radioco.apps.global_settings.models.SiteConfiguration (id, site_name,  
about_footer,  
more_about_us,  
google_analytics_id,  
address, twitter_address,  
facebook_address)  
  
Bases: radioco.apps.global_settings.models.SingletonModel  
  
exception DoesNotExist  
Bases: django.core.exceptions.ObjectDoesNotExist  
  
exception SiteConfiguration.MultipleObjectsReturned  
Bases: django.core.exceptions.MultipleObjectsReturned  
  
SiteConfiguration.objects = <radioco.apps.global_settings.models.SingletonModelManager object>
```

## Module contents

### radioco.apps.programmes package



### radioco.apps.programmes.migrations.0005\_auto\_20150531\_1734 module

```
class radioco.apps.programmes.migrations.0005_auto_20150531_1734.Migration (name,
                                                                    app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'programmes', u'0004_change_photo_url')]
    operations = [<AlterField field=<django.db.models.fields.related.OneToOneField>, preserve_default=True, name=u'ep
```

### radioco.apps.programmes.migrations.0006\_auto\_20160104\_2029 module

```
class radioco.apps.programmes.migrations.0006_auto_20160104_2029.Migration (name,
                                                                    app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'programmes', u'0005_auto_20150531_1734')]
    operations = [<AlterField field=<django.db.models.fields.files.ImageField>, name=u'photo', model_name=u'programm
```

### radioco.apps.programmes.migrations.0007\_change\_default\_image module

```
class radioco.apps.programmes.migrations.0007_change_default_image.Migration (name,
                                                                    app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'programmes', u'0006_auto_20160104_2029'), (u'users', u'0003_auto_20160104_2029')]
    operations = [<RunPython <function change_programmes>>, <RunPython <function change_users>>]
radioco.apps.programmes.migrations.0007_change_default_image.change_programmes (apps,
                                                                    schema_editor)
radioco.apps.programmes.migrations.0007_change_default_image.change_users (apps,
                                                                    schema_editor)
```

### radioco.apps.programmes.migrations.0008\_auto\_20160116\_1509 module

```
class radioco.apps.programmes.migrations.0008_auto_20160116_1509.Migration (name,
                                                                    app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'programmes', u'0007_change_default_image')]
    operations = [<AlterField field=<django.db.models.fields.CharField>, name=u'language', model_name=u'programme
```

### radioco.apps.programmes.migrations.0009\_\_v3\_0\_\_small\_tweaks module

```
class radioco.apps.programmes.migrations.0009__v3_0__small_tweaks.Migration (name,
                                                                    app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'programmes', u'0008_auto_20160116_1509'), (u'radioco', u'0001__v3_0__mysql_timezone')]
    operations = [<AlterField field=<django.db.models.fields.DateField>, name=u'end_date', model_name=u'programme
```

### `radioco.apps.programmes.migrations.0010__v3_2__convert_role` module

```
class radioco.apps.programmes.migrations.0010__v3_2__convert_role.Migration(name,
                                                                    app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'programmes', u'0009__v3_0__small_tweaks')]
    operations = [<AlterField field=<django.db.models.fields.CharField>, name=u'role', model_name=u'participant'>, <A
radioco.apps.programmes.migrations.0010__v3_2__convert_role.convert_role_choice_to_value (ap
                                                                    sc
```

### `radioco.apps.programmes.migrations.0011__v3_2__ensure_one_user_role` module

```
class radioco.apps.programmes.migrations.0011__v3_2__ensure_one_user_role.Migration(name,
                                                                    app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'programmes', u'0010__v3_2__convert_role')]
    operations = [<AlterUniqueTogether unique_together=set([(u'person', u'episode')]), name=u'participant'>, <AlterUn
```

## Module contents

### Submodules

#### `radioco.apps.programmes.admin` module

```
class radioco.apps.programmes.admin.NonStaffEpisodeAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    fields = ['programme', 'title', 'summary', 'issue_date', 'season', 'number_in_season']
    form
        alias of NonStaffEpisodeAdminForm
    formfield_for_foreignkey (db_field, request=None, **kwargs)
    get_queryset (request)
    get_readonly_fields (request, obj=None)
    inlines = [<class 'radioco.apps.programmes.admin.NonStaffParticipantInline'>, <class 'radioco.apps.programmes.ad
    list_display = ('__unicode__', 'season', 'number_in_season', 'issue_date', 'programme')
    list_filter = ['issue_date', <class 'radioco.apps.programmes.admin.OwnEpisodeProgrammeListFilter'>, <class 'rad
    list_select_related = True
    media
    ordering = ['-season', '-number_in_season']
    save_formset (request, form, formset, change)
    save_model (request, obj, form, change)
    search_fields = ['programme__name']
```

```
class radioco.apps.programmes.admin.NonStaffEpisodeAdminForm (*args, **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta

        fields = '__all__'

        model
            alias of Episode

NonStaffEpisodeAdminForm.base_fields = OrderedDict([('title', <django.forms.fields.CharField object>), ('peo
NonStaffEpisodeAdminForm.clean_programme ()
NonStaffEpisodeAdminForm.declared_fields = OrderedDict()
NonStaffEpisodeAdminForm.media

class radioco.apps.programmes.admin.NonStaffParticipantInline (parent_model,      ad-
                                                                min_site)
    Bases: django.contrib.admin.options.StackedInline

    extra = 0

    form
        alias of NonStaffParticipantInlineForm

    formset
        alias of NonStaffParticipantInlineFormset

    get_formset (request, obj=None, **kwargs)

    get_queryset (request)

    media

    model
        alias of Participant

class radioco.apps.programmes.admin.NonStaffParticipantInlineForm (*args,
                                                                    **kwargs)
    Bases: django.forms.models.ModelForm

    base_fields = OrderedDict()

    clean ()

    declared_fields = OrderedDict()

    media

class radioco.apps.programmes.admin.NonStaffParticipantInlineFormset (data=None,
                                                                    files=None,
                                                                    in-
                                                                    stance=None,
                                                                    save_as_new=False,
                                                                    pre-
                                                                    fix=None,
                                                                    query-
                                                                    set=None,
                                                                    **kwargs)

    Bases: django.forms.models.BaseInlineFormSet

class radioco.apps.programmes.admin.NonStaffProgrammeAdmin (model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
```



```

get_fieldsets (request, obj=None)
get_prepopulated_fields (request, obj=None)
get_queryset (request)
inlines = [<class 'radioco.apps.programmes.admin.NonStaffRoleInline'>]
list_display = ('name', '_runtime', 'current_season', 'category', 'start_date', 'end_date')
list_filter = ['_runtime', 'category']
media
save_formset (request, form, formset, change)
search_fields = ['name']
class radioco.apps.programmes.admin.NonStaffRoleInline (parent_model, admin_site)
    Bases: django.contrib.admin.options.StackedInline
    extra = 0
    form
        alias of NonStaffRoleInlineForm
    formset
        alias of NonStaffRoleInlineFormset
    get_formset (request, obj=None, **kwargs)
    get_queryset (request)
    list_select_related = True
    media
    model
        alias of Role
class radioco.apps.programmes.admin.NonStaffRoleInlineForm (*args, **kwargs)
    Bases: django.forms.models.ModelForm
    base_fields = OrderedDict()
    clean ()
        Check unique together: person, role, programme
    declared_fields = OrderedDict()
    media
class radioco.apps.programmes.admin.NonStaffRoleInlineFormset (data=None,
                                                                    files=None,      in-
                                                                    stance=None,
                                                                    save_as_new=False,
                                                                    prefix=None,
                                                                    queryset=None,
                                                                    **kwargs)
    Bases: django.forms.models.BaseInlineFormSet
class radioco.apps.programmes.admin.OwnEpisodeIssueDateListFilter (request,
                                                                    params, model,
                                                                    model_admin)
    Bases: django.contrib.admin.filters.SimpleListFilter
    lookups (request, model_admin)

```

```
parameter_name = 'date'
queryset (request, queryset)
title = <django.utils.functional.__proxy__ object>
class radioco.apps.programmes.admin.OwnEpisodeProgrammeListFilter (request,
                                                                    params, model,
                                                                    model_admin)
    Bases: django.contrib.admin.filters.SimpleListFilter
    Check people in programmes besides episodes, better performance
    lookups (request, model_admin)
    parameter_name = 'programme'
    queryset (request, queryset)
    title = <django.utils.functional.__proxy__ object>
class radioco.apps.programmes.admin.PodcastInline (parent_model, admin_site)
    Bases: django.contrib.admin.options.StackedInline
    extra = 0
    inline_classes = ('grp-collapse grp-open',)
    media
    model
        alias of Podcast
```

### **radioco.apps.programmes.feeds module**

```
class radioco.apps.programmes.feeds.ProgrammeFeed
    Bases: django.contrib.syndication.views.Feed
    description (programme)
    feed_extra_kwargs (programme)
    get_object (request, slug)
    item_description (podcast)
    item_enclosure_length (podcast)
    item_enclosure_mime_type (podcast)
    item_enclosure_url (podcast)
    item_extra_kwargs (podcast)
    item_keywords (podcast)
    item_pubdate (podcast)
    item_title (podcast)
    items (programme)
    link (programme)
    title (programme)
```

```

class radioco.apps.programmes.feeds.RssProgrammeFeed
    Bases: radioco.apps.programmes.feeds.ProgrammeFeed

    feed_type
        alias of iTunesFeed

    item_guid (podcast)

class radioco.apps.programmes.feeds.iTunesFeed (title, link, description, language=None,
                                                author_email=None, author_name=None,
                                                author_link=None, subtitle=None,
                                                categories=None, feed_url=None,
                                                feed_copyright=None, feed_guid=None,
                                                ttl=None, **kwargs)
    Bases: django.utils.feedgenerator.Rss201rev2Feed

    add_item_elements (handler, item)

    add_root_elements (handler)

    rss_attributes ()

```

### radioco.apps.programmes.models module

```

class radioco.apps.programmes.models.Episode (id, title, programme, summary, issue_date, sea-
                                                son, number_in_season)
    Bases: django.db.models.base.Model

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

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

    Episode.get_absolute_url ()

    Episode.objects = <radioco.apps.programmes.models.EpisodeManager object>

    Episode.participant_set

    Episode.people

    Episode.podcast

    Episode.programme

    Episode.runtime

    Episode.summary_text

class radioco.apps.programmes.models.EpisodeManager
    Bases: django.db.models.manager.Manager

    create_episode (date, programme, last_episode=None, episode=None)

    static last (programme)

    static unfinished (programme, after=None)

class radioco.apps.programmes.models.Participant (id, person, episode, role, description)
    Bases: django.db.models.base.Model

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

```

**exception** `Participant.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`Participant.episode`

`Participant.objects = <django.db.models.manager.Manager object>`

`Participant.person`

**class** `radioco.apps.programmes.models.Podcast` (*episode, url, mime\_type, length, duration*)

Bases: `django.db.models.base.Model`

**exception** `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

**exception** `Podcast.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`Podcast.episode`

`Podcast.get_absolute_url()`

`Podcast.objects = <django.db.models.manager.Manager object>`

**class** `radioco.apps.programmes.models.Programme` (*id, name, synopsis, photo, language, current\_season, category, slug, \_runtime, start\_date, end\_date*)

Bases: `django.db.models.base.Model`

`CATEGORY_CHOICES = (('Arts', <django.utils.functional.__proxy__ object>), ('Business', <django.utils.functional.__proxy__ object>))`

**exception** `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

**exception** `Programme.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`Programme.announcers`

`Programme.end_dt`

`Programme.episode_set`

`Programme.get_absolute_url()`

`Programme.get_category_display(*moreargs, **morekwargs)`

`Programme.get_language_display(*moreargs, **morekwargs)`

`Programme.objects = <django.db.models.manager.Manager object>`

`Programme.photo`

Just like the `FileDescriptor`, but for `ImageFields`. The only difference is assigning the width/height to the `width_field/height_field`, if appropriate.

`Programme.rearrange_episodes` (*after, calendar*)

Update the `issue_date` of episodes from a given date

`Programme.role_set`

`Programme.runtime`

`Programme.save(*args, **kwargs)`

`Programme.schedule_set`

`Programme.start_dt`

Programme.**synopsis\_text**

```
class radioco.apps.programmes.models.Role(id, person, programme, role, description,
                                         date_joined)
```

Bases: django.db.models.base.Model

**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception Role.MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

Role.**get\_next\_by\_date\_joined**(\*moreargs, \*\*morekwargs)

Role.**get\_previous\_by\_date\_joined**(\*moreargs, \*\*morekwargs)

Role.**objects** = <django.db.models.manager.Manager object>

Role.**person**

Role.**programme**

```
radioco.apps.programmes.models.update_schedule_if_dt_has_changed(sender,
                                                                    instance,
                                                                    **kwargs)
```

```
radioco.apps.programmes.models.update_schedule_performance(programme)
```

**radioco.apps.programmes.urls module**

**radioco.apps.programmes.utils module**

**radioco.apps.programmes.views module**

```
radioco.apps.programmes.views.episode_detail(request, slug, season_number,
                                                episode_number)
```

```
radioco.apps.programmes.views.programme_detail(request, slug)
```

**Module contents**

**radioco.apps.radioco package**

**Subpackages**

**radioco.apps.radioco.management package**

**Subpackages**

**radioco.apps.radioco.management.commands package**

**Submodules**

## radioco.apps.radioco.management.commands.create\_example\_data module

```
class radioco.apps.radioco.management.commands.create_example_data.Command (stdout=None,
stderr=None,
no_color=False)

    Bases: django.core.management.base.BaseCommand

    handle (*args, **options)
```

### Module contents

### Module contents

## radioco.apps.radioco.migrations package

### Submodules

## radioco.apps.radioco.migrations.0001\_\_v3\_0\_\_mysql\_timezone module

```
class radioco.apps.radioco.migrations.0001__v3_0__mysql_timezone.Migration (name,
app_label)

    Bases: django.db.migrations.migration.Migration

    dependencies = [(u'schedules', u'0001_initial'), (u'programmes', u'0008_auto_20160116_1509'), (u'global_settings', u'0001_initial')]

    operations = [<RunPython <function migrate_datetime_to_utc>>]

radioco.apps.radioco.migrations.0001__v3_0__mysql_timezone.migrate_datetime_to_utc (apps,
schema_editor)

    version 3 introduces timezone support, this migration will fix the Episode issue_date Since Postgres is the only
    database which is already storing the datetimes in utc we don't need to run on it
```

### Module contents

## radioco.apps.radioco.tests package

### Submodules

## radioco.apps.radioco.tests.tests module

```
class radioco.apps.radioco.tests.tests.RadioIntegrationTests (methodName='runTest')
    Bases: radioco.apps.radioco.test_utils.TestDataMixin, django.test.testcases.TestCase

    test_index ()

class radioco.apps.radioco.tests.tests.UtilsTest (methodName='runTest')
    Bases: django.test.testcases.TestCase

    test_dictionary_key ()
        Making sure that is safe to use a dt tz aware object in a different timezone to access a dictionary

    test_example_data ()
        Running example data function, If nothing crash we are happy
```

## Module contents

### Submodules

#### **radioco.apps.radioco.context\_processors module**

`radioco.apps.radioco.context_processors.settings` (*request*)

#### **radioco.apps.radioco.test\_utils module**

**class** `radioco.apps.radioco.test_utils.TestDataMixin`

Bases: `object`

**classmethod** `setUpTestData()`

`radioco.apps.radioco.test_utils.create_test_data()`

`radioco.apps.radioco.test_utils.spain_tz()`

#### **radioco.apps.radioco.tz\_utils module**

**class** `radioco.apps.radioco.tz_utils.GMT` (*seconds*)

Bases: `dateutil.tz.tz.tzoffset`

GMT implementation, it has a fixed offset

**localize** (*dt, is\_dst=False*)

Convert naive time to local time

**normalize** (*dt, is\_dst=False*)

Correct the timezone information on the given datetime

`radioco.apps.radioco.tz_utils.fix_recurrence_date` (*start\_dt, dt*)

Fix for django-recurrence 1.3 rdates and exdates needs a datetime, we are combining the date with the time from start\_date.

Return: A datetime in the default timezone with the offset required to work in the recurrence

`radioco.apps.radioco.tz_utils.fix_recurrence_dst` (*dt*)

Fix for django-recurrence 1.3 Function to fix a datetime tz aware with an incorrect offset

Returns: A datetime in the same timezone but with the offset fixed

`radioco.apps.radioco.tz_utils.get_active_timezone` ()

Same method as `timezone.get_current_timezone` but returning utc if nothing was set

`radioco.apps.radioco.tz_utils.recurrence_after` (*recurrence, after\_dt, start\_dt*)

Fix for django-recurrence 1.3 Avoid outputting a impossible dt

`radioco.apps.radioco.tz_utils.recurrence_before` (*recurrence, before\_dt, start\_dt*)

Fix for django-recurrence 1.3 Avoid outputting a impossible dt

`radioco.apps.radioco.tz_utils.transform_datetime_tz` (*dt, tz=None*)

Transform a datetime in other timezone to the current one

`radioco.apps.radioco.tz_utils.transform_dt_to_default_tz` (*dt*)

Transform a datetime in other timezone to the current one

## radioco.apps.radioco.utils module

**class** radioco.apps.radioco.utils.**DeletePermissionMixin**  
Bases: object

**dispatch** (*request*, \*args, \*\*kwargs)

**model** = None

**class** radioco.apps.radioco.utils.**GetObjectMixin**  
Bases: django.views.generic.detail.SingleObjectMixin

**dispatch** (\*args, \*\*kwargs)

**object** = None

radioco.apps.radioco.utils.**check\_delete\_permission** (*user*, *model*)

radioco.apps.radioco.utils.**create\_example\_data** ()

radioco.apps.radioco.utils.**field\_has\_changed** (*\_object*, *field*)

**class** radioco.apps.radioco.utils.**memorize** (*func*)  
Bases: dict

A simple cache system, use as decorator

## radioco.apps.radioco.views module

radioco.apps.radioco.views.**index** (*request*)

radioco.apps.radioco.views.**user\_logout** (*request*)

## Module contents

### radioco.apps.schedules package

#### Subpackages

### radioco.apps.schedules.migrations package

#### Submodules

### radioco.apps.schedules.migrations.0001\_initial module

**class** radioco.apps.schedules.migrations.0001\_initial.**Migration** (*name*, *app\_label*)  
Bases: django.db.migrations.migration.Migration

**dependencies** = [(u'programmes', u'0001\_initial')]

**operations** = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'day', <django.db.models.fields.Int



**radioco.apps.schedules.migrations.0002\_\_v3\_0\_\_renaming\_calendar\_model module**

```
class radioco.apps.schedules.migrations.0002__v3_0__renaming_calendar_model.Migration (name,
                                                                    app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'schedules', u'0001_initial'), (u'radioco', u'0001__v3_0__mysql_timezone')]
    operations = [<RenameField new_name=u'calendar', model_name=u'schedule', old_name=u'schedule_board'>, <RemoveField name=u'schedule_board', model_name=u'schedule'>]
radioco.apps.schedules.migrations.0002__v3_0__renaming_calendar_model.migrate_board (apps,
                                                                    schema_editor)
    Renaming previous Calendars and creating a new one
```

**radioco.apps.schedules.migrations.0003\_\_v3\_0\_\_create\_excludeddates\_model module**

```
class radioco.apps.schedules.migrations.0003__v3_0__create_excludeddates_model.Migration (name,
                                                                    app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'schedules', u'0002__v3_0__renaming_calendar_model')]
    operations = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'datetime', <django.db.models.fields.DateTimeField>)]>]
```

**radioco.apps.schedules.migrations.0004\_\_v3\_0\_\_adding\_rrules module**

```
class radioco.apps.schedules.migrations.0004__v3_0__adding_rrules.Migration (name,
                                                                    app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'schedules', u'0003__v3_0__create_excludeddates_model')]
    operations = [<AddField field=<django.db.models.fields.DateTimeField>, name=u'end_date', model_name=u'schedule'>]
radioco.apps.schedules.migrations.0004__v3_0__adding_rrules.migrate_daily_recurrences (apps,
                                                                    schema_editor)
    Migration to convert weekly recurrences into new complex recurrences
```

**radioco.apps.schedules.migrations.0005\_\_v3\_0\_\_migrating\_schedules\_to\_unique\_calendar module**

```
class radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_unique_calendar.Migration (name,
                                                                    app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'schedules', u'0004__v3_0__adding_rrules'), (u'programmes', u'0009__v3_0__small_tweaks')]
    operations = [<RunPython <function migrate_schedules>>, <RemoveField name=u'day', model_name=u'schedule'>, <RemoveField name=u'end_date', model_name=u'schedule'>]
radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_unique_calendar.calculate_last_end_date
    Calculation of the last end date to improve performance
radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_unique_calendar.calculate_first_start_date
    Calculation of the first start date to improve performance
radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_unique_calendar.migrate_schedules
    Final Migration to v3.0
```

Before this migration Calendar (ScheduleBoard) had a date limit and all the schedules were repeated weekly  
We want to move constraint dates from Calendar to programmes and clone the active schedules into the active calendar

```
radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_unique_calendar.recurr
```

Fix for django-recurrence 1.3 Avoid outputting a non possible dt

```
radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_unique_calendar.recurr
```

Fix for django-recurrence 1.3 Avoid outputting a non possible dt

### Module contents

#### radioco.apps.schedules.tests package

#### Submodules

#### radioco.apps.schedules.tests.test\_recurrences module

```
class radioco.apps.schedules.tests.test_recurrences.EmptyRecurrenceTests (methodName='runTest')  
    Bases: radioco.apps.radioco.test_utils.TestDataMixin, django.test.testcases.TestCase
```

Tests to check django-recurrence behaviour when there are no recurrences

```
setUp ()
```

```
test_after ()
```

```
test_after_inclusive ()
```

```
test_before ()
```

```
test_before_inclusive ()
```

```
test_no_values ()
```

```
class radioco.apps.schedules.tests.test_recurrences.RecurrenceTests (methodName='runTest')  
    Bases: radioco.apps.radioco.test_utils.TestDataMixin, django.test.testcases.TestCase
```

Tests to check django-recurrence behaviour

```
setUp ()
```

```
test_after ()
```

```
test_after_inclusive ()
```

```
test_before_dt ()
```

```
test_before_inclusive ()
```

```
test_impossible_recurrence_after ()
```

Testing error calling after and function wrapper to solve it (recurrence\_after)

```

test_impossible_recurrence_before ()
    Testing error calling before and function wrapper to solve it (recurrence_before)

test_last_date_is_equals_before_limit ()

test_start_date_is_before_start_limit ()

test_when_date_doesnt_exist_on_february ()

```

### **radioco.apps.schedules.tests.test\_schedules** module

```

class radioco.apps.schedules.tests.test_schedules.CalendarAdminTests (methodName='runTest')
    Bases: radioco.apps.radioco.test_utils.TestDataMixin, django.test.testcases.
    TestCase

    setUp ()

    test_clone_calendar ()

class radioco.apps.schedules.tests.test_schedules.CalendarManagerTests (methodName='runTest')
    Bases: radioco.apps.radioco.test_utils.TestDataMixin, django.test.testcases.
    TestCase

    setUp ()

    test_current ()

    test_current_no_date ()

class radioco.apps.schedules.tests.test_schedules.CalendarModelTests (methodName='runTest')
    Bases: radioco.apps.radioco.test_utils.TestDataMixin, django.test.testcases.
    TestCase

    test_str ()

class radioco.apps.schedules.tests.test_schedules.CalendarValidationTests (methodName='runTest')
    Bases: django.test.testcases.TestCase

    setUp ()

    test_only_one_calendar_active ()

class radioco.apps.schedules.tests.test_schedules.ScheduleBetweenTests (methodName='runTest')
    Bases: radioco.apps.radioco.test_utils.TestDataMixin, django.test.testcases.
    TestCase

    setUp ()

    test_between_includes_started_episode ()

    test_dates_between_includes_started_episode ()

class radioco.apps.schedules.tests.test_schedules.ScheduleModelTests (methodName='runTest')
    Bases: radioco.apps.radioco.test_utils.TestDataMixin, django.test.testcases.
    TestCase

    setUp ()

    test_date_after ()

    test_date_after_exclude ()

    test_date_before ()

    test_dates_between ()

```

```
test_dates_between_complex_ruleset ()
test_dates_between_earlier_end_by_programme ()
test_dates_between_later_start_by_programme ()
test_end_gt_calendar ()
test_recurrence_rules ()
test_runtime ()
test_runtime_not_set ()
test_save_rearrange_episodes (*args, **keywargs)
test_start ()
test_start_lt_calendar ()
test_unicode ()

class radioco.apps.schedules.tests.test_schedules.ScheduleUtilsTests (methodName='runTest')
  Bases: radioco.apps.radioco.test_utils.TestDataMixin, django.test.testcases.
  TestCase

  setUp ()

  test_available_dates_after ()
  test_available_dates_none ()
  test_rearrange_episodes ()
  test_rearrange_episodes_new_schedule (*args, **keywargs)
  test_rearrange_only_non_emited_episodes (*args, **keywargs)

class radioco.apps.schedules.tests.test_schedules.ScheduleValidationTests (methodName='runTest')
  Bases: radioco.apps.radioco.test_utils.TestDataMixin, django.test.testcases.
  TestCase

  test_fields ()

class radioco.apps.schedules.tests.test_schedules.TransmissionModelTests (methodName='runTest')
  Bases: radioco.apps.radioco.test_utils.TestDataMixin, django.test.testcases.
  TestCase

  setUp ()

  test_at ()
  test_between ()
  test_between_by_queryset ()
  test_end ()
  test_episode__url ()
  test_name ()
  test_programme_url ()
  test_slug ()
  test_start ()

radioco.apps.schedules.tests.test_schedules.mock_now (dt=datetime.datetime(2014, 1,
1, 13, 30, tzinfo=<UTC>))
```

## `radioco.apps.schedules.tests.test_timezone` module

```
class radioco.apps.schedules.tests.test_timezone.ScheduleModelTests (methodName='runTest')
    Bases: radioco.apps.radioco.test_utils.TestDataMixin, django.test.testcases.TestCase
    setUp()
    test_CEST_transition()
    test_CET_transition()
    test_cleaned_internal_recurrence_dates()
    test_transform_dt_to_default_tz()
```

```
radioco.apps.schedules.tests.test_timezone.test_CET_transitions (self)
```

## Module contents

### Submodules

## `radioco.apps.schedules.admin` module

```
class radioco.apps.schedules.admin.CalendarAdmin (model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    actions = ['clone_calendar', 'set_active']
    clone_calendar (request, queryset)
    list_display = ('name', 'is_active')
    list_filter = ['is_active']
    media
    ordering = ['name']
    search_fields = ['name']
    set_active (request, queryset)
```

```
class radioco.apps.schedules.admin.ScheduleAdmin (model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    change_list_template = 'admin/schedules/calendar.html'
    changelist_view (request, extra_context=None)
    fieldsets = ((None, {'fields': ('programme', 'type', 'start_dt', 'recurrences')}), (<django.utils.functional.__proxy__ ob
    has_add_permission (request)
    media
    readonly_fields = ('effective_start_dt', 'effective_end_dt', 'source', 'from_collection')
```

**radioco.apps.schedules.forms module**

```
class radioco.apps.schedules.forms.DeleteScheduleForm (has_recurrences,          *args,  
                                                    **kwargs)  
    Bases: django.forms.forms.Form  
  
    DELETE_ALL = 'all'  
    DELETE_ONLY_THIS = 'only_this'  
    DELETE_THIS_AND_FOLLOWING = 'this_and_following'  
    base_fields = OrderedDict([('transmission_dt', <django.forms.fields.DateTimeField object>), ('schedule', <django.forms.fields.DateTimeField object>)])  
    declared_fields = OrderedDict([('transmission_dt', <django.forms.fields.DateTimeField object>), ('schedule', <django.forms.fields.DateTimeField object>)])  
    media
```

**radioco.apps.schedules.models module**

```
class radioco.apps.schedules.models.Calendar (id, name, is_active)  
    Bases: django.db.models.base.Model  
  
    exception DoesNotExist  
        Bases: django.core.exceptions.ObjectDoesNotExist  
  
    exception Calendar.MultipleObjectsReturned  
        Bases: django.core.exceptions.MultipleObjectsReturned  
  
    classmethod Calendar.get_active ()  
  
    Calendar.objects = <django.db.models.manager.Manager object>  
  
    Calendar.rearrange_episodes ()  
  
    Calendar.save (*args, **kwargs)  
  
    Calendar.schedule_set  
  
class radioco.apps.schedules.models.CalendarManager  
    Bases: django.db.models.manager.Manager  
  
    current ()  
  
class radioco.apps.schedules.models.ExcludedDates (*args, **kwargs)  
    Bases: django.db.models.base.Model  
  
    Helper to improve performance  
  
    exception DoesNotExist  
        Bases: django.core.exceptions.ObjectDoesNotExist  
  
    exception ExcludedDates.MultipleObjectsReturned  
        Bases: django.core.exceptions.MultipleObjectsReturned  
  
    ExcludedDates.date  
  
    ExcludedDates.get_new_excluded_datetime (new_dt)  
        Returns: A new dt to be excluded in that date  
  
    ExcludedDates.get_next_by_datetime (*moreargs, **morekwargs)  
  
    ExcludedDates.get_previous_by_datetime (*moreargs, **morekwargs)  
  
    ExcludedDates.objects = <django.db.models.manager.Manager object>
```

ExcludedDates.**schedule**

```
class radioco.apps.schedules.models.Schedule(id, programme, type, calendar, recurrences,
                                             start_dt, effective_start_dt, effective_end_dt,
                                             from_collection, source)
```

Bases: django.db.models.base.Model

**exception DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

**exception Schedule.MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

Schedule.**calendar**

Schedule.**child\_schedules**

Schedule.**date\_after** (*after*)

Schedule.**date\_before** (*before*)

Schedule.**dates\_between** (*after, before*)

Return a sorted list of dates between after and before

Schedule.**exclude\_date** (*dt*)

Schedule.**excludeddates\_set**

Schedule.**from\_collection**

Schedule.**get\_next\_by\_start\_dt** (*\*moreargs, \*\*morekwargs*)

Schedule.**get\_previous\_by\_start\_dt** (*\*moreargs, \*\*morekwargs*)

**static** Schedule.**get\_schedule\_which\_excluded\_dt** (*programme, dt*)

Schedule.**get\_type\_display** (*\*moreargs, \*\*morekwargs*)

Schedule.**has\_recurrences** ()

Schedule.**include\_date** (*dt*)

Schedule.**objects** = <django.db.models.manager.Manager object>

Schedule.**programme**

Schedule.**recurrences**

A placeholder class that provides a way to set the attribute on the model.

Schedule.**runtime**

Schedule.**save** (*\*args, \*\*kwargs*)

Schedule.**schedule\_set**

Schedule.**source**

```
class radioco.apps.schedules.models.Transmission(schedule, date, episode=None)
```

Bases: object

Temporal object generated according to recurrence rules or schedule information It contains concrete dates

**classmethod** **at** (*at*)

**classmethod** **between** (*after, before, schedules=None*)

Return a tuple of Schedule and Transmissions sorted by date

**end**

**episode\_url**

**name**

**programme\_url**

**slug**

`radioco.apps.schedules.models.calculate_effective_schedule_end_dt` (*schedule*)  
Calculation of the last end date to improve performance

`radioco.apps.schedules.models.calculate_effective_schedule_start_dt` (*schedule*)  
Calculation of the first start date to improve performance

### **radioco.apps.schedules.urls module**

### **radioco.apps.schedules.utils module**

`radioco.apps.schedules.utils.next_dates` (*calendar, programme, after*)  
Returns: A generator with the next dates of a given programme

### **radioco.apps.schedules.views module**

**class** `radioco.apps.schedules.views.DeleteScheduleView` (\*\*kwargs)  
Bases: `radioco.apps.radioco.utils.GetObjectMixin`, `radioco.apps.radioco.utils.DeletePermissionMixin`, `django.views.generic.edit.FormView`

**form\_class**  
alias of `DeleteScheduleForm`

**form\_valid** (*form*)

**get** (*request, \*args, \*\*kwargs*)

**get\_context\_data** (\*\*kwargs)

**get\_form\_kwargs** ()

**get\_initial** ()

**model**  
alias of `Schedule`

**schedule\_id** = None

**template\_name** = 'admin/schedules/delete\_modal.html'

**transmission\_dt** = None

`radioco.apps.schedules.views.schedule_list` (*request*)

### **Module contents**

#### **radioco.apps.users package**

#### **Subpackages**





```
clean_username ()
declared_fields = OrderedDict([('username', <django.forms.fields.CharField object>), ('first_name', <django.forms
media
save (force_insert=False, force_update=False, commit=True)
class radioco.apps.users.admin.SingletonProfileAdmin (model, admin_site)
Bases: django.contrib.admin.options.ModelAdmin
change_view (request, object_id, extra_context=None)
fields = ['username', 'first_name', 'last_name', 'email', 'bio', 'avatar', 'display_personal_page']
form
    alias of NonStaffUserProfileForm
get_urls ()
has_add_permission (request)
has_delete_permission (request, obj=None)
media
response_change (request, obj)
save_model (request, obj, form, change)
class radioco.apps.users.admin.UserProfileAdmin (model, admin_site)
Bases: django.contrib.auth.admin.UserAdmin
inlines = (<class 'radioco.apps.users.admin.UserProfileInline'>,)
list_filter = ('is_staff', 'is_superuser', 'is_active', 'groups', 'userprofile__display_personal_page')
media
class radioco.apps.users.admin.UserProfileInline (parent_model, admin_site)
Bases: django.contrib.admin.options.StackedInline
can_delete = False
exclude = ('slug',)
extra = 1
inline_classes = ('grp-collapse grp-open',)
media
model
    alias of UserProfile
```

### **radioco.apps.users.models module**

```
class radioco.apps.users.models.UserProfile (id, user, bio, avatar, display_personal_page,
slug)
Bases: django.db.models.base.Model
exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist
exception UserProfile.MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned
```

`UserProfile.avatar`

Just like the `FileDescriptor`, but for `ImageFields`. The only difference is assigning the width/height to the `width_field/height_field`, if appropriate.

`UserProfile.get_absolute_url()`

`UserProfile.objects = <django.db.models.manager.Manager object>`

`UserProfile.save(*args, **kwargs)`

`UserProfile.user`

`radioco.apps.users.models.save_slug(sender, instance=None, **kwargs)`

### **radioco.apps.users.tests module**

**class** `radioco.apps.users.tests.UserProfileMethodTests` (*methodName='runTest'*)

Bases: `django.test.testcases.TestCase`

**test\_get\_announcers\_and\_profile()**

**test\_save()**

### **radioco.apps.users.urls module**

### **radioco.apps.users.views module**

**class** `radioco.apps.users.views.UsersView` (*\*\*kwargs*)

Bases: `django.views.generic.list.ListView`

**get\_queryset()**

**model**

alias of `UserProfile`

`radioco.apps.users.views.userprofile_detail(request, slug)`

## **Module contents**

### **Module contents**

### **radioco.configs package**

#### **Subpackages**

### **radioco.configs.base package**

#### **Submodules**

### **radioco.configs.base.manage module**

### **radioco.configs.base.settings module**

`radioco.configs.base.settings.gettext_noop(s)`

### **radioco.configs.base.test\_runner module**

```
class radioco.configs.base.test_runner.MyTestSuiteRunner (*args, **kwargs)
    Bases: django.test.runner.DiscoverRunner
```

### **radioco.configs.base.urls module**

#### **Module contents**

### **radioco.configs.development package**

#### **Submodules**

### **radioco.configs.development.manage module**

### **radioco.configs.development.settings module**

#### **Module contents**

### **radioco.configs.heroku package**

#### **Submodules**

### **radioco.configs.heroku.manage module**

### **radioco.configs.heroku.settings module**

### **radioco.configs.heroku.wsgi module**

#### **Module contents**

### **radioco.configs.staging package**

#### **Submodules**

### **radioco.configs.staging.manage module**

### **radioco.configs.staging.settings module**

### **radioco.configs.staging.wsgi module**

Module contents

Module contents

Module contents

tasks package

Submodules

tasks.docker module

tasks.docs module

tasks.heroku module

tasks.locale module

tasks.radioco module

tasks.utils module

Module contents

## Release notes & upgrade information

Some versions of RadioCo present more complex upgrade paths than others, and some **require** you to take action. It is strongly recommended to read the release notes carefully when upgrading.

It goes without saying that you should **backup your database** before embarking on any process that makes changes to your database.

### 1.1 release notes

#### What's new in 1.1

- Numerous updates to the documentation
- Updates to facilitate the starting-up (radioco\_recorder\_user and a empty shedule\_board are self-created)
- Added custom languages to programmes (by default, all django languages)

#### How this affects you

If you're starting with a new installation, you don't need to worry about this. Don't even bother reading this section; it's for upgraders.

Activate your virtualenv and do the following in your project main directory:

```
pip install django-radio==1.1
python manage.py migrate
```

## 1.1.1 release notes

### What's new in 1.1.1

- Added CKEditor to programme's synopsis, episode's summary and user's biography
- Added default settings to the project
- Added customisable footer

### How this affects you

If you're starting with a new installation, you don't need to worry about this. Don't even bother reading this section; it's for upgraders.

Activate your virtualenv and do the following in your project main directory:

```
pip install django-radio==1.1.1
python manage.py collectstatic
python manage.py migrate
```

In your settings.py import default settings:

```
from radio.settings_base import *
```

## 1.2 release notes

### What's new in 1.2

- Fixed project urls when RadioCo is deployed in a subdirectory
- Fixed default url of photos when STATIC\_URL is not "/static/"
- Fixed errors 404 when users don't have a personal webpage.

### How this affects you

If you're starting with a new installation, you don't need to worry about this. Don't even bother reading this section; it's for upgraders.

Activate your virtualenv and do the following in your project main directory:

```
pip install django-radio==1.2
python manage.py migrate
```

## 2.1 release notes

### What's new in 2.1

- New layout! The whole site has been redesigned
- Added image versions, RadioCo will adapt your images to fit in the new template
- Added comments support.

- Added analytics support.

### How this affects you

If you're starting with a new installation, you don't need to worry about this. Don't even bother reading this section; it's for upgraders.

You need to replace your current source with the content of <https://github.com/iago1460/django-radio>. To setup your settings please read the configuration section.

You should be able to keep your current database but make sure to create a backup before start.

```
pip install -r radio/configs/common/requirements.txt
python manage.py collectstatic
python manage.py migrate
```

## 3.0 release notes

### What's new in 3.0

In this version RadioCo has almost been rewritten completely, internally the major improvements are:

- Use of time-zone-aware datetime objects ([more info](#))
- Added good amount of tests
- Changed calendar to use API methods
- Using bower to download js and css dependencies
- Renamed internal folders and models
  - django-radio/radio -> django-radio/radioco
  - django-radio/radio/configs/common -> django-radio/radioco/configs/base
  - ScheduleBoard model has been renamed to Calendar
  - dashboard app has been merged into schedules

Special mention to [@stefan-walluhn](#) - Radio Corax who helped in this release, specifically adding Bower, Rest Framework API, tests and complex recurrence rules.

And the visible features are:

- Added download button to episodes ([#38](#) Credit to [@AlexCortinas](#))
- Removed unnecessary Scroll configuration setting ([#13](#) Credit to [@AlexCortinas](#))
- Improvements in title representation of episodes when not set ([#17](#) Credit to [@txenoo](#))
- Removed warnings when compiling ([#32](#) Thanks to [@txenoo](#))
- Added Radiocom endpoints (Thanks to [@pablogrela](#))
- Added upload feature to CKEditor ([#22](#))
- Added new calendar setting `slotDuration`
- Added a suite of helper tasks using `invoke` (Easy deploy to docker)
- Added public API to access to programmes, episodes, schedules and transmissions ([#2](#))
- The schedule manager has received important updates ([#20](#), [#4](#))

Now managing schedules should be a lot more simple. From the manager is possible to create and edit programmes in addition to schedules.

The forced weekly recurrence has been removed in favor of complex recurrence rules, having a unique calendar should be enough.

The Calendar model doesn't have date constraints anymore and only one can be active at the same time. If you want to restrict a programme you should do it filling the start and end date fields inside itself.

### How this affects you

If you're starting with a new installation, you don't need to worry about this. Don't even bother reading this section; it's for upgraders.

Please **make sure of having a copy of your database** before even started to look to the next steps.

If you have custom code or templates I recommend putting them in a safe place, special mention to `local_settings.py` files, your are going to need to move them to the correct folder.

Since this migration introduces timezone support, it's necessary to set the timezone variable:

1. Find your timezone in [this list](#)
2. Go to the config folder and create if not exists a `local_settings.py` in the same directory than `settings.py`
3. Add to the local settings the variable timezone, for example: `TIME_ZONE = "Europe/Madrid"`

**Warning:** Using an incorrect timezone will cause incorrect dates, you should set/override that setting in your `local_settings.py`.

**Warning:** If you have a database backend [different to PostgreSQL](#) you must convert your data from local time to UTC. Fortunately for you a migration was written to fix episode `issue_dates`, other dates will suffer alterations.

**Warning:** We strongly recommend to use PostgreSQL as backend. Other database engines **will not be supported** in the future.

You have to install [bower](#) before running the following commands in the project root folder:

```
bower install
pip install -r radioco/configs/base/requirements.txt
python manage.py collectstatic
python manage.py migrate
```

After the migration process you will find that there is a new calendar set to active that contains all your schedules and your other calendars have been renamed with the prefix "Legacy - ". Feel free of delete these, they are not require anymore.

---

**Note:** Warning messages could appear during the migration process, it's usually safe to ignore them.

---



### 3.0.1 release notes

#### What's new in 3.0.1

- Fixing invoke commands.
- Simplified docker setup.

Take a look to our installation section.

#### How this affects you

If you're starting with a new installation, you don't need to worry about this. Don't even bother reading this section; it's for upgraders.

You need to replace your current source with the content of <https://github.com/iago1460/django-radio>. To setup your settings please read the configuration section.

This update doesn't require any further steps.

### 3.1 release notes

#### What's new in 3.1

- Fixing staging docker port issue. (Thanks to @mmontes11 for report)
- Adding stable Radiocom API
- Solving ticket #43
- Upgrading libraries to last version
- Small fixes to invoke scripts
- Improving API (Outputting absolute urls and adding some fields)

#### How this affects you

If you're starting with a new installation, you don't need to worry about this. Don't even bother reading this section; it's for upgraders.

You need to replace your current source with the content of <https://github.com/iago1460/django-radio>. To setup your settings please read the configuration section.

You should be able to keep your current database but make sure to create a backup before start.

```
pip install -r radioco/configs/base/requirements.txt
python manage.py collectstatic
python manage.py migrate # check the following note in case of error
```

---

**Note:** When migrating from version 3.0 to version 3.1 the following error will appear during the migration process `django.db.utils.ProgrammingError: relation "recurrence_date" already exists` to solve this please run the following command:

```
python manage.py migrate --fake-initial
```

---

Or if you are using our docker setup:

```
inv docker.clean -e staging
inv docker.setup -e staging # check the following note in case of error
```

---

**Note:** If a migration error appear during the setup please run:

```
inv docker.manage -e staging -c "migrate --fake-initial"
```

---

## 3.2 release notes

### What's new in 3.2

- Solving ticket #5
- Improving rss (adding images and fixing itunes encoding)
- Added rss autodiscovery
- Documenting `django.contrib.sites`
- Improving ORM queries
- Adding cache to singleton models using memcached
- Improve performance in `staging config` using nginx cache, uwsgi and caching templates
- Updating Spanish and Galician translations
- Fixing progress bar in the main page

### How this affects you

If you're starting with a new installation, you don't need to worry about this. Don't even bother reading this section; it's for upgraders.

You need to replace your current source with the content of <https://github.com/iago1460/django-radio>. To setup your settings please read the configuration section.

You should be able to keep your current database but make sure to create a backup before start.

**Warning:** You must ensure your episodes and programmes only have one role per user. A database constraint has been put in place as part of this migration.

**Warning:** This migration depends on `settings.LANGUAGE_CODE`, it will convert the `role` field into a editable CharField using your preferred language. Note you can change this value temporally to run this migration. [More info](#).

```
pip install -r radioco/configs/base/requirements.txt
python manage.py collectstatic
python manage.py migrate
```

**Note:** If you have a custom production environment please consider to apply our [staging setup](#) to improve performance.

---

Or if you are using our docker setup:

```
inv docker.clean -e staging
inv docker.setup -e staging
```

## 3.2.1 release notes

### What's new in 3.2.1

- Making rss links protocol aware so they can be opened by podcast apps.

### How this affects you

If you're starting with a new installation, you don't need to worry about this. Don't even bother reading this section; it's for upgraders.

You need to replace your current source with the content of <https://github.com/iago1460/django-radio>. To setup your settings please read the configuration section.

This update doesn't require any further steps.

## 3.2.2 release notes

### What's new in 3.2.2

- Fixing bug that made impossible deleting inline objects.
- Removing undocumented switch user functionality.
- Tweaking calendar css to improve readability.
- Filtering for active programmes in the home page.

Thanks to [@beaulalume](#) for reporting the previous issues.

### How this affects you

If you're starting with a new installation, you don't need to worry about this. Don't even bother reading this section; it's for upgraders.

You need to replace your current source with the content of <https://github.com/iago1460/django-radio>. To setup your settings please read the configuration section.

You should be able to keep your current database but make sure to create a backup before start.

```
python manage.py collectstatic
```

Or if you are using our docker setup:

```
inv docker.clean -e staging
inv docker.setup -e staging
```

## Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

### m

manage, 12

### r

radioco, 49

radioco.apps, 47

radioco.apps.api, 20

radioco.apps.api.recorder\_views, 15

radioco.apps.api.serializers, 15

radioco.apps.api.tests, 15

radioco.apps.api.tests.test\_api, 12

radioco.apps.api.tests.test\_fullcalendar,  
13

radioco.apps.api.tests.test\_programmes,  
13

radioco.apps.api.tests.test\_radiocom\_api,  
14

radioco.apps.api.tests.test\_recorder,  
14

radioco.apps.api.tests.test\_schedules,  
14

radioco.apps.api.urls, 17

radioco.apps.api.views, 17

radioco.apps.api.viewsets, 20

radioco.apps.global\_settings, 24

radioco.apps.global\_settings.admin, 22

radioco.apps.global\_settings.migrations,  
22

radioco.apps.global\_settings.migrations.0001\_initial,  
21

radioco.apps.global\_settings.migrations.0002\_remove\_calendarconfiguration\_display\_next\_week,  
21

radioco.apps.global\_settings.migrations.0003\_siteconfiguration\_footer,  
21

radioco.apps.global\_settings.migrations.0004\_auto\_20150606\_1335,  
21

radioco.apps.global\_settings.migrations.0005\_auto\_20150606\_1445, urls, 33

radioco.apps.global\_settings.migrations.0006\_auto\_20160006\_1509, views, 33

22

radioco.apps.global\_settings.migrations.0007\_\_v3\_0,  
22

radioco.apps.global\_settings.migrations.0008\_\_v3\_0,  
22

radioco.apps.global\_settings.models, 23

radioco.apps.global\_settings.templatetags,  
22

radioco.apps.global\_settings.templatetags.global\_se,  
22

radioco.apps.programmes, 33

radioco.apps.programmes.admin, 27

radioco.apps.programmes.feeds, 30

radioco.apps.programmes.migrations, 27

radioco.apps.programmes.migrations.0001\_initial,  
25

radioco.apps.programmes.migrations.0002\_change\_lang,  
25

radioco.apps.programmes.migrations.0003\_change\_text,  
25

radioco.apps.programmes.migrations.0004\_change\_phot,  
25

radioco.apps.programmes.migrations.0005\_auto\_20150,  
26

radioco.apps.programmes.migrations.0006\_auto\_20160,  
26

radioco.apps.programmes.migrations.0007\_change\_defa,  
26

radioco.apps.programmes.migrations.0008\_auto\_20160,  
26

radioco.apps.programmes.migrations.0009\_\_v3\_0\_\_sma,  
26

radioco.apps.programmes.migrations.0010\_\_v3\_2\_\_con,  
27

radioco.apps.programmes.migrations.0011\_\_v3\_2\_\_ens,  
27

radioco.apps.programmes.models, 31

radioco.apps.programmes.urls, 33

radioco.apps.programmes.utils, 33

radioco.apps.programmes.views, 33

radioco.apps.radioco, 36

radioco.apps.radioco.context\_processors, radioco.apps.users.urls, 47  
35 radioco.apps.users.views, 47  
radioco.apps.radioco.management, 34 radioco.configs, 49  
radioco.apps.radioco.management.commands radioco.configs.base, 48  
34 radioco.configs.base.manage, 47  
radioco.apps.radioco.management.commands radioco.configs.base.manage, 47  
34 radioco.configs.base.settings, 48  
radioco.apps.radioco.migrations, 34 radioco.configs.base.test\_runner, 48  
radioco.apps.radioco.migrations.0001\_\_v3\_\_radioco\_\_initial, 34 radioco.configs.base.urls, 48  
34 radioco.configs.base.urls, 48  
radioco.apps.radioco.test\_utils, 35 radioco.configs.development, 48  
radioco.apps.radioco.tests, 35 radioco.configs.development.manage, 48  
radioco.apps.radioco.tests.tests, 34 radioco.configs.heroku, 48  
radioco.apps.radioco.tz\_utils, 35 radioco.configs.heroku.manage, 48  
radioco.apps.radioco.utils, 36 radioco.configs.heroku.settings, 48  
radioco.apps.radioco.views, 36 radioco.configs.heroku.wsgi, 48  
radioco.apps.schedules, 44 radioco.configs.staging, 49  
radioco.apps.schedules.admin, 41 radioco.configs.staging.manage, 48  
radioco.apps.schedules.forms, 42 radioco.configs.staging.wsgi, 48  
radioco.apps.schedules.migrations, 38  
radioco.apps.schedules.migrations.0001\_initial, 36  
radioco.apps.schedules.migrations.0002\_\_v3\_0\_\_renaming\_calendar\_model, 37  
radioco.apps.schedules.migrations.0003\_\_v3\_0\_\_create\_excludeddates\_model, 37  
radioco.apps.schedules.migrations.0004\_\_v3\_0\_\_adding\_rrules, 37  
radioco.apps.schedules.migrations.0005\_\_v3\_0\_\_migrating\_schedules\_to\_unique\_calendar, 37  
radioco.apps.schedules.models, 42  
radioco.apps.schedules.tests, 41  
radioco.apps.schedules.tests.test\_recurrences, 38  
radioco.apps.schedules.tests.test\_schedules, 39  
radioco.apps.schedules.tests.test\_timezone, 41  
radioco.apps.schedules.urls, 44  
radioco.apps.schedules.utils, 44  
radioco.apps.schedules.views, 44  
radioco.apps.users, 47  
radioco.apps.users.admin, 45  
radioco.apps.users.migrations, 45  
radioco.apps.users.migrations.0001\_initial, 45  
radioco.apps.users.migrations.0002\_change\_textfield\_to\_richtextfield, 45  
radioco.apps.users.migrations.0003\_auto\_20160104\_2029, 45  
radioco.apps.users.migrations.0004\_\_v3\_0\_\_ckeditor\_upload\_field, 45  
radioco.apps.users.models, 46  
radioco.apps.users.tests, 47

**A**

AbsoluteURLField (class in radioco.apps.api.serializers), 15

abstract (radioco.apps.global\_settings.models.SingletonModel.Meta attribute), 24

actions (radioco.apps.schedules.admin.CalendarAdmin attribute), 41

add\_item\_elements() (radioco.apps.programmes.feeds.iTunesFeed method), 31

add\_root\_elements() (radioco.apps.programmes.feeds.iTunesFeed method), 31

announcers (radioco.apps.programmes.models.Programme attribute), 32

at() (radioco.apps.schedules.models.Transmission class method), 43

avatar (radioco.apps.users.models.UserProfile attribute), 46

**B**

base\_fields (radioco.apps.api.views.ProgrammeFilterForm attribute), 18

base\_fields (radioco.apps.api.views.TimezoneForm attribute), 19

base\_fields (radioco.apps.api.views.TransmissionForm attribute), 20

base\_fields (radioco.apps.programmes.admin.NonStaffEpisodeAdminForm attribute), 28

base\_fields (radioco.apps.programmes.admin.NonStaffParticipantInlineForm attribute), 28

base\_fields (radioco.apps.programmes.admin.NonStaffRoleInlineForm attribute), 29

base\_fields (radioco.apps.schedules.forms.DeleteScheduleForm attribute), 42

base\_fields (radioco.apps.users.admin.NonStaffUserProfileForm attribute), 45

base\_filters (radioco.apps.api.views.EpisodeFilter attribute), 17

base\_filters (radioco.apps.api.views.ProgrammeFilter attribute), 18

base\_filters (radioco.apps.api.views.ScheduleFilter attribute), 19

between() (radioco.apps.schedules.models.Transmission class method), 43

**C**

calculate\_effective\_schedule\_end\_dt() (in module radioco.apps.schedules.migrations.0005\_\_v3\_0\_\_migrating\_schedules), 37

calculate\_effective\_schedule\_end\_dt() (in module radioco.apps.schedules.models), 44

calculate\_effective\_schedule\_start\_dt() (in module radioco.apps.schedules.migrations.0005\_\_v3\_0\_\_migrating\_schedules), 37

calculate\_effective\_schedule\_start\_dt() (in module radioco.apps.schedules.models), 44

Calendar (class in radioco.apps.schedules.models), 42

calendar (radioco.apps.schedules.models.Schedule attribute), 43

Calendar.DoesNotExist, 42

Calendar.MultipleObjectsReturned, 42

CalendarAdmin (class in radioco.apps.schedules.admin), 41

CalendarAdminTests (class in radioco.apps.schedules.tests.test\_schedules), 39

CalendarConfiguration (class in radioco.apps.global\_settings.models), 23

CalendarConfiguration.DoesNotExist, 23

CalendarConfiguration.MultipleObjectsReturned, 23

CalendarManager (class in radioco.apps.schedules.models), 42

CalendarManagerTests (class in radioco.apps.schedules.tests.test\_schedules), 39

CalendarModelTests (class in radioco.apps.schedules.tests.test\_schedules), 39

CalendarValidationTests (class in radioco.apps.schedules.tests.test\_schedules), 39

can\_delete (radioco.apps.users.admin.UserProfileInline attribute), 46

CATEGORY\_CHOICES (radioco.apps.programmes.models.Programme attribute), 32

change\_form\_template (radioco.apps.global\_settings.admin.SingletonModelAdmin attribute), 23

change\_list\_template (radioco.apps.schedules.admin.ScheduleAdmin attribute), 41

change\_photo\_url() (in module radioco.apps.programmes.migrations.0004\_change\_photo\_url), 25

change\_programmes() (in module radioco.apps.programmes.migrations.0007\_change\_default\_image), 26

change\_users() (in module radioco.apps.programmes.migrations.0007\_change\_default\_image), 26

change\_view() (radioco.apps.global\_settings.admin.SingletonModelAdmin method), 23

change\_view() (radioco.apps.users.admin.SingletonProfileAdmin method), 46

changelist\_view() (radioco.apps.schedules.admin.ScheduleAdmin method), 41

check\_delete\_permission() (in module radioco.apps.radioco.utils), 36

check\_recorder\_program() (in module radioco.apps.api.recorder\_views), 15

child\_schedules (radioco.apps.schedules.models.Schedule attribute), 43

clean() (radioco.apps.api.views.ProgrammeFilterForm method), 18

clean() (radioco.apps.api.views.TransmissionForm method), 20

clean() (radioco.apps.programmes.admin.NonStaffParticipantInlineForm attribute), 28

clean() (radioco.apps.programmes.admin.NonStaffRoleInlineForm method), 29

clean\_programme() (radioco.apps.programmes.admin.NonStaffEpisodeAdminForm attribute), 28

clean\_timezone() (radioco.apps.api.views.TimezoneForm method), 19

clean\_username() (radioco.apps.users.admin.NonStaffUserProfileForm method), 45

clone\_calendar() (radioco.apps.schedules.admin.CalendarAdmin method), 41

Command (class in radioco.apps.radioco.management.commands.create\_example\_data), 34

convert\_role\_choice\_to\_value() (in module radioco.apps.programmes.migrations.0010\_\_v3\_2\_\_convert\_role), 27

create\_episode() (radioco.apps.programmes.models.EpisodeManager method), 31

create\_example\_data() (in module radioco.apps.radioco.utils), 36

create\_test\_data() (in module radioco.apps.radioco.test\_utils), 35

current() (radioco.apps.schedules.models.CalendarManager method), 42

## D

date (radioco.apps.schedules.models.ExcludedDates attribute), 42

date\_after() (radioco.apps.schedules.models.Schedule method), 43

date\_before() (radioco.apps.schedules.models.Schedule method), 43

dates\_between() (radioco.apps.schedules.models.Schedule method), 43

DateTimeFieldTz (class in radioco.apps.api.serializers), 43

declared\_fields (radioco.apps.api.views.ProgrammeFilterForm attribute), 18

declared\_fields (radioco.apps.api.views.TimezoneForm attribute), 19

declared\_fields (radioco.apps.api.views.TransmissionForm attribute), 20

declared\_fields (radioco.apps.programmes.admin.NonStaffEpisodeAdminForm attribute), 28

declared\_fields (radioco.apps.programmes.admin.NonStaffParticipantInlineForm attribute), 28

declared\_fields (radioco.apps.programmes.admin.NonStaffRoleInlineForm attribute), 29

declared\_fields (radioco.apps.schedules.forms.DeleteScheduleForm attribute), 42

declared\_fields (radioco.apps.users.admin.NonStaffUserProfileForm attribute), 46

declared\_filters (radioco.apps.api.views.EpisodeFilter attribute), 17

declared\_filters (radioco.apps.api.views.ProgrammeFilter attribute), 18

declared\_filters (radioco.apps.api.views.ScheduleFilter attribute), 19

delete() (radioco.apps.global\_settings.models.SingletonModel method), 24

DELETE\_ALL (radioco.apps.schedules.forms.DeleteScheduleForm attribute), 42

DELETE\_ONLY\_THIS (radioco.apps.schedules.forms.DeleteScheduleForm attribute), 42





- ExcludedDates (class in radioco.apps.schedules.models), 42
- ExcludedDates.DoesNotExist, 42
- ExcludedDates.MultipleObjectsReturned, 42
- excludeddates\_set (radioco.apps.schedules.models.Schedulefilter\_class attribute), 43
- extra (radioco.apps.programmes.admin.NonStaffParticipantInline attribute), 28
- extra (radioco.apps.programmes.admin.NonStaffRoleInline attribute), 29
- extra (radioco.apps.programmes.admin.PodcastInline attribute), 30
- extra (radioco.apps.users.admin.UserProfileInline attribute), 46
- F**
- feed\_extra\_kwargs() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- feed\_type (radioco.apps.programmes.feeds.RssProgrammeFeed attribute), 31
- field\_has\_changed() (in module radioco.apps.radioco.utils), 36
- fields (radioco.apps.api.serializers.EpisodeSerializer.Meta attribute), 16
- fields (radioco.apps.api.serializers.ProgrammeSerializer.Meta attribute), 16
- fields (radioco.apps.api.serializers.RadiocomConfigurationSerializer.Meta attribute), 16
- fields (radioco.apps.api.serializers.RadiocomProgrammeSerializer.Meta attribute), 16
- fields (radioco.apps.api.serializers.ScheduleSerializer.Meta attribute), 17
- fields (radioco.apps.api.views.EpisodeFilter.Meta attribute), 17
- fields (radioco.apps.api.views.ProgrammeFilter.Meta attribute), 18
- fields (radioco.apps.api.views.ScheduleFilter.Meta attribute), 19
- fields (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 27
- fields (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 28
- fields (radioco.apps.users.admin.SingletonProfileAdmin attribute), 46
- fieldsets (radioco.apps.schedules.admin.ScheduleAdmin attribute), 41
- filter\_backends (radioco.apps.api.views.EpisodeViewSet attribute), 17
- filter\_backends (radioco.apps.api.views.ProgrammeViewSet attribute), 18
- filter\_backends (radioco.apps.api.views.ScheduleViewSet attribute), 19
- filter\_backends (radioco.apps.api.views.TransmissionViewSet attribute), 20
- filter\_class (radioco.apps.api.views.EpisodeViewSet attribute), 17
- filter\_class (radioco.apps.api.views.ProgrammeViewSet attribute), 18
- filter\_class (radioco.apps.api.views.ScheduleViewSet attribute), 19
- filter\_class (radioco.apps.api.views.TransmissionViewSet attribute), 20
- fix\_recurrence\_date() (in module radioco.apps.radioco.tz\_utils), 35
- fix\_recurrence\_dst() (in module radioco.apps.radioco.tz\_utils), 35
- form (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 27
- form (radioco.apps.programmes.admin.NonStaffParticipantInline attribute), 28
- form (radioco.apps.programmes.admin.NonStaffRoleInline attribute), 29
- form (radioco.apps.users.admin.SingletonProfileAdmin attribute), 46
- form\_class (radioco.apps.schedules.views.DeleteScheduleView attribute), 44
- form\_valid() (radioco.apps.schedules.views.DeleteScheduleView method), 44
- formfield\_for\_foreignkey() (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 27
- formset (radioco.apps.programmes.admin.NonStaffParticipantInline attribute), 28
- formset (radioco.apps.programmes.admin.NonStaffRoleInline attribute), 29
- from\_collection (radioco.apps.schedules.models.Schedule attribute), 43
- G**
- get() (radioco.apps.global\_settings.models.SingletonModelManager method), 24
- get() (radioco.apps.schedules.views.DeleteScheduleView method), 44
- get\_absolute\_url() (radioco.apps.programmes.models.Episode attribute), 31
- get\_absolute\_url() (radioco.apps.programmes.models.Podcast attribute), 32
- get\_absolute\_url() (radioco.apps.programmes.models.Programme attribute), 32
- get\_absolute\_url() (radioco.apps.users.models.UserProfile attribute), 47
- get\_active() (radioco.apps.schedules.models.Calendar class method), 42
- get\_active\_timezone() (in module radioco.apps.radioco.tz\_utils), 35

get\_category\_display() (radioco.apps.programmes.models.Programme method), 32  
 get\_context\_data() (radioco.apps.schedules.views.DeleteScheduleView method), 44  
 get\_facebook\_url() (radioco.apps.api.serializers.RadiocomConfigurationSerializer method), 16  
 get\_fieldsets() (radioco.apps.programmes.admin.NonStaffProgrammeAdmin method), 28  
 get\_first\_day\_display() (radioco.apps.global\_settings.models.CalendarConfiguration method), 23  
 get\_form\_kwargs() (radioco.apps.schedules.views.DeleteScheduleView method), 44  
 get\_formset() (radioco.apps.programmes.admin.NonStaffParticipantInline method), 28  
 get\_formset() (radioco.apps.programmes.admin.NonStaffRoleInline method), 29  
 get\_global() (radioco.apps.global\_settings.models.SingletonModel class method), 24  
 get\_global\_model() (in module radioco.apps.global\_settings.templatetags.global\_settings\_twitter\_tag), 22  
 get\_initial() (radioco.apps.schedules.views.DeleteScheduleView method), 44  
 get\_language\_display() (radioco.apps.programmes.models.Programme method), 32  
 get\_new\_excluded\_datetime() (radioco.apps.schedules.models.ExcludedDates method), 42  
 get\_next\_by\_date\_joined() (radioco.apps.programmes.models.Role method), 33  
 get\_next\_by\_datetime() (radioco.apps.schedules.models.ExcludedDates method), 42  
 get\_next\_by\_start\_dt() (radioco.apps.schedules.models.Schedule method), 43  
 get\_object() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30  
 get\_prepopulated\_fields() (radioco.apps.programmes.admin.NonStaffProgrammeAdmin method), 29  
 get\_previous\_by\_date\_joined() (radioco.apps.programmes.models.Role method), 33  
 get\_previous\_by\_datetime() (radioco.apps.schedules.models.ExcludedDates method), 42  
 get\_previous\_by\_start\_dt() (radioco.apps.schedules.models.ExcludedDates method), 42  
 dioco.apps.schedules.models.Schedule method), 43  
 get\_queryset() (radioco.apps.programmes.admin.NonStaffEpisodeAdmin method), 27  
 get\_queryset() (radioco.apps.programmes.admin.NonStaffParticipantInline method), 28  
 get\_queryset() (radioco.apps.programmes.admin.NonStaffProgrammeAdmin method), 29  
 get\_queryset() (radioco.apps.programmes.admin.NonStaffRoleInline method), 29  
 get\_queryset() (radioco.apps.users.views.UsersView method), 47  
 get\_readonly\_fields() (radioco.apps.programmes.admin.NonStaffEpisodeAdmin method), 27  
 get\_schedule\_which\_excluded\_dt() (radioco.apps.schedules.models.Schedule static method), 43  
 get\_instance\_photos() (radioco.apps.api.serializers.RadiocomConfigurationSerializer method), 16  
 get\_title() (radioco.apps.api.serializers.ScheduleSerializer method), 17  
 get\_twitter\_url() (radioco.apps.api.serializers.RadiocomConfigurationSerializer method), 16  
 view\_type\_display() (radioco.apps.schedules.models.Schedule method), 43  
 get\_urls() (radioco.apps.global\_settings.admin.SingletonModelAdmin method), 23  
 get\_urls() (radioco.apps.users.admin.SingletonProfileAdmin method), 46  
 GetObjectMixin (class in radioco.apps.radioco.utils), 36  
 gettext\_noop() (in module radioco.configs.base.settings), 48  
 GMT (class in radioco.apps.radioco.tz\_utils), 35

## H

handle() (radioco.apps.radioco.management.commands.create\_example\_data method), 34  
 has\_add\_permission() (radioco.apps.global\_settings.admin.SingletonModelAdmin method), 23  
 has\_add\_permission() (radioco.apps.schedules.admin.ScheduleAdmin method), 41  
 has\_delete\_permission() (radioco.apps.users.admin.SingletonProfileAdmin method), 46  
 has\_delete\_permission() (radioco.apps.global\_settings.admin.SingletonModelAdmin method), 23  
 has\_delete\_permission() (radioco.apps.users.admin.SingletonProfileAdmin method), 46

- has\_reurrences() (radioco.apps.schedules.models.Schedule method), 43
- I
- include\_date() (radioco.apps.schedules.models.Schedule method), 43
- index() (in module radioco.apps.radioco.views), 36
- inline\_classes (radioco.apps.programmes.admin.PodcastInline attribute), 30
- inline\_classes (radioco.apps.users.admin.UserProfileInline attribute), 46
- inlines (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 27
- inlines (radioco.apps.programmes.admin.NonStaffProgrammeAdmin attribute), 29
- inlines (radioco.apps.users.admin.UserProfileAdmin attribute), 46
- item\_description() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- item\_enclosure\_length() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- item\_enclosure\_mime\_type() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- item\_enclosure\_url() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- item\_extra\_kwargs() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- item\_guid() (radioco.apps.programmes.feeds.RssProgrammeFeed method), 31
- item\_keywords() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- item\_pubdate() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- item\_title() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- items() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- iTunesFeed (class in radioco.apps.programmes.feeds), 31
- L
- last() (radioco.apps.programmes.models.EpisodeManager static method), 31
- link() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- list() (radioco.apps.api.views.ProgrammeViewSet method), 18
- list() (radioco.apps.api.views.RadiocomStation method), 19
- list() (radioco.apps.api.views.TransmissionViewSet method), 20
- list\_display (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 27
- list\_display (radioco.apps.programmes.admin.NonStaffProgrammeAdmin attribute), 29
- list\_display (radioco.apps.schedules.admin.CalendarAdmin attribute), 41
- list\_filter (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 27
- list\_filter (radioco.apps.programmes.admin.NonStaffProgrammeAdmin attribute), 29
- list\_filter (radioco.apps.schedules.admin.CalendarAdmin attribute), 41
- list\_filter (radioco.apps.users.admin.UserProfileAdmin attribute), 46
- list\_select\_related (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 27
- list\_select\_related (radioco.apps.programmes.admin.NonStaffRoleInline attribute), 29
- localize() (radioco.apps.radioco.tz\_utils.GMT method), 35
- lookup\_field (radioco.apps.api.views.ProgrammeViewSet attribute), 18
- lookups() (radioco.apps.programmes.admin.OwnEpisodeIssueDateListFilter method), 29
- lookups() (radioco.apps.programmes.admin.OwnEpisodeProgrammeListFilter method), 30
- lowercase\_language() (in module radioco.apps.programmes.migrations.0002\_change\_language\_choices), 25
- M
- media (module), 12
- media (radioco.apps.api.views.ProgrammeFilterForm attribute), 18
- media (radioco.apps.api.views.TimezoneForm attribute), 19
- media (radioco.apps.api.views.TransmissionForm attribute), 20
- media (radioco.apps.global\_settings.admin.PodcastConfigurationAdmin attribute), 22
- media (radioco.apps.global\_settings.admin.SingletonModelAdmin attribute), 23
- media (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 27
- media (radioco.apps.programmes.admin.NonStaffEpisodeAdminForm attribute), 28
- media (radioco.apps.programmes.admin.NonStaffParticipantInline attribute), 28
- media (radioco.apps.programmes.admin.NonStaffParticipantInlineForm attribute), 28
- media (radioco.apps.programmes.admin.NonStaffProgrammeAdmin attribute), 29
- media (radioco.apps.programmes.admin.NonStaffRoleInline attribute), 29

- media (radioco.apps.programmes.admin.NonStaffRoleInlineForm attribute), 29
- media (radioco.apps.programmes.admin.PodcastInline attribute), 30
- media (radioco.apps.schedules.admin.CalendarAdmin attribute), 41
- media (radioco.apps.schedules.admin.ScheduleAdmin attribute), 41
- media (radioco.apps.schedules.forms.DeleteScheduleForm attribute), 42
- media (radioco.apps.users.admin.NonStaffUserProfileForm attribute), 46
- media (radioco.apps.users.admin.SingletonProfileAdmin attribute), 46
- media (radioco.apps.users.admin.UserProfileAdmin attribute), 46
- media (radioco.apps.users.admin.UserProfileInline attribute), 46
- memorize (class in radioco.apps.radioco.utils), 36
- migrate\_board() (in module radioco.apps.schedules.migrations.0002\_\_v3\_0\_\_renaming\_calendar\_apps), 37
- migrate\_daily\_recurrences() (in module radioco.apps.schedules.migrations.0004\_\_v3\_0\_\_adding\_rrules), 37
- migrate\_datetime\_to\_utc() (in module radioco.apps.radioco.migrations.0001\_\_v3\_0\_\_mysql\_timezone), 34
- migrate\_schedules() (in module radioco.apps.schedules.migrations.0005\_\_v3\_0\_\_migrating\_schedules\_to\_programmes), 37
- Migration (class in radioco.apps.global\_settings.migrations.0001\_initial), 21
- Migration (class in radioco.apps.global\_settings.migrations.0002\_remove\_calendar\_apps), 21
- Migration (class in radioco.apps.global\_settings.migrations.0003\_siteconfiguration\_defaults), 21
- Migration (class in radioco.apps.global\_settings.migrations.0004\_auto\_20150606\_1335), 21
- Migration (class in radioco.apps.global\_settings.migrations.0005\_auto\_20150606\_1415), 21
- Migration (class in radioco.apps.global\_settings.migrations.0006\_auto\_20160116\_1509), 22
- Migration (class in radioco.apps.global\_settings.migrations.0007\_\_v3\_0\_\_calendar\_apps\_site\_tweaks), 22
- Migration (class in radioco.apps.global\_settings.migrations.0008\_\_v3\_0\_\_radioco\_users), 21
- Migration (class in radioco.apps.programmes.migrations.0001\_initial), 25
- Migration (class in radioco.apps.programmes.migrations.0002\_change\_language\_choices), 25
- Migration (class in radioco.apps.programmes.migrations.0003\_change\_textfield\_to\_rich\_text), 25
- Migration (class in radioco.apps.programmes.migrations.0004\_change\_photo\_url), 25
- Migration (class in radioco.apps.programmes.migrations.0005\_auto\_20150531\_1734), 26
- Migration (class in radioco.apps.programmes.migrations.0006\_auto\_20160104\_2029), 26
- Migration (class in radioco.apps.programmes.migrations.0007\_change\_default\_image), 26
- Migration (class in radioco.apps.programmes.migrations.0008\_auto\_20160116\_1509), 26
- Migration (class in radioco.apps.programmes.migrations.0009\_\_v3\_0\_\_small\_tweaks), 26
- Migration (class in radioco.apps.programmes.migrations.0010\_\_v3\_2\_\_convert\_role), 27
- Migration (class in radioco.apps.programmes.migrations.0011\_\_v3\_2\_\_ensure\_one\_user\_per\_programme), 27
- Migration (class in radioco.apps.radioco.migrations.0001\_\_v3\_0\_\_mysql\_timezone), 34
- Migration (class in radioco.apps.schedules.migrations.0001\_initial), 36
- Migration (class in radioco.apps.schedules.migrations.0002\_\_v3\_0\_\_renaming\_calendar\_apps), 37
- Migration (class in radioco.apps.schedules.migrations.0003\_\_v3\_0\_\_create\_excludedddays), 37
- Migration (class in radioco.apps.schedules.migrations.0004\_\_v3\_0\_\_adding\_rrules), 37
- Migration (class in radioco.apps.schedules.migrations.0005\_\_v3\_0\_\_migrating\_schedules\_to\_programmes), 37
- Migration (class in radioco.apps.schedules.migrations.0006\_\_v3\_0\_\_calendar\_apps\_site\_tweaks), 37
- Migration (class in radioco.apps.schedules.migrations.0007\_change\_default\_image), 37
- Migration (class in radioco.apps.schedules.migrations.0008\_\_v3\_0\_\_radioco\_users), 37



	tribute), 44
Migration (class in radioco.apps.users.migrations.0002_change_textfield_to_staff_episode_admin), 45	next_dates() (in module radioco.apps.schedules.utils), 44
Migration (class in radioco.apps.users.migrations.0003_auto_20161014_2029), 45	NonStaffEpisodeAdmin (class in radioco.apps.programmes.admin), 27
Migration (class in radioco.apps.users.migrations.0004__v3_0__ckeditor_non_staff_episode_admin), 45	NonStaffEpisodeAdminForm (class in radioco.apps.programmes.admin), 27
	NonStaffEpisodeAdminForm.Meta (class in radioco.apps.programmes.admin), 28
mock_now() (in module radioco.apps.api.tests.test_radiocom_api), 14	NonStaffParticipantInline (class in radioco.apps.programmes.admin), 28
mock_now() (in module radioco.apps.api.tests.test_schedules), 15	NonStaffParticipantInlineForm (class in radioco.apps.programmes.admin), 28
mock_now() (in module radioco.apps.schedules.tests.test_schedules), 40	NonStaffParticipantInlineFormset (class in radioco.apps.programmes.admin), 28
model (radioco.apps.api.serializers.EpisodeSerializer.Meta attribute), 16	NonStaffProgrammeAdmin (class in radioco.apps.programmes.admin), 28
model (radioco.apps.api.serializers.ProgrammeSerializer.Meta attribute), 16	NonStaffRoleInline (class in radioco.apps.programmes.admin), 29
model (radioco.apps.api.serializers.RadiocomConfigurationSerializer.Meta attribute), 16	NonStaffRoleInlineForm (class in radioco.apps.programmes.admin), 29
model (radioco.apps.api.serializers.RadiocomProgrammeSerializer.Meta attribute), 16	NonStaffRoleInlineFormset (class in radioco.apps.programmes.admin), 29
model (radioco.apps.api.serializers.ScheduleSerializer.Meta attribute), 17	NonStaffRoleProfileForm (class in radioco.apps.users.admin), 45
model (radioco.apps.api.views.EpisodeFilter.Meta attribute), 17	now() (radioco.apps.radioco.tz_utils.GMT method), 35
model (radioco.apps.api.views.ProgrammeFilter.Meta attribute), 18	now() (radioco.apps.api.views.TransmissionViewSet method), 20
model (radioco.apps.api.views.ScheduleFilter.Meta attribute), 19	
model (radioco.apps.programmes.admin.NonStaffEpisodeAdminForm.Meta attribute), 28	<b>O</b>
model (radioco.apps.programmes.admin.NonStaffParticipantInline attribute), 28	object (radioco.apps.radioco.utils.GetObjectMixin attribute), 36
model (radioco.apps.programmes.admin.NonStaffRoleInline attribute), 29	object_history_template (radioco.apps.global_settings.admin.SingletonModelAdmin attribute), 23
model (radioco.apps.programmes.admin.PodcastInline attribute), 30	objects (radioco.apps.global_settings.models.CalendarConfiguration attribute), 23
model (radioco.apps.radioco.utils.DeletePermissionMixin attribute), 36	objects (radioco.apps.global_settings.models.PodcastConfiguration attribute), 23
model (radioco.apps.schedules.views.DeleteScheduleView attribute), 44	objects (radioco.apps.global_settings.models.RadiocomConfiguration attribute), 24
model (radioco.apps.users.admin.UserProfileInline attribute), 46	objects (radioco.apps.global_settings.models.SingletonModel attribute), 24
model (radioco.apps.users.views.UsersView attribute), 47	objects (radioco.apps.global_settings.models.SiteConfiguration attribute), 24
MyTestSuiteRunner (class in radioco.configs.base.test_runner), 48	objects (radioco.apps.programmes.models.Episode attribute), 31
	objects (radioco.apps.programmes.models.Participant attribute), 32
	objects (radioco.apps.programmes.models.Podcast attribute), 32
	objects (radioco.apps.programmes.models.Programme attribute), 32
<b>N</b>	
name (radioco.apps.schedules.models.Transmission at-	



podcast (radioco.apps.programmes.models.Episode attribute), 31

Podcast.DoesNotExist, 32

Podcast.MultipleObjectsReturned, 32

PodcastConfiguration (class in radioco.apps.global\_settings.models), 23

PodcastConfiguration.DoesNotExist, 23

PodcastConfiguration.MultipleObjectsReturned, 23

PodcastConfigurationAdmin (class in radioco.apps.global\_settings.admin), 22

PodcastInline (class in radioco.apps.programmes.admin), 30

PodcastMock (class in radioco.apps.api.tests.test\_recorder), 14

Programme (class in radioco.apps.programmes.models), 32

programme (radioco.apps.programmes.models.Episode attribute), 31

programme (radioco.apps.programmes.models.Role attribute), 33

programme (radioco.apps.schedules.models.Schedule attribute), 43

Programme.DoesNotExist, 32

Programme.MultipleObjectsReturned, 32

programme\_detail() (in module radioco.apps.programmes.views), 33

programme\_url (radioco.apps.schedules.models.Transmission attribute), 44

ProgrammeFeed (class in radioco.apps.programmes.feeds), 30

ProgrammeFilter (class in radioco.apps.api.views), 18

ProgrammeFilter.Meta (class in radioco.apps.api.views), 18

ProgrammeFilterForm (class in radioco.apps.api.views), 18

ProgrammeSerializer (class in radioco.apps.api.serializers), 16

ProgrammeSerializer.Meta (class in radioco.apps.api.serializers), 16

ProgrammeViewSet (class in radioco.apps.api.views), 18

## Q

queryset (radioco.apps.api.views.EpisodeViewSet attribute), 18

queryset (radioco.apps.api.views.ProgrammeViewSet attribute), 18

queryset (radioco.apps.api.views.ScheduleViewSet attribute), 19

queryset (radioco.apps.api.views.TransmissionOperationViewSet attribute), 20

queryset (radioco.apps.api.views.TransmissionViewSet attribute), 20

queryset() (radioco.apps.programmes.admin.OwnEpisodeIssue attribute), 30

queryset() (radioco.apps.programmes.admin.OwnEpisodeProgrammeListFilter attribute), 30

## R

radioco (module), 49

radioco.apps (module), 47

radioco.apps.api (module), 20

radioco.apps.api.recorder\_views (module), 15

radioco.apps.api.serializers (module), 15

radioco.apps.api.tests (module), 15

radioco.apps.api.tests.test\_api (module), 12

radioco.apps.api.tests.test\_fullcalendar (module), 13

radioco.apps.api.tests.test\_programmes (module), 13

radioco.apps.api.tests.test\_radiocom\_api (module), 14

radioco.apps.api.tests.test\_recorder (module), 14

radioco.apps.api.tests.test\_schedules (module), 14

radioco.apps.api.urls (module), 17

radioco.apps.api.views (module), 17

radioco.apps.api.viewsets (module), 20

radioco.apps.global\_settings (module), 24

radioco.apps.global\_settings.admin (module), 22

radioco.apps.global\_settings.migrations (module), 22

radioco.apps.global\_settings.migrations.0001\_initial (module), 21

radioco.apps.global\_settings.migrations.0002\_remove\_calendarconfiguration (module), 21

radioco.apps.global\_settings.migrations.0003\_siteconfiguration\_footer (module), 21

radioco.apps.global\_settings.migrations.0004\_auto\_20150606\_1335 (module), 21

radioco.apps.global\_settings.migrations.0005\_auto\_20150606\_1415 (module), 21

radioco.apps.global\_settings.migrations.0006\_auto\_20160116\_1509 (module), 22

radioco.apps.global\_settings.migrations.0007\_\_v3\_0\_\_calendarconfiguration (module), 22

radioco.apps.global\_settings.migrations.0008\_\_v3\_0\_\_radiocomconfiguration (module), 22

radioco.apps.global\_settings.models (module), 23

radioco.apps.global\_settings.templatetags (module), 22

radioco.apps.global\_settings.templatetags.global\_settings\_extras (module), 22

radioco.apps.programmes (module), 33

radioco.apps.programmes.admin (module), 27

radioco.apps.programmes.feeds (module), 30

radioco.apps.programmes.migrations (module), 27

radioco.apps.programmes.migrations.0001\_initial (module), 25

radioco.apps.programmes.migrations.0002\_change\_language\_choices (module), 25

radioco.apps.programmes.migrations.0003\_change\_textfield\_to\_richtextfield (module), 25

radioco.apps.programmes.migrations.0004\_change\_photo\_url (module), 25



- radioco.apps.programmes.migrations.0005\_auto\_20150531\_1734 (module), 26
- radioco.apps.programmes.migrations.0006\_auto\_20160104\_2029 (module), 26
- radioco.apps.programmes.migrations.0007\_change\_default\_image (module), 26
- radioco.apps.programmes.migrations.0008\_auto\_20160116\_1509 (module), 26
- radioco.apps.programmes.migrations.0009\_\_v3\_0\_\_small\_tweaks (module), 26
- radioco.apps.programmes.migrations.0010\_\_v3\_2\_\_convert\_role (module), 27
- radioco.apps.programmes.migrations.0011\_\_v3\_2\_\_ensure\_one\_user (module), 27
- radioco.apps.programmes.models (module), 31
- radioco.apps.programmes.urls (module), 33
- radioco.apps.programmes.utils (module), 33
- radioco.apps.programmes.views (module), 33
- radioco.apps.radioco (module), 36
- radioco.apps.radioco.context\_processors (module), 35
- radioco.apps.radioco.management (module), 34
- radioco.apps.radioco.management.commands (module), 34
  - radioco.apps.radioco.management.commands.create\_example\_data (module), 34
- radioco.apps.radioco.migrations (module), 34
- radioco.apps.radioco.migrations.0001\_\_v3\_0\_\_mysql\_timezone (module), 34
- radioco.apps.radioco.test\_utils (module), 35
- radioco.apps.radioco.tests (module), 35
- radioco.apps.radioco.tests.tests (module), 34
- radioco.apps.radioco.tz\_utils (module), 35
- radioco.apps.radioco.utils (module), 36
- radioco.apps.radioco.views (module), 36
- radioco.apps.schedules (module), 44
- radioco.apps.schedules.admin (module), 41
- radioco.apps.schedules.forms (module), 42
- radioco.apps.schedules.migrations (module), 38
- radioco.apps.schedules.migrations.0001\_initial (module), 36
- radioco.apps.schedules.migrations.0002\_\_v3\_0\_\_renaming\_RadiocomProgramme (module), 37
- radioco.apps.schedules.migrations.0003\_\_v3\_0\_\_create\_example\_data (module), 37
- radioco.apps.schedules.migrations.0004\_\_v3\_0\_\_adding\_rules (module), 37
- radioco.apps.schedules.migrations.0005\_\_v3\_0\_\_migrating\_RadiocomStation (module), 37
- radioco.apps.schedules.models (module), 42
- radioco.apps.schedules.tests (module), 41
- radioco.apps.schedules.tests.test\_reurrences (module), 38
- radioco.apps.schedules.tests.test\_schedules (module), 39
- radioco.apps.schedules.tests.test\_timezone (module), 41
- radioco.apps.schedules.urls (module), 44
- radioco.apps.schedules.utils (module), 44
- radioco.apps.schedules.views (module), 44
- radioco.apps.users (module), 47
- radioco.apps.users.admin (module), 45
- radioco.apps.users.migrations (module), 45
- radioco.apps.users.migrations.0001\_initial (module), 45
- radioco.apps.users.migrations.0002\_change\_textfield\_to\_richtextfield (module), 45
- radioco.apps.users.migrations.0003\_auto\_20160104\_2029 (module), 45
- radioco.apps.users.migrations.0004\_\_v3\_0\_\_ckeditor\_upload\_field (module), 45
- radioco.apps.users.models (module), 46
- radioco.apps.users.tests (module), 47
- radioco.apps.users.urls (module), 47
- radioco.apps.users.views (module), 47
- radioco.configs (module), 49
- radioco.configs.base (module), 48
- radioco.configs.base.manage (module), 47
- radioco.configs.base.settings (module), 48
- radioco.configs.base.test\_runner (module), 48
- radioco.configs.base.urls (module), 48
- radioco.configs.development (module), 48
- radioco.configs.development.manage (module), 48
- radioco.configs.heroku (module), 48
- radioco.configs.heroku.manage (module), 48
- radioco.configs.heroku.settings (module), 48
- radioco.configs.heroku.wsgi (module), 48
- radioco.configs.staging (module), 49
- radioco.configs.staging.manage (module), 48
- radioco.configs.staging.wsgi (module), 48
- RadiocomConfiguration (class in radioco.apps.global\_settings.models), 23
- RadiocomConfiguration.DoesNotExist, 24
- RadiocomConfiguration.MultipleObjectsReturned, 24
- RadiocomConfigurationSerializer (class in radioco.apps.api.serializers), 16
- RadiocomConfigurationSerializer.Meta (class in radioco.apps.api.serializers), 16
- RadiocomProgrammeSerializer (class in radioco.apps.api.serializers), 16
- RadiocomProgrammeSerializer.Meta (class in radioco.apps.api.serializers), 16
- RadiocomProgrammeViewSet (class in radioco.apps.api.views), 18
- RadiocomStation (class in radioco.apps.api.views), 19
- RadiocomTransmissionSerializer (class in radioco.apps.api.serializers), 16
- RadiocomTransmissionViewSet (class in radioco.apps.api.views), 19
- RadioIntegrationTests (class in radioco.apps.radioco.tests.tests), 34



- 40
- ScheduleViewSet (class in radioco.apps.api.views), 19
- search\_fields (radioco.apps.programmes.admin.NonStaffEpisodeAdmin class attribute), 27
- search\_fields (radioco.apps.programmes.admin.NonStaffProgrammeAdmin class attribute), 29
- search\_fields (radioco.apps.schedules.admin.CalendarAdmin class attribute), 41
- serializer\_class (radioco.apps.api.views.EpisodeViewSet attribute), 18
- serializer\_class (radioco.apps.api.views.ProgrammeViewSet attribute), 18
- serializer\_class (radioco.apps.api.views.RadiocomProgrammeViewSet attribute), 18
- serializer\_class (radioco.apps.api.views.RadiocomStationViewSet attribute), 19
- serializer\_class (radioco.apps.api.views.RadiocomTransmissionViewSet attribute), 19
- serializer\_class (radioco.apps.api.views.ScheduleViewSet attribute), 19
- serializer\_class (radioco.apps.api.views.TransmissionOperationViewSet attribute), 20
- serializer\_class (radioco.apps.api.views.TransmissionViewSet attribute), 20
- set\_active() (radioco.apps.schedules.admin.CalendarAdmin method), 41
- settings() (in module radioco.apps.radioco.context\_processors), 35
- setUp() (radioco.apps.api.tests.test\_fullcalendar.TestFullCalendarApi method), 13
- setUp() (radioco.apps.api.tests.test\_programmes.TestProgrammesAPI method), 13
- setUp() (radioco.apps.api.tests.test\_radiocom\_api.TestProgramme method), 14
- setUp() (radioco.apps.schedules.tests.test\_reurrences.EmptyRecurrenceTests method), 38
- setUp() (radioco.apps.schedules.tests.test\_reurrences.RecurrenceTests method), 38
- setUp() (radioco.apps.schedules.tests.test\_schedules.CalendarAdminTests method), 39
- setUp() (radioco.apps.schedules.tests.test\_schedules.CalendarManagerTests method), 39
- setUp() (radioco.apps.schedules.tests.test\_schedules.CalendarValidationTests method), 39
- setUp() (radioco.apps.schedules.tests.test\_schedules.ScheduleBetweenTests method), 39
- setUp() (radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests method), 39
- setUp() (radioco.apps.schedules.tests.test\_schedules.ScheduleUtilsTests method), 40
- setUp() (radioco.apps.schedules.tests.test\_schedules.TransmissionModelTests method), 40
- setUp() (radioco.apps.schedules.tests.test\_timezone.ScheduleModelTests method), 38
- setUpTestData() (radioco.apps.api.tests.test\_recorder.TestProgrammesAPI class method), 14
- setUpTestData() (radioco.apps.radioco.test\_utils.TestDataMixin class method), 35
- SingletonModel (class in radioco.apps.global\_settings.models), 24
- SingletonModel.Meta (class in radioco.apps.global\_settings.models), 24
- SingletonModelAdmin (class in radioco.apps.global\_settings.admin), 23
- SingletonModelManager (class in radioco.apps.global\_settings.models), 24
- SingletonProfileAdmin (class in radioco.apps.users.admin), 46
- SiteConfiguration (class in radioco.apps.global\_settings.models), 24
- SiteConfiguration.DoesNotExist, 24
- SiteConfiguration.MultipleObjectsReturned, 24
- slug (radioco.apps.schedules.models.Transmission class attribute), 44
- source (radioco.apps.schedules.models.Schedule class attribute), 43
- spain\_tz() (in module radioco.apps.radioco.test\_utils), 35
- start\_dt (radioco.apps.programmes.models.Programme class attribute), 32
- submit\_recorder() (in module radioco.apps.api.recorder\_views), 15
- suffix (radioco.apps.api.views.EpisodeViewSet attribute), 18
- suffix (radioco.apps.api.views.ProgrammeViewSet attribute), 18
- suffix (radioco.apps.api.views.RadiocomProgrammeViewSet attribute), 19
- suffix (radioco.apps.api.views.RadiocomStationViewSet attribute), 19
- suffix (radioco.apps.api.views.RadiocomTransmissionViewSet attribute), 19
- suffix (radioco.apps.api.views.ScheduleViewSet attribute), 19
- suffix (radioco.apps.api.views.TransmissionOperationViewSet attribute), 20
- suffix (radioco.apps.api.views.TransmissionViewSet attribute), 20
- summary\_text (radioco.apps.programmes.models.Episode class attribute), 31
- synopsis\_text (radioco.apps.programmes.models.Programme class attribute), 32

test\_after() (radioco.apps.schedules.tests.test\_reurrences.RecurrenceTests method), 38

test\_after\_inclusive() (radioco.apps.schedules.tests.test\_reurrences.EmptyRecurrenceTests method), 38

test\_after\_inclusive() (radioco.apps.schedules.tests.test\_reurrences.RecurrenceTests method), 38

test\_api() (radioco.apps.api.tests.test\_api.TestApi method), 12

test\_at() (radioco.apps.schedules.tests.test\_schedules.TransmissionModelTests method), 40

test\_available\_dates\_after() (radioco.apps.schedules.tests.test\_schedules.ScheduleUtilsTests method), 40

test\_available\_dates\_none() (radioco.apps.schedules.tests.test\_schedules.ScheduleUtilsTests method), 40

test\_before() (radioco.apps.schedules.tests.test\_reurrences.EmptyRecurrenceTests method), 38

test\_before\_dt() (radioco.apps.schedules.tests.test\_reurrences.RecurrenceTests method), 38

test\_before\_inclusive() (radioco.apps.schedules.tests.test\_reurrences.EmptyRecurrenceTests method), 38

test\_before\_inclusive() (radioco.apps.schedules.tests.test\_reurrences.RecurrenceTests method), 38

test\_between() (radioco.apps.schedules.tests.test\_schedules.TransmissionModelTests method), 40

test\_between\_by\_queryset() (radioco.apps.schedules.tests.test\_schedules.TransmissionModelTests method), 40

test\_between\_includes\_started\_episode() (radioco.apps.schedules.tests.test\_schedules.ScheduleBetweenTests method), 39

test\_CEST\_transition() (radioco.apps.schedules.tests.test\_timezone.ScheduleModelTests method), 41

test\_CET\_transition() (radioco.apps.schedules.tests.test\_timezone.ScheduleModelTests method), 41

test\_CET\_transitions() (in module radioco.apps.schedules.tests.test\_timezone), 41

test\_cleaned\_internal\_recurrence\_dates() (radioco.apps.schedules.tests.test\_timezone.ScheduleModelTests method), 41

test\_clone\_calendar() (radioco.apps.schedules.tests.test\_schedules.CalendarAdminTests method), 39

test\_current() (radioco.apps.schedules.tests.test\_schedules.CalendarManagerTests method), 39

test\_current\_no\_date() (radioco.apps.schedules.tests.test\_schedules.CalendarManagerTests method), 39

test\_date\_after() (radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests method), 39

test\_date\_after\_exclude() (radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests method), 39

test\_date\_before() (radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests method), 39

test\_dates\_between() (radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests method), 39

test\_dates\_between\_complex\_ruleset() (radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests method), 39

test\_dates\_between\_earlier\_end\_by\_programme() (radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests method), 40

test\_dates\_between\_later\_start\_by\_programme() (radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests method), 40

test\_dictionary\_key() (radioco.apps.radioco.tests.test\_utils.UtilsTest method), 34

test\_end() (radioco.apps.schedules.tests.test\_schedules.TransmissionModelTests method), 40

test\_end\_gt\_calendar() (radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests method), 40

test\_episode() (radioco.apps.api.tests.test\_api.TestSerializers method), 12

test\_episode\_url() (radioco.apps.schedules.tests.test\_schedules.TransmissionModelTests method), 40

test\_episode\_by\_programme() (radioco.apps.api.tests.test\_api.TestSerializers method), 12

test\_episodes\_delete() (radioco.apps.api.tests.test\_programmes.TestNotAllowedMethodsPr method), 13

test\_episodes\_get\_all() (radioco.apps.api.tests.test\_programmes.TestProgrammesAPI method), 13

test\_episodes\_get\_by\_programme() (radioco.apps.api.tests.test\_programmes.TestProgrammesAPI method), 13

test\_episodes\_post() (radioco.apps.api.tests.test\_programmes.TestNotAllowedMethodsPr method), 13

test\_episodes\_put() (radioco.apps.api.tests.test\_programmes.TestNotAllowedMethodsPr method), 13

method), 13		dioco.apps.api.tests.test_programmes.TestProgrammesAPI
test_example_data()	(ra-	method), 13
dioco.apps.radioco.tests.tests.UtilsTest		test_programmes_before()
method), 34		(ra-
test_fields()	(radioco.apps.schedules.tests.test_schedules.ScheduleValidationTests	dioco.apps.api.tests.test_programmes.TestProgrammesAPI
method), 40		method)
test_get_announcers_and_profile()	(ra-	test_programmes_between()
dioco.apps.users.tests.UserProfileMethodTests		(ra-
method), 47		dioco.apps.api.tests.test_programmes.TestProgrammesAPI
test_impossible_recurrence_after()	(ra-	method), 13
dioco.apps.schedules.tests.test_reurrences.RecurrenceTests		test_programmes_delete()
method), 38		(ra-
test_impossible_recurrence_before()	(ra-	dioco.apps.api.tests.test_programmes.TestNotAllowedMethodsPr
dioco.apps.schedules.tests.test_reurrences.RecurrenceTests		method), 13
method), 38		test_programmes_get_all()
test_incorrect_transmission_queries()	(ra-	(ra-
dioco.apps.api.tests.test_schedules.TestTransmissionAPI		dioco.apps.api.tests.test_programmes.TestProgrammesAPI
method), 15		method), 13
test_index()	(radioco.apps.radioco.tests.tests.RadioIntegrationTests	test_programmes_put()
method), 34		(ra-
test_last_date_is_equals_before_limit()	(ra-	dioco.apps.api.tests.test_programmes.TestNotAllowedMethodsPr
dioco.apps.schedules.tests.test_reurrences.RecurrenceTests		method), 13
method), 39		test_programmes_post()
test_move_schedule()	(ra-	(ra-
dioco.apps.api.tests.test_fullcalendar.TestFullCalendarApi		dioco.apps.api.tests.test_programmes.TestNotAllowedMethodsPr
method), 13		method), 13
test_move_schedule_in_tz()	(ra-	test_rearrange_episodes()
dioco.apps.api.tests.test_fullcalendar.TestFullCalendarApi		(ra-
method), 13		test_rearrange_episodes_new_schedule()
test_move_schedule_with_schedules()	(ra-	dioco.apps.schedules.tests.test_schedules.ScheduleUtilsTests
dioco.apps.api.tests.test_fullcalendar.TestFullCalendarApi		method), 40
method), 13		test_rearrange_episodes_only_non_emited_episodes()
test_move_schedule_with_schedules_in_tz()	(ra-	(ra-
dioco.apps.api.tests.test_fullcalendar.TestFullCalendarApi		dioco.apps.schedules.tests.test_schedules.ScheduleUtilsTests
method), 13		method), 40
test_name()	(radioco.apps.schedules.tests.test_schedules.TransmissionModelTests	test_recording_schedules()
method), 40		(ra-
test_no_values()	(radioco.apps.schedules.tests.test_recurrentEmptyRecorsetTests	dioco.apps.api.tests.test_recorder.TestProgrammesAPI
method), 38		method), 14
test_only_one_calendar_active()	(ra-	test_recurrence_rules()
dioco.apps.schedules.tests.test_schedules.CalendarValidationTests		(ra-
method), 39		dioco.apps.schedules.tests.test_schedules.ScheduleModelTests
test_programme()	(radioco.apps.api.tests.test_api.TestSerializers	method), 40
method), 12		dioco.apps.schedules.tests.test_schedules.ScheduleModelTests
test_programme_in_list()	(ra-	method), 40
dioco.apps.api.tests.test_radiocom_api.TestProgrammes		test_rearrange_episodes()
method), 14		(ra-
test_programme_photo_url()	(ra-	dioco.apps.schedules.tests.test_schedules.ScheduleModelTests
dioco.apps.api.tests.test_api.TestSerializers		method), 40
method), 12		test_schedules_delete()
test_programme_url()	(ra-	(ra-
dioco.apps.schedules.tests.test_schedules.TransmissionModelTests		dioco.apps.api.tests.test_schedules.TestRestrictedMethodsSchedu
method), 40		method), 14
test_programmes_after()	(ra-	test_schedules_get_all()
		(ra-
		dioco.apps.api.tests.test_schedules.TestSchedulesAPI
		method), 14
		test_schedules_get_by_calendar()
		(ra-



dioco.apps.api.tests.test\_schedules.TestSchedulesAPI method), 15  
 method), 14  
 test\_schedules\_get\_by\_nonexisting\_calendar() (ra-  
 dioco.apps.api.tests.test\_schedules.TestSchedulesAPI method), 15  
 method), 14  
 test\_schedules\_get\_by\_nonexisting\_programme() (ra-  
 dioco.apps.api.tests.test\_schedules.TestSchedulesAPI method), 14  
 method), 14  
 test\_schedules\_get\_by\_nonexisting\_type() (ra-  
 dioco.apps.api.tests.test\_schedules.TestSchedulesAPI method), 15  
 method), 14  
 test\_schedules\_get\_by\_programme() (ra-  
 dioco.apps.api.tests.test\_schedules.TestSchedulesAPI method), 15  
 method), 14  
 test\_schedules\_get\_by\_type() (ra-  
 dioco.apps.api.tests.test\_schedules.TestSchedulesAPI method), 15  
 method), 15  
 test\_schedules\_post() (ra-  
 dioco.apps.api.tests.test\_fullcalendar.TestFullCalendarApi method), 13  
 method), 13  
 test\_schedules\_post() (ra-  
 dioco.apps.api.tests.test\_schedules.TestRestrictedMethodsScheduleAPI method), 14  
 method), 14  
 test\_schedules\_post\_in\_tz() (ra-  
 dioco.apps.api.tests.test\_fullcalendar.TestFullCalendarApi method), 13  
 method), 13  
 test\_schedules\_put() (ra-  
 dioco.apps.api.tests.test\_schedules.TestRestrictedMethodsScheduleAPI method), 14  
 method), 14  
 test\_slug() (radioco.apps.schedules.tests.test\_schedules.TransmissionModelTests (class in radioco.apps.api.tests.test\_schedules.TestRestrictedMethodsScheduleAPI method), 40  
 method), 40  
 test\_start() (radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests (class in radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests method), 40  
 method), 40  
 test\_start() (radioco.apps.schedules.tests.test\_schedules.TransmissionModelTests (class in radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests method), 40  
 method), 40  
 test\_start\_date\_is\_before\_start\_limit() (ra-  
 dioco.apps.schedules.tests.test\_recurrences.RecurrenceTests (class in radioco.apps.api.tests.test\_api), 12  
 method), 39  
 method), 39  
 test\_start\_lt\_calendar() (ra-  
 dioco.apps.schedules.tests.test\_schedules.ScheduleModelTests (class in radioco.apps.schedules.tests.test\_fullcalendar), 13  
 method), 40  
 method), 40  
 test\_str() (radioco.apps.schedules.tests.test\_schedules.CalendarModelTests (class in radioco.apps.schedules.tests.test\_recurrences.RecurrenceTests (class in radioco.apps.api.tests.test\_api), 12  
 method), 39  
 method), 39  
 test\_submit\_recorder() (ra-  
 dioco.apps.api.tests.test\_recorder.TestProgrammesAPI dioco.apps.api.tests.test\_radiocom\_api),  
 method), 14  
 method), 14  
 test\_transform\_dt\_to\_default\_tz() (ra-  
 dioco.apps.schedules.tests.test\_timezone.ScheduleModelTests (class in radioco.apps.schedules.tests.test\_recorder), 14  
 method), 41  
 method), 41  
 test\_transmission() (ra-  
 dioco.apps.api.tests.test\_api.TestSerializers TestRestrictedMethodsScheduleAPI (class in ra-  
 method), 13  
 method), 13  
 test\_transmission\_before() (ra-  
 dioco.apps.api.tests.test\_schedules.TestTransmissionAPI dioco.apps.api.tests.test\_schedules), 14  
 dioco.apps.api.tests.test\_schedules.TestTransmissionAPI dioco.apps.api.tests.test\_schedules), 14

- TestSerializers (class in radioco.apps.api.tests.test\_api), 12
- TestTransmission (class in radioco.apps.api.tests.test\_radiocom\_api), 14
- TestTransmissionAPI (class in radioco.apps.api.tests.test\_schedules), 15
- TimezoneForm (class in radioco.apps.api.views), 19
- title (radioco.apps.programmes.admin.OwnEpisodeIssueDateListFilter attribute), 30
- title (radioco.apps.programmes.admin.OwnEpisodeProgrammeIssueDateListFilter attribute), 30
- title() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- to\_representation() (radioco.apps.api.serializers.AbsoluteURLField method), 15
- to\_representation() (radioco.apps.api.serializers.DateTimeFieldTz method), 16
- transform\_datetime\_tz() (in module radioco.apps.radioco.tz\_utils), 35
- transform\_dt\_to\_default\_tz() (in module radioco.apps.radioco.tz\_utils), 35
- Transmission (class in radioco.apps.schedules.models), 43
- transmission\_dt (radioco.apps.schedules.views.DeleteScheduleView attribute), 44
- TransmissionForm (class in radioco.apps.api.views), 19
- TransmissionModelTests (class in radioco.apps.schedules.tests.test\_schedules), 40
- TransmissionOperationViewSet (class in radioco.apps.api.views), 20
- TransmissionSerializer (class in radioco.apps.api.serializers), 17
- TransmissionSerializerLight (class in radioco.apps.api.serializers), 17
- TransmissionViewSet (class in radioco.apps.api.views), 20
- UserProfile (class in radioco.apps.users.models), 46
- UserProfile.DoesNotExist, 46
- UserProfile.MultipleObjectsReturned, 46
- userprofile\_detail() (in module radioco.apps.users.views), 47
- UserProfileAdmin (class in radioco.apps.users.admin), 46
- UserProfileInline (class in radioco.apps.users.admin), 46
- UserProfileMethodTests (class in radioco.apps.users.tests), 47
- UsersView (class in radioco.apps.users.views), 47
- UnitTests (class in radioco.apps.radioco.tests.tests), 34

## V

- validate() (radioco.apps.api.serializers.ScheduleSerializer method), 17
- validate() (radioco.apps.api.serializers.TransmissionSerializerLight method), 17

## X

- x (radioco.apps.api.views.TimezoneForm attribute), 19

## U

- unfinished() (radioco.apps.programmes.models.EpisodeManager static method), 31
- update() (radioco.apps.api.views.TransmissionOperationViewSet method), 20
- update\_schedule\_if\_dt\_has\_changed() (in module radioco.apps.programmes.models), 33
- update\_schedule\_performance() (in module radioco.apps.programmes.models), 33
- UpdateOnlyModelViewSet (class in radioco.apps.api.viewsets), 20
- user (radioco.apps.users.models.UserProfile attribute), 47
- user\_logout() (in module radioco.apps.radioco.views), 36