
radioco Documentation

Release 3.1

Iago Veloso Abalo

Jun 10, 2017

Contents

1	Overview	1
2	Release Notes	3
3	Table of contents	5
3.1	Installation Guide	5
3.2	Customize	7
3.3	API	9
3.4	Development & community	10
3.5	django-radio	12
3.6	Release notes & upgrade information	49
3.7	Indices and tables	54
	Python Module Index	55

CHAPTER 1

Overview

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

Here are a few of the key features:

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

Note: This software is not for live streaming

CHAPTER 2

Release Notes

This document refers to version 3.1

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.

Installing web application

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

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

Installing locally

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

Ubuntu

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

Open a terminal and introduce the following commands:

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

Next, download the project and cd into it:

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

Install the python invoke library:

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

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

```
inv quickstart
```

Warning: If you have faced the error “ValueError: unknown locale: UTF-8” on MacOS X, execute:
export LC_ALL=en_US.UTF-8 export LANG=en_US.UTF-8

Testing

Let’s verify your installation

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

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

Using RadioCo on production

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

Locally

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

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

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

To create a superuser you still can use management commands:

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

Installing recorder program

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

Installing on Ubuntu

We'll get started by setting up our environment.

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

Next, download the project and create the virtual environment:

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

Create and activate a virtual env:

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

Install the requirements:

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

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

Launch the program

```
python main.py
```

Communication Setup

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

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

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

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

Customize

Technical reference material.

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

Application Setup

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

Global settings

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

Global Configuration

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

Calendar Configuration

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

Podcast Configuration

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

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'
```

Recorder Program Setup

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

API

Api documentation.

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
```

Radiocom API

New in version 3.1

[Radiocom](#) is an Android application for Community Media Stations. RadioCo has the [Radiocom API](#) under the following urls:

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

Development & community

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

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

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

Branch policy

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

Contributing Translation

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

Contributing Documentation

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

Documentation should be:

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

Merging documentation is pretty fast and painless.

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

Documentation markup

Sections

We use Python documentation conventions for section marking:

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

Inline markup

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

References

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

django-radio

manage module

radioco package

Subpackages

radioco.apps package

Subpackages

radioco.apps.api package

Subpackages

radioco.apps.api.tests package

Submodules

radioco.apps.api.tests.test_api module

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

    test_api()

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

    test_episode()

    test_episode_programme()

    test_programme()

    test_programme_photo_url()

    test_schedule()

    test_transmission()
```

radioco.apps.api.tests.test_fullcalendar module

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

    setUp()

    test_move_schedule(*args, **kwargs)

    test_move_schedule_in_tz(*args, **kwargs)
```



```

test_move_schedule_with_schedules (*args, **kwargs)
test_move_schedule_with_schedules_in_tz (*args, **kwargs)
test_schedules_post (*args, **kwargs)
test_schedules_post_in_tz (*args, **kwargs)

```

radioco.apps.api.tests.test_programmes module

```

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

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

```

```

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

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

```

radioco.apps.api.tests.test_radiocom_api module

```

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

    setUp ()
    test_programme_in_list ()

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

    test_transmission_now (*args, **kwargs)

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

```

radioco.apps.api.tests.test_recorder module

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

    classmethod setUpTestData ()

    test_recording_schedules (*args, **kwargs)

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

radioco.apps.api.tests.test_schedules module

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

    test_schedules_delete ()

    test_schedules_post ()

    test_schedules_put ()

    test_transmissions_delete ()

    test_transmissions_post ()

    test_transmissions_put ()

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

    test_schedules_get_all ()

    test_schedules_get_by_calendar ()

    test_schedules_get_by_nonexisting_calendar ()

    test_schedules_get_by_nonexisting_programme ()

    test_schedules_get_by_nonexisting_type ()

    test_schedules_get_by_programme ()

    test_schedules_get_by_type ()

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

    test_incorrect_transmission_queries ()

    test_transmission_before (*args, **kwargs)

    test_transmission_list_filter_non_active_calendar (*args, **kwargs)

    test_transmission_now (*args, **kwargs)

    test_transmission_same_day ()

    test_transmissions_between (*args, **kwargs)

    test_transmissions_between_requesting_tz ()

    test_transmissions_filter_calendar_nonexistend ()
```

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

Module contents

Submodules

radioco.apps.api.recorder_views module

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

radioco.apps.api.serializers module

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

Bases: `rest_framework.fields.URLField`

A field similar to `URLField` but returning always the absolute url

For example:

```
class ExampleSerializer(self):
    episode_url = AbsoluteURLField()
    photo_url = AbsoluteURLField(source='photo.url', read_only=True)
    rss_url = AbsoluteURLField(source='slug', reverse_url='programmes:rss')
```

to_representation (*value*)

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

Bases: `rest_framework.fields.DateTimeField`

Field to display the datetime in the current timezone

to_representation (*date*)

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

Bases: `rest_framework.serializers.ModelSerializer`

class Meta

```
fields = ('title', 'programme', 'summary', 'issue_date', 'season', 'number_in_season')
```

model

alias of `Episode`

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

Bases: `rest_framework.serializers.ModelSerializer`

class Meta

fields = ('id', 'slug', 'name', 'synopsis', 'runtime', 'photo_url', 'rss_url', 'language', 'category')

model

alias of Programme

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

Bases: *rest_framework.serializers.ModelSerializer*

class Meta

fields = ('station_name', 'icon_url', 'big_icon_url', 'station_photos', 'history', 'latitude', 'longitude', 'news_rss', '')

model

alias of RadiocomConfiguration

RadiocomConfigurationSerializer.get_facebook_url (*obj*)

RadiocomConfigurationSerializer.get_station_photos (*obj*)

RadiocomConfigurationSerializer.get_twitter_url (*obj*)

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

Bases: *rest_framework.serializers.ModelSerializer*

class Meta

fields = ('id', 'name', 'description', 'logo_url', 'rss_url')

model

alias of Programme

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

Bases: *rest_framework.serializers.Serializer*

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

Bases: *rest_framework.serializers.ModelSerializer*

class Meta

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

model

alias of Schedule

ScheduleSerializer.get_title (*schedule*)

ScheduleSerializer.validate (*attrs*)

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

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

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

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

```
    validate (attrs)
```

radioco.apps.api.urls module

radioco.apps.api.views module

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

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

```
    class Meta
```

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

```
        model
```

```
            alias of Episode
```

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

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

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

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

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

```
    filter_class
```

```
        alias of EpisodeFilter
```

```
    queryset
```

```
    serializer_class
```

```
        alias of EpisodeSerializer
```

```
    suffix = None
```

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

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

```
    class Meta
```

```
        fields = ('name', 'category')
```

```
        model
```

```
            alias of Programme
```

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

```
    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>), ('before', <django.forms.fields.DateField object>)]

clean ()

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

media

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

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

filter_class
    alias of ProgrammeFilter

list (request, *args, **kwargs)

lookup_field = 'slug'

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

queryset

serializer_class
    alias of ProgrammeSerializer

suffix = None

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

serializer_class
    alias of RadiocomProgrammeSerializer

suffix = None

class radioco.apps.api.views.RadiocomStation (**kwargs)
Bases: rest_framework.viewsets.ReadOnlyModelViewSet

list (request, *args, **kwargs)

serializer_class
    alias of RadiocomConfigurationSerializer

suffix = None

class radioco.apps.api.views.RadiocomTransmissionViewSet (**kwargs)
Bases: radioco.apps.api.views.TransmissionViewSet

serializer_class
    alias of RadiocomTransmissionSerializer

suffix = None

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

```

class Meta

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

    model
        alias of Schedule

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

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

    filter_class
        alias of ScheduleFilter

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

    queryset

    serializer_class
        alias of ScheduleSerializer

    suffix = None

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

    Bases: django.forms.forms.Form

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

    clean_timezone ()

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

    media

    x = 'Zulu'

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

    Bases: radioco.apps.api.views.TimezoneForm

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

    clean ()

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

    media

class radioco.apps.api.views.TransmissionOperationViewSet (**kwargs)
    Bases: radioco.apps.api.viewsets.UpdateOnlyModelViewSet

    perform_update (serializer)

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

```

```
queryset  
serializer_class  
    alias of TransmissionSerializerLight  
suffix = None  
update (request, *args, **kwargs)
```

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

radioco.apps.api.viewsets module

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

Module contents

radioco.apps.global_settings package

Subpackages

radioco.apps.global_settings.migrations package

Submodules

radioco.apps.global_settings.migrations.0001_initial module

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


radioco.apps.global_settings.migrations.0002_remove_calendarconfiguration_display_next_weeks module

```
class radioco.apps.global_settings.migrations.0002_remove_calendarconfiguration_display_next_
```

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

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

```
    operations = [<RemoveField name=u'display_next_weeks', model_name=u'calendarconfiguration'>]
```

radioco.apps.global_settings.migrations.0003_siteconfiguration_footer module

```
class radioco.apps.global_settings.migrations.0003_siteconfiguration_footer.Migration (name, app_label)
```

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

```
    dependencies = [(u'global_settings', u'0002_remove_calendarconfiguration_display_next_weeks')]
```

```
    operations = [<AddField field=<django.db.models.fields.TextField>, preserve_default=True, name=u'footer', model_name=
```

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_name=
```

radioco.apps.global_settings.migrations.0005_auto_20150606_1415 module

```
class radioco.apps.global_settings.migrations.0005_auto_20150606_1415.Migration (name, app_label)
```

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

```
    dependencies = [(u'global_settings', u'0004_auto_20150606_1335')]
```

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

radioco.apps.global_settings.migrations.0006_auto_20160116_1509 module

```
class radioco.apps.global_settings.migrations.0006_auto_20160116_1509.Migration (name, app_label)
```

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

```
    dependencies = [(u'global_settings', u'0005_auto_20150606_1415')]
```

```
    operations = [<AlterField field=<django.db.models.fields.CharField>, name=u'google_analytics_id', model_name=u'si
```

radioco.apps.global_settings.migrations.0007__v3_0__calendarconfiguration_tweaks module

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

radioco.apps.global_settings.migrations.0008__v3_0__radiocomconfiguration module

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

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.objects
```

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

```
class radioco.apps.global_settings.models.SingletonModelManager
```

```
    Bases: django.db.models.manager.Manager
```

```
    get(*args, **kwargs)
```

```
class radioco.apps.global_settings.models.SiteConfiguration(id, site_name,
                                                           about_footer,
                                                           more_about_us,
                                                           google_analytics_id,
                                                           address, twitter_address, facebook_address)
```

```
    Bases: radioco.apps.global_settings.models.SingletonModel
```

```
exception DoesNotExist
```

```
    Bases: django.core.exceptions.ObjectDoesNotExist
```

```
exception SiteConfiguration.MultipleObjectsReturned
```

```
    Bases: django.core.exceptions.MultipleObjectsReturned
```

```
SiteConfiguration.objects = <radioco.apps.global_settings.models.SingletonModelManager object>
```

Module contents

[radioco.apps.programmes package](#)

[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>>, <AlterField field=<django.db.models.fields.CharField>, name=<django.db.models.fields.CharField>, to_type=<django.db.models.fields.CharField>>]
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=u'programme'>]
```

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>>]
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'episode'>]
```

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'programme'>]
```

radioco.apps.programmes.migrations.0007_change_default_image module

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

radioco.apps.programmes.migrations.0008_auto_20160116_1509 module

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

radioco.apps.programmes.migrations.0009_v3_0_small_tweaks module

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

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.adm
```

```

list_display = ('__unicode__', 'season', 'number_in_season', 'issue_date', 'programme')
list_filter = ['issue_date', <class 'radioco.apps.programmes.admin.OwnEpisodeProgrammeListFilter'>, <class 'rad
list_select_related = True
media
ordering = ['-season', '-number_in_season']
save_formset (request, form, formset, change)
save_model (request, obj, form, change)
search_fields = ['programme__name']
class radioco.apps.programmes.admin.NonStaffEpisodeAdminForm (*args, **kwargs)
Bases: django.forms.models.ModelForm
class Meta
    fields = '__all__'
    model
        alias of Episode
NonStaffEpisodeAdminForm.base_fields = OrderedDict([('title', <django.forms.fields.CharField object>), ('peo
NonStaffEpisodeAdminForm.clean_programme ()
NonStaffEpisodeAdminForm.declared_fields = OrderedDict()
NonStaffEpisodeAdminForm.media
class radioco.apps.programmes.admin.NonStaffParticipantInline (parent_model, ad-
min_site)
Bases: django.contrib.admin.options.StackedInline
extra = 0
form
    alias of NonStaffParticipantInlineForm
formset
    alias of NonStaffParticipantInlineFormset
get_formset (request, obj=None, **kwargs)
get_queryset (request)
media
model
    alias of Participant
class radioco.apps.programmes.admin.NonStaffParticipantInlineForm (*args,
**kwargs)
Bases: django.forms.models.ModelForm
base_fields = OrderedDict()
clean ()
declared_fields = OrderedDict()
media

```

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

```
Bases: django.forms.models.BaseInlineFormSet
```

```
class radioco.apps.programmes.admin.NonStaffProgrammeAdmin (model, admin_site)  
Bases: django.contrib.admin.options.ModelAdmin  
get_fieldsets (request, obj=None)  
get_prepopulated_fields (request, obj=None)  
get_queryset (request)  
inlines = [<class 'radioco.apps.programmes.admin.NonStaffRoleInline'>]  
list_display = ('name', '_runtime', 'current_season', 'category', 'start_date', 'end_date')  
list_filter = ['_runtime', 'category']  
media  
save_formset (request, form, formset, change)  
search_fields = ['name']
```

```
class radioco.apps.programmes.admin.NonStaffRoleInline (parent_model, admin_site)  
Bases: django.contrib.admin.options.StackedInline  
extra = 0  
form  
    alias of NonStaffRoleInlineForm  
formset  
    alias of NonStaffRoleInlineFormset  
get_formset (request, obj=None, **kwargs)  
get_queryset (request)  
list_select_related = True  
media  
model  
    alias of Role
```

```
class radioco.apps.programmes.admin.NonStaffRoleInlineForm (*args, **kwargs)  
Bases: django.forms.models.ModelForm  
base_fields = OrderedDict()  
clean ()  
    Check unique together: person, role, programme  
declared_fields = OrderedDict()  
media
```



```
class radioco.apps.programmes.admin.NonStaffRoleInlineFormset (data=None,
                                                             files=None,      in-
                                                             stance=None,
                                                             save_as_new=False,
                                                             prefix=None,
                                                             queryset=None,
                                                             **kwargs)
```

Bases: django.forms.models.BaseInlineFormSet

```
class radioco.apps.programmes.admin.OwnEpisodeIssueDateListFilter (request,
                                                                    params, model,
                                                                    model_admin)
```

Bases: django.contrib.admin.filters.SimpleListFilter

lookups (*request, model_admin*)

parameter_name = 'date'

queryset (*request, queryset*)

title = <django.utils.functional.__proxy__ object>

```
class radioco.apps.programmes.admin.OwnEpisodeProgrammeListFilter (request,
                                                                    params, model,
                                                                    model_admin)
```

Bases: django.contrib.admin.filters.SimpleListFilter

Check people in programmes besides episodes, better performance

lookups (*request, model_admin*)

parameter_name = 'programme'

queryset (*request, queryset*)

title = <django.utils.functional.__proxy__ object>

```
class radioco.apps.programmes.admin.PodcastInline (parent_model, admin_site)
```

Bases: django.contrib.admin.options.StackedInline

extra = 0

inline_classes = ('grp-collapse grp-open',)

media

model

alias of Podcast

radioco.apps.programmes.feeds module

```
class radioco.apps.programmes.feeds.ProgrammeFeed
```

Bases: django.contrib.syndication.views.Feed

description (*programme*)

feed_extra_kwargs (*programme*)

get_object (*request, slug*)

item_description (*podcast*)

item_enclosure_length (*podcast*)

item_enclosure_mime_type (*podcast*)

`item_enclosure_url` (*podcast*)

`item_extra_kwargs` (*podcast*)

`item_keywords` (*podcast*)

`item_pubdate` (*podcast*)

`item_title` (*podcast*)

`items` (*programme*)

`link` (*programme*)

`title` (*programme*)

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

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

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

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

    Programme.announcers
    Programme.end_dt
    Programme.episode_set
    Programme.get_absolute_url ()
    Programme.get_category_display (*moreargs, **morekwargs)
    Programme.get_language_display (*moreargs, **morekwargs)
    Programme.objects = <django.db.models.manager.Manager object>

```

Programme.**photo**

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

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

Update the issue_date of episodes from a given date

Programme.**role_set**

Programme.**runtime**

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

Programme.**schedule_set**

Programme.**start_dt**

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.migrations package](#)

Submodules

[radioco.apps.radioco.migrations.0001__v3_0__mysql_timezone module](#)

```
class radioco.apps.radioco.migrations.0001__v3_0__mysql_timezone.Migration (name,
                                                                    app_label)
    Bases: django.db.migrations.migration.Migration
    dependencies = [(u'schedules', u'0001_initial'), (u'programmes', u'0008_auto_20160116_1509'), (u'global_settings', u'0001_initial')]
    operations = [<RunPython <function migrate_datetime_to_utc>>]
```

```
radioco.apps.radioco.migrations.0001__v3_0__mysql_timezone.migrate_datetime_to_utc (apps,
                                                                    schema_editor)
    version 3 introduces timezone support, this migration will fix the Episode issue_date Since Postgres is the only
    database which is already storing the datetimes in utc we don't need to run on it
```

Module contents

[radioco.apps.radioco.tests package](#)

Submodules

radioco.apps.radioco.tests.tests module

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

test_index()

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

test_dictionary_key()

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

test_example_data()

Running example data function, If nothing crash we are happy

Module contents

Submodules

radioco.apps.radioco.context_processors module

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

radioco.apps.radioco.test_utils module

class radioco.apps.radioco.test_utils.**TestDataMixin**
 Bases: *object*

classmethod **setUpTestData**()

radioco.apps.radioco.test_utils.**create_test_data**()

radioco.apps.radioco.test_utils.**spain_tz**()

radioco.apps.radioco.tz_utils module

class radioco.apps.radioco.tz_utils.**GMT** (*seconds*)
 Bases: *dateutil.tz.tz.tzoffset*

GMT implementation, it has a fixed offset

localize (*dt, is_dst=False*)

Convert naive time to local time

normalize (*dt, is_dst=False*)

Correct the timezone information on the given datetime

radioco.apps.radioco.tz_utils.**fix_recurrence_date** (*start_dt, dt*)

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

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

`radioco.apps.radioco.tz_utils.fix_recurrence_dst(dt)`
Fix for django-recurrence 1.3 Function to fix a datetime tz aware with an incorrect offset
Returns: A datetime in the same timezone but with the offset fixed

`radioco.apps.radioco.tz_utils.get_active_timezone()`
Same method as `timezone.get_current_timezone` but returning `utc` if nothing was set

`radioco.apps.radioco.tz_utils.recurrence_after(recurrence, after_dt, start_dt)`
Fix for django-recurrence 1.3 Avoid outputting a impossible dt

`radioco.apps.radioco.tz_utils.recurrence_before(recurrence, before_dt, start_dt)`
Fix for django-recurrence 1.3 Avoid outputting a impossible dt

`radioco.apps.radioco.tz_utils.transform_datetime_tz(dt, tz=None)`
Transform a datetime in other timezone to the current one

`radioco.apps.radioco.tz_utils.transform_dt_to_default_tz(dt)`
Transform a datetime in other timezone to the current one

radioco.apps.radioco.utils module

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

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

model = `None`

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

dispatch (*args, **kwargs)

object = `None`

`radioco.apps.radioco.utils.check_delete_permission(user, model)`

`radioco.apps.radioco.utils.create_example_data()`

`radioco.apps.radioco.utils.field_has_changed(_object, field)`

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

A simple cache system, use as decorator

radioco.apps.radioco.views module

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

`radioco.apps.radioco.views.user_logout(request)`

Module contents

radioco.apps.schedules package

Subpackages

radioco.apps.schedules.migrations package

Submodules

radioco.apps.schedules.migrations.0001_initial module

class radioco.apps.schedules.migrations.0001_initial.**Migration** (*name*, *app_label*)

Bases: django.db.migrations.migration.Migration

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

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

radioco.apps.schedules.migrations.0002__v3_0__renaming_calendar_model module

class radioco.apps.schedules.migrations.0002__v3_0__renaming_calendar_model.**Migration** (*name*, *app_label*)

Bases: django.db.migrations.migration.Migration

dependencies = [(u'schedules', u'0001_initial'), (u'radioco', u'0001__v3_0__mysql_timezone')]

operations = [<RenameField new_name=u'calendar', model_name=u'schedule', old_name=u'schedule_board'>, <Ren

radioco.apps.schedules.migrations.0002__v3_0__renaming_calendar_model.**migrate_board** (*apps*, *schema_e*)

Renaming previous Calendars and creating a new one

radioco.apps.schedules.migrations.0003__v3_0__create_excludeddates_model module

class radioco.apps.schedules.migrations.0003__v3_0__create_excludeddates_model.**Migration** (*name*, *app_label*)

Bases: django.db.migrations.migration.Migration

dependencies = [(u'schedules', u'0002__v3_0__renaming_calendar_model')]

operations = [<CreateModel fields=[(u'id', <django.db.models.fields.AutoField>), (u'datetime', <django.db.models.fiel

radioco.apps.schedules.migrations.0004__v3_0__adding_rrules module

class radioco.apps.schedules.migrations.0004__v3_0__adding_rrules.**Migration** (*name*, *app_label*)

Bases: django.db.migrations.migration.Migration

dependencies = [(u'schedules', u'0003__v3_0__create_excludeddates_model')]

operations = [<AddField field=<django.db.models.fields.DateTimeField>, name=u'end_date', model_name=u'schedule

radioco.apps.schedules.migrations.0004__v3_0__adding_rrules.**migrate_daily_recurrences** (*apps*, *schema*)

Migration to convert weekly recurrences into new complex recurrences

radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_unique_calendar module

class radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_unique_calendar.**Mi**

Bases: django.db.migrations.migration.Migration

dependencies = [(u'schedules', u'0004__v3_0__adding_rrules'), (u'programmes', u'0009__v3_0__small_tweaks')]

operations = [<RunPython <function migrate_schedules>>, <RemoveField name=u'day', model_name=u'schedule'>, <

radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_unique_calendar.**calcul**

Calculation of the last end date to improve performance

radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_unique_calendar.**calcul**

Calculation of the first start date to improve performance

radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_unique_calendar.**migrat**

Final Migration to v3.0

Before this migration Calendar (ScheduleBoard) had a date limit and all the schedules were repeated weekly

We want to move constraint dates from Calendar to programmes and clone the active schedules into the active calendar

radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_unique_calendar.**recur**

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

radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_unique_calendar.**recur**

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

Module contents

radioco.apps.schedules.tests package

Submodules

radioco.apps.schedules.tests.test_recurrences module

class radioco.apps.schedules.tests.test_recurrences.**EmptyRecurrenceTests** (*methodName='runTest'*)

Bases: *radioco.apps.radioco.test_utils.TestDataMixin*, *django.test.testcases.TestCase*

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

setUp ()

test_after ()

test_after_inclusive ()

test_before ()

test_before_inclusive ()

`test_no_values()`

class `radioco.apps.schedules.tests.test_reurrences.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, **kwargs)

    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>), ('schedule', <django.for

declared_fields = OrderedDict([('transmission_dt', <django.forms.fields.DateTimeField object>), ('schedule', <djan

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.DeletePermissionMixin, django.views.generic.edit.ModelFormView

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__v3_0__ckeditor_upload_field module

```
class radioco.apps.users.migrations.0004__v3_0__ckeditor_upload_field.Migration (name,  
                                                                    app_label)  
    Bases: django.db.migrations.migration.Migration  
    dependencies = [(u'users', u'0003_auto_20160104_2029'), (u'radioco', u'0001__v3_0__mysql_timezone')]  
    operations = [<AlterField field=<ckeditor_uploader.fields.RichTextUploadingField>, name=u'bio', model_name=u'usr
```

Module contents

Submodules

radioco.apps.users.admin module

```
class radioco.apps.users.admin.NonStaffUserProfileForm (*args, **kwargs)  
    Bases: django.forms.models.ModelForm  
    base_fields = OrderedDict([(u'username', <django.forms.fields.CharField object>), (u'first_name', <django.forms.fields  
    clean_username ()  
    declared_fields = OrderedDict([(u'username', <django.forms.fields.CharField object>), (u'first_name', <django.forms  
    media  
    save (force_insert=False, force_update=False, commit=True)  
class radioco.apps.users.admin.SingletonProfileAdmin (model, admin_site)  
    Bases: django.contrib.admin.options.ModelAdmin  
    change_view (request, object_id, extra_context=None)  
    fields = [u'username', u'first_name', u'last_name', u'email', u'bio', u'avatar', u'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 = (u'is_staff', u'is_superuser', u'is_active', u'groups', u'userprofile__display_personal_page')  
    media  
class radioco.apps.users.admin.UserProfileInline (parent_model, admin_site)  
    Bases: django.contrib.admin.options.StackedInline  
    can_delete = False
```

```

exclude = ('slug',)
extra = 1
inline_classes = ('grp-collapse grp-open',)
media
model
    alias of UserProfile

```

radioco.apps.users.models module

```

class radioco.apps.users.models.UserProfile (id, user, bio, avatar, display_personal_page,
                                             slug)
    Bases: django.db.models.base.Model

exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

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

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

UserProfile.get_absolute_url()

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

UserProfile.save(*args, **kwargs)

UserProfile.user

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

```

radioco.apps.users.tests module

```

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

test_get_announcers_and_profile()

test_save()

```

radioco.apps.users.urls module

radioco.apps.users.views module

```

class radioco.apps.users.views.UsersView (**kwargs)
    Bases: django.views.generic.list.ListView

get_queryset()

model
    alias of UserProfile

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

```

Module contents

Module contents

radioco.configs package

Subpackages

radioco.configs.base package

Submodules

radioco.configs.base.manage module

radioco.configs.base.settings module

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

radioco.configs.base.test_runner module

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

radioco.configs.base.urls module

Module contents

radioco.configs.development package

Submodules

radioco.configs.development.manage module

radioco.configs.development.settings module

`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

`radioco.configs.staging` package

Submodules

`radioco.configs.staging.manage` module

`radioco.configs.staging.settings` module

`radioco.configs.staging.wsgi` module

Module contents

Module contents

Module contents

tasks package

Submodules

`tasks.docker` module

`tasks.docs` module

`tasks.heroku` module

`tasks.locale` module

`tasks.radioco` module

`tasks.utils` module

Module contents

Release notes & upgrade information

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

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

1.1 release notes

What's new in 1.1

- Numerous updates to the documentation
- Updates to facilitate the starting-up (radioco_recorder_user and a empty shedule_board are self-created)
- Added custom languages to programmes (by default, all django languages)

How this affects you

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

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

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

1.1.1 release notes

What's new in 1.1.1

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

How this affects you

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

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

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

In your settings.py import default settings:

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

1.2 release notes

What's new in 1.2

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

How this affects you

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

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

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

2.1 release notes

What's new in 2.1

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

How this affects you

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

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

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

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

3.0 release notes

What's new in 3.0

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

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

- dashboard app has been merged into schedules

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

And the visible features are:

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

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

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

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

How this affects you

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

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

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

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

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

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

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

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

You have to install `bower` before running the following commands in the project root folder:

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

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

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

3.0.1 release notes

What’s new in 3.0.1

- Fixing invoke commands.
- Simplified docker setup.

Take a look to our installation section.

How this affects you

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

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

This update doesn’t require any further steps.

3.1 release notes

What’s new in 3.1

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

How this affects you

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

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

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

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

Or if you are using our docker setup:

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

Indices and tables

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

Python Module Index

m

manage, 12

r

radioco, 49

radioco.apps, 48

radioco.apps.api, 20

radioco.apps.api.recorder_views, 15

radioco.apps.api.serializers, 15

radioco.apps.api.tests, 15

radioco.apps.api.tests.test_api, 12

radioco.apps.api.tests.test_fullcalendar,
12

radioco.apps.api.tests.test_programmes,
13

radioco.apps.api.tests.test_radiocom_api,
13

radioco.apps.api.tests.test_recorder,
14

radioco.apps.api.tests.test_schedules,
14

radioco.apps.api.urls, 17

radioco.apps.api.views, 17

radioco.apps.api.viewsets, 20

radioco.apps.global_settings, 24

radioco.apps.global_settings.admin, 22

radioco.apps.global_settings.migrations,
22

radioco.apps.global_settings.migrations.0001_initial,
20

radioco.apps.global_settings.migrations.0002_remove_calendarconfiguration_display_next_week,
21

radioco.apps.global_settings.migrations.0003_set_apps_programmes_feeds,
21

radioco.apps.global_settings.migrations.0004_auto_20150606_0436,
21

radioco.apps.global_settings.migrations.0005_auto_20150606_0436,
21

radioco.apps.global_settings.migrations.0006_auto_20160116_1509,
21

radioco.apps.global_settings.migrations.0007__v3_0,
22

radioco.apps.global_settings.migrations.0008__v3_0,
22

radioco.apps.global_settings.models, 23

radioco.apps.global_settings.templatetags,
22

radioco.apps.global_settings.templatetags.global_se,
22

radioco.apps.programmes, 34

radioco.apps.programmes.admin, 26

radioco.apps.programmes.feeds, 29

radioco.apps.programmes.migrations, 26

radioco.apps.programmes.migrations.0001_initial,
24

radioco.apps.programmes.migrations.0002_change_lang,
25

radioco.apps.programmes.migrations.0003_change_text,
25

radioco.apps.programmes.migrations.0004_change_phot,
25

radioco.apps.programmes.migrations.0005_auto_20150,
25

radioco.apps.programmes.migrations.0006_auto_20160,
25

radioco.apps.programmes.migrations.0007_change_defa,
26

radioco.apps.programmes.migrations.0008_auto_20160,
26

radioco.apps.programmes.migrations.0009__v3_0__sma,
26

radioco.apps.programmes.models, 30

radioco.apps.programmes.urls, 33

radioco.apps.programmes.utils, 33

radioco.apps.programmes.views, 33

radioco.apps.radioco.context_processors,
35

radioco.apps.radioco.management, 34

- [radioco.apps.radioco.management.commandsradioco.configs.base](#), 48
 - [radioco.configs.base.manage](#), 48
- [radioco.apps.radioco.management.commandsradioco.create_example_data](#), settings, 48
- [radioco.configs.base.test_runner](#), 48
 - [radioco.configs.base.urls](#), 48
- [radioco.apps.radioco.migrations](#), 34
- [radioco.apps.radioco.migrations.0001__v3__adding_sqlite_configs](#), development, 48
- [radioco.configs.development.manage](#), 48
 - [radioco.configs.development.settings](#), 48
- [radioco.apps.radioco.test_utils](#), 35
- [radioco.apps.radioco.tests](#), 35
- [radioco.apps.radioco.tests.tests](#), 35
- [radioco.apps.radioco.tz_utils](#), 35
- [radioco.apps.radioco.utils](#), 36
- [radioco.apps.radioco.views](#), 36
- [radioco.apps.schedules](#), 45
- [radioco.apps.schedules.admin](#), 41
 - [radioco.apps.schedules.forms](#), 42
 - [radioco.apps.schedules.migrations](#), 38
 - [radioco.apps.schedules.migrations.0001_initial](#), 37
 - [radioco.apps.schedules.migrations.0002__v3_0__renaming_calendar_model](#), 37
 - [radioco.apps.schedules.migrations.0003__v3_0__create_excludeddates_model](#), 37
 - [radioco.apps.schedules.migrations.0004__v3_0__adding_rrules](#), 37
 - [radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_unique_calendar](#), 38
 - [radioco.apps.schedules.models](#), 42
 - [radioco.apps.schedules.tests](#), 41
 - [radioco.apps.schedules.tests.test_recurrences](#), 38
 - [radioco.apps.schedules.tests.test_schedules](#), 39
 - [radioco.apps.schedules.tests.test_timezone](#), 41
 - [radioco.apps.schedules.urls](#), 44
 - [radioco.apps.schedules.utils](#), 44
 - [radioco.apps.schedules.views](#), 44
- [radioco.apps.users](#), 48
- [radioco.apps.users.admin](#), 46
 - [radioco.apps.users.migrations](#), 46
 - [radioco.apps.users.migrations.0001_initial](#), 45
 - [radioco.apps.users.migrations.0002_change_textfield_to_richtextfield](#), 45
 - [radioco.apps.users.migrations.0003_auto_20160104_2029](#), 45
 - [radioco.apps.users.migrations.0004__v3_0__ckeditor_upload_field](#), 46
 - [radioco.apps.users.models](#), 47
 - [radioco.apps.users.tests](#), 47
 - [radioco.apps.users.urls](#), 47
 - [radioco.apps.users.views](#), 47
- [radioco.configs](#), 49

A

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

abstract (radioco.apps.global_settings.models.SingletonModel.Meta attribute), 24

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

add_item_elements() (radioco.apps.programmes.feeds.iTunesFeed method), 30

add_root_elements() (radioco.apps.programmes.feeds.iTunesFeed method), 30

announcers (radioco.apps.programmes.models.Programme attribute), 31

at() (radioco.apps.schedules.models.Transmission class method), 44

avatar (radioco.apps.users.models.UserProfile attribute), 47

B

base_fields (radioco.apps.api.views.ProgrammeFilterForm attribute), 18

base_fields (radioco.apps.api.views.TimezoneForm attribute), 19

base_fields (radioco.apps.api.views.TransmissionForm attribute), 19

base_fields (radioco.apps.programmes.admin.NonStaffEpisodeAdminForm attribute), 27

base_fields (radioco.apps.programmes.admin.NonStaffParticipantInlineForm attribute), 27

base_fields (radioco.apps.programmes.admin.NonStaffRoleInlineForm attribute), 28

base_fields (radioco.apps.schedules.forms.DeleteScheduleForm attribute), 42

base_fields (radioco.apps.users.admin.NonStaffUserProfileForm attribute), 46

base_filters (radioco.apps.api.views.EpisodeFilter attribute), 17

base_filters (radioco.apps.api.views.ProgrammeFilter attribute), 17

base_filters (radioco.apps.api.views.ScheduleFilter attribute), 19

between() (radioco.apps.schedules.models.Transmission class method), 44

C

calculate_effective_schedule_end_dt() (in module radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules), 38

calculate_effective_schedule_end_dt() (in module radioco.apps.schedules.models), 44

calculate_effective_schedule_start_dt() (in module radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules), 38

calculate_effective_schedule_start_dt() (in module radioco.apps.schedules.models), 44

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

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

Calendar.DoesNotExist, 42

Calendar.MultipleObjectsReturned, 42

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

CalendarAdminTests (class in radioco.apps.schedules.tests.test_schedules), 39

CalendarConfiguration (class in radioco.apps.global_settings.models), 23

CalendarConfiguration.DoesNotExist, 23

CalendarConfiguration.MultipleObjectsReturned, 23

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

CalendarManagerTests (class in radioco.apps.schedules.tests.test_schedules), 39

CalendarModelTests (class in radioco.apps.schedules.tests.test_schedules), 39

CalendarValidationTests	(class in radioco.apps.schedules.tests.test_schedules), 39	34
can_delete	(radioco.apps.users.admin.UserProfileInline attribute), 46	create_episode() (radioco.apps.programmes.models.EpisodeManager method), 31
CATEGORY_CHOICES	(radioco.apps.programmes.models.Programme attribute), 31	create_example_data() (in module radioco.apps.radioco.utils), 36
change_form_template	(radioco.apps.global_settings.admin.SingletonModelAdmin attribute), 22	create_test_data() (in module radioco.apps.radioco.test_utils), 35
change_list_template	(radioco.apps.schedules.admin.ScheduleAdmin attribute), 42	current() (radioco.apps.schedules.models.CalendarManager method), 42
change_photo_url()	(in module radioco.apps.programmes.migrations.0004_change_photo_url), 25	
change_programmes()	(in module radioco.apps.programmes.migrations.0007_change_default_image), 26	
change_users()	(in module radioco.apps.programmes.migrations.0007_change_default_image), 26	
change_view()	(radioco.apps.global_settings.admin.SingletonModelAdmin method), 22	
change_view()	(radioco.apps.users.admin.SingletonProfileAdmin method), 46	
changelist_view()	(radioco.apps.schedules.admin.ScheduleAdmin method), 42	
check_delete_permission()	(in module radioco.apps.radioco.utils), 36	
check_recorder_program()	(in module radioco.apps.api.recorder_views), 15	
child_schedules	(radioco.apps.schedules.models.Schedule attribute), 43	
clean()	(radioco.apps.api.views.ProgrammeFilterForm method), 18	
clean()	(radioco.apps.api.views.TransmissionForm method), 19	
clean()	(radioco.apps.programmes.admin.NonStaffParticipantInlineForm method), 27	
clean()	(radioco.apps.programmes.admin.NonStaffRoleInlineForm method), 28	
clean_programme()	(radioco.apps.programmes.admin.NonStaffEpisodeAdminForm method), 27	
clean_timezone()	(radioco.apps.api.views.TimezoneForm method), 19	
clean_username()	(radioco.apps.users.admin.NonStaffUserProfileForm method), 46	
clone_calendar()	(radioco.apps.schedules.admin.CalendarAdmin method), 41	
Command	(class in radioco.apps.radioco.management.commands.create_example_data),	

D

date	(radioco.apps.schedules.models.ExcludedDates attribute), 43
date_after()	(radioco.apps.schedules.models.Schedule method), 43
date_before()	(radioco.apps.schedules.models.Schedule method), 43
dates_between()	(radioco.apps.schedules.models.Schedule method), 43
DateTimeFieldTz	(class in radioco.apps.api.serializers), 15
default_image	(radioco.apps.api.views.ProgrammeFilterForm attribute), 18
declared_fields	(radioco.apps.api.views.TimezoneForm attribute), 19
declared_fields	(radioco.apps.api.views.TransmissionForm attribute), 19
declared_fields	(radioco.apps.programmes.admin.NonStaffEpisodeAdminForm attribute), 27
declared_fields	(radioco.apps.programmes.admin.NonStaffParticipantInlineForm attribute), 27
declared_fields	(radioco.apps.programmes.admin.NonStaffRoleInlineForm attribute), 28
declared_fields	(radioco.apps.schedules.forms.DeleteScheduleForm attribute), 42
declared_fields	(radioco.apps.users.admin.NonStaffUserProfileForm attribute), 46
declared_filters	(radioco.apps.api.views.EpisodeFilter attribute), 17
declared_filters	(radioco.apps.api.views.ProgrammeFilter attribute), 17
declared_filters	(radioco.apps.api.views.ScheduleFilter attribute), 19
delete()	(radioco.apps.global_settings.models.SingletonModelAdmin method), 24
DELETE_ALL	(radioco.apps.schedules.forms.DeleteScheduleForm attribute), 42
DELETE_ONLY_THIS	(radioco.apps.schedules.forms.DeleteScheduleForm attribute), 42
DELETE_THIS_AND_FOLLOWING	(radioco.apps.schedules.forms.DeleteScheduleForm attribute), 42

ExcludedDates (class in radioco.apps.schedules.models), 43

ExcludedDates.DoesNotExist, 43

ExcludedDates.MultipleObjectsReturned, 43

excludeddates_set (radioco.apps.schedules.models.Schedulefilter_class attribute), 43

extra (radioco.apps.programmes.admin.NonStaffParticipantInline attribute), 27

extra (radioco.apps.programmes.admin.NonStaffRoleInline attribute), 28

extra (radioco.apps.programmes.admin.PodcastInline attribute), 29

extra (radioco.apps.users.admin.UserProfileInline attribute), 47

F

feed_extra_kwargs() (radioco.apps.programmes.feeds.ProgrammeFeed method), 29

feed_type (radioco.apps.programmes.feeds.RssProgrammeFeed attribute), 30

field_has_changed() (in module radioco.apps.radioco.utils), 36

fields (radioco.apps.api.serializers.EpisodeSerializer.Meta attribute), 15

fields (radioco.apps.api.serializers.ProgrammeSerializer.Meta attribute), 16

fields (radioco.apps.api.serializers.RadiocomConfigurationSerializer.Meta attribute), 16

fields (radioco.apps.api.serializers.RadiocomProgrammeSerializer.Meta attribute), 16

fields (radioco.apps.api.serializers.ScheduleSerializer.Meta attribute), 16

fields (radioco.apps.api.views.EpisodeFilter.Meta attribute), 17

fields (radioco.apps.api.views.ProgrammeFilter.Meta attribute), 17

fields (radioco.apps.api.views.ScheduleFilter.Meta attribute), 19

fields (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 26

fields (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 27

fields (radioco.apps.users.admin.SingletonProfileAdmin attribute), 46

fieldsets (radioco.apps.schedules.admin.ScheduleAdmin attribute), 42

filter_backends (radioco.apps.api.views.EpisodeViewSet attribute), 17

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

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

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

filter_class (radioco.apps.api.views.EpisodeViewSet attribute), 17

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

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

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

fix_recurrence_date() (in module radioco.apps.radioco.tz_utils), 35

fix_recurrence_dst() (in module radioco.apps.radioco.tz_utils), 36

form (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 26

form (radioco.apps.programmes.admin.NonStaffParticipantInline attribute), 27

form (radioco.apps.programmes.admin.NonStaffRoleInline attribute), 28

form (radioco.apps.users.admin.SingletonProfileAdmin attribute), 46

form_class (radioco.apps.schedules.views.DeleteScheduleView attribute), 44

form_valid() (radioco.apps.schedules.views.DeleteScheduleView method), 44

formfield_for_foreignkey() (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 26

formset (radioco.apps.programmes.admin.NonStaffParticipantInline attribute), 27

formset (radioco.apps.programmes.admin.NonStaffRoleInline attribute), 28

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

G

get() (radioco.apps.global_settings.models.SingletonModelManager method), 24

get() (radioco.apps.schedules.views.DeleteScheduleView method), 44

get_absolute_url() (radioco.apps.programmes.models.Episode attribute), 30

get_absolute_url() (radioco.apps.programmes.models.Podcast attribute), 31

get_absolute_url() (radioco.apps.programmes.models.Programme attribute), 31

get_absolute_url() (radioco.apps.users.models.UserProfile attribute), 47

get_active() (radioco.apps.schedules.models.Calendar class method), 42

get_active_timezone() (in module radioco.apps.radioco.tz_utils), 36

get_category_display() (radioco.apps.programmes.models.Programme method), 31
 get_context_data() (radioco.apps.schedules.views.DeleteScheduleView method), 44
 get_facebook_url() (radioco.apps.api.serializers.RadiocomConfigurationSerializer method), 16
 get_fieldsets() (radioco.apps.programmes.admin.NonStaffProgrammeAdmin method), 28
 get_first_day_display() (radioco.apps.global_settings.models.CalendarConfiguration method), 23
 get_form_kwargs() (radioco.apps.schedules.views.DeleteScheduleView method), 44
 get_formset() (radioco.apps.programmes.admin.NonStaffParticipantInline method), 27
 get_formset() (radioco.apps.programmes.admin.NonStaffRoleInline method), 28
 get_global() (radioco.apps.global_settings.models.SingletonModel class method), 24
 get_global_model() (in module radioco.apps.global_settings.templatetags.global_settings_extends), 22
 get_initial() (radioco.apps.schedules.views.DeleteScheduleView method), 44
 get_language_display() (radioco.apps.programmes.models.Programme method), 31
 get_new_excluded_datetime() (radioco.apps.schedules.models.ExcludedDates method), 43
 get_next_by_date_joined() (radioco.apps.programmes.models.Role method), 32
 get_next_by_datetime() (radioco.apps.schedules.models.ExcludedDates method), 43
 get_next_by_start_dt() (radioco.apps.schedules.models.Schedule method), 43
 get_object() (radioco.apps.programmes.feeds.ProgrammeFeed method), 29
 get_prepopulated_fields() (radioco.apps.programmes.admin.NonStaffProgrammeAdmin method), 28
 get_previous_by_date_joined() (radioco.apps.programmes.models.Role method), 32
 get_previous_by_datetime() (radioco.apps.schedules.models.ExcludedDates method), 43
 get_previous_by_start_dt() (radioco.apps.global_settings.admin.SingletonModelAdmin method), 22
 get_queryset() (radioco.apps.programmes.admin.NonStaffEpisodeAdmin method), 43
 get_queryset() (radioco.apps.programmes.admin.NonStaffParticipantInline method), 28
 get_queryset() (radioco.apps.programmes.admin.NonStaffProgrammeAdmin method), 28
 get_queryset() (radioco.apps.programmes.admin.NonStaffRoleInline method), 28
 get_queryset() (radioco.apps.users.views.UsersView method), 47
 get_readonly_fields() (radioco.apps.programmes.admin.NonStaffEpisodeAdmin method), 26
 get_role_display() (radioco.apps.programmes.models.Participant method), 31
 get_role_display() (radioco.apps.programmes.models.Role method), 32
 get_schedule_which_excluded_dt() (radioco.apps.schedules.models.Schedule static method), 43
 get_station_photos() (radioco.apps.api.serializers.RadiocomConfigurationSerializer method), 16
 get_title() (radioco.apps.api.serializers.ScheduleSerializer method), 16
 get_twitter_url() (radioco.apps.api.serializers.RadiocomConfigurationSerializer method), 16
 get_type_display() (radioco.apps.schedules.models.Schedule method), 43
 get_urls() (radioco.apps.global_settings.admin.SingletonModelAdmin method), 22
 get_urls() (radioco.apps.users.admin.SingletonProfileAdmin method), 46
 GetObjectMixin (class in radioco.apps.radioco.utils), 36
 gettext_noop() (in module radioco.configs.base.settings), 48
 GMT (class in radioco.apps.radioco.tz_utils), 35

H

handle() (radioco.apps.radioco.management.commands.create_example_data method), 34
 has_add_permission() (radioco.apps.global_settings.admin.SingletonModelAdmin method), 22
 has_add_permission() (radioco.apps.schedules.admin.ScheduleAdmin method), 42
 has_add_permission() (radioco.apps.users.admin.SingletonProfileAdmin method), 46
 has_delete_permission() (radioco.apps.global_settings.admin.SingletonModelAdmin method), 22

- method), 22
- has_delete_permission() (radioco.apps.users.admin.SingletonProfileAdmin method), 46
- has_recurrences() (radioco.apps.schedules.models.Schedule method), 43
- I**
- include_date() (radioco.apps.schedules.models.Schedule method), 43
- index() (in module radioco.apps.radioco.views), 36
- inline_classes (radioco.apps.programmes.admin.PodcastInline attribute), 29
- inline_classes (radioco.apps.users.admin.UserProfileInline attribute), 47
- inlines (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 26
- inlines (radioco.apps.programmes.admin.NonStaffProgrammeAdmin attribute), 28
- inlines (radioco.apps.users.admin.UserProfileAdmin attribute), 46
- item_description() (radioco.apps.programmes.feeds.ProgrammeFeed method), 29
- item_enclosure_length() (radioco.apps.programmes.feeds.ProgrammeFeed method), 29
- item_enclosure_mime_type() (radioco.apps.programmes.feeds.ProgrammeFeed method), 29
- item_enclosure_url() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- item_extra_kwargs() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- item_guid() (radioco.apps.programmes.feeds.RssProgrammeFeed method), 30
- item_keywords() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- item_pubdate() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- item_title() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- items() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- iTunesFeed (class in radioco.apps.programmes.feeds), 30
- L**
- last() (radioco.apps.programmes.models.EpisodeManager static method), 31
- link() (radioco.apps.programmes.feeds.ProgrammeFeed method), 30
- list() (radioco.apps.api.views.ProgrammeViewSet method), 18
- list() (radioco.apps.api.views.RadiocomStation method), 18
- list() (radioco.apps.api.views.TransmissionViewSet method), 20
- list_display (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 26
- list_display (radioco.apps.programmes.admin.NonStaffProgrammeAdmin attribute), 28
- list_display (radioco.apps.schedules.admin.CalendarAdmin attribute), 41
- list_filter (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 27
- list_filter (radioco.apps.programmes.admin.NonStaffProgrammeAdmin attribute), 28
- list_filter (radioco.apps.schedules.admin.CalendarAdmin attribute), 41
- list_filter (radioco.apps.users.admin.UserProfileAdmin attribute), 46
- list_select_related (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 27
- list_select_related (radioco.apps.programmes.admin.NonStaffRoleInline attribute), 28
- localize() (radioco.apps.radioco.tz_utils.GMT method), 35
- lookup_field (radioco.apps.api.views.ProgrammeViewSet attribute), 18
- lookups() (radioco.apps.programmes.admin.OwnEpisodeIssueDateListFilter method), 29
- lookups() (radioco.apps.programmes.admin.OwnEpisodeProgrammeListFilter method), 29
- lowercase_language() (in module radioco.apps.programmes.migrations.0002_change_language_choices), 25
- M**
- media (module), 12
- media (radioco.apps.api.views.ProgrammeFilterForm attribute), 18
- media (radioco.apps.api.views.TimezoneForm attribute), 19
- media (radioco.apps.api.views.TransmissionForm attribute), 19
- media (radioco.apps.global_settings.admin.PodcastConfigurationAdmin attribute), 22
- media (radioco.apps.global_settings.admin.SingletonModelAdmin attribute), 23
- media (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 27
- media (radioco.apps.programmes.admin.NonStaffEpisodeAdminForm attribute), 27
- media (radioco.apps.programmes.admin.NonStaffParticipantInline attribute), 27
- media (radioco.apps.programmes.admin.NonStaffParticipantInlineForm attribute), 27

media (radioco.apps.programmes.admin.NonStaffProgrammeAdmin attribute), 28

media (radioco.apps.programmes.admin.NonStaffRoleInlineMigration attribute), 28

media (radioco.apps.programmes.admin.NonStaffRoleInlineForm attribute), 28

media (radioco.apps.programmes.admin.PodcastInline attribute), 29

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

media (radioco.apps.schedules.admin.ScheduleAdmin attribute), 42

media (radioco.apps.schedules.forms.DeleteScheduleForm attribute), 42

media (radioco.apps.users.admin.NonStaffUserProfileForm attribute), 46

media (radioco.apps.users.admin.SingletonProfileAdmin attribute), 46

media (radioco.apps.users.admin.UserProfileAdmin attribute), 46

media (radioco.apps.users.admin.UserProfileInline attribute), 47

memorize (class in radioco.apps.radioco.utils), 36

migrate_board() (in module radioco.apps.schedules.migrations.0002__v3_0__renaming_calendar_apps), 37

migrate_daily_recurrences() (in module radioco.apps.schedules.migrations.0004__v3_0__adding_rrules), 37

migrate_datetime_to_utc() (in module radioco.apps.radioco.migrations.0001__v3_0__mysql_timezone), 34

migrate_schedules() (in module radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_radioco_migrations), 38

Migration (class in radioco.apps.global_settings.migrations.0001_initial), 20

Migration (class in radioco.apps.global_settings.migrations.0002_remove_calendar_apps), 21

Migration (class in radioco.apps.global_settings.migrations.0003_siteconfiguration_content), 21

Migration (class in radioco.apps.global_settings.migrations.0004_auto_20150606_1533), 21

Migration (class in radioco.apps.global_settings.migrations.0005_auto_20150606_1545), 21

Migration (class in radioco.apps.global_settings.migrations.0006_auto_20160116_1509), 21

Migration (class in radioco.apps.global_settings.migrations.0007__v3_0__calendar_apps), 22

Migration (class in radioco.apps.global_settings.migrations.0008__v3_0__radiocommon), 22

Migration (class in radioco.apps.programmes.migrations.0001_initial), 24

Migration (class in radioco.apps.programmes.migrations.0002_change_language_choices), 25

Migration (class in radioco.apps.programmes.migrations.0003_change_textfield_to_rich_text), 25

Migration (class in radioco.apps.programmes.migrations.0004_change_photo_url), 25

Migration (class in radioco.apps.programmes.migrations.0005_auto_20150531_1734), 25

Migration (class in radioco.apps.programmes.migrations.0006_auto_20160104_2029), 25

Migration (class in radioco.apps.programmes.migrations.0007_change_default_image), 26

Migration (class in radioco.apps.programmes.migrations.0008_auto_20160116_1509), 26

Migration (class in radioco.apps.programmes.migrations.0009__v3_0__small_tweaks), 26

Migration (class in radioco.apps.schedules.migrations.0001_initial), 37

Migration (class in radioco.apps.schedules.migrations.0002__v3_0__renaming_calendar_apps), 37

Migration (class in radioco.apps.schedules.migrations.0003__v3_0__create_excluded_rules), 37

Migration (class in radioco.apps.schedules.migrations.0004__v3_0__adding_rrules), 38

Migration (class in radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_radioco_migrations), 38

Migration (class in radioco.apps.users.migrations.0001_initial), 45

Migration (class in radioco.apps.users.migrations.0002__v3_0__adding_profile_picture), 45

dioco.apps.users.migrations.0002_change_textfield_to_staff_episode_admin (class in radioco.apps.programmes.admin), 26	
45	
Migration (class in radioco.apps.users.migrations.0003_auto_20160104_2029), dioco.apps.programmes.admin), 27	NonStaffEpisodeAdminForm (class in radioco.apps.programmes.admin), 27
45	NonStaffEpisodeAdminForm.Meta (class in radioco.apps.programmes.admin), 27
Migration (class in radioco.apps.users.migrations.0004__v3_0__ckeditor_staff_participant_inline (class in radioco.apps.programmes.admin), 27	
46	
mock_now() (in module radioco.apps.api.tests.test_radiocom_api), 13	NonStaffParticipantInlineForm (class in radioco.apps.programmes.admin), 27
	NonStaffParticipantInlineFormset (class in radioco.apps.programmes.admin), 27
mock_now() (in module radioco.apps.api.tests.test_schedules), 15	NonStaffProgrammeAdmin (class in radioco.apps.programmes.admin), 28
	NonStaffRoleInline (class in radioco.apps.programmes.admin), 28
mock_now() (in module radioco.apps.schedules.tests.test_schedules), 41	
	NonStaffRoleInlineForm (class in radioco.apps.programmes.admin), 28
model (radioco.apps.api.serializers.EpisodeSerializer.Meta attribute), 15	NonStaffRoleInlineFormset (class in radioco.apps.programmes.admin), 28
model (radioco.apps.api.serializers.ProgrammeSerializer.Meta attribute), 16	SocialSizeMetaProfileForm (class in radioco.apps.users.admin), 46
model (radioco.apps.api.serializers.RadiocomConfigurationSerializer.Meta attribute), 16	normalize_timezone (radioco.apps.radioco.tz_utils.GMT method), 35
model (radioco.apps.api.serializers.RadiocomProgrammeSerializer.Meta attribute), 16	now() (radioco.apps.api.views.TransmissionViewSet method), 20
model (radioco.apps.api.serializers.ScheduleSerializer.Meta attribute), 16	
model (radioco.apps.api.views.EpisodeFilter.Meta attribute), 17	O
model (radioco.apps.api.views.ProgrammeFilter.Meta attribute), 17	object (radioco.apps.radioco.utils.GetObjectMixin attribute), 36
model (radioco.apps.api.views.ScheduleFilter.Meta attribute), 19	object_history_template (radioco.apps.global_settings.admin.SingletonModelAdmin attribute), 23
model (radioco.apps.programmes.admin.NonStaffEpisodeAdminForm.Meta attribute), 27	objects (radioco.apps.global_settings.models.CalendarConfiguration attribute), 23
model (radioco.apps.programmes.admin.NonStaffParticipantInline attribute), 27	objects (radioco.apps.global_settings.models.PodcastConfiguration attribute), 23
model (radioco.apps.programmes.admin.NonStaffRoleInline attribute), 28	objects (radioco.apps.global_settings.models.RadiocomConfiguration attribute), 23
model (radioco.apps.programmes.admin.PodcastInline attribute), 29	objects (radioco.apps.global_settings.models.SingletonModel attribute), 24
model (radioco.apps.radioco.utils.DeletePermissionMixin attribute), 36	objects (radioco.apps.global_settings.models.SiteConfiguration attribute), 24
model (radioco.apps.schedules.views.DeleteScheduleView attribute), 45	objects (radioco.apps.programmes.models.Episode attribute), 30
model (radioco.apps.users.admin.UserProfileInline attribute), 47	objects (radioco.apps.programmes.models.Participant attribute), 31
model (radioco.apps.users.views.UsersView attribute), 47	objects (radioco.apps.programmes.models.Podcast attribute), 31
MyTestSuiteRunner (class in radioco.configs.base.test_runner), 48	objects (radioco.apps.programmes.models.Programme attribute), 31
	objects (radioco.apps.programmes.models.Role attribute), 32
N	
name (radioco.apps.schedules.models.Transmission attribute), 44	
next_dates() (in module radioco.apps.schedules.utils), 44	

- objects (radioco.apps.schedules.models.Calendar attribute), 42
- objects (radioco.apps.schedules.models.ExcludedDates attribute), 43
- objects (radioco.apps.schedules.models.Schedule attribute), 43
- objects (radioco.apps.users.models.UserProfile attribute), 47
- operations (radioco.apps.global_settings.migrations.0001_initial_migration attribute), 20
- operations (radioco.apps.global_settings.migrations.0002_reordering_end_date_attribute_display_name_calendar_migration attribute), 21
- operations (radioco.apps.global_settings.migrations.0003_scheduling_rules_migration (class in radioco.apps.programmes.admin), 29
- operations (radioco.apps.global_settings.migrations.0004_auto_20150606_1435_migration_filter (class in radioco.apps.programmes.admin), 29
- operations (radioco.apps.global_settings.migrations.0005_auto_20150606_1415_migration attribute), 21
- operations (radioco.apps.global_settings.migrations.0006_auto_20160116_1509_migration parameter_name (radioco.apps.programmes.admin.OwnEpisodeIssueDateL attribute), 29
- operations (radioco.apps.global_settings.migrations.0007__v3_0__calendar_configuration_tweaks_migration parameter_name (radioco.apps.programmes.admin.OwnEpisodeProgramme attribute), 29
- operations (radioco.apps.global_settings.migrations.0008__v3_0__radioco_configuration_migration participant (class in radioco.apps.programmes.models), 31
- operations (radioco.apps.programmes.migrations.0001_initial_migration Participant.DoesNotExist, 31
- operations (radioco.apps.programmes.migrations.0002_change_language_choices_migration Participant.MultipleObjectsReturned, 31
- operations (radioco.apps.programmes.migrations.0003_change_textfield_to_richtextfield_migration participant_set (radioco.apps.programmes.models.Episode attribute), 30
- operations (radioco.apps.programmes.migrations.0004_change_photo_url_migration people (radioco.apps.programmes.models.Episode attribute), 30
- operations (radioco.apps.programmes.migrations.0005_auto_20150531_1734_migration person_update (radioco.apps.api.views.TransmissionOperationViewSet method), 19
- operations (radioco.apps.programmes.migrations.0006_auto_20160104_2029_migration permission_classes (radioco.apps.api.views.ProgrammeViewSet attribute), 19
- operations (radioco.apps.programmes.migrations.0007_change_default_image_migration permission_classes (radioco.apps.api.views.ScheduleViewSet attribute), 19
- operations (radioco.apps.programmes.migrations.0008_auto_20160116_1509_migration permission_classes (radioco.apps.api.views.TransmissionOperationViewSet attribute), 19
- operations (radioco.apps.programmes.migrations.0009__v3_0__small_tweaks_migration person (radioco.apps.programmes.models.Participant attribute), 31
- operations (radioco.apps.radioco.migrations.0001__v3_0__mysql_timezone_migration attribute), 34
- operations (radioco.apps.schedules.migrations.0001_initial_migration person (radioco.apps.programmes.models.Role attribute), 32
- operations (radioco.apps.schedules.migrations.0002__v3_0__renaming_calendar_model_migration photo (radioco.apps.programmes.models.Programme attribute), 31
- operations (radioco.apps.schedules.migrations.0003__v3_0__create_excludeddates_model_migration Podcast (class in radioco.apps.programmes.models), 31
- operations (radioco.apps.schedules.migrations.0004__v3_0__padding_rules_migration podcast (radioco.apps.programmes.models.Episode attribute), 30
- operations (radioco.apps.schedules.migrations.0005__v3_0__podcast_configuration Podcast.DoesNotExist, 31
- operations (radioco.apps.schedules.migrations.0005__v3_0__migrating_schedules_to_unique_calendar_migration Podcast.MultipleObjectsReturned, 31
- operations (radioco.apps.schedules.migrations.0005__v3_0__podcast_configuration migrating_schedules_to_unique_calendar_migration (class in radioco.apps.global_settings.models), 23

PodcastConfiguration.DoesNotExist, 23
 PodcastConfiguration.MultipleObjectsReturned, 23
 PodcastConfigurationAdmin (class in radioco.apps.global_settings.admin), 22
 PodcastInline (class in radioco.apps.programmes.admin), 29
 Programme (class in radioco.apps.programmes.models), 31
 programme (radioco.apps.programmes.models.Episode attribute), 30
 programme (radioco.apps.programmes.models.Role attribute), 32
 programme (radioco.apps.schedules.models.Schedule attribute), 43
 programme (radioco.apps.schedules.models.Transmission attribute), 44
 Programme.DoesNotExist, 31
 Programme.MultipleObjectsReturned, 31
 programme_detail() (in module radioco.apps.programmes.views), 33
 programme_url (radioco.apps.schedules.models.Transmission attribute), 44
 ProgrammeFeed (class in radioco.apps.programmes.feeds), 29
 ProgrammeFilter (class in radioco.apps.api.views), 17
 ProgrammeFilter.Meta (class in radioco.apps.api.views), 17
 ProgrammeFilterForm (class in radioco.apps.api.views), 17
 ProgrammeModelAdminTests (class in radioco.apps.programmes.tests), 33
 ProgrammeModelTests (class in radioco.apps.programmes.tests), 33
 ProgrammeSerializer (class in radioco.apps.api.serializers), 15
 ProgrammeSerializer.Meta (class in radioco.apps.api.serializers), 15
 ProgrammeViewSet (class in radioco.apps.api.views), 18

Q

queryset (radioco.apps.api.views.EpisodeViewSet attribute), 17
 queryset (radioco.apps.api.views.ProgrammeViewSet attribute), 18
 queryset (radioco.apps.api.views.ScheduleViewSet attribute), 19
 queryset (radioco.apps.api.views.TransmissionOperationViewSet attribute), 19
 queryset (radioco.apps.api.views.TransmissionViewSet attribute), 20
 queryset() (radioco.apps.programmes.admin.OwnEpisodeIssueDateListFilter method), 29
 queryset() (radioco.apps.programmes.admin.OwnEpisodeProgrammeListFilter method), 29

R

radioco (module), 49
 radioco.apps (module), 48
 radioco.apps.api (module), 20
 radioco.apps.api.recorder_views (module), 15
 radioco.apps.api.serializers (module), 15
 radioco.apps.api.tests (module), 15
 radioco.apps.api.tests.test_api (module), 12
 radioco.apps.api.tests.test_fullcalendar (module), 12
 radioco.apps.api.tests.test_programmes (module), 13
 radioco.apps.api.tests.test_radiocom_api (module), 13
 radioco.apps.api.tests.test_recorder (module), 14
 radioco.apps.api.tests.test_schedules (module), 14
 radioco.apps.api.urls (module), 17
 radioco.apps.api.views (module), 17
 radioco.apps.api.viewsets (module), 20
 radioco.apps.global_settings (module), 24
 radioco.apps.global_settings.admin (module), 22
 radioco.apps.global_settings.migrations (module), 22
 radioco.apps.global_settings.migrations.0001_initial (module), 20
 radioco.apps.global_settings.migrations.0002_remove_calendarconfiguration (module), 21
 radioco.apps.global_settings.migrations.0003_siteconfiguration_footer (module), 21
 radioco.apps.global_settings.migrations.0004_auto_20150606_1335 (module), 21
 radioco.apps.global_settings.migrations.0005_auto_20150606_1415 (module), 21
 radioco.apps.global_settings.migrations.0006_auto_20160116_1509 (module), 21
 radioco.apps.global_settings.migrations.0007__v3_0__calendarconfiguration (module), 22
 radioco.apps.global_settings.migrations.0008__v3_0__radiocomconfiguration (module), 22
 radioco.apps.global_settings.models (module), 23
 radioco.apps.global_settings.templatetags (module), 22
 radioco.apps.global_settings.templatetags.global_settings_extras (module), 22
 radioco.apps.programmes (module), 34
 radioco.apps.programmes.admin (module), 26
 radioco.apps.programmes.feeds (module), 29
 radioco.apps.programmes.migrations (module), 26
 radioco.apps.programmes.migrations.0001_initial (module), 24
 radioco.apps.programmes.migrations.0002_change_language_choices (module), 25
 radioco.apps.programmes.migrations.0003_change_textfield_to_richtextfield (module), 25
 radioco.apps.programmes.migrations.0004_change_photo_url (module), 25
 radioco.apps.programmes.migrations.0005_auto_20150531_1734 (module), 25

- radioco.apps.programmes.migrations.0006_auto_20160104_2029 (module), 25
- radioco.apps.programmes.migrations.0007_change_default_ridige (module), 26
- radioco.apps.programmes.migrations.0008_auto_20160116_1509 (module), 26
- radioco.apps.programmes.migrations.0009__v3_0__small_tweets (module), 26
- radioco.apps.programmes.models (module), 30
- radioco.apps.programmes.tests (module), 32
- radioco.apps.programmes.urls (module), 33
- radioco.apps.programmes.utils (module), 33
- radioco.apps.programmes.views (module), 33
- radioco.apps.radioco (module), 36
- radioco.apps.radioco.context_processors (module), 35
- radioco.apps.radioco.management (module), 34
- radioco.apps.radioco.management.commands (module), 34
- radioco.apps.radioco.management.commands.create_example_data (module), 34
- radioco.apps.radioco.migrations (module), 34
- radioco.apps.radioco.migrations.0001__v3_0__mysql_timezone (module), 34
- radioco.apps.radioco.test_utils (module), 35
- radioco.apps.radioco.tests (module), 35
- radioco.apps.radioco.tests.tests (module), 35
- radioco.apps.radioco.tz_utils (module), 35
- radioco.apps.radioco.utils (module), 36
- radioco.apps.radioco.views (module), 36
- radioco.apps.schedules (module), 45
- radioco.apps.schedules.admin (module), 41
- radioco.apps.schedules.forms (module), 42
- radioco.apps.schedules.migrations (module), 38
- radioco.apps.schedules.migrations.0001_initial (module), 37
- radioco.apps.schedules.migrations.0002__v3_0__renaming_RadiocomConfiguration (module), 37
- radioco.apps.schedules.migrations.0003__v3_0__create_example_data (module), 37
- radioco.apps.schedules.migrations.0004__v3_0__adding_rules (module), 37
- radioco.apps.schedules.migrations.0005__v3_0__migrating_RadiocomProgrammeViews (module), 38
- radioco.apps.schedules.models (module), 42
- radioco.apps.schedules.tests (module), 41
- radioco.apps.schedules.tests.test_reurrences (module), 38
- radioco.apps.schedules.tests.test_schedules (module), 39
- radioco.apps.schedules.tests.test_timezone (module), 41
- radioco.apps.schedules.urls (module), 44
- radioco.apps.schedules.utils (module), 44
- radioco.apps.schedules.views (module), 44
- radioco.apps.users (module), 48
- radioco.apps.users.admin (module), 46
- radioco.apps.users.migrations (module), 46
- radioco.apps.users.migrations.0001_initial (module), 45
- radioco.apps.users.migrations.0002_change_textfield_to_richtextfield (module), 45
- radioco.apps.users.migrations.0003_auto_20160104_2029 (module), 45
- radioco.apps.users.migrations.0004__v3_0__ckeditor_upload_field (module), 46
- radioco.apps.users.models (module), 47
- radioco.apps.users.tests (module), 47
- radioco.apps.users.urls (module), 47
- radioco.apps.users.views (module), 47
- radioco.configs (module), 49
- radioco.configs.base (module), 48
- radioco.configs.base.manage (module), 48
- radioco.configs.base.settings (module), 48
- radioco.configs.base.test_runner (module), 48
- radioco.configs.base.urls (module), 48
- radioco.configs.development (module), 48
- radioco.configs.development.manage (module), 48
- radioco.configs.development.settings (module), 48
- radioco.configs.heroku (module), 49
- radioco.configs.heroku.manage (module), 48
- radioco.configs.heroku.settings (module), 49
- radioco.configs.heroku.wsgi (module), 49
- radioco.configs.staging (module), 49
- radioco.configs.staging.manage (module), 49
- radioco.configs.staging.settings (module), 49
- radioco.configs.staging.wsgi (module), 49
- RadiocomConfiguration (class in radioco.apps.global_settings.models), 23
- RadiocomConfiguration.DoesNotExist, 23
- RadiocomConfiguration.MultipleObjectsReturned, 23
- RadiocomConfigurationSerializer (class in radioco.apps.api.serializers), 16
- RadiocomConfigurationSerializer.Meta (class in radioco.apps.api.serializers), 16
- RadiocomProgrammeSerializer (class in radioco.apps.api.serializers), 16
- RadiocomProgrammeSerializer.Meta (class in radioco.apps.api.serializers), 16
- RadiocomProgrammeViewSet (class in radioco.apps.api.views), 18
- RadiocomStation (class in radioco.apps.api.views), 18
- RadiocomTransmissionSerializer (class in radioco.apps.api.serializers), 16
- RadiocomTransmissionViewSet (class in radioco.apps.api.views), 18
- RadioIntegrationTests (class in radioco.apps.radioco.tests.tests), 35
- readonly_fields (radioco.apps.global_settings.admin.PodcastConfiguration attribute), 22
- readonly_fields (radioco.apps.schedules.admin.ScheduleAdmin attribute), 42

rearrange_episodes() (radioco.apps.programmes.models.Programme method), 32
 rearrange_episodes() (radioco.apps.schedules.models.Calendar method), 42
 recorder_token (radioco.apps.global_settings.models.PodcastConfiguration attribute), 23
 recording_schedules() (in module radioco.apps.api.recorder_views), 15
 recurrence_after() (in module radioco.apps.radioco.tz_utils), 36
 recurrence_after() (in module radioco.apps.schedules.migrations.0005__v3_0__migrations class in radioco.apps.schedules), 38
 recurrence_before() (in module radioco.apps.radioco.tz_utils), 36
 recurrence_before() (in module radioco.apps.schedules.migrations.0005__v3_0__migrations class in radioco.apps.schedules), 38
 recurrences (radioco.apps.schedules.models.Schedule attribute), 44
 RecurrenceTests (class in radioco.apps.schedules.tests.test_recurrences), 39
 response_change() (radioco.apps.global_settings.admin.SingletonModelAdmin method), 23
 response_change() (radioco.apps.users.admin.SingletonProfileAdmin method), 46
 Role (class in radioco.apps.programmes.models), 32
 Role.DoesNotExist, 32
 Role.MultipleObjectsReturned, 32
 role_set (radioco.apps.programmes.models.Programme attribute), 32
 rss_attributes() (radioco.apps.programmes.feeds.iTunesFeed method), 30
 RssProgrammeFeed (class in radioco.apps.programmes.feeds), 30
 runtime (radioco.apps.programmes.models.Episode attribute), 30
 runtime (radioco.apps.programmes.models.Programme attribute), 32
 runtime (radioco.apps.schedules.models.Schedule attribute), 44
S
 save() (radioco.apps.global_settings.models.SingletonModelAdmin method), 24
 save() (radioco.apps.programmes.models.Programme method), 32
 save() (radioco.apps.schedules.models.Calendar method), 42
 save() (radioco.apps.schedules.models.Schedule method), 44
 save() (radioco.apps.users.admin.NonStaffUserProfileForm method), 46
 save() (radioco.apps.users.models.UserProfile method), 47
 save_formset() (radioco.apps.programmes.admin.NonStaffEpisodeAdmin method), 27
 save_formset() (radioco.apps.programmes.admin.NonStaffProgrammeAdmin method), 28
 save_model() (radioco.apps.programmes.admin.NonStaffEpisodeAdmin method), 27
 save_model() (radioco.apps.users.admin.SingletonProfileAdmin method), 46
 save_slug() (in module radioco.apps.users.models), 47
 Schedules (class in radioco.apps.schedules), 43
 schedule (radioco.apps.schedules.models.ExcludedDates attribute), 43
 Schedule.DoesNotExist, 43
 Schedule.MultipleObjectsReturned, 43
 schedule_set (radioco.apps.schedules.admin.DeleteScheduleView attribute), 45
 schedule_list() (in module radioco.apps.schedules.views), 45
 schedule_set (radioco.apps.programmes.models.Programme attribute), 32
 schedule_set (radioco.apps.schedules.models.Calendar attribute), 42
 schedule_set (radioco.apps.schedules.models.Schedule attribute), 44
 ScheduleAdmin (class in radioco.apps.schedules.admin), 42
 ScheduleBetweenTests (class in radioco.apps.schedules.tests.test_schedules), 39
 ScheduleFilter (class in radioco.apps.api.views), 18
 ScheduleFilter.Meta (class in radioco.apps.api.views), 18
 ScheduleModelTests (class in radioco.apps.schedules.tests.test_schedules), 40
 ScheduleModelTests (class in radioco.apps.schedules.tests.test_timezone), 41
 ScheduleSerializer (class in radioco.apps.api.serializers), 16
 ScheduleSerializer.Meta (class in radioco.apps.api.serializers), 16
 ScheduleUtilsTests (class in radioco.apps.schedules.tests.test_schedules), 40
 ScheduleValidationTests (class in radioco.apps.schedules.tests.test_schedules), 40
 ScheduleViewSet (class in radioco.apps.api.views), 19
 search_fields (radioco.apps.programmes.admin.NonStaffEpisodeAdmin attribute), 27

[search_fields \(radioco.apps.programmes.admin.NonStaffProgrammeAdmin attribute\)](#), 28
[search_fields \(radioco.apps.schedules.admin.CalendarAdmin attribute\)](#), 42
[serializer_class \(radioco.apps.api.views.EpisodeViewSet attribute\)](#), 17
[serializer_class \(radioco.apps.api.views.ProgrammeViewSet attribute\)](#), 18
[serializer_class \(radioco.apps.api.views.RadiocomProgrammeViewSet attribute\)](#), 18
[serializer_class \(radioco.apps.api.views.RadiocomStationViewSet attribute\)](#), 18
[serializer_class \(radioco.apps.api.views.RadiocomTransmissionViewSet attribute\)](#), 18
[serializer_class \(radioco.apps.api.views.ScheduleViewSet attribute\)](#), 19
[serializer_class \(radioco.apps.api.views.TransmissionOperationViewSet attribute\)](#), 20
[serializer_class \(radioco.apps.api.views.TransmissionViewSet attribute\)](#), 20
[set_active\(\) \(radioco.apps.schedules.admin.CalendarAdmin method\)](#), 42
[settings\(\) \(in module radioco.apps.radioco.context_processors\)](#), 35
[setUp\(\) \(radioco.apps.api.tests.test_fullcalendar.TestFullCalendarApi method\)](#), 12
[setUp\(\) \(radioco.apps.api.tests.test_programmes.TestProgrammesAPI method\)](#), 13
[setUp\(\) \(radioco.apps.api.tests.test_radiocom_api.TestRadiocomAPI method\)](#), 13
[setUp\(\) \(radioco.apps.programmes.tests.EpisodeManagerTests method\)](#), 32
[setUp\(\) \(radioco.apps.programmes.tests.EpisodeModelTests method\)](#), 33
[setUp\(\) \(radioco.apps.programmes.tests.ProgrammeModelAdminTests method\)](#), 33
[setUp\(\) \(radioco.apps.programmes.tests.ProgrammeModelTests method\)](#), 33
[setUp\(\) \(radioco.apps.schedules.tests.test_reurrences.EmptyRecurrenceTests method\)](#), 38
[setUp\(\) \(radioco.apps.schedules.tests.test_reurrences.RecurrenceTests method\)](#), 39
[setUp\(\) \(radioco.apps.schedules.tests.test_schedules.CalendarAdminTests method\)](#), 39
[setUp\(\) \(radioco.apps.schedules.tests.test_schedules.CalendarManagerTests method\)](#), 39
[setUp\(\) \(radioco.apps.schedules.tests.test_schedules.CalendarValidationTests method\)](#), 39
[setUp\(\) \(radioco.apps.schedules.tests.test_schedules.ScheduleBetweenTests method\)](#), 40
[setUp\(\) \(radioco.apps.schedules.tests.test_schedules.ScheduleModelTests method\)](#), 40
[setUp\(\) \(radioco.apps.schedules.tests.test_schedules.ScheduleUtilsTests method\)](#), 45
[setUp\(\) \(radioco.apps.schedules.tests.test_schedules.TransmissionModelTests method\)](#), 41
[setUp\(\) \(radioco.apps.schedules.tests.test_timezone.ScheduleModelTests method\)](#), 41
[setUpTestData\(\) \(radioco.apps.api.tests.test_recorder.TestProgrammesAPI class method\)](#), 14
[setUpTestData\(\) \(radioco.apps.radioco.test_utils.TestDataMixin class method\)](#), 35
[show_toolbar\(\) \(in module radioco.configs.development.settings\)](#), 48
[SingletonModel \(class in module radioco.apps.global_settings.models\)](#), 23
[SingletonModel.Meta \(class in module radioco.apps.global_settings.models\)](#), 23
[SingletonModelAdmin \(class in module radioco.apps.global_settings.admin\)](#), 22
[SingletonModelManager \(class in module radioco.apps.global_settings.models\)](#), 24
[SingletonProfileAdmin \(class in module radioco.apps.users.admin\)](#), 46
[SiteConfiguration \(class in module radioco.apps.global_settings.models\)](#), 24
[SiteConfiguration.DoesNotExist](#), 24
[SiteConfiguration.MultipleObjectsReturned](#), 24
[slug_api \(radioco.apps.schedules.models.Transmission attribute\)](#), 44
[source_api \(radioco.apps.schedules.models.Schedule attribute\)](#), 44
[start_tz\(\) \(in module radioco.apps.radioco.test_utils\)](#), 35
[start_dt \(radioco.apps.programmes.models.Programme attribute\)](#), 32
[submit_recorder\(\) \(in module radioco.apps.api.recorder_views\)](#), 15
[suffix \(radioco.apps.api.views.EpisodeViewSet attribute\)](#), 18
[suffix \(radioco.apps.api.views.ProgrammeViewSet attribute\)](#), 18
[suffix \(radioco.apps.api.views.RadiocomProgrammeViewSet attribute\)](#), 18
[suffix \(radioco.apps.api.views.RadiocomStation attribute\)](#), 18
[suffix \(radioco.apps.api.views.RadiocomTransmissionViewSet attribute\)](#), 19
[suffix \(radioco.apps.api.views.ScheduleViewSet attribute\)](#), 19
[suffix \(radioco.apps.api.views.TransmissionOperationViewSet attribute\)](#), 20
[suffix \(radioco.apps.api.views.TransmissionViewSet attribute\)](#), 20
[template_name \(radioco.apps.schedules.views.DeleteScheduleView attribute\)](#), 45

test_absolute_url() (radioco.apps.programmes.tests.ProgrammeModelTests.method), 33

test_absolute_url() (radioco.apps.programmes.tests.EpisodeModelTests.method), 33

test_after() (radioco.apps.schedules.tests.test_reurrences.EmptyRecurrenceTests.method), 38

test_after() (radioco.apps.schedules.tests.test_reurrences.RecurrenceTests.method), 39

test_after_inclusive() (radioco.apps.schedules.tests.test_reurrences.EmptyRecurrenceTests.method), 38

test_after_inclusive() (radioco.apps.schedules.tests.test_reurrences.RecurrenceTests.method), 39

test_api() (radioco.apps.api.tests.test_api.TestApi.method), 12

test_at() (radioco.apps.schedules.tests.test_schedules.TransmissionModelTests.method), 41

test_available_dates_after() (radioco.apps.schedules.tests.test_schedules.ScheduleUtilsTests.method), 40

test_available_dates_none() (radioco.apps.schedules.tests.test_schedules.ScheduleUtilsTests.method), 40

test_before() (radioco.apps.schedules.tests.test_reurrences.EmptyRecurrenceTests.method), 38

test_before_dt() (radioco.apps.schedules.tests.test_reurrences.RecurrenceTests.method), 39

test_before_inclusive() (radioco.apps.schedules.tests.test_reurrences.EmptyRecurrenceTests.method), 38

test_before_inclusive() (radioco.apps.schedules.tests.test_reurrences.RecurrenceTests.method), 39

test_between() (radioco.apps.schedules.tests.test_schedules.TransmissionModelTests.method), 41

test_between_by_queryset() (radioco.apps.schedules.tests.test_schedules.TransmissionModelTests.method), 41

test_between_includes_started_episode() (radioco.apps.schedules.tests.test_schedules.ScheduleBetweenTests.method), 40

test_CEST_transition() (radioco.apps.schedules.tests.test_timezone.ScheduleModelTests.method), 41

test_CET_transition() (radioco.apps.schedules.tests.test_timezone.ScheduleModelTests.method), 41

test_CET_transitions() (in module radioco.apps.schedules.tests.test_timezone), 41

test_cleaned_internal_recurrence_dates() (radioco.apps.schedules.tests.test_timezone.ScheduleModelTests.method), 41

test_create_episode() (radioco.apps.programmes.tests.EpisodeManagerTests.method), 32

test_create_episode() (radioco.apps.schedules.tests.test_schedules.CalendarAdminTests.method), 39

test_create_episode() (radioco.apps.schedules.tests.test_schedules.CalendarManagerTests.method), 39

test_current_no_date() (radioco.apps.schedules.tests.test_schedules.CalendarManagerTests.method), 39

test_date_after() (radioco.apps.schedules.tests.test_schedules.ScheduleModelTests.method), 40

test_date_after_exclude() (radioco.apps.schedules.tests.test_schedules.ScheduleModelTests.method), 40

test_dates_between() (radioco.apps.schedules.tests.test_schedules.ScheduleModelTests.method), 40

test_dates_between_complex_ruleset() (radioco.apps.schedules.tests.test_schedules.ScheduleModelTests.method), 40

test_dates_between_earlier_end_by_programme() (radioco.apps.schedules.tests.test_schedules.ScheduleModelTests.method), 40

test_dates_between_includes_started_episode() (radioco.apps.schedules.tests.test_schedules.ScheduleBetweenTests.method), 40

test_dates_between_later_start_by_programme() (radioco.apps.schedules.tests.test_schedules.ScheduleModelTests.method), 40

test_dictionary_key() (radioco.tests.test_utils.UtilsTest.method), 35

test_end() (radioco.apps.schedules.tests.test_schedules.TransmissionModelTests.method), 41

test_end_gt_calendar() (radioco.apps.schedules.tests.test_schedules.ScheduleModelTests.method), 40

test_episode() (radioco.apps.api.tests.test_api.TestSerializers.method), 12

test_episode_url() (radioco.apps.schedules.tests.test_schedules.TransmissionModelTests.method), 41

test_episode_programme() (radioco.apps.api.tests.test_api.TestSerializers.method), 12

test_episodes_delete() (radioco.apps.api.tests.test_programmes.TestNotAllowedMethodsPr.method), 13

test_episodes_get_all() (radioco.apps.api.tests.test_programmes.TestProgrammesAPI.method), 13

method), 13	method), 13
test_episodes_get_by_programme() dioco.apps.api.tests.test_programmes.TestProgrammesAPI method), 13	(ra- test_name() (radioco.apps.schedules.tests.test_schedules.TransmissionModelTests) method), 41
test_episodes_post() dioco.apps.api.tests.test_programmes.TestNotAllowedMethodsProgrammesAPI method), 13	test_no_values() (radioco.apps.schedules.tests.test_reurrences.EmptyRecurrenceTests) method), 39
test_episodes_put() dioco.apps.api.tests.test_programmes.TestNotAllowedMethodsProgrammesAPI method), 13	test_programme_in_list() (radioco.apps.programmes.tests.EpisodeManagerTests) method), 32
test_example_data() dioco.apps.radioco.tests.tests.UtilsTest method), 35	test_programme_photo_url() (radioco.apps.programmes.tests.EpisodeManagerTests) method), 32
test_fields() (radioco.apps.schedules.tests.test_schedules.ScheduleValidationTests) method), 40	test_programme_url() (radioco.apps.schedules.tests.test_schedules.TransmissionModelTests) method), 41
test_fieldset() (radioco.apps.programmes.tests.ProgrammeModelAdminTests) method), 33	test_programmes_after() (radioco.apps.api.tests.test_programmes.TestProgrammesAPI) method), 13
test_get_announcers_and_profile() dioco.apps.users.tests.UserProfileMethodTests method), 47	test_programmes_before() (radioco.apps.api.tests.test_programmes.TestProgrammesAPI) method), 13
test_impossible_recurrence_after() dioco.apps.schedules.tests.test_reurrences.RecurrenceTests method), 39	test_programmes_between() (radioco.apps.api.tests.test_programmes.TestProgrammesAPI) method), 13
test_impossible_recurrence_before() dioco.apps.schedules.tests.test_reurrences.RecurrenceTests method), 39	test_programmes_delete() (radioco.apps.api.tests.test_programmes.TestNotAllowedMethodsProgrammesAPI) method), 13
test_incorrect_transmission_queries() dioco.apps.api.tests.test_schedules.TestTransmissionAPI method), 14	test_programmes_get_all() (radioco.apps.api.tests.test_programmes.TestProgrammesAPI) method), 13
test_index() (radioco.apps.radioco.tests.tests.RadioIntegrationTests) method), 35	test_programmes_post() (radioco.apps.api.tests.test_programmes.TestNotAllowedMethodsProgrammesAPI) method), 13
test_issue_date() (radioco.apps.programmes.tests.EpisodeManagerTests) method), 32	test_programmes_put() (radioco.apps.api.tests.test_programmes.TestNotAllowedMethodsProgrammesAPI) method), 13
test_last() (radioco.apps.programmes.tests.EpisodeManagerTests) method), 32	test_programmes_range_episodes() (radioco.apps.schedules.tests.test_schedules.ScheduleUtilsTests) method), 40
test_last_date_is_equals_before_limit() dioco.apps.schedules.tests.test_reurrences.RecurrenceTests method), 39	test_programmes_range_episodes_new_schedule() (radioco.apps.schedules.tests.test_schedules.ScheduleUtilsTests) method), 40
test_last_none() (radioco.apps.programmes.tests.EpisodeManagerTests) method), 32	test_programmes_range_only_non_emited_episodes() (radioco.apps.schedules.tests.test_schedules.ScheduleUtilsTests) method), 40
test_model_manager() dioco.apps.programmes.tests.EpisodeModelTests method), 33	
test_move_schedule() dioco.apps.api.tests.test_fullcalendar.TestFullCalendarAPI method), 12	
test_move_schedule_in_tz() dioco.apps.api.tests.test_fullcalendar.TestFullCalendarAPI method), 12	
test_move_schedule_with_schedules() dioco.apps.api.tests.test_fullcalendar.TestFullCalendarAPI method), 12	
test_move_schedule_with_schedules_in_tz() dioco.apps.api.tests.test_fullcalendar.TestFullCalendarAPI method), 12	

dioco.apps.schedules.tests.test_schedules.ScheduleModelTests	test_schedules_get_by_programme()	(ra-	dioco.apps.schedules.tests.test_schedules.ScheduleModelTests	method), 40
test_recording_schedules()	(ra-	dioco.apps.api.tests.test_schedules.TestSchedulesAPI	method), 14	
dioco.apps.api.tests.test_recorder.TestProgrammesAPI	test_schedules_get_by_type()	(ra-	dioco.apps.api.tests.test_schedules.TestSchedulesAPI	method), 14
test_recurrence_rules()	(ra-	dioco.apps.schedules.tests.test_schedules.ScheduleModelTests	method), 14	
dioco.apps.schedules.tests.test_schedules.ScheduleModelTests	test_schedules_post()	(ra-	dioco.apps.api.tests.test_fullcalendar.TestFullCalendarApi	method), 40
test_runtime() (radioco.apps.programmes.tests.EpisodeModelTests	method), 13			
method), 33	test_schedules_post()	(ra-	dioco.apps.schedules.tests.test_schedules.ScheduleModelTests	method), 33
test_runtime() (radioco.apps.programmes.tests.ProgrammeModelTests	dioco.apps.api.tests.test_schedules.TestRestrictedMethodsSchedu			method), 33
method), 33	method), 14			
test_runtime() (radioco.apps.schedules.tests.test_schedules.ScheduleModelTests	test_schedules_post_in_tz()	(ra-	dioco.apps.api.tests.test_fullcalendar.TestFullCalendarApi	method), 40
method), 40	dioco.apps.api.tests.test_fullcalendar.TestFullCalendarApi			method), 13
test_runtime_is_zero()	(ra-	dioco.apps.programmes.tests.ProgrammeModelTests	test_schedules_put()	(ra-
method), 33	method), 14	dioco.apps.api.tests.test_schedules.TestRestrictedMethodsSchedu		method), 33
test_runtime_not_get()	(ra-	dioco.apps.programmes.tests.ProgrammeModelTests	test_season() (radioco.apps.programmes.tests.EpisodeManagerTests	method), 33
method), 33	method), 32			
test_runtime_not_set()	(ra-	dioco.apps.programmes.tests.ProgrammeModelTests	test_slug() (radioco.apps.programmes.tests.ProgrammeModelTests	method), 33
method), 33	method), 33			
test_runtime_not_set()	(ra-	dioco.apps.schedules.tests.test_schedules.TransmissionModelTests	test_slug() (radioco.apps.schedules.tests.test_schedules.TransmissionModelTests	method), 41
method), 40	method), 40	dioco.apps.schedules.tests.test_schedules.ScheduleModelTests	test_start() (radioco.apps.schedules.tests.test_schedules.TransmissionModelTests	method), 40
test_save() (radioco.apps.users.tests.UserProfileMethodTests	test_start() (radioco.apps.schedules.tests.test_schedules.TransmissionModelTests			method), 47
method), 47	method), 41			
test_save_programme()	(ra-	dioco.apps.programmes.tests.ProgrammeModelTests	test_start_date_is_before_start_limit() (ra-	dioco.apps.schedules.tests.test_reurrences.RecurrenceTests
method), 33	method), 39		method), 39	
test_save_rearrange_episodes()	(ra-	dioco.apps.schedules.tests.test_schedules.ScheduleModelTests	test_start_lt_calendar() (ra-	dioco.apps.schedules.tests.test_schedules.ScheduleModelTests
method), 40	method), 40		method), 40	
test_schedule() (radioco.apps.api.tests.test_api.TestSerializers	test_str() (radioco.apps.programmes.tests.EpisodeModelTests			method), 12
method), 12	method), 33			
test_schedules_delete()	(ra-	dioco.apps.api.tests.test_schedules.TestRestrictedMethodsSchedu	test_str() (radioco.apps.programmes.tests.ProgrammeModelTests	method), 14
method), 14	method), 39		method), 39	
test_schedules_get_all()	(ra-	dioco.apps.api.tests.test_schedules.TestSchedulesAPI	test_submit_recorder() (ra-	dioco.apps.api.tests.test_recorder.TestProgrammesAPI
method), 14	method), 14		method), 14	
test_schedules_get_by_calendar()	(ra-	dioco.apps.api.tests.test_schedules.TestSchedulesAPI	test_transform_dt_to_default_tz() (ra-	dioco.apps.schedules.tests.test_timezone.ScheduleModelTests
method), 14	method), 41		method), 41	
test_schedules_get_by_nonexisting_calendar()	(ra-	dioco.apps.api.tests.test_schedules.TestSchedulesAPI	test_transmission() (ra-	dioco.apps.api.tests.test_api.TestSerializers
method), 14	method), 12		method), 12	
test_schedules_get_by_nonexisting_programme()	(ra-	dioco.apps.api.tests.test_schedules.TestSchedulesAPI	test_transmission_before() (ra-	dioco.apps.api.tests.test_schedules.TestTransmissionAPI
method), 14	method), 14		method), 14	
test_schedules_get_by_nonexisting_type()	(ra-	dioco.apps.api.tests.test_schedules.TestSchedulesAPI	test_transmission_list_filter_non_active_calendar() (ra-	dioco.apps.api.tests.test_schedules.TestTransmissionAPI
method), 14	method), 14		method), 14	

method), 14

test_transmission_now() (radioco.apps.api.tests.test_schedules), 14

dioco.apps.api.tests.test_radiocom_api.TestTransmissionSerializers (class in radioco.apps.api.tests.test_api), method), 13

test_transmission_now() (radioco.apps.api.tests.test_schedules.TestTransmissionAPI (class in radioco.apps.api.tests.test_radiocom_api), method), 14

test_transmission_same_day() (radioco.apps.api.tests.test_schedules.TestTransmissionAPI (class in radioco.apps.api.tests.test_schedules), method), 14

test_transmissions_between() (radioco.apps.api.tests.test_schedules.TestTransmissionAPI (class in radioco.apps.api.tests.test_schedules), method), 14

dioco.apps.api.tests.test_schedules.TestTransmissionAPI (class in radioco.apps.api.tests.test_schedules), 14

TimezoneForm (class in radioco.apps.api.views), 19

test_transmissions_between() (radioco.apps.programmes.admin.OwnEpisodeIssueDateListFilter (class in radioco.apps.programmes.admin.views), method), 14

title (radioco.apps.programmes.admin.OwnEpisodeIssueDateListFilter (class in radioco.apps.programmes.admin.views), attribute), 29

test_transmissions_between_requesting_tz() (radioco.apps.api.tests.test_schedules.TestTransmissionAPI (class in radioco.apps.api.tests.test_schedules), method), 14

title (radioco.apps.programmes.admin.OwnEpisodeProgrammeListFilter (class in radioco.apps.programmes.admin.views), attribute), 29

test_transmissions_between_requesting_tz() (radioco.apps.api.tests.test_schedules.TestTransmissionAPI (class in radioco.apps.api.tests.test_schedules), method), 14

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

method), 30

test_transmissions_delete() (radioco.apps.api.tests.test_schedules.TestRestrictedMethodsScheduleAPI (class in radioco.apps.api.tests.test_schedules), method), 14

to_representation() (radioco.apps.api.tests.test_schedules.TestRestrictedMethodsScheduleAPI (class in radioco.apps.api.tests.test_schedules), method), 15

test_transmissions_delete() (radioco.apps.api.tests.test_schedules.TestRestrictedMethodsScheduleAPI (class in radioco.apps.api.tests.test_schedules), method), 14

to_representation() (radioco.apps.api.tests.test_schedules.TestTransmissionAPI (class in radioco.apps.api.tests.test_schedules), method), 15

test_transmissions_filter_calendar_nonexistend() (radioco.apps.api.tests.test_schedules.TestTransmissionAPI (class in radioco.apps.api.tests.test_schedules), method), 14

to_representation() (radioco.apps.api.tests.test_schedules.TestTransmissionAPI (class in radioco.apps.api.tests.test_schedules), method), 15

test_transmissions_post() (radioco.apps.api.tests.test_schedules.TestRestrictedMethodsScheduleAPI (class in radioco.apps.api.tests.test_schedules), method), 14

transform_datetime_tz() (in radioco.apps.programmes.feeds.ProgrammeFeed (class in radioco.apps.programmes.feeds.views), module radioco.tz_utils), 36

test_transmissions_post() (radioco.apps.api.tests.test_schedules.TestRestrictedMethodsScheduleAPI (class in radioco.apps.api.tests.test_schedules), method), 14

transform_dt_to_default_tz() (in radioco.apps.programmes.feeds.ProgrammeFeed (class in radioco.apps.programmes.feeds.views), module radioco.tz_utils), 36

test_transmissions_put() (radioco.apps.api.tests.test_schedules.TestRestrictedMethodsScheduleAPI (class in radioco.apps.api.tests.test_schedules), method), 14

to_representation() (radioco.apps.api.tests.test_schedules.TestRestrictedMethodsScheduleAPI (class in radioco.apps.api.tests.test_schedules), method), 44

test_unfinished() (radioco.apps.programmes.tests.EpisodeManagerTests (class in radioco.apps.programmes.tests.test_episodes), method), 33

Transmission_dt (radioco.apps.schedules.views.DeleteScheduleView (class in radioco.apps.schedules.views), attribute), 45

test_unfinished_none() (radioco.apps.programmes.tests.EpisodeManagerTests (class in radioco.apps.programmes.tests.test_episodes), method), 33

TransmissionForm (class in radioco.apps.api.views), 19

test_unfinished_none() (radioco.apps.programmes.tests.EpisodeManagerTests (class in radioco.apps.programmes.tests.test_episodes), method), 33

TransmissionModelTests (class in radioco.apps.schedules.tests.test_schedules), 44

test_unicode() (radioco.apps.schedules.tests.test_schedules.ScheduleModelTests (class in radioco.apps.schedules.tests.test_schedules), method), 40

TransmissionOperationViewSet (class in radioco.apps.schedules.views), 19

test_when_date_doesnt_exist_on_february() (radioco.apps.schedules.tests.test_reurrences.RecurrentTransmissionSerializer (class in radioco.apps.schedules.tests.test_reurrences), method), 39

TrueFastTransmissionSerializer (class in radioco.apps.schedules.tests.test_reurrences), 16

TestApi (class in radioco.apps.api.tests.test_api), 12

TransmissionSerializerLight (class in radioco.apps.schedules.tests.test_reurrences), 17

TestDataMixin (class in radioco.apps.radioco.test_utils), 35

TransmissionViewSet (class in radioco.apps.api.views), 20

TestFullCalendarApi (class in radioco.apps.api.tests.test_fullcalendar), 12

TestNotAllowedMethodsProgrammesAPI (class in radioco.apps.api.tests.test_programmes), 13

TestProgramme (class in radioco.apps.api.tests.test_radiocom_api), 13

TestProgrammesAPI (class in radioco.apps.api.tests.test_programmes), 13

TestProgrammesAPI (class in radioco.apps.api.tests.test_recorder), 14

TestRestrictedMethodsScheduleAPI (class in radioco.apps.api.tests.test_schedules), 14

U

unfinished() (radioco.apps.programmes.models.EpisodeManager (class in radioco.apps.programmes.models), static method), 31

update() (radioco.apps.api.views.TransmissionOperationViewSet (class in radioco.apps.schedules.views), method), 20

update_schedule_if_dt_has_changed() (in radioco.apps.programmes.models), 32

update_schedule_performance() (in radioco.apps.programmes.models), 32

UpdateOnlyModelViewSet (class in radioco.apps.api.viewsets), 20

user (radioco.apps.users.models.UserProfile attribute), 47
user_logout() (in module radioco.apps.radioco.views), 36
UserProfile (class in radioco.apps.users.models), 47
UserProfile.DoesNotExist, 47
UserProfile.MultipleObjectsReturned, 47
userprofile_detail() (in module radioco.apps.users.views),
47
UserProfileAdmin (class in radioco.apps.users.admin), 46
UserProfileInline (class in radioco.apps.users.admin), 46
UserProfileMethodTests (class in radioco.apps.users.tests), 47
UsersView (class in radioco.apps.users.views), 47
UtilsTest (class in radioco.apps.radioco.tests.tests), 35

V

validate() (radioco.apps.api.serializers.ScheduleSerializer
method), 16
validate() (radioco.apps.api.serializers.TransmissionSerializerLight
method), 17

X

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