# django-applepodcast Documentation

Release 0.1

**Richard Cornish** 

# Contents

1	Install	3
2	Usage	5
3	Contents	7
	3.1 Install	7
	3.2 Usage	8
	3.3 Settings	11
	3.4 Feed	12
	3.5 Admin	16
	3.6 Documentation	30
	3.7 Tests	31
	3.8 Resources	31
4	Indices and tables	33
Pv	ython Module Index	35

**Django Apple Podcast** is a Django podcast application optimized for Apple Podcasts. Formerly *Django iTunes Podcast*.

- Package distribution
- Code repository
- Documentation
- Tests

An online demo also exists.

- Online demo
- Online demo code repository

Contents 1

2 Contents

# CHAPTER 1

Install

```
$ pipenv install django-applepodcast
```

Add to settings.py.

```
INSTALLED_APPS = [
    # ...
    'podcast',
]
```

Add to urls.py.

```
from django.urls import include, path

urlpatterns = [
    # ...
    path('podcast/', include('podcast.urls')),
]
```

Migrate the database.

```
$ pipenv run python manage.py migrate
```

Load the fixtures.

```
$ pipenv run python manage.py loaddata podcast_category.json
```

4 Chapter 1. Install

# CHAPTER 2

Usage

### Run the local server.

\$ pipenv run python manage.py runserver

Visit either the show view or the admin.

- http://127.0.0.1:8000/podcast/
- http://127.0.0.1:8000/admin/podcast/

6 Chapter 2. Usage

# CHAPTER 3

Contents

### 3.1 Install

Install with Pipenv.

```
$ pipenv install django django-applepodcast
```

After creating a project, add podcast to INSTALLED\_APPS in settings.py.

```
INSTALLED_APPS = [
# ...
'podcast',
]
```

Because the app is primarily model driven, you will want to expose the URL of the show's feed for submission to Apple Podcasts. Add the URL conf to urls.py.

```
from django.urls import include, path

urlpatterns = [
    # ...
    path('podcast/', include('podcast.urls')),
]
```

If you're on Django 1.11, 1.10, or 1.9, use the older, regex-based syntax instead.

```
from django.conf.urls import include, url

urlpatterns = [
    # ...
    url(r'^podcast/', include('podcast.urls')),
]
```

If you're on Django 1.8, you will additionally need to add the namespace keyword argument to the include () method manually because the convenient app\_name attribute in urls.py wasn't added until Django 1.9.

```
from django.conf.urls import include, url

urlpatterns = [
    # ...
    url(r'^podcast/', include('podcast.urls', namespace='podcast')),
]
```

Add the models to your project by migrating the database.

```
$ pipenv run python manage.py migrate
```

Add the default Apple Podcasts categories by loading the fixtures.

```
$ pipenv run python manage.py loaddata podcast_category.json
```

### 3.2 Usage

### 3.2.1 Structure

The best way to understand the app is to simply browse the online demo or locally after adding a show and a few episodes in the admin. But for reference, assuming you chose podcast as the base of your URL conf, the structure is:

URL	URL name	View	Mod	elTemplate	Context	Absolute URLs	
/podcast/	podcast:sl	qwo_dbetsti.l	Show	podcast/	show,	{{ show.	
		views.		show_detail	.episode	_ges <u>t</u> absolute_url	
		ShowDetailVie	W	html		} }	
/podcast/	podcast:sl	pwodeest.	Show	7		{{ show.	
feed/		views.				get_absolute_feed_ur	1
		ShowFeed				} }	
/podcast/	podcast:ep	ipsoodbea <u>s</u> de.tail	Epis	opobedcast/	episode	{{ episode.	
<episode-slug< td=""><td>&gt;/</td><td>views.</td><td></td><td>episode_det</td><td>ail.</td><td>get_absolute_url</td><td></td></episode-slug<>	>/	views.		episode_det	ail.	get_absolute_url	
		EpisodeDetail	View	html		} }	
/podcast/	podcast:ep	ipsoodbea_soto.wnloa	d Epis	ode		{{ episode.	
<episode-slug< td=""><td>&gt;/</td><td>views.</td><td></td><td></td><td></td><td>get_absolute_downloa</td><td>d_url</td></episode-slug<>	>/	views.				get_absolute_downloa	d_url
download/		EpisodeDownlo	adVi	ew		} }	

And if you chose podcasts and set PODCAST\_SINGULAR = False to display multiple shows:

URL	URL name	View	Mod	elTemplate	Con-	Absolute URLs	
					text		
/podcasts/	podcast:s	hpw <u>d</u> dàst.	Show	podcast/	show_li	st	
		views.		show_list.			
		ShowListView	7	html			
/podcasts/	podcast:s	h <b>pw<u>d</u>desai</b> l	Show	podcast/	show,	{{ show.	
<show-slug>/</show-slug>		views.		show_detai	lepisode	<u>gleitst</u> absolute_u	rl
		ShowDetailVi	ew	html		} }	
/podcasts/	podcast:s	hpw <u>d</u> feed.	Show	J		{{ show.	
<show-slug>/</show-slug>		views.				get_absolute_f	eed_url
feed/		ShowFeed				} }	
/podcasts/	podcast:e	p <b>ṗodde<u>s</u>tl</b> etail	Epis	opobedcast/	episode	e {{ episode.	
<show-slug>/</show-slug>		views.		episode_de	tail.	get_absolute_u	rl
<episode-slug>/</episode-slug>		EpisodeDetai	lVie	v html		} }	
/podcasts/	podcast:e	p <b>psdde<u>s</u>do</b> wnlo	aĦpis	ode		{{ episode.	
<show-slug>/</show-slug>		views.				get_absolute_d	ownload_url
<episode-slug>/</episode-slug>		EpisodeDownl	oadV:	ew		} }	
download/							

#### 3.2.2 Enclosures

The show feed is a subclass of Django's Rss201rev2Feed feed generator class, which prohibits the use of more than one enclosure. Although Django's syndication documentation doesn't directly state it, the use of multiple enclosures are meant for only subclasses of Django's Atom1Feed feed generator class.

The RSS Advisory Board states:

Support for the enclosure element in RSS software varies significantly because of disagreement over whether the specification permits more than one enclosure per item. Although the author intended to permit no more than one enclosure in each item, this limit is not explicit in the specification.

For best support in the widest number of aggregators, an item should not contain more than one enclosure.

Therefore, enclosures are modeled as OneToOneField's off of episodes, limiting episodes to one and only one enclosure.

Apple Podcasts does not host enclosure files; it is the responsibility of the developer to host them. Because an enclosure's file is a FileField, files are uploaded to your MEDIA\_ROOT setting. If you haven't already, your urls.py should include patterns for interfacing with files in local development.

```
from django.conf import settings
from django.conf.urls.static import static
from django.contrib.staticfiles.urls import staticfiles_urlpatterns
from django.urls import include, path

urlpatterns = [
    # ...
    path('podcast/', include('podcast.urls')),
]

# Static/media for local development
if getattr(settings, 'DEBUG', False):
    urlpatterns += staticfiles_urlpatterns()
    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

3.2. Usage 9

Although file management in production is out of scope of this documentation, consider using Amazon Web Service's S3 (Simple Storage Service) to host files and Django Storages and Boto (and, if using Python 3, Boto 3), to interface with them.

The code repository of the online demo is worth looking at for a complete implementation, especially the settings file if you expect to use AWS's S3.

#### 3.2.3 Protocols

Depending on the operating system and/or web browser, it's possible to immediately subscribe a user to a podcast in iTunes or the iOS Podcasts app by using itpc:// or feed:// protcol-based URLs rather than the standard http:// or https://.

The app comes with the {% show\_url %} template tag to help create these URLs.

```
{% load podcast_tags %}
{% show_url protocol='itpc' url=show.get_absolute_feed_url %}">
```

#### Result:

```
itpc://127.0.0.1:8000/podcast/feed/
```

Beware that these URLs are purely interaction based; you would not be required to submit the show feed to Apple Podcasts, but you would also not be able to track users' behavior in Podcasts Connect. For this reason, you're probably better off in a traditional submission to Apple Podcasts, saving your new URL in the Show model, and using the show apple variable in your template.

```
{{ show.apple }}
```

#### 3.2.4 Submission

The show feed URL is:

```
/podcast/feed/
```

If you have multiple shows, each respective show feed URL is:

```
/podcasts/<show-slug>/feed/
```

Submit the show feed to Podcasts Connect.

### 3.2.5 Badges

After Apple Podcasts approves your podcast, feel free to use the "Listen on Apple Podcasts" badge or icon, the U.S. versions of which are included as minified static files. You can also download them from the Apple Podcasts Identity Guidelines. The SVGs were minified with SVGO.

#### **Badge**

**Icon** 

```
{% load i18n static %}
<img src="{% static 'podcast/img/icon.svg' %}" alt="{% trans 'Listen on Apple Podcasts
→' %}">
```

Load the white or black icons similarly.

```
{% load i18n static %}
<img src="{% static 'podcast/img/icon-black.svg' %}" alt="{% trans 'Listen on Apple_
→Podcasts' %}">
```

### 3.3 Settings

The app offers several settings. By default, they are:

```
PODCAST_SINGULAR = True

PODCAST_ID = 1

PODCAST_EPISODE_LIMIT = None

PODCAST_NO_ARTWORK = 'podcast/img/no_artwork.png'

PODCAST_PAGINATE_BY = 10

PODCAST_ALLOWED_TAGS = ['p', 'ol', 'ul', 'li', 'a', 'i', 'em', 'b', 'strong']
```

#### 3.3.1 PODCAST SINGULAR

A boolean indicating display of multiple podcast shows.

The app displays a single show by default. If you would like to display multiple shows, set the PODCAST\_SINGULAR variable in settings.py.

```
PODCAST_SINGULAR = False
```

If you have multiple shows, you might want to edit the URL pattern in urls.py from podcast/ to podcasts/, although the difference is purely cosmetic.

3.3. Settings

```
from django.urls import include, path

urlpatterns = [
    # ...
    path('podcasts/', include('podcast.urls')),
]
```

### 3.3.2 PODCAST ID

An integer indicating the primary key of the show to display; used when PODCAST\_SINGULAR is True. Concept modeled after the SITE\_ID setting used in the Sites application.

The app displays the first show by default.

#### 3.3.3 PODCAST EPISODE LIMIT

An integer indicating the number of episodes to display in a show feed or None to display all episodes. Formerly the limit was an arbitrary 50 episodes.

The app displays all episodes in a show feed by default.

#### 3.3.4 PODCAST NO ARTWORK

A string indicating the path to an image used when artwork is lacking; used for shows, episodes, and video enclosures.

Although the path can be customized in the setting, you're probably better off overriding the image look up at the project level; that is, creating a new image at myproject/static/podcast/img/no\_artwork.png.

#### 3.3.5 PODCAST PAGINATE BY

An integer indicating how many items to display in a list view or in a detail view of related objects; used for shows and episodes.

### 3.3.6 PODCAST\_ALLOWED\_TAGS

A list indicating which HTML tags are allowed for display in output; used for show summaries and episode notes. The database can store HTML, but tags not specified in the list are stripped out, and the remaining output is wrapped in <! [CDATA[...]] > tags. Uses the Bleach Python package.

The list includes HTML tags specified in the iOS 11 Apple Podcasts update by default. The list also includes the conspicuously absent and <strong> tags, which appear to be an oversight on Apple's part because , , and <b> are present in the original list. Note that the specification confusingly specifies a subset of fewer tags: , , , and <a>..

#### 3.4 Feed

### **3.4.1 Sample**

The show feed is optimized for submission to the Apple Podcasts by adding additional Apple Podcast-specific tags.

The following is the direct output of a show feed following the RSS feed sample in the Podcasts Connect documentation as closely as possible.

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0" xmlns:itunes="http://www.itunes.com/dtds/podcast-1.0.dtd",</pre>
→xmlns:atom="http://www.w3.org/2005/Atom">
    <channel>
        <title>All About Everything</title>
        <link>http://testserver/podcast/</link>
        <description>All About Everything is a show about everything. Each week we_
→dive into any subject known to man and talk about it as much as we can. Look for
→our podcast in the Podcasts app or in the iTunes Store</description>
        <atom:link rel="self" href="http://testserver/podcast/feed/" />
        <language>en-us</language>
        <lastBuildDate>Thu, 10 Mar 2017 22:15:00 +0000</lastBuildDate>
        <copyright>&#x2117; &amp; &#xA9; 2017 John Doe &amp; Family</copyright>
        <itunes:subtitle><![CDATA[A show about everything]]></itunes:subtitle>
        <itunes:summary><![CDATA[All About Everything is a show about everything...</pre>
→Each week we dive into any subject known to man and talk about it as much as we can.
→ Look for our podcast in the Podcasts app or in the iTunes Store]]></itunes:summary>
        <itunes:author>John Doe</itunes:author>
        <itunes:owner>
            <itunes:name>John Doe</itunes:name>
            <itunes:email>john.doe@example.com</itunes:email>
        <itunes:image href="http://testserver/podcast/tests/static/everything/</pre>
→AllAboutEverything.jpg" />
        <itunes:category text="Arts">
            <itunes:category text="Food" />
        </itunes:category>
        <itunes:category text="TV &amp; Film" />
        <itunes:category text="Technology">
            <itunes:category text="Gadgets" />
        </itunes:category>
        <itunes:explicit>no</itunes:explicit>
        <item>
            <title>Red, Whine, &amp; Blue</title>
            <link>http://testserver/podcast/red-whine-blue/</link>
            <description>This week we talk about surviving in a Red state if you are,
→a Blue person. Or vice versa.</description>
            <pubDate>Thu, 10 Mar 2016 22:15:00 +0000</pubDate>
            <guid>http://testserver/podcast/tests/static/everything/
→AllAboutEverythingEpisode4.mp3</guid>
            <enclosure length="75232" type="audio/mpeg" url="http://testserver/</pre>
→podcast/tests/static/everything/AllAboutEverythingEpisode4.mp3" />
            <itunes:subtitle><![CDATA[Red + Blue != Purple]]></itunes:subtitle>
            <itunes:summary><![CDATA[This week we talk about surviving in a Red state.]</pre>
→ if you are a Blue person. Or vice versa.]]></itunes:summary>
            <itunes:author>Various
            <itunes:image href="http://testserver/podcast/tests/static/everything/</pre>
→AllAboutEverything/Episode4.jpg" />
            <itunes:explicit>no</itunes:explicit>
            <itunes:duration>00:03</itunes:duration>
        </item>
        <item>
            <title>The Best Chili</title>
            <link>http://testserver/podcast/best-chili/</link>
            <description>This week we talk about the best Chili in the world. Which,

→chili is better?</description>

                                                                          (continues on next page)
```

3.4. Feed 13

(continued from previous page)

```
<pubDate>Thu, 10 Mar 2016 09:00:00 +0000</pubDate>
           <guid>http://testserver/podcast/tests/static/everything/
→AllAboutEverythingEpisode2.m4v</guid>
           <enclosure length="25725" type="video/x-m4v" url="http://testserver/</pre>
→podcast/tests/static/everything/AllAboutEverythingEpisode2.m4v" />
           <itunes:subtitle><![CDATA[Jane and Eric]]></itunes:subtitle>
           <itunes:summary><![CDATA[This week we talk about the best Chili in the...</pre>
→world. Which chili is better?]]></itunes:summary>
           <itunes:author>Jane Doe</itunes:author>
           <itunes:image href="http://testserver/podcast/tests/static/everything/</pre>
→AllAboutEverything/Episode3.jpg" />
           <itunes:explicit>no</itunes:explicit>
           <itunes:duration>00:02</itunes:duration>
           <itunes:isClosedCaptioned>ves</itunes:isClosedCaptioned>
       </item>
       <item>
           <title>Socket Wrench Shootout</title>
           <link>http://testserver/podcast/socket-wrench-shootout/</link>
           <description>This week we talk about metric vs. Old English socket_
→wrenches. Which one is better? Do you really need both? Get all of your answers.
→here.</description>
           <pubDate>Wed, 09 Mar 2016 18:00:00 +0000</pubDate>
           <guid>http://testserver/podcast/tests/static/everything/
→AllAboutEverythingEpisode2.mp4</guid>
           <enclosure length="28355" type="video/mp4" url="http://testserver/podcast/</pre>
→tests/static/everything/AllAboutEverythingEpisode2.mp4" />
           <itunes:subtitle><![CDATA[Comparing socket wrenches is fun!]]>
→itunes:subtitle>
           <itunes:summary><![CDATA[This week we talk about metric vs. Old English]</pre>
→ socket wrenches. Which one is better? Do you really need both? Get all of your_
→ answers here.]]></itunes:summary>
           <itunes:author>Jane Doe</itunes:author>
           <itunes:image href="http://testserver/podcast/tests/static/everything/"</pre>
→AllAboutEverything/Episode2.jpg" />
           <itunes:explicit>no</itunes:explicit>
           <itunes:duration>00:03</itunes:duration>
       </item>
       <item>
           <title>Shake Shake Your Spices</title>
           <link>http://testserver/podcast/shake-shake-your-spices/</link>
           <description>This week we talk about &lt;a href="https://itunes/apple.com/
→us/book/antique-trader-salt-pepper/id429691295?mt=11">salt and pepper shakers<
→/a>, comparing and contrasting pour rates, construction materials, and overall_
→aesthetics. Come and join the party!</description>
           <pubDate>Tue, 08 Mar 2016 12:00:00 +0000</pubDate>
            <quid>http://testserver/podcast/tests/static/everything/
→AllAboutEverythingEpisode3.m4a</quid>
           <enclosure length="46862" type="audio/x-m4a" url="http://testserver/</pre>
→podcast/tests/static/everything/AllAboutEverythingEpisode3.m4a" />
           <itunes:subtitle><![CDATA[A short primer on table spices]]>
→itunes:subtitle>
           <itunes:summary><![CDATA[This week we talk about <a href="https://itunes/")</pre>
→apple.com/us/book/antique-trader-salt-pepper/id429691295?mt=11">salt and pepper.
→ shakers </a>, comparing and contrasting pour rates, construction materials, and
→overall aesthetics. Come and join the party!||></itunes:summary>
           <itunes:author>John Doe</itunes:author>
           <itunes:image href="http://testserver/podcast/tests/static/everything/</pre>
                                                                         (continues on next page)
→AllAboutEverything/Episodel.jpg" />
```

(continued from previous page)

### 3.4.2 Sample differences

Although every effort was made to recreate the RSS feed sample on Podcasts Connnect as closely as possible, the limitations of the way in which Django creates feeds and the occassional stray error in the feed sample itself required small changes:

- The RssFeed class in Django's deep syndication class hierarchy adds an <atom:link> to the <channel> element that would require a significant code duplication and rewrite to eliminate. It does not affect Apple Podcasts compatibility and thus remains in the show feed.
- The <atom:link> previously mentioned can only exist in a correponding XML namespace; i.e. the attribute xmlns:atom="http://www.w3.org/2005/Atom" in the <rss> element. The attribute could be easily removed, but would prevent the feed from achieving XML validation. The Atom XML namespace thus remains in the show feed.
- The RssFeed class adds a <lastBuildDate> to the <channel> element that corresponds to the <pubDate> of the latest <item>. Due to Django's deep syndication class hierarchy, it remains in the show feed.
- In the RSS feed sample, the <copyright> element contains a year of 2014. The sample is replaced with the current year, at the time of this writing, 2017.
- In the RSS feed sample, <itunes:summary> tag in the "Shake Shake Your Spices" episode has an errant space in its <! [CDATA[...]] > tag. The sample displays <! [CDATA[...]] >. The show feed removes the errant space.
- In the RSS feed sample, the domain in URLs is www.example.com or example.com. Django's testing framework uses the server name testserver. The feed test replaces www.example.com with testserver.
- In the RSS feed sample, the absolute URL of the show is /podcasts/everything/index.html. In the interest of clean URLs, the feed test removes index.html.
- In the RSS feed sample, only instances of <itunes:summary> or <itunes:subtitle> that have HTML contain <! [CDATA[...]]> tags to escape the HTML. Rather than conditionally insert <! [CDATA[...]]> tags, they are inserted in all instances of <itunes:summary> and <itunes:subtitle>.
- In the RSS feed sample, the enclosure url of an <item> is often different from the <guid>, e.g. http://example.com/podcasts/everything/AllAboutEverythingEpisode3.m4a vs. http://example.com/podcasts/archive/aae20140615.m4a. The <guid> of an <item> is normalized to return the enclosure URL and eliminate a competing, arbitrary URL.
- In the RSS feed sample, the (fake) enclosure files have accompanying fake values in <itunes:duration> elements. The app automatically reads the duration of media files using the Python Mutagen package, and their durations are not subject to manual editing.
- In the RSS feed sample, the enclosure length of an <item> is similarly determined by automatically reading the enclosure size of the file.
- In the RSS feed sample, the paths to media files were changed to reflect a more typical Django file path.

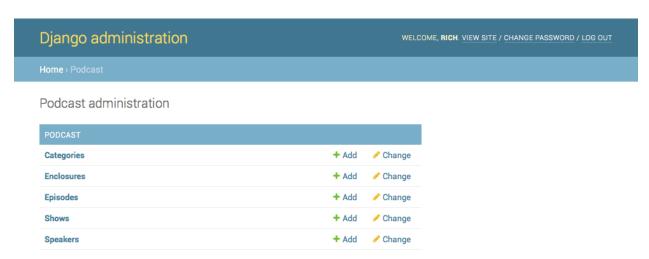
3.4. Feed 15

- In the RSS feed sample, <item> elements omit <link> and <description> elements. While technically valid, Django encourages its use by automatically querying for an item's absolute URL, and thus each item's <link> and <description> are preserved.
- In the RSS feed sample, the <item> elements contain <pubDate> values whose time zones are inconsistent: GMT (which is obsolete), EST, -0700, and +3000 (which should be +0300). Because Django defines TIME\_ZONE at the project level in settings, it's impossible to display datetimes in the show feed with different time zones. For example, given a datetime 2016-03-11T01:15:00+0300 (which might be, say, 'Europe/Moscow'), a setting of TIME\_ZONE = 'UTC' would ultimately result in a display of Thu, 10 Mar 2016 22:15:00 +0000, that is, moving three hours backward to achieve UTC, which would be around 10 p.m. the prior evening. All values of <pubDate> elements have been converted to their UTC-time zone equivalents.
- In the RSS feed sample, elements that correspond to boolean values are inconsistently capitalized. The values of <itunes:explicit> elements are no, but the value of <itunes:isClosedCaptioned> is Yes. The sample was changed to yes.
- In the RSS feed, the episode "Shake Shake Shake Your Spices" has a <description> and <itunes:summary> element whose value contains a malformed URL, i.e. https://itunes/apple.com. The error has been preserved.
- In the RSS feed, the episode "Red, Whine, & Blue" is missing a space after the first comma. The error has been preserved.
- The show feed and RSS feed sample only compare semantic differences, i.e. parsed content, and not syntax differences, i.e. various orderings of elements, capitalization, orderings of attributes, and spaces, etc. Django's assertXMLEqual is used to assert equality.

### 3.5 Admin

The data in the screenshots of the admin mimics the RSS feed sample of the Podcasts Connect documentation.

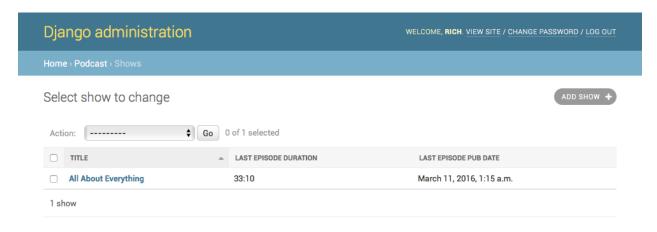
### 3.5.1 App



3.5. Admin 17

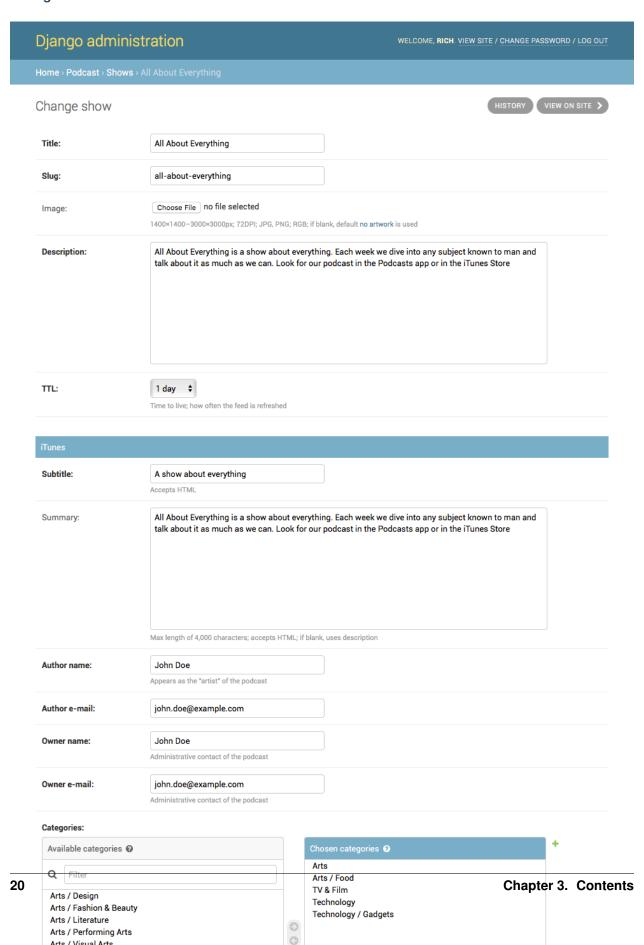
### 3.5.2 Shows

### **List view**



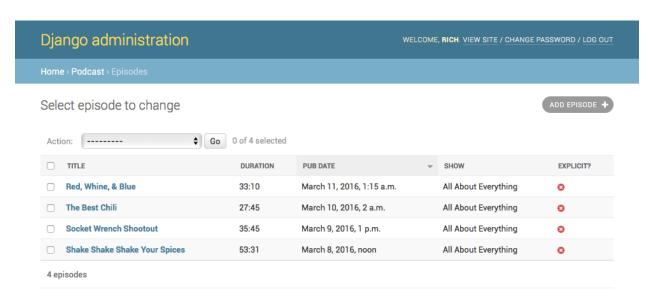
3.5. Admin 19

### Change view



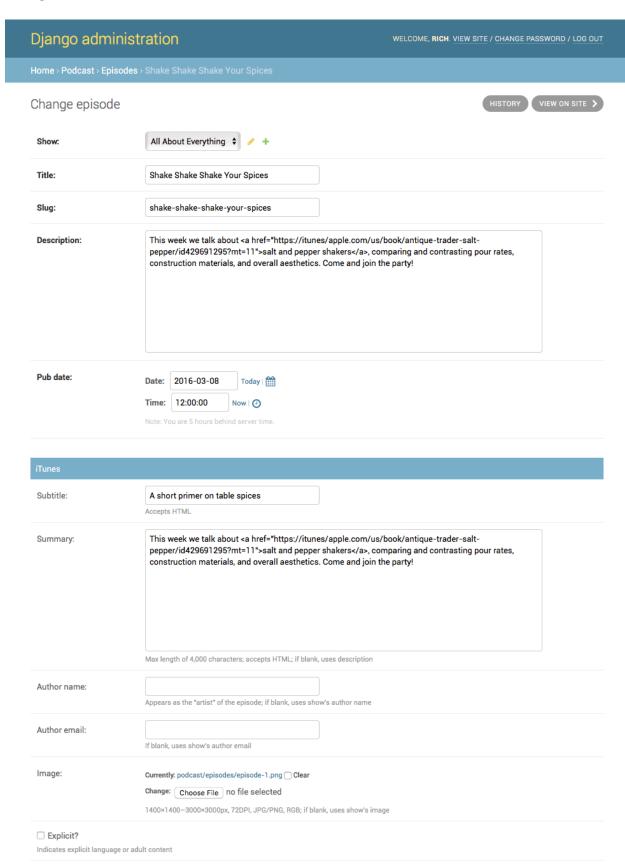
### 3.5.3 Episodes

### List view



3.5. Admin 21

### Change view

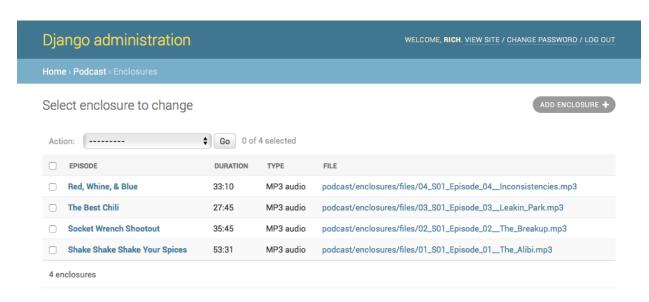


3.5. Admin Prevents episode from appearing on the iTunes Store

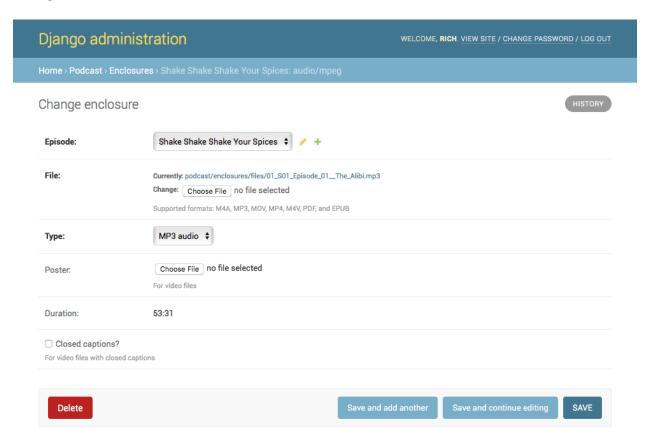
23

### 3.5.4 Enclosures

### List view



### Change view



3.5. Admin 25

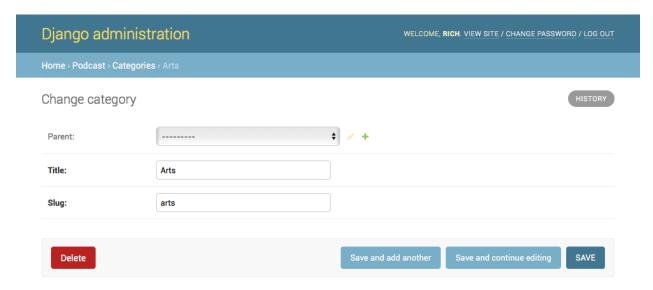
26 Chapter 3. Contents

## 3.5.5 Categories

### List view

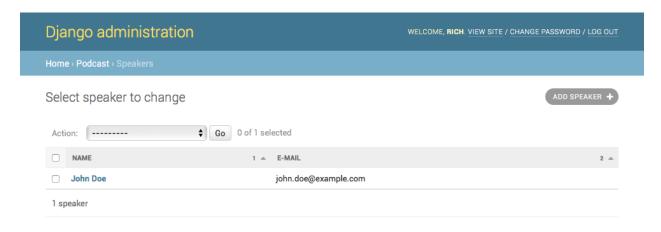
	Dja	ingo administration	WELCOME, <b>RICH</b> . VIEW SITE / CHANGE PASSWORD / LOG OUT
	Home	e > Podcast > Categories	
	Sele	ect category to change	ADD CATEGORY +
	Acti	on: Go 0 of 67 selected	
		FULL TITLE	PARENT
		Arts	
		Arts / Design	Arts
		Arts / Fashion & Beauty	Arts
		Arts / Food	Arts
		Arts / Literature	Arts
		Arts / Performing Arts	Arts
		Arts / Visual Arts	Arts
		Business	
		Business / Business News	Business
		Business / Careers	Business
		Business / Investing	Business
		Business / Management & Marketing	Business
		Business / Shopping	Business
		Comedy	-
		Education	-
		Education / Educational Technology	Education
		Education / Higher Education	Education
		Education / K-12	Education
		Education / Language Courses	Education
		Education / Training	Education
		Games & Hobbies	-
		Games & Hobbies / Automotive	Games & Hobbies
		Games & Hobbies / Aviation	Games & Hobbies
		Games & Hobbies / Hobbies	Games & Hobbies
		Games & Hobbies / Other Games	Games & Hobbies
		Games & Hobbies / Video Games	Games & Hobbies
		Government & Organizations	-
		Government & Organizations / Local	Government & Organizations
		Government & Organizations / National	Government & Organizations
		Government & Organizations / Non-Profit	Government & Organizations
	0	Government & Organizations / Regional	Government & Organizations
3.5.	Ad	Health Imin	2
J. <b>J</b> .		Health / Alternative Health	Health
		Health / Fitness & Nutrition	Health
		Health / Self-Help	Health

### Change view



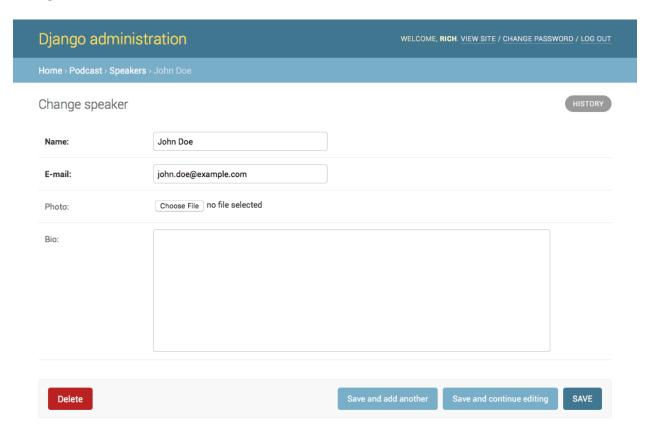
### 3.5.6 Speakers

### **List view**



3.5. Admin 29

### Change view



### 3.6 Documentation

Full documentation is available online.

However, you can also build the documentation from source.

Clone the code repository.

```
$ git clone git@github.com:richardcornish/django-applepodcast.git
$ cd django-applepodcast/
```

Install Sphinx, sphinx-autobuild, and sphinx\_rtd\_theme.

```
$ pipenv install sphinx sphinx-autobuild sphinx_rtd_theme --three
```

Create an HTML build.

```
$ (cd docs/ && make html)
```

Or use sphinx-autobuild to watch for live changes.

```
$ sphinx-autobuild docs/ docs/_build_html
```

Open 127.0.0.1:8000.

### 3.7 Tests

Continuous integration test results are available online.

However, you can also test the source code. Note that because the app stores media files, you will have to change to the directory above the source code directory before testing, typically site-packages.

```
$ cd ~/.virtualenvs/myenv/lib/python3.6/site-packages/
$ django-admin test podcast.tests --settings="podcast.tests.settings"

Creating test database for alias 'default'...

Ran 1 test in 0.119s

OK

Destroying test database for alias 'default'...
```

A bundled settings file allows you to test the code without even creating a Django project.

### 3.8 Resources

Some of the helpful resources used to create the app.

#### 3.8.1 Documentation

· Django's syndication feed framework

### 3.8.2 Specifications

- RSS 2.0 specification
- RSS tags for Podcasts Connect
- Apple iOS 11 specification update
- Google Play Music
- ID3 standards
- · iTunes metadata

### 3.8.3 Validators

- W3C Feed Validation Service
- Podba.se Podcast Validator
- · Cast Feed Validator
- · Feed Validator

3.7. Tests 31

### 3.8.4 Resources

- Apple iTunes Connect Resources and Help
- Apple Podcasts Connect Help

### 3.8.5 Marketing

• Apple Podcasts Identity Guidelines

### 3.8.6 Submission

- Apple Podcasts Connect
- Google Play Music Podcast Portal

# $\mathsf{CHAPTER}\, 4$

# Indices and tables

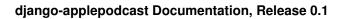
- genindex
- modindex
- search

django-applepodcast Documentation, R	elease 0.1	

# Python Module Index

## р

podcast, 1



36 Python Module Index

I	n	d	ΔY
		u	

### Р

podcast (module), 1