

---

# **radioco Documentation**

***Release 3.0***

**Iago Veloso Abalo**

December 24, 2016



<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 . . . . .	9
3.5	django-radio . . . . .	11
3.6	Release notes & upgrade information . . . . .	43
3.7	Indices and tables . . . . .	46
	<b>Python Module Index</b>	<b>49</b>



### 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

---



---

**Release Notes**

---

This document refers to version 3.0





---

## Table of contents

---

### 3.1 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.

#### 3.1.1 Installing web application

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

##### Installing on Ubuntu

The easiest way of installing the app is using [Docker engine](#)

Follow the [installation steps](#)

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"

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
```

### 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](#).

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

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

**Note:** If you see an error related with postgres sql wait a minute and just run the command again

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"
```

### 3.1.2 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
```

### 3.1.3 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/
```

## 3.2 Customize

Technical reference material.

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

### 3.2.1 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

### 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.

---

### 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'
```

## 3.2.2 Recorder Program Setup

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

## 3.3 API

Api documentation.

### 3.3.1 Browsable API

*New 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
```

## 3.4 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!

### 3.4.1 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.

### 3.4.2 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.

### 3.4.3 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.

## 3.5 django-radio

### 3.5.1 manage module

### 3.5.2 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.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_api ()
    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\_recorder module**

**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.radiocom\_views module

radioco.apps.api.radiocom\_views.**programmes\_json** (*request*)

radioco.apps.api.radiocom\_views.**station\_json** (*request*)

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

radioco.apps.api.recorder\_views.**check\_recorder\_program** (*user*)

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

```
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', 'language', 'category')

**model**

alias of Programme

```
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 at 0x7fba4EpisodeFilter.**declared\_filters** = OrderedDict([(u'programme', <django\_filters.filters.CharFilter object at 0x7f

```
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 at 0x7fba445284d0>),
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([(‘after’, <django.forms.fields.DateField object at 0x7fba445284d0>), (‘before’, <django.forms.fields.DateField object at 0x7fba445284d0>)]
clean ()
declared_fields = OrderedDict([(‘after’, <django.forms.fields.DateField object at 0x7fba445284d0>), (‘before’, <django.forms.fields.DateField object at 0x7fba445284d0>)]
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.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 at 0x7fba445284d0>),
ScheduleFilter.declared_filters = OrderedDict([(‘programme’, <django_filters.filters.CharFilter object at 0x7fba445284d0>),
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.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: `django.forms.forms.Form`

**base\_fields = OrderedDict([('after', <django.forms.fields.DateField object at 0x7fba444d3150>), ('before', <django.f**

**clean()**

**clean\_timezone()**

**declared\_fields = OrderedDict([('after', <django.forms.fields.DateField object at 0x7fba444d3150>), ('before', <dja**

**media**

**x = 'Zulu'**

```
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's

**radioco.apps.global\_settings.migrations.0007\_dev\_calendarconfiguration\_tweaks module****radioco.apps.global\_settings.migrations.0008\_dev\_radiocomconfiguration module****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)  
  
classmethod SingletonModel.get_global ()  
  
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
    ter_address, face-
    book_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

#### Subpackages

### radioco.apps.programmes.migrations package

#### Submodules

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

```

class radioco.apps.programmes.migrations.0001_initial.Migration (name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [(u'auth', u'__first__')]

    operations = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'title', <django.db.models.fields.CharField>)]>]

```

#### radioco.apps.programmes.migrations.0002\_change\_language\_choices module

```

class radioco.apps.programmes.migrations.0002_change_language_choices.Migration (name, app_label)
    Bases: django.db.migrations.migration.Migration

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

    operations = [<RunPython <function lowercase_language at 0x7fba42f3e5f0>>, <AlterField field=<django.db.models.fields.CharField> name='language' type='text' to_type='text'>]

radioco.apps.programmes.migrations.0002_change_language_choices.lowercase_language (apps, schema_editor)

```

**radioco.apps.programmes.migrations.0003\_change\_textfield\_to\_richtextfield module**

**class** radioco.apps.programmes.migrations.0003\_change\_textfield\_to\_richtextfield.**Migration** (*name, app\_label*)

Bases: django.db.migrations.migration.Migration

**dependencies** = [(u'programmes', u'0002\_change\_language\_choices')]

**operations** = [<AlterField field=<ckeditor.fields.RichTextField>, preserve\_default=True, name=u'summary', model\_name=...

**radioco.apps.programmes.migrations.0004\_change\_photo\_url module**

**class** radioco.apps.programmes.migrations.0004\_change\_photo\_url.**Migration** (*name, app\_label*)

Bases: django.db.migrations.migration.Migration

**dependencies** = [(u'programmes', u'0003\_change\_textfield\_to\_richtextfield')]

**operations** = [<RunPython <function change\_photo\_url at 0x7fba42f3eed8>>]

radioco.apps.programmes.migrations.0004\_change\_photo\_url.**change\_photo\_url** (*apps, schema\_editor*)

**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'program...

**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 at 0x7fba42f3ef50>>, <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\_dev\_auto\_20160820\_1634** module

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 at 0x7fb
```

```
NonStaffEpisodeAdminForm.clean_programme()
```

```
NonStaffEpisodeAdminForm.declared_fields = OrderedDict()
```

```
NonStaffEpisodeAdminForm.media
```

```
class radioco.apps.programmes.admin.NonStaffParticipantInline(parent_model, admin_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

    fieldsets = ((None, {'fields': ('name', 'slug', 'synopsis', 'category', 'current_season', 'photo', 'language', '_runtime')})

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

    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

    description (programme)

    feed_type
        alias of iTunesFeed

    item_guid (episode)

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, season, 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**

**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.**get\_role\_display** (*\*moreargs, \*\*morekwargs*)

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 at 0x7fba4a486410>), ('Business', <django.ut  
  
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.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  
  
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.get_role_display(*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.tests module**

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

    setUp()

    test_create_episode()

    test_issue_date()

    test_last()

    test_last_none()

    test_number_in_season()

    test_people()

    test_programme()

    test_season()

    test_unfinished()

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

    setUp()

    test_absoulte_url()

    test_model_manager()

    test_runtime()

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

    setUp()

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

    setUp()

    test_absolute_url()

    test_runtime()

    test_runtime_is_zero()

    test_runtime_not_get()

    test_runtime_not_set()

    test_save_programme()

    test_slug()

    test_str()
```

**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.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 non 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.In

### radioco.apps.schedules.migrations.0002\_dev\_schedules module

#### 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, **kwargs)
    test_rearrange_only_non_emited_episodes (*args, **kwargs)
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 at 0x7fba4386f4d0>), ('sch**declared\_fields** = OrderedDict([('transmission\_dt', <django.forms.fields.DateTimeField object at 0x7fba4386f4d0>)**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**

**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.DeletePermissionsMixin`

`django.views.generic.edit.ModelForm`

**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

**radioco.apps.users.migrations** package

## Submodules

### **radioco.apps.users.migrations.0001\_initial** module

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

```
    Bases: django.db.migrations.migration.Migration
```

```
    dependencies = [(u'auth', u'__first__')]
```

```
    operations = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'bio', <django.db.models.fields.TextField>)]>]
```

### **radioco.apps.users.migrations.0002\_change\_textfield\_to\_richtextfield** module

```
class radioco.apps.users.migrations.0002_change_textfield_to_richtextfield.Migration (name, app_label)
```

```
    Bases: django.db.migrations.migration.Migration
```

```
    dependencies = [(u'users', u'0001_initial')]
```

```
    operations = [<AlterField field=<ckeditor.fields.RichTextField>, preserve_default=True, name=u'bio', model_name=u'UserProfile'>]
```

### **radioco.apps.users.migrations.0003\_auto\_20160104\_2029** module

```
class radioco.apps.users.migrations.0003_auto_20160104_2029.Migration (name, app_label)
```

```
    Bases: django.db.migrations.migration.Migration
```

```
    dependencies = [(u'users', u'0002_change_textfield_to_richtextfield')]
```

```
    operations = [<AlterField field=<django.db.models.fields.files.ImageField>, name=u'avatar', model_name=u'UserProfile'>]
```

## radioco.apps.users.migrations.0004\_dev\_ckeditor\_upload\_field module

### Module contents

#### Submodules

#### radioco.apps.users.admin module

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

    base_fields = OrderedDict([('username', <django.forms.fields.CharField object at 0x7fba4a3929d0>), ('first_name', <django.forms.fields.CharField object at 0x7fba4a3929d0>), ('last_name', <django.forms.fields.CharField object at 0x7fba4a3929d0>), ('email', <django.forms.fields.EmailField object at 0x7fba4a3929d0>), ('bio', <django.forms.fields.CharField object at 0x7fba4a3929d0>), ('avatar', <django.forms.fields.ImageField object at 0x7fba4a3929d0>), ('display_personal_page', <django.forms.fields.BooleanField object at 0x7fba4a3929d0>)])
    clean_username()
    declared_fields = OrderedDict([('username', <django.forms.fields.CharField object at 0x7fba4a3929d0>), ('first_name', <django.forms.fields.CharField object at 0x7fba4a3929d0>), ('last_name', <django.forms.fields.CharField object at 0x7fba4a3929d0>), ('email', <django.forms.fields.EmailField object at 0x7fba4a3929d0>), ('bio', <django.forms.fields.CharField object at 0x7fba4a3929d0>), ('avatar', <django.forms.fields.ImageField object at 0x7fba4a3929d0>), ('display_personal_page', <django.forms.fields.BooleanField object at 0x7fba4a3929d0>)])
    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.**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.urls module**

**Module contents**

**radioco.configs.development package**

**Submodules**

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

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

`radioco.configs.development.settings.show_toolbar(request)`

**Module contents**

**radioco.configs.heroku package**

**Submodules**

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

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

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

**Module contents**

**Module contents**

## Module contents

### 3.5.3 tasks package

#### Submodules

tasks.docker module

tasks.docs module

tasks.heroku module

tasks.locale module

tasks.radioco module

tasks.utils module

## Module contents

## 3.6 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.

### 3.6.1 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
```

## 3.6.2 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 *
```

## 3.6.3 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
```

## 3.6.4 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
```

## 3.6.5 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.

---

## 3.7 Indices and tables

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

- search



## m

manage, 11

## r

radioco, 43

radioco.apps, 41

radioco.apps.api, 18

radioco.apps.api.radiocom\_views, 14

radioco.apps.api.recorder\_views, 14

radioco.apps.api.serializers, 14

radioco.apps.api.tests, 14

radioco.apps.api.tests.test\_api, 11

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

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

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

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

radioco.apps.api.urls, 15

radioco.apps.api.views, 15

radioco.apps.api.viewsets, 17

radioco.apps.global\_settings, 21

radioco.apps.global\_settings.admin, 19

radioco.apps.global\_settings.migrations,  
19

radioco.apps.global\_settings.migrations.0001\_initial,  
18

radioco.apps.global\_settings.migrations.0002\_remove\_calendarconfiguration\_display\_next\_week,  
18

radioco.apps.global\_settings.migrations.0003\_siteconfiguration\_footer,  
18

radioco.apps.global\_settings.migrations.0004\_auto\_20150808\_1335,  
18

radioco.apps.global\_settings.migrations.0005\_auto\_20150808\_1413,  
18

radioco.apps.global\_settings.migrations.0006\_auto\_20160118\_1509,  
19

radioco.apps.global\_settings.models, 20

radioco.apps.global\_settings.templatetags,  
19

radioco.apps.global\_settings.templatetags.global\_se,  
19

radioco.apps.programmes, 30

radioco.apps.programmes.admin, 23

radioco.apps.programmes.feeds, 26

radioco.apps.programmes.migrations, 23

radioco.apps.programmes.migrations.0001\_initial,  
21

radioco.apps.programmes.migrations.0002\_change\_lang,  
21

radioco.apps.programmes.migrations.0003\_change\_text,  
22

radioco.apps.programmes.migrations.0004\_change\_phot,  
22

radioco.apps.programmes.migrations.0005\_auto\_20150808\_1335,  
22

radioco.apps.programmes.migrations.0006\_auto\_20160118\_1509,  
22

radioco.apps.programmes.migrations.0007\_change\_defa,  
22

radioco.apps.programmes.migrations.0008\_auto\_20160118\_1509,  
22

radioco.apps.programmes.models, 27

radioco.apps.programmes.tests, 29

radioco.apps.programmes.urls, 29

radioco.apps.programmes.utils, 30

radioco.apps.programmes.views, 30

radioco.apps.radioco, 32

radioco.apps.radioco.configuration\_display\_next\_week,  
31

radioco.apps.radioco.management, 30

radioco.apps.radioco.management.commands,  
30

radioco.apps.radioco.management.commands.create\_ex,  
30

radioco.apps.radioco.test\_utils, 31

radioco.apps.radioco.tests, 31

radioco.apps.radioco.tests.tests, 30

radioco.apps.radioco.tz\_utils, 31

- radioco.apps.radioco.utils, 32
- radioco.apps.radioco.views, 32
- radioco.apps.schedules, 39
- radioco.apps.schedules.admin, 36
- radioco.apps.schedules.forms, 36
- radioco.apps.schedules.migrations, 32
- radioco.apps.schedules.migrations.0001\_initial, 32
- radioco.apps.schedules.models, 36
- radioco.apps.schedules.tests, 35
- radioco.apps.schedules.tests.test\_recurrences, 33
- radioco.apps.schedules.tests.test\_schedules, 33
- radioco.apps.schedules.tests.test\_timezone, 35
- radioco.apps.schedules.urls, 38
- radioco.apps.schedules.utils, 38
- radioco.apps.schedules.views, 38
- radioco.apps.users, 41
- radioco.apps.users.admin, 40
- radioco.apps.users.migrations, 40
- radioco.apps.users.migrations.0001\_initial, 39
- radioco.apps.users.migrations.0002\_change\_textfield\_to\_richtextfield, 39
- radioco.apps.users.migrations.0003\_auto\_20160104\_2029, 39
- radioco.apps.users.models, 41
- radioco.apps.users.tests, 41
- radioco.apps.users.urls, 41
- radioco.apps.users.views, 41
- radioco.configs, 42
- radioco.configs.base, 42
- radioco.configs.base.manage, 41
- radioco.configs.base.settings, 42
- radioco.configs.base.urls, 42
- radioco.configs.development, 42
- radioco.configs.development.manage, 42
- radioco.configs.development.settings, 42
- radioco.configs.heroku, 42
- radioco.configs.heroku.manage, 42
- radioco.configs.heroku.settings, 42
- radioco.configs.heroku.wsgi, 42

## A

abstract (radioco.apps.global\_settings.models.SingletonModelAdmin attribute), 20

actions (radioco.apps.schedules.admin.CalendarAdmin attribute), 36

add\_item\_elements() (radioco.apps.programmes.feeds.iTunesFeed method), 26

add\_root\_elements() (radioco.apps.programmes.feeds.iTunesFeed method), 26

announcers (radioco.apps.programmes.models.Programme attribute), 28

at() (radioco.apps.schedules.models.Transmission class method), 38

## B

base\_fields (radioco.apps.api.views.ProgrammeFilterForm attribute), 16

base\_fields (radioco.apps.api.views.TransmissionForm attribute), 17

base\_fields (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 23

base\_fields (radioco.apps.programmes.admin.NonStaffParticipantInlineForm attribute), 24

base\_fields (radioco.apps.programmes.admin.NonStaffRoleInlineForm attribute), 25

base\_fields (radioco.apps.schedules.forms.DeleteScheduleForm attribute), 36

base\_fields (radioco.apps.users.admin.NonStaffUserProfileForm attribute), 40

base\_filters (radioco.apps.api.views.EpisodeFilter attribute), 15

base\_filters (radioco.apps.api.views.ProgrammeFilter attribute), 16

base\_filters (radioco.apps.api.views.ScheduleFilter attribute), 16

between() (radioco.apps.schedules.models.Transmission class method), 38

## C

calculate\_effective\_schedule\_end\_dt() (in module radioco.apps.schedules.models), 38

calculate\_effective\_schedule\_start\_dt() (in module radioco.apps.schedules.models), 38

Calendar (class in radioco.apps.schedules.models), 36

calendar (radioco.apps.schedules.models.Schedule attribute), 37

Calendar.DoesNotExist, 36

Calendar.MultipleObjectsReturned, 36

CalendarAdmin (class in radioco.apps.schedules.admin), 36

CalendarAdminTests (class in radioco.apps.schedules.tests.test\_schedules), 33

CalendarConfiguration (class in radioco.apps.global\_settings.models), 20

CalendarConfiguration.DoesNotExist, 20

CalendarConfiguration.MultipleObjectsReturned, 20

CalendarManager (class in radioco.apps.schedules.models), 37

CalendarManagerTests (class in radioco.apps.schedules.tests.test\_schedules), 37

CalendarModelTests (class in radioco.apps.schedules.tests.test\_schedules), 34

CalendarValidationTests (class in radioco.apps.schedules.tests.test\_schedules), 34

can\_delete (radioco.apps.users.admin.UserProfileInline attribute), 40

CATEGORY\_CHOICES (radioco.apps.programmes.models.Programme attribute), 28

change\_form\_template (radioco.apps.global\_settings.admin.SingletonModelAdmin attribute), 19

change\_list\_template (radioco.apps.schedules.admin.ScheduleAdmin attribute), 19

attribute), 36

change\_photo\_url() (in module radioco.apps.programmes.migrations.0004\_change\_photo\_url), 22

change\_programmes() (in module radioco.apps.programmes.migrations.0007\_change\_default\_image), 22

change\_users() (in module radioco.apps.programmes.migrations.0007\_change\_default\_image), 22

change\_view() (radioco.apps.global\_settings.admin.SingletonModelAdmin), 19

change\_view() (radioco.apps.users.admin.SingletonProfileAdmin), 40

changelist\_view() (radioco.apps.schedules.admin.ScheduleAdmin), 36

check\_delete\_permission() (in module radioco.apps.radioco.utils), 32

check\_recorder\_program() (in module radioco.apps.api.recorder\_views), 14

child\_schedules (radioco.apps.schedules.models.Schedule attribute), 37

clean() (radioco.apps.api.views.ProgrammeFilterForm method), 16

clean() (radioco.apps.api.views.TransmissionForm method), 17

clean() (radioco.apps.programmes.admin.NonStaffParticipantInlineForm), 24

clean() (radioco.apps.programmes.admin.NonStaffRoleInlineForm), 25

clean\_programme() (radioco.apps.programmes.admin.NonStaffEpisodeAdminForm method), 23

clean\_timezone() (radioco.apps.api.views.TransmissionForm method), 17

clean\_username() (radioco.apps.users.admin.NonStaffUserProfileForm), 40

clone\_calendar() (radioco.apps.schedules.admin.CalendarAdmin), 36

Command (class in radioco.apps.radioco.management.commands.create\_schedule), 30

create\_episode() (radioco.apps.programmes.models.EpisodeAdminForm method), 27

create\_example\_data() (in module radioco.apps.radioco.utils), 32

create\_test\_data() (in module radioco.apps.radioco.test\_utils), 31

current() (radioco.apps.schedules.models.CalendarManager), 37

date\_after() (radioco.apps.schedules.models.Schedule method), 37

date\_before() (radioco.apps.schedules.models.Schedule method), 37

dates\_between() (radioco.apps.schedules.models.Schedule method), 37

DateTimeFieldTz (class in radioco.apps.api.serializers), 14

declared\_fields (radioco.apps.api.views.ProgrammeFilterForm attribute), 16

declared\_fields (radioco.apps.api.views.TransmissionForm attribute), 17

declared\_fields (radioco.apps.programmes.admin.NonStaffEpisodeAdminForm attribute), 23

declared\_fields (radioco.apps.programmes.admin.NonStaffParticipantInlineForm attribute), 24

declared\_fields (radioco.apps.programmes.admin.NonStaffRoleInlineForm attribute), 25

declared\_fields (radioco.apps.schedules.forms.DeleteScheduleForm attribute), 36

declared\_fields (radioco.apps.users.admin.NonStaffUserProfileForm attribute), 40

declared\_filters (radioco.apps.api.views.EpisodeFilter attribute), 15

declared\_filters (radioco.apps.api.views.ProgrammeFilter attribute), 16

declared\_filters (radioco.apps.api.views.ScheduleFilter attribute), 16

DELETE\_ALL (radioco.apps.schedules.forms.DeleteScheduleForm attribute), 36

DELETE\_ONLY\_THIS (radioco.apps.schedules.forms.DeleteScheduleForm attribute), 36

DELETE\_THIS\_AND\_FOLLOWING (radioco.apps.schedules.forms.DeleteScheduleForm attribute), 36

DeletePermissionMixin (class in radioco.apps.radioco.utils), 32

DeleteScheduleForm (class in radioco.apps.schedules.forms), 36

DeleteScheduleView (class in radioco.apps.schedules.views), 38

dependencies (radioco.apps.global\_settings.migrations.0001\_initial.Migration attribute), 18

dependencies (radioco.apps.global\_settings.migrations.0002\_remove\_calendar attribute), 18

dependencies (radioco.apps.global\_settings.migrations.0003\_siteconfiguration attribute), 18

dependencies (radioco.apps.global\_settings.migrations.0004\_auto\_20150606 attribute), 18

dependencies (radioco.apps.global\_settings.migrations.0005\_auto\_20150606 attribute), 18

## D





fields (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 23

fields (radioco.apps.programmes.admin.NonStaffEpisodeAdminForm attribute), 23

fields (radioco.apps.users.admin.SingletonProfileAdmin attribute), 40

fieldsets (radioco.apps.programmes.admin.NonStaffProgrammeAdmin attribute), 24

fieldsets (radioco.apps.schedules.admin.ScheduleAdmin attribute), 36

filter\_backends (radioco.apps.api.views.EpisodeViewSet attribute), 15

filter\_backends (radioco.apps.api.views.ProgrammeViewSet attribute), 16

filter\_backends (radioco.apps.api.views.ScheduleViewSet attribute), 16

filter\_backends (radioco.apps.api.views.TransmissionViewSet attribute), 17

filter\_class (radioco.apps.api.views.EpisodeViewSet attribute), 15

filter\_class (radioco.apps.api.views.ProgrammeViewSet attribute), 16

filter\_class (radioco.apps.api.views.ScheduleViewSet attribute), 16

filter\_class (radioco.apps.api.views.TransmissionViewSet attribute), 17

fix\_recurrence\_date() (in module radioco.apps.radioco.tz\_utils), 31

fix\_recurrence\_dst() (in module radioco.apps.radioco.tz\_utils), 31

form (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 23

form (radioco.apps.programmes.admin.NonStaffParticipantInline attribute), 24

form (radioco.apps.programmes.admin.NonStaffRoleInline attribute), 24

form (radioco.apps.users.admin.SingletonProfileAdmin attribute), 40

form\_class (radioco.apps.schedules.views.DeleteScheduleView attribute), 38

form\_valid() (radioco.apps.schedules.views.DeleteScheduleView method), 38

formfield\_for\_foreignkey() (radioco.apps.programmes.admin.NonStaffEpisodeAdmin method), 23

formset (radioco.apps.programmes.admin.NonStaffParticipantInline attribute), 24

formset (radioco.apps.programmes.admin.NonStaffRoleInline attribute), 25

from\_collection (radioco.apps.schedules.models.Schedule attribute), 37

get() (radioco.apps.schedules.views.DeleteScheduleView method), 39

get\_absolute\_url() (radioco.apps.programmes.models.Episode method), 27

get\_absolute\_url() (radioco.apps.programmes.models.Podcast method), 28

get\_absolute\_url() (radioco.apps.programmes.models.Programme method), 28

get\_absolute\_url() (radioco.apps.users.models.UserProfile method), 41

get\_active() (radioco.apps.schedules.models.Calendar class method), 36

get\_active\_timezone() (in module radioco.apps.radioco.tz\_utils), 31

get\_category\_display() (radioco.apps.programmes.models.Programme method), 28

get\_context\_data() (radioco.apps.schedules.views.DeleteScheduleView method), 39

get\_first\_day\_display() (radioco.apps.global\_settings.models.CalendarConfiguration method), 20

get\_form() (radioco.apps.programmes.admin.NonStaffProgrammeAdmin method), 24

get\_form\_kwargs() (radioco.apps.schedules.views.DeleteScheduleView method), 39

get\_formset() (radioco.apps.programmes.admin.NonStaffParticipantInline method), 24

get\_formset() (radioco.apps.programmes.admin.NonStaffRoleInline method), 25

get\_global() (radioco.apps.global\_settings.models.SingletonModel class method), 20

get\_global\_model() (in module radioco.apps.global\_settings.templatetags.global\_settings\_extras), 19

get\_initial() (radioco.apps.schedules.views.DeleteScheduleView method), 39

get\_language\_display() (radioco.apps.programmes.models.Programme method), 28

get\_new\_excluded\_datetime() (radioco.apps.schedules.models.ExcludedDates method), 37

get\_next\_by\_date\_joined() (radioco.apps.programmes.models.Role method), 28

get\_next\_by\_datetime() (radioco.apps.schedules.models.ExcludedDates method), 37

get\_next\_by\_start\_dt() (radioco.apps.schedules.models.Schedule method), 37

get() (radioco.apps.global\_settings.models.SingletonModelManager method), 37

## G

[get\\_object\(\)](#) (radioco.apps.programmes.feeds.ProgrammeFeed method), 26  
[get\\_previous\\_by\\_date\\_joined\(\)](#) (radioco.apps.programmes.models.Role method), 28  
[get\\_previous\\_by\\_datetime\(\)](#) (radioco.apps.schedules.models.ExcludedDates method), 37  
[get\\_previous\\_by\\_start\\_dt\(\)](#) (radioco.apps.schedules.models.Schedule method), 37  
[get\\_queryset\(\)](#) (radioco.apps.programmes.admin.NonStaffEpisodeAdmin method), 23  
[get\\_queryset\(\)](#) (radioco.apps.programmes.admin.NonStaffParticipantAdmin method), 24  
[get\\_queryset\(\)](#) (radioco.apps.programmes.admin.NonStaffProgrammeAdmin method), 24  
[get\\_queryset\(\)](#) (radioco.apps.programmes.admin.NonStaffRoleAdmin method), 25  
[get\\_queryset\(\)](#) (radioco.apps.users.views.UsersView method), 41  
[get\\_readonly\\_fields\(\)](#) (radioco.apps.programmes.admin.NonStaffEpisodeAdmin method), 23  
[get\\_role\\_display\(\)](#) (radioco.apps.programmes.models.ParticipantAdmin method), 27  
[get\\_role\\_display\(\)](#) (radioco.apps.programmes.models.RoleAdmin method), 28  
[get\\_schedule\\_which\\_excluded\\_dt\(\)](#) (radioco.apps.schedules.models.Schedule static method), 37  
[get\\_title\(\)](#) (radioco.apps.api.serializers.ScheduleSerializer method), 15  
[get\\_type\\_display\(\)](#) (radioco.apps.schedules.models.Schedule method), 37  
[get\\_urls\(\)](#) (radioco.apps.global\_settings.admin.SingletonModelAdmin method), 19  
[get\\_urls\(\)](#) (radioco.apps.users.admin.SingletonProfileAdmin method), 40  
[GetObjectMixin](#) (class in radioco.apps.radioco.utils), 32  
[gettext\\_noop\(\)](#) (in module radioco.configs.base.settings), 42  
[GMT](#) (class in radioco.apps.radioco.tz\_utils), 31

## H

[handle\(\)](#) (radioco.apps.radioco.management.commands.create\_example\_media\_command method), 30  
[has\\_add\\_permission\(\)](#) (radioco.apps.global\_settings.admin.SingletonModelAdmin method), 19  
[has\\_add\\_permission\(\)](#) (radioco.apps.schedules.admin.ScheduleAdmin method), 36

## L

[last\(\)](#) (radioco.apps.programmes.models.EpisodeManager

- static method), 27
  - link() (radioco.apps.programmes.feeds.ProgrammeFeed method), 26
  - list() (radioco.apps.api.views.ProgrammeViewSet method), 16
  - list() (radioco.apps.api.views.TransmissionViewSet method), 17
  - list\_display (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 23
  - list\_display (radioco.apps.programmes.admin.NonStaffProgrammeAdmin attribute), 24
  - list\_display (radioco.apps.schedules.admin.CalendarAdmin attribute), 36
  - list\_filter (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 23
  - list\_filter (radioco.apps.programmes.admin.NonStaffProgrammeAdmin attribute), 24
  - list\_filter (radioco.apps.schedules.admin.CalendarAdmin attribute), 36
  - list\_filter (radioco.apps.users.admin.UserProfileAdmin attribute), 40
  - list\_select\_related (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 23
  - list\_select\_related (radioco.apps.programmes.admin.NonStaffRoleInlineAdmin attribute), 25
  - localize() (radioco.apps.radioco.tz\_utils.GMT method), 31
  - lookup\_field (radioco.apps.api.views.ProgrammeViewSet attribute), 16
  - lookups() (radioco.apps.programmes.admin.OwnEpisodeIssueDateListFilter method), 25
  - lookups() (radioco.apps.programmes.admin.OwnEpisodeProgrammeListFilter method), 25
  - lowercase\_language() (in module radioco.apps.programmes.migrations.0002\_change\_language\_choices), 21
- M**
- manage (module), 11
  - media (radioco.apps.api.views.ProgrammeFilterForm attribute), 16
  - media (radioco.apps.api.views.TransmissionForm attribute), 17
  - media (radioco.apps.global\_settings.admin.PodcastConfigurationAdmin attribute), 19
  - media (radioco.apps.global\_settings.admin.SingletonModelAdmin attribute), 19
  - media (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 23
  - media (radioco.apps.programmes.admin.NonStaffEpisodeAdminForm attribute), 23
  - media (radioco.apps.programmes.admin.NonStaffParticipantInline attribute), 24
  - media (radioco.apps.programmes.admin.NonStaffParticipantInlineForm attribute), 24
  - media (radioco.apps.programmes.admin.NonStaffProgrammeAdmin attribute), 24
  - media (radioco.apps.programmes.admin.NonStaffRoleInline attribute), 25
  - media (radioco.apps.programmes.admin.NonStaffRoleInlineForm attribute), 25
  - media (radioco.apps.programmes.admin.PodcastInline attribute), 26
  - media (radioco.apps.schedules.admin.CalendarAdmin attribute), 36
  - media (radioco.apps.schedules.admin.ScheduleAdmin attribute), 36
  - media (radioco.apps.schedules.forms.DeleteScheduleForm attribute), 36
  - media (radioco.apps.users.admin.NonStaffUserProfileForm attribute), 40
  - media (radioco.apps.users.admin.SingletonProfileAdmin attribute), 40
  - media (radioco.apps.users.admin.UserProfileAdmin attribute), 40
  - media (radioco.apps.users.admin.UserProfileInline attribute), 40
  - memorize (class in radioco.apps.radioco.utils), 32
  - Migration (class in radioco.apps.global\_settings.migrations.0001\_initial), 18
  - Migration (class in radioco.apps.global\_settings.migrations.0002\_remove\_calendarconform), 18
  - MigrationListFilter (class in radioco.apps.global\_settings.migrations.0003\_siteconfiguration\_form), 18
  - Migration (class in radioco.apps.global\_settings.migrations.0004\_auto\_20150606\_1330), 18
  - Migration (class in radioco.apps.global\_settings.migrations.0005\_auto\_20150606\_1411), 18
  - Migration (class in radioco.apps.global\_settings.migrations.0006\_auto\_20160116\_1509), 19
  - MigrationAdmin (class in radioco.apps.programmes.migrations.0001\_initial), 21
  - Migration (class in radioco.apps.programmes.migrations.0002\_change\_language\_choices), 21
  - MigrationForm (class in radioco.apps.programmes.migrations.0003\_change\_textfield\_to\_richtext), 22
  - Migration (class in radioco.apps.programmes.migrations.0004\_change\_photo\_url), 22

- 22 Migration (class in radioco.apps.programmes.migrations.0005\_auto\_20160104\_2029) (radioco.apps.users.admin.UserProfileInline attribute), 40
- 22 Migration (class in radioco.apps.programmes.migrations.0006\_auto\_20160104\_2029) (radioco.apps.users.views.UsersView attribute), 41
- 22 Migration (class in radioco.apps.programmes.migrations.0007\_change\_default\_episode\_admin) (radioco.apps.schedules.models.Transmission attribute), 38
- 22 Migration (class in radioco.apps.programmes.migrations.0008\_auto\_20160116\_1509) (radioco.apps.schedules.util.next\_dates() (in module radioco.apps.schedules.util)), 38
- 22 Migration (class in radioco.apps.schedules.migrations.0001\_initial) (radioco.apps.programmes.admin.NonStaffEpisodeAdmin (class in radioco.apps.programmes.admin)), 23
- 32 Migration (class in radioco.apps.users.migrations.0001\_initial) (radioco.apps.programmes.admin.NonStaffEpisodeAdminForm (class in radioco.apps.programmes.admin)), 23
- 39 Migration (class in radioco.apps.users.migrations.0002\_change\_textfield\_to\_right\_textfield) (radioco.apps.programmes.admin.NonStaffEpisodeAdminForm.Meta (class in radioco.apps.programmes.admin)), 23
- 39 Migration (class in radioco.apps.users.migrations.0003\_auto\_20160104\_2029) (radioco.apps.programmes.admin.NonStaffParticipantInline (class in radioco.apps.programmes.admin)), 23
- 35 mock\_now() (in module radioco.apps.api.tests.test\_schedules), 13 (radioco.apps.programmes.admin.NonStaffParticipantInlineForm (class in radioco.apps.programmes.admin)), 24
- 35 mock\_now() (in module radioco.apps.schedules.tests.test\_schedules), 15 (radioco.apps.programmes.admin.NonStaffParticipantInlineFormset (class in radioco.apps.programmes.admin)), 24
- model (radioco.apps.api.serializers.EpisodeSerializer.Meta attribute), 14 (radioco.apps.programmes.admin.NonStaffProgrammeAdmin (class in radioco.apps.programmes.admin)), 24
- model (radioco.apps.api.serializers.ProgrammeSerializer.Meta attribute), 14 (radioco.apps.programmes.admin.NonStaffRoleInline (class in radioco.apps.programmes.admin)), 24
- model (radioco.apps.api.serializers.ScheduleSerializer.Meta attribute), 14 (radioco.apps.programmes.admin.NonStaffRoleInlineForm (class in radioco.apps.programmes.admin)), 25
- model (radioco.apps.api.views.EpisodeFilter.Meta attribute), 15 (radioco.apps.programmes.admin.NonStaffRoleInlineFormset (class in radioco.apps.programmes.admin)), 25
- model (radioco.apps.api.views.ProgrammeFilter.Meta attribute), 15 (radioco.apps.users.admin.NonStaffUserProfileForm (class in radioco.apps.users.admin)), 40
- model (radioco.apps.api.views.ScheduleFilter.Meta attribute), 16 normalize() (radioco.apps.radioco.tz\_utils.GMT method), 31
- model (radioco.apps.programmes.admin.NonStaffEpisodeAdminForm.Meta attribute), 23 new() (radioco.apps.api.views.TransmissionViewSet method), 17
- model (radioco.apps.programmes.admin.NonStaffParticipantInline attribute), 24 object (radioco.apps.radioco.util.GetObjectMixin attribute), 32
- model (radioco.apps.programmes.admin.NonStaffRoleInline attribute), 25 object\_history\_template (radioco.apps.global\_settings.admin.SingletonModelAdmin attribute), 19
- model (radioco.apps.programmes.admin.PodcastInline attribute), 26 objects (radioco.apps.global\_settings.models.CalendarConfiguration attribute), 20
- model (radioco.apps.radioco.util.DeletePermissionMixin attribute), 32 objects (radioco.apps.global\_settings.models.PodcastConfiguration attribute), 20
- model (radioco.apps.schedules.views.DeleteScheduleView attribute), 39 objects (radioco.apps.global\_settings.models.RadiocomConfiguration attribute), 20
- objects (radioco.apps.global\_settings.models.SiteConfiguration attribute), 21
- objects (radioco.apps.programmes.models.Episode attribute), 27
- objects (radioco.apps.programmes.models.Participant attribute), 27



objects (radioco.apps.programmes.models.Podcast attribute), 27	OwnEpisodeIssueDateListFilter (class in radioco.apps.programmes.admin), 25
objects (radioco.apps.programmes.models.Programme attribute), 28	OwnEpisodeProgrammeListFilter (class in radioco.apps.programmes.admin), 25
objects (radioco.apps.programmes.models.Role attribute), 28	<b>P</b>
objects (radioco.apps.schedules.models.Calendar attribute), 36	parameter_name (radioco.apps.programmes.admin.OwnEpisodeIssueDateListFilter attribute), 25
objects (radioco.apps.schedules.models.ExcludedDates attribute), 37	parameter_name (radioco.apps.programmes.admin.OwnEpisodeProgrammeListFilter attribute), 25
objects (radioco.apps.schedules.models.Schedule attribute), 38	Participant (class in radioco.apps.programmes.models), 27
objects (radioco.apps.users.models.UserProfile attribute), 41	Participant.DoesNotExist, 27
operations (radioco.apps.global_settings.migrations.0001_initial_migration attribute), 18	Participant.MultipleObjectsReturned, 27
operations (radioco.apps.global_settings.migrations.0002_remove_calendar_configuration_display_next_weeks attribute), 18	participant_set (radioco.apps.programmes.models.Episode attribute), 27
operations (radioco.apps.global_settings.migrations.0003_site_configuration_footer attribute), 18	people (radioco.apps.programmes.models.Episode attribute), 27
operations (radioco.apps.global_settings.migrations.0004_auto_20150606_1335 attribute), 18	perform_update (radioco.apps.api.views.TransmissionOperationViewSet method), 17
operations (radioco.apps.global_settings.migrations.0005_auto_20150606_1415 attribute), 18	permission_classes (radioco.apps.api.views.ProgrammeViewSet attribute), 16
operations (radioco.apps.global_settings.migrations.0006_auto_20160116_1509 attribute), 19	permission_classes (radioco.apps.api.views.ScheduleViewSet attribute), 16
operations (radioco.apps.programmes.migrations.0001_initial_migration attribute), 21	permission_classes (radioco.apps.api.views.TransmissionOperationViewSet attribute), 17
operations (radioco.apps.programmes.migrations.0002_change_language_choices attribute), 21	person (radioco.apps.programmes.models.Participant attribute), 21
operations (radioco.apps.programmes.migrations.0003_change_textfield_to_richtextfield attribute), 22	person (radioco.apps.programmes.models.Role attribute), 21
operations (radioco.apps.programmes.migrations.0004_change_photo_url attribute), 22	Podcast (class in radioco.apps.programmes.models), 27
operations (radioco.apps.programmes.migrations.0005_auto_20150531_1734 attribute), 22	podcast (radioco.apps.programmes.models.Episode attribute), 27
operations (radioco.apps.programmes.migrations.0006_auto_20160104_2029 attribute), 22	Podcast.DoesNotExist, 27
operations (radioco.apps.programmes.migrations.0007_change_default_image attribute), 22	Podcast.MultipleObjectsReturned, 27
operations (radioco.apps.programmes.migrations.0008_auto_20160116_1509 attribute), 22	PodcastConfiguration (class in radioco.apps.global_settings.models), 20
operations (radioco.apps.schedules.migrations.0001_initial_migration attribute), 32	PodcastConfiguration.DoesNotExist, 20
operations (radioco.apps.users.migrations.0001_initial_migration attribute), 39	PodcastConfiguration.MultipleObjectsReturned, 20
operations (radioco.apps.users.migrations.0002_change_textfield_to_richtextfield attribute), 39	PodcastConfigurationAdmin (class in radioco.apps.global_settings.admin), 19
operations (radioco.apps.users.migrations.0003_auto_20160104_2029 attribute), 39	PodcastInline (class in radioco.apps.programmes.admin), 25
ordering (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 23	Programme (class in radioco.apps.programmes.models), 27
ordering (radioco.apps.schedules.admin.CalendarAdmin attribute), 36	programme (radioco.apps.programmes.models.Episode attribute), 27
	programme (radioco.apps.programmes.models.Role attribute), 28
	programme (radioco.apps.schedules.models.Schedule attribute), 38

- programme (radioco.apps.schedules.models.Transmission attribute), 38
- Programme.DoesNotExist, 28
- Programme.MultipleObjectsReturned, 28
- programme\_detail() (in module radioco.apps.programmes.views), 30
- programme\_url (radioco.apps.schedules.models.Transmission attribute), 38
- ProgrammeFeed (class in radioco.apps.programmes.feeds), 26
- ProgrammeFilter (class in radioco.apps.api.views), 15
- ProgrammeFilter.Meta (class in radioco.apps.api.views), 15
- ProgrammeFilterForm (class in radioco.apps.api.views), 16
- ProgrammeModelAdminTests (class in radioco.apps.programmes.tests), 29
- ProgrammeModelTests (class in radioco.apps.programmes.tests), 29
- programmes\_json() (in module radioco.apps.api.radiocom\_views), 14
- ProgrammeSerializer (class in radioco.apps.api.serializers), 14
- ProgrammeSerializer.Meta (class in radioco.apps.api.serializers), 14
- ProgrammeViewSet (class in radioco.apps.api.views), 16
- ## Q
- queryset (radioco.apps.api.views.EpisodeViewSet attribute), 15
- queryset (radioco.apps.api.views.ProgrammeViewSet attribute), 16
- queryset (radioco.apps.api.views.ScheduleViewSet attribute), 16
- queryset (radioco.apps.api.views.TransmissionOperationViewSet attribute), 17
- queryset (radioco.apps.api.views.TransmissionViewSet attribute), 17
- queryset() (radioco.apps.programmes.admin.OwnEpisodeIssueDateFilter method), 25
- queryset() (radioco.apps.programmes.admin.OwnEpisodeProgrammeListFilter method), 25
- ## R
- radioco (module), 43
- radioco.apps (module), 41
- radioco.apps.api (module), 18
- radioco.apps.api.radiocom\_views (module), 14
- radioco.apps.api.recorder\_views (module), 14
- radioco.apps.api.serializers (module), 14
- radioco.apps.api.tests (module), 14
- radioco.apps.api.tests.test\_api (module), 11
- radioco.apps.api.tests.test\_fullcalendar (module), 12
- radioco.apps.api.tests.test\_programmes (module), 12
- radioco.apps.api.tests.test\_recorder (module), 13
- radioco.apps.api.tests.test\_schedules (module), 13
- radioco.apps.api.urls (module), 15
- radioco.apps.api.views (module), 15
- radioco.apps.api.viewsets (module), 17
- radioco.apps.global\_settings (module), 21
- radioco.apps.global\_settings.admin (module), 19
- radioco.apps.global\_settings.migrations (module), 19
- radioco.apps.global\_settings.migrations.0001\_initial (module), 18
- radioco.apps.global\_settings.migrations.0002\_remove\_calendarconfiguration (module), 18
- radioco.apps.global\_settings.migrations.0003\_siteconfiguration\_footer (module), 18
- radioco.apps.global\_settings.migrations.0004\_auto\_20150606\_1335 (module), 18
- radioco.apps.global\_settings.migrations.0005\_auto\_20150606\_1415 (module), 18
- radioco.apps.global\_settings.migrations.0006\_auto\_20160116\_1509 (module), 19
- radioco.apps.global\_settings.models (module), 20
- radioco.apps.global\_settings.templatetags (module), 19
- radioco.apps.global\_settings.templatetags.global\_settings\_extras (module), 19
- radioco.apps.programmes (module), 30
- radioco.apps.programmes.admin (module), 23
- radioco.apps.programmes.feeds (module), 26
- radioco.apps.programmes.migrations (module), 23
- radioco.apps.programmes.migrations.0001\_initial (module), 21
- radioco.apps.programmes.migrations.0002\_change\_language\_choices (module), 21
- radioco.apps.programmes.migrations.0003\_change\_textfield\_to\_richtextfield (module), 22
- radioco.apps.programmes.migrations.0004\_change\_photo\_url (module), 22
- radioco.apps.programmes.migrations.0005\_auto\_20150531\_1734 (module), 22
- radioco.apps.programmes.migrations.0006\_auto\_20160104\_2029 (module), 22
- radioco.apps.programmes.migrations.0007\_change\_default\_image (module), 22
- radioco.apps.programmes.migrations.0008\_auto\_20160116\_1509 (module), 22
- radioco.apps.programmes.models (module), 27
- radioco.apps.programmes.tests (module), 29
- radioco.apps.programmes.urls (module), 29
- radioco.apps.programmes.utils (module), 30
- radioco.apps.programmes.views (module), 30
- radioco.apps.radioco (module), 32
- radioco.apps.radioco.context\_processors (module), 31
- radioco.apps.radioco.management (module), 30
- radioco.apps.radioco.management.commands (module), 30

- radioco.apps.radioco.management.commands.create\_example (module), 30
- radioco.apps.radioco.test\_utils (module), 31
- radioco.apps.radioco.tests (module), 31
- radioco.apps.radioco.tests.tests (module), 30
- radioco.apps.radioco.tz\_utils (module), 31
- radioco.apps.radioco.utils (module), 32
- radioco.apps.radioco.views (module), 32
- radioco.apps.schedules (module), 39
- radioco.apps.schedules.admin (module), 36
- radioco.apps.schedules.forms (module), 36
- radioco.apps.schedules.migrations (module), 32
- radioco.apps.schedules.migrations.0001\_initial (module), 32
- radioco.apps.schedules.models (module), 36
- radioco.apps.schedules.tests (module), 35
- radioco.apps.schedules.tests.test\_reurrences (module), 33
- radioco.apps.schedules.tests.test\_schedules (module), 33
- radioco.apps.schedules.tests.test\_timezone (module), 35
- radioco.apps.schedules.urls (module), 38
- radioco.apps.schedules.utils (module), 38
- radioco.apps.schedules.views (module), 38
- radioco.apps.users (module), 41
- radioco.apps.users.admin (module), 40
- radioco.apps.users.migrations (module), 40
- radioco.apps.users.migrations.0001\_initial (module), 39
- radioco.apps.users.migrations.0002\_change\_textfield\_to\_rich\_text (module), 39
- radioco.apps.users.migrations.0003\_auto\_20160104\_2029 (module), 39
- radioco.apps.users.models (module), 41
- radioco.apps.users.tests (module), 41
- radioco.apps.users.urls (module), 41
- radioco.apps.users.views (module), 41
- radioco.configs (module), 42
- radioco.configs.base (module), 42
- radioco.configs.base.manage (module), 41
- radioco.configs.base.settings (module), 42
- radioco.configs.base.urls (module), 42
- radioco.configs.development (module), 42
- radioco.configs.development.manage (module), 42
- radioco.configs.development.settings (module), 42
- radioco.configs.heroku (module), 42
- radioco.configs.heroku.manage (module), 42
- radioco.configs.heroku.settings (module), 42
- radioco.configs.heroku.wsgi (module), 42
- RadiocomConfiguration (class in radioco.apps.global\_settings.models), 20
- RadiocomConfiguration.DoesNotExist, 20
- RadiocomConfiguration.MultipleObjectsReturned, 20
- RadioIntegrationTests (class in radioco.apps.radioco.tests.tests), 30
- readonly\_fields (radioco.apps.global\_settings.admin.PodcastConfiguration attribute), 19
- readonly\_fields (radioco.apps.schedules.admin.ScheduleAdmin attribute), 36
- rearrange\_episodes() (radioco.apps.programmes.models.Programme method), 28
- rearrange\_episodes() (radioco.apps.schedules.models.Calendar method), 36
- recorder\_token (radioco.apps.global\_settings.models.PodcastConfiguration attribute), 20
- recurrence\_after() (in module radioco.apps.radioco.tz\_utils), 31
- recurrence\_before() (in module radioco.apps.radioco.tz\_utils), 31
- recurrences (radioco.apps.schedules.models.Schedule attribute), 38
- RecurrenceTests (class in radioco.apps.schedules.tests.test\_reurrences), 33
- response\_change() (radioco.apps.global\_settings.admin.SingletonModelAdmin method), 19
- response\_change() (radioco.apps.users.admin.SingletonProfileAdmin method), 40
- Role (class in radioco.apps.programmes.models), 28
- Role.DoesNotExist, 28
- Role.MultipleObjectsReturned, 28
- role\_set (radioco.apps.programmes.models.Programme attribute), 28
- rss\_attributes() (radioco.apps.programmes.feeds.iTunesFeed method), 26
- RssProgrammeFeed (class in radioco.apps.programmes.feeds), 26
- runtime (radioco.apps.programmes.models.Episode attribute), 27
- runtime (radioco.apps.programmes.models.Programme attribute), 28
- runtime (radioco.apps.schedules.models.Schedule attribute), 38

## S

- save() (radioco.apps.global\_settings.models.SingletonModel method), 20
- save() (radioco.apps.programmes.models.Programme method), 28
- save() (radioco.apps.schedules.models.Calendar method), 37
- save() (radioco.apps.schedules.models.Schedule method), 38
- save() (radioco.apps.users.admin.NonStaffUserProfileForm method), 40
- save() (radioco.apps.users.models.UserProfile method), 41



- save\_formset() (radioco.apps.programmes.admin.NonStaffEpisodeAdmin class (radioco.apps.api.views.EpisodeViewSet method), 23
- save\_formset() (radioco.apps.programmes.admin.NonStaffProgrammeAdmin class (radioco.apps.api.views.ProgrammeViewSet method), 24
- save\_model() (radioco.apps.programmes.admin.NonStaffEpisodeAdmin class (radioco.apps.api.views.ScheduleViewSet method), 23
- save\_model() (radioco.apps.users.admin.SingletonProfileAdmin class (radioco.apps.api.views.TransmissionOperationViewSet method), 40
- save\_slug() (in module radioco.apps.users.models), 41
- Schedule (class in radioco.apps.schedules.models), 37
- schedule (radioco.apps.schedules.models.ExcludedDates attribute), 37
- Schedule.DoesNotExist, 37
- Schedule.MultipleObjectsReturned, 37
- schedule\_id (radioco.apps.schedules.views.DeleteScheduleView attribute), 39
- schedule\_list() (in module radioco.apps.schedules.views), 39
- schedule\_set (radioco.apps.programmes.models.Programme attribute), 28
- schedule\_set (radioco.apps.schedules.models.Calendar attribute), 37
- schedule\_set (radioco.apps.schedules.models.Schedule attribute), 38
- ScheduleAdmin (class in radioco.apps.schedules.admin), 36
- ScheduleBetweenTests (class in radioco.apps.schedules.tests.test\_schedules), 34
- ScheduleFilter (class in radioco.apps.api.views), 16
- ScheduleFilter.Meta (class in radioco.apps.api.views), 16
- ScheduleModelTests (class in radioco.apps.schedules.tests.test\_schedules), 34
- ScheduleModelTests (class in radioco.apps.schedules.tests.test\_timezone), 35
- ScheduleSerializer (class in radioco.apps.api.serializers), 14
- ScheduleSerializer.Meta (class in radioco.apps.api.serializers), 14
- ScheduleUtilsTests (class in radioco.apps.schedules.tests.test\_schedules), 34
- ScheduleValidationTests (class in radioco.apps.schedules.tests.test\_schedules), 35
- ScheduleViewSet (class in radioco.apps.api.views), 16
- search\_fields (radioco.apps.programmes.admin.NonStaffEpisodeAdmin class attribute), 23
- search\_fields (radioco.apps.programmes.admin.NonStaffProgrammeAdmin class attribute), 24
- search\_fields (radioco.apps.schedules.admin.CalendarAdmin attribute), 36
- serializer\_class (radioco.apps.api.views.EpisodeViewSet attribute), 15
- serializer\_class (radioco.apps.api.views.ProgrammeViewSet attribute), 16
- serializer\_class (radioco.apps.api.views.ScheduleViewSet attribute), 16
- serializer\_class (radioco.apps.api.views.TransmissionOperationViewSet attribute), 17
- serializer\_class (radioco.apps.api.views.TransmissionViewSet attribute), 17
- set\_active() (radioco.apps.schedules.admin.CalendarAdmin method), 36
- settings() (in module radioco.apps.radioco.context\_processors), 31
- setUp() (radioco.apps.api.tests.test\_fullcalendar.TestFullCalendarApi method), 12
- setUp() (radioco.apps.api.tests.test\_programmes.TestProgrammesAPI method), 12
- setUp() (radioco.apps.programmes.tests.EpisodeManagerTests method), 29
- setUp() (radioco.apps.programmes.tests.EpisodeModelTests method), 29
- setUp() (radioco.apps.programmes.tests.ProgrammeModelAdminTests method), 29
- setUp() (radioco.apps.programmes.tests.ProgrammeModelTests method), 29
- setUp() (radioco.apps.schedules.tests.test\_recurrences.EmptyRecurrenceTests method), 33
- setUp() (radioco.apps.schedules.tests.test\_recurrences.RecurrenceTests method), 33
- setUp() (radioco.apps.schedules.tests.test\_schedules.CalendarAdminTests method), 33
- setUp() (radioco.apps.schedules.tests.test\_schedules.CalendarManagerTests method), 33
- setUp() (radioco.apps.schedules.tests.test\_schedules.CalendarValidationTests method), 34
- setUp() (radioco.apps.schedules.tests.test\_schedules.ScheduleBetweenTests method), 34
- setUp() (radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests method), 34
- setUp() (radioco.apps.schedules.tests.test\_schedules.ScheduleUtilsTests method), 34
- setUp() (radioco.apps.schedules.tests.test\_schedules.TransmissionModelTests method), 35
- setUp() (radioco.apps.schedules.tests.test\_timezone.ScheduleModelTests method), 35
- setUpTestData() (radioco.apps.api.tests.test\_recorder.TestProgrammesAPI method), 13
- setUpTestData() (radioco.apps.radioco.test\_utils.TestDataMixin method), 31
- show\_toolbar() (in module radioco.configs.development.settings), 42
- SingletonModel (class in ra-

dioco.apps.global\_settings.models), 20  
 SingletonModel.Meta (class in ra-  
 dioco.apps.global\_settings.models), 20  
 SingletonModelAdmin (class in ra-  
 dioco.apps.global\_settings.admin), 19  
 SingletonModelManager (class in ra-  
 dioco.apps.global\_settings.models), 20  
 SingletonProfileAdmin (class in ra-  
 dioco.apps.users.admin), 40  
 SiteConfiguration (class in ra-  
 dioco.apps.global\_settings.models), 21  
 SiteConfiguration.DoesNotExist, 21  
 SiteConfiguration.MultipleObjectsReturned, 21  
 slug (radioco.apps.schedules.models.Transmission  
 attribute), 38  
 source (radioco.apps.schedules.models.Schedule at-  
 tribute), 38  
 spain\_tz() (in module radioco.apps.radioco.test\_utils), 31  
 start\_dt (radioco.apps.programmes.models.Programme  
 attribute), 28  
 station\_json() (in module ra-  
 dioco.apps.api.radiocom\_views), 14  
 suffix (radioco.apps.api.views.EpisodeViewSet at-  
 tribute), 15  
 suffix (radioco.apps.api.views.ProgrammeViewSet  
 attribute), 16  
 suffix (radioco.apps.api.views.ScheduleViewSet at-  
 tribute), 17  
 suffix (radioco.apps.api.views.TransmissionOperationViewSet  
 attribute), 17  
 suffix (radioco.apps.api.views.TransmissionViewSet at-  
 tribute), 17  
**T**  
 template\_name (radioco.apps.schedules.views.DeleteScheduleView  
 attribute), 39  
 test\_absolute\_url() (radioco.apps.programmes.tests.ProgrammeModelTests  
 method), 29  
 test\_absoulte\_url() (radioco.apps.programmes.tests.EpisodeModelTests  
 method), 29  
 test\_after() (radioco.apps.schedules.tests.test\_recurrences.EmptyRecurrenceTests  
 method), 33  
 test\_after() (radioco.apps.schedules.tests.test\_recurrences.RecurrenceTests  
 method), 33  
 test\_after\_inclusive() (ra-  
 dioco.apps.schedules.tests.test\_recurrences.EmptyRecurrenceTests  
 method), 33  
 test\_after\_inclusive() (ra-  
 dioco.apps.schedules.tests.test\_recurrences.RecurrenceTests  
 method), 33  
 test\_api() (radioco.apps.api.tests.test\_programmes.TestProgrammesAPI  
 method), 12  
 test\_at() (radioco.apps.schedules.tests.test\_schedules.TransmissionModelTests  
 method), 35  
 test\_available\_dates\_after() (ra-  
 dioco.apps.schedules.tests.test\_schedules.ScheduleUtilsTests  
 method), 34  
 test\_available\_dates\_none() (ra-  
 dioco.apps.schedules.tests.test\_schedules.ScheduleUtilsTests  
 method), 34  
 test\_before() (radioco.apps.schedules.tests.test\_recurrences.EmptyRecurrenceTests  
 method), 33  
 test\_before\_dt() (radioco.apps.schedules.tests.test\_recurrences.RecurrenceTests  
 method), 33  
 test\_before\_inclusive() (ra-  
 dioco.apps.schedules.tests.test\_recurrences.EmptyRecurrenceTests  
 method), 33  
 test\_before\_inclusive() (ra-  
 dioco.apps.schedules.tests.test\_recurrences.RecurrenceTests  
 method), 33  
 test\_between() (radioco.apps.schedules.tests.test\_schedules.TransmissionModelTests  
 method), 35  
 test\_between\_by\_queryset() (ra-  
 dioco.apps.schedules.tests.test\_schedules.TransmissionModelTests  
 method), 35  
 test\_between\_includes\_started\_episode() (ra-  
 dioco.apps.schedules.tests.test\_schedules.ScheduleBetweenTests  
 method), 34  
 test\_CEST\_transition() (ra-  
 dioco.apps.schedules.tests.test\_timezone.ScheduleModelTests  
 method), 35  
 test\_CET\_transition() (ra-  
 dioco.apps.schedules.tests.test\_timezone.ScheduleModelTests  
 method), 35  
 test\_CET\_transitions() (in module ra-  
 dioco.apps.schedules.tests.test\_timezone),  
 35  
 test\_cleaned\_internal\_recurrence\_dates() (ra-  
 dioco.apps.schedules.tests.test\_timezone.ScheduleModelTests  
 method), 35  
 test\_clone\_calendar() (ra-  
 dioco.apps.schedules.tests.test\_schedules.CalendarAdminTests  
 method), 33  
 test\_create\_episode() (ra-  
 dioco.apps.programmes.tests.EpisodeManagerTests  
 method), 29  
 test\_create\_episode() (radioco.apps.schedules.tests.test\_schedules.CalendarManagerTests  
 method), 33  
 test\_current\_no\_date() (ra-  
 dioco.apps.schedules.tests.test\_schedules.CalendarManagerTests  
 method), 33  
 test\_date\_after() (radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests  
 method), 34  
 test\_date\_after\_exclude() (ra-  
 dioco.apps.schedules.tests.test\_schedules.ScheduleModelTests  
 method), 34  
 test\_data\_before() (radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests  
 method), 34

test_dates_between()	(ra- dioco.apps.schedules.tests.test_schedules.ScheduleModelTests method), 41	dioco.apps.users.tests.UserProfileMethodTests test_impossible_recurrence_after() (ra-
test_dates_between_complex_ruleset()	(ra- dioco.apps.schedules.tests.test_schedules.ScheduleModelTests method), 34	dioco.apps.schedules.tests.test_recurrences.RecurrenceTests test_impossible_recurrence_before() (ra-
test_dates_between_earlier_end_by_programme()	(ra- dioco.apps.schedules.tests.test_schedules.ScheduleModelTests method), 34	dioco.apps.schedules.tests.test_recurrences.RecurrenceTests test_incorrect_transmission_queries() (ra-
test_dates_between_includes_started_episode()	(ra- dioco.apps.schedules.tests.test_schedules.ScheduleBetweenTests method), 34	dioco.apps.api.tests.test_schedules.TestTransmissionAPI test_index() (radioco.apps.radioco.tests.tests.RadioIntegrationTests method), 30
test_dates_between_later_start_by_programme()	(ra- dioco.apps.schedules.tests.test_schedules.ScheduleModelTests method), 34	test_isolate_state() (radioco.apps.programmes.tests.EpisodeManagerTests method), 29
test_dictionary_key()	(ra- dioco.apps.radioco.tests.tests.UtilsTest method), 30	test_last() (radioco.apps.programmes.tests.EpisodeManagerTests method), 29
test_end()	(radioco.apps.schedules.tests.test_schedules.TransmissionModelTests method), 35	test_last_date_is_equals_before_limit() (ra- dioco.apps.schedules.tests.test_recurrences.RecurrenceTests method), 33
test_end_gt_calendar()	(ra- dioco.apps.schedules.tests.test_schedules.ScheduleModelTests method), 34	test_last_none() (radioco.apps.programmes.tests.EpisodeManagerTests method), 29
test_episode()	(radioco.apps.api.tests.test_api.TestSerializers method), 11	test_model_manager() (ra- dioco.apps.programmes.tests.EpisodeModelTests method), 29
test_episode_url()	(ra- dioco.apps.schedules.tests.test_schedules.TransmissionModelTests method), 35	test_move_schedule() (ra- dioco.apps.api.tests.test_fullcalendar.TestFullCalendarApi method), 12
test_episode_programme()	(ra- dioco.apps.api.tests.test_api.TestSerializers method), 11	test_move_schedule_in_tz() (ra- dioco.apps.api.tests.test_fullcalendar.TestFullCalendarApi method), 12
test_episodes_delete()	(ra- dioco.apps.api.tests.test_programmes.TestNotAllowedMethodsProgrammesAPI method), 12	test_move_schedule_with_schedules() (ra- dioco.apps.api.tests.test_fullcalendar.TestFullCalendarApi method), 12
test_episodes_get_all()	(ra- dioco.apps.api.tests.test_programmes.TestProgrammesAPI method), 12	test_move_schedule_with_schedules_in_tz() (ra- dioco.apps.api.tests.test_fullcalendar.TestFullCalendarApi method), 12
test_episodes_get_by_programme()	(ra- dioco.apps.api.tests.test_programmes.TestProgrammesAPI method), 12	test_name() (radioco.apps.schedules.tests.test_schedules.TransmissionModelTests method), 35
test_episodes_post()	(ra- dioco.apps.api.tests.test_programmes.TestNotAllowedMethodsProgrammesAPI method), 12	test_no_values() (radioco.apps.schedules.tests.test_recurrences.EmptyRecurrenceTests method), 33
test_episodes_put()	(ra- dioco.apps.api.tests.test_programmes.TestNotAllowedMethodsProgrammesAPI method), 12	test_not_allowed_programmes_api() (ra- dioco.apps.programmes.tests.EpisodeManagerTests method), 29
test_example_data()	(ra- dioco.apps.radioco.tests.tests.UtilsTest method), 31	test_not_allowed_programmes_api() (ra- dioco.apps.schedules.tests.test_schedules.CalendarValidationTests method), 34
test_fields()	(radioco.apps.schedules.tests.test_schedules.ScheduleValidationTests method), 35	test_people() (radioco.apps.programmes.tests.EpisodeManagerTests method), 29
test_fieldset()	(radioco.apps.programmes.tests.ProgrammeModelAdminTests method), 29	test_programme_photo_url() (ra-
test_get_announcers_and_profile()	(ra-	

dioco.apps.api.tests.test_api.TestSerializers method), 11	dioco.apps.programmes.tests.ProgrammeModelTests method), 29
test_programme_url() (radioco.apps.schedules.tests.test_schedules.TransmissionModelTests method), 35	test_runtime_not_set() (radioco.apps.schedules.tests.test_schedules.ScheduleModelTests method), 34
test_programmes_after() (radioco.apps.api.tests.test_programmes.TestProgrammesAPI method), 12	test_save() (radioco.apps.users.tests.UserProfileMethodTests method), 41
test_programmes_before() (radioco.apps.api.tests.test_programmes.TestProgrammesAPI method), 12	test_save_programme() (radioco.apps.programmes.tests.ProgrammeModelTests method), 29
test_programmes_between() (radioco.apps.api.tests.test_programmes.TestProgrammesAPI method), 12	test_save_rearrange_episodes() (radioco.apps.schedules.tests.test_schedules.ScheduleModelTests method), 34
test_programmes_delete() (radioco.apps.api.tests.test_programmes.TestNotAllowedMethodsProgrammesAPI method), 12	test_schedule() (radioco.apps.api.tests.test_api.TestSerializers method), 12
test_programmes_get_all() (radioco.apps.api.tests.test_programmes.TestProgrammesAPI method), 12	test_schedules_get_all() (radioco.apps.schedules.tests.test_schedules.TestRestrictedMethodsSchedulesAPI method), 13
test_programmes_post() (radioco.apps.api.tests.test_programmes.TestNotAllowedMethodsProgrammesAPI method), 12	test_schedules_get_by_name() (radioco.apps.schedules.tests.test_schedules.TestSchedulesAPI method), 13
test_programmes_put() (radioco.apps.api.tests.test_programmes.TestNotAllowedMethodsProgrammesAPI method), 12	test_schedules_get_by_name_api() (radioco.apps.schedules.tests.test_schedules.TestSchedulesAPI method), 13
test_rearrange_episodes() (radioco.apps.schedules.tests.test_schedules.ScheduleModelTests method), 35	test_schedules_get_by_nonexisting_programme() (radioco.apps.schedules.tests.test_schedules.TestSchedulesAPI method), 13
test_rearrange_episodes_new_schedule() (radioco.apps.schedules.tests.test_schedules.ScheduleModelTests method), 35	test_schedules_get_by_nonexisting_type() (radioco.apps.schedules.tests.test_schedules.TestSchedulesAPI method), 13
test_rearrange_only_non_emited_episodes() (radioco.apps.schedules.tests.test_schedules.ScheduleModelTests method), 35	test_schedules_get_by_programme() (radioco.apps.schedules.tests.test_schedules.TestSchedulesAPI method), 13
test_recording_schedules() (radioco.apps.api.tests.test_recorder.TestProgrammesAPI method), 13	test_schedules_get_by_type() (radioco.apps.schedules.tests.test_schedules.TestSchedulesAPI method), 13
test_recurrence_rules() (radioco.apps.schedules.tests.test_schedules.ScheduleModelTests method), 34	test_schedules_post() (radioco.apps.schedules.tests.test_schedules.TestRestrictedMethodsSchedulesAPI method), 12
test_runtime() (radioco.apps.programmes.tests.EpisodeModelTests method), 29	test_schedules_post() (radioco.apps.schedules.tests.test_schedules.TestRestrictedMethodsSchedulesAPI method), 13
test_runtime() (radioco.apps.programmes.tests.ProgrammeModelTests method), 29	test_schedules_test_in_tz() (radioco.apps.schedules.tests.test_schedules.TestSchedulesAPI method), 12
test_runtime() (radioco.apps.schedules.tests.test_schedules.ScheduleModelTests method), 34	test_schedules_put() (radioco.apps.schedules.tests.test_schedules.TestRestrictedMethodsSchedulesAPI method), 12
test_runtime_is_zero() (radioco.apps.programmes.tests.ProgrammeModelTests method), 29	test_season() (radioco.apps.programmes.tests.EpisodeManagerTests method), 29
test_runtime_not_get() (radioco.apps.programmes.tests.ProgrammeModelTests method), 29	test_slug() (radioco.apps.programmes.tests.ProgrammeModelTests method), 29
test_runtime_not_set() (radioco.apps.programmes.tests.ProgrammeModelTests method), 29	

method), 29

test\_slug() (radioco.apps.schedules.tests.test\_schedules.TransmissionModelTests method), 35

test\_start() (radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests method), 34

test\_start() (radioco.apps.schedules.tests.test\_schedules.TransmissionModelTests method), 35

test\_start\_date\_is\_before\_start\_limit() (radioco.apps.schedules.tests.test\_recurrences.RecurrenceTests method), 33

test\_start\_lt\_calendar() (radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests method), 34

test\_str() (radioco.apps.programmes.tests.EpisodeModelTests method), 29

test\_str() (radioco.apps.programmes.tests.ProgrammeModelTests method), 29

test\_str() (radioco.apps.schedules.tests.test\_schedules.CalendarModelTests method), 34

test\_submit\_recorder() (radioco.apps.api.tests.test\_recorder.TestProgrammesAPI method), 13

test\_transform\_dt\_to\_default\_tz() (radioco.apps.schedules.tests.test\_timezone.ScheduleModelTests method), 35

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

test\_transmission\_before() (radioco.apps.api.tests.test\_schedules.TestTransmissionAPI method), 13

test\_transmission\_list\_filter\_non\_active\_calendar() (radioco.apps.api.tests.test\_schedules.TestTransmissionAPI method), 13

test\_transmission\_now() (radioco.apps.api.tests.test\_schedules.TestTransmissionAPI method), 13

test\_transmission\_same\_day() (radioco.apps.api.tests.test\_schedules.TestTransmissionAPI method), 13

test\_transmissions\_between() (radioco.apps.api.tests.test\_schedules.TestTransmissionAPI method), 13

test\_transmissions\_between\_requesting\_tz() (radioco.apps.api.tests.test\_schedules.TestTransmissionAPI method), 13

test\_transmissions\_delete() (radioco.apps.api.tests.test\_schedules.TestRestrictedMethodsScheduleAPI method), 13

test\_transmissions\_filter\_calendar\_nonexistend() (radioco.apps.api.tests.test\_schedules.TestTransmissionAPI method), 13

test\_transmissions\_post() (radioco.apps.api.tests.test\_schedules.TestRestrictedMethodsScheduleAPI method), 13

method), 13

test\_transmission\_model\_fixture() (radioco.apps.api.tests.test\_schedules.TestRestrictedMethodsScheduleAPI method), 13

test\_unfinished() (radioco.apps.programmes.tests.EpisodeManagerTests method), 29

test\_unfinished\_none() (radioco.apps.programmes.tests.EpisodeManagerTests method), 29

test\_unicode() (radioco.apps.schedules.tests.test\_schedules.ScheduleModelTests method), 34

test\_validate\_date\_doesnt\_exist\_on\_february() (radioco.apps.schedules.tests.test\_recurrences.RecurrenceTests method), 33

TestDataMixin (class in radioco.apps.radioco.test\_utils), 31

TestFullCalendarApi (class in radioco.apps.api.tests.test\_fullcalendar), 12

TestNotAllowedMethodsProgrammesAPI (class in radioco.apps.api.tests.test\_programmes), 12

TestProgrammesAPI (class in radioco.apps.api.tests.test\_programmes), 12

TestProgrammesAPI (class in radioco.apps.api.tests.test\_recorder), 13

TestRestrictedMethodsScheduleAPI (class in radioco.apps.api.tests.test\_schedules), 13

TestSchedulesAPI (class in radioco.apps.api.tests.test\_schedules), 13

TestSerializers (class in radioco.apps.api.tests.test\_api), 11

TestTransmissionAPI (class in radioco.apps.api.tests.test\_schedules), 13

OwnEpisodeIssueDateListFilter (attribute), 25

title (radioco.apps.programmes.admin.OwnEpisodeProgrammeListFilter attribute), 25

title() (radioco.apps.programmes.feeds.ProgrammeFeed method), 26

TransmissionAPIPresentation() (radioco.apps.api.serializers.DateTimeFieldTz method), 14

TransmissionForm\_datetime\_tz() (in module radioco.apps.radioco.tz\_utils), 31

transform\_dt\_to\_default\_tz() (in module radioco.apps.radioco.tz\_utils), 31

Transmission (class in radioco.apps.schedules.models), 38

MethodsScheduleAPI (class in radioco.apps.schedules.views.DeleteScheduleView attribute), 39

TransmissionForm (class in radioco.apps.api.views), 17

TransmissionModelTests (class in radioco.apps.schedules.tests.test\_schedules), 35

MethodsScheduleAPI (class in radioco.apps.schedules.views.DeleteScheduleView attribute), 39

dioco.apps.api.views), 17  
TransmissionSerializer (class in radioco.apps.api.serializers), 15  
TransmissionSerializerLight (class in radioco.apps.api.serializers), 15  
TransmissionViewSet (class in radioco.apps.api.views), 17

## U

unfinished() (radioco.apps.programmes.models.EpisodeManager static method), 27  
update() (radioco.apps.api.views.TransmissionOperationViewSet method), 17  
update\_schedule\_if\_dt\_has\_changed() (in module radioco.apps.programmes.models), 28  
update\_schedule\_performance() (in module radioco.apps.programmes.models), 28  
UpdateOnlyModelViewSet (class in radioco.apps.api.viewsets), 17  
user (radioco.apps.users.models.UserProfile attribute), 41  
user\_logout() (in module radioco.apps.radioco.views), 32  
UserProfile (class in radioco.apps.users.models), 41  
UserProfile.DoesNotExist, 41  
UserProfile.MultipleObjectsReturned, 41  
userprofile\_detail() (in module radioco.apps.users.views), 41  
UserProfileAdmin (class in radioco.apps.users.admin), 40  
UserProfileInline (class in radioco.apps.users.admin), 40  
UserProfileMethodTests (class in radioco.apps.users.tests), 41  
UsersView (class in radioco.apps.users.views), 41  
UtilsTest (class in radioco.apps.radioco.tests.tests), 30

## V

validate() (radioco.apps.api.serializers.ScheduleSerializer method), 15  
validate() (radioco.apps.api.serializers.TransmissionSerializerLight method), 15

## X

x (radioco.apps.api.views.TransmissionForm attribute), 17