
Wagtail Podcast Documentation

Release 0.1

Daniel F. Meyer

Oct 03, 2018

Contents:

1	Getting Started	3
1.1	Installation	3
1.2	Usage	3
1.3	Analytics	4
1.4	Customising	4
1.5	Caveats	4
2	Podcast Options	5
3	Customising	7
4	Internationalisation	9
4.1	Languages	9
4.2	Caveats	9
5	API Documentation	11
6	Proposed Roadmap	15
7	Changelog	17
8	Thanks	19
9	Indices and tables	21
	Python Module Index	23

This is a very basic podcast app that integrates with Wagtail. It allows a person to upload a mp3 and then play it on the website. It generates a rss/xml feed that you can submit to Itunes, Google and/or Blubrry.

This app is simple, and not feature rich. It does the basics and thats it. I will probably add more features and options to it as time permits, but only if requests are made.

It is also not production ready as there are **no tests currently written** for it. I did put this up on a site and they have been using it extensively and haven't had any reported issues. That said, *apps without tests should always be treated sceptically*.

I hope to write tests in a month or so when I get a bit of free time. I also hope to add a bunch of features in the near future. See: [Proposed Roadmap](#)

Please report any errors you encounter. I will try resolve them quickly and then add tests for them as things come up so it doesn't reoccur. Please visit [wagtail_podcast git](#) to make pull requests or log issues etc. Documentation is at [readthedocs.io: wagtail_podcast documentation](#)

1.1 Installation

To install run `pip install wagtail_podcast`

It should automatically install mutagen; however, if it doesn't then you will need to install it manually with: `pip install mutagen`

Remember to add `wagtail_podcast` to your installed apps in `settings.py` i.e.

```
INSTALLED_APPS = [  
    ...  
    'wagtail_podcast',  
]
```

Requirements:

```
python3  
mutagen  
wagtail  
django  
django-social-share
```

I'm not quite sure how far back this app works; however, it should work going back quite far. It's currently tested on Python3 with Wagtail >2 and Django >2 on openSUSE. It should work on all platforms and shouldn't break anytime soon. Let me know if you have a combo that doesn't work and I'll see what I can do to support it.

Please see the [Caveats](#) below

1.2 Usage

To use the app just add a Podcast Root Page to your page hierarchy and fill in the details as necessary.

Try use a very high resolution image for the default image as that is the image used for your Podcast on iTunes/Google/etc. I also recommend high resolution images for the podcasts themselves. Don't worry about the high resolution, iTunes will cache it their side, and `wagtail_podcast` reduces image sizes when serving for you.

Look at the Podcast Root Page url and append `podcast/` to the end. Submit this URL to iTunes/Google/Blubrry. You will then get a podcast page on their platforms. Copy the url of your page to the relevent urls in the podcast root page and then the sidebar widget will be rendered.

To add a podcast just add a child page to the parent page and fill in the details. See [Podcast Options](#) for more information.

1.3 Analytics

90% of your users and listeners will consume your podcast through an app. Putting in analytics will create code bloat and probably not be meaningfully useful to you. If you need some analytics I highly recommend using Google Analytics as that will give you a good indication of web listeners, but for apps just use the analytics Google/iTunes/Blubrry provide through their platforms.

1.4 Customising

Honestly I give a really basic thing. No menus, no title bar, just some bootstrap cards, pagination, and a side bar, all of which is responsive. I highly recommend you edit the templates. I do plan on creating template tags at a later stage that should make customising much easier in the future. This is a rough, ugly framework just to get you on your feet. Go nuts! I recommend cheaking out the beautiful themes (based on Bootstrap) made by [Creative Tim](#)

For more about customising see: [Customising](#)

1.5 Caveats

1. I haven't implemented categories at all so that is rather broken. You need to go into `wagtail_podcast/templates/wagtail_podcast/feed.xml` and change it to something more appropriate

```
<itunes:category text="{{ page.category }}">
  <itunes:category text="Christianity"/>
</itunes:category>
```

2. There is currently no handling of ogg files though I plan on doing auto conversion to them at some point. All major browsers support mp3 these days.
3. This is made only for audio podcasts though I may support video podcasts at a later stage since that doesn't require much changing.
4. Seasons are currently not supported. This is really high on my list, but I haven't figured out a really great way to do it. The page hierarchy is already going to be a bit of a mess.
5. This currently only supports one podcast per wagtail site. It should be relatively easy to make it support multiple; however, I'll only do it if there is demand for the feature.

1. Episode Type:

This is what kind of episode you are publishing. It should usually be *Full*. Options are:

- (a) Full - Normal full length episode. Appears as expected in apps and on website.
- (b) Trailer - Teaser episode typically of a short length. Apps don't usually show them normally. Appears in normal list of podcasts on website.
- (c) Bonus - Special releases outside of your normal cycle and will be marked as such by most apps. Appears in normal list of podcasts on website

2. **Author:** The author of the podcast. List the institution if unsure

3. **Title:** Title of the podcast

4. **Subtitle:** Subtitle of the podcast

5. **Description:** Short description of the podcast. Used to attract users on the app.

6. **Recording:** The audio file you want to publish. It needs to be an MP3. Other formats will raise exceptions

7. **Explicit:** Whether the podcast episode contains explicit content or not. Make sure you check the iTunes guidelines to help decide if you need to tick this. **If this is unticked and your content contains explicit content then your account may be suspended on iTunes/Google/Whatever.**

8. **Page Text:** Text that accompanies your podcast. Shown on website and in app. Apps do not always display this.

9. **Image:** Image that you want to use to represent your particular podcast episode. Use a high quality image as *wagtail_podcast* will deal with the resizing automatically.

CHAPTER 3

Customising

Edit the template files. I do plan on doing template tags in the *future* for a better experience but this is just how it is for now.

All that you absolutely need is to load the `wagtail_podcast/css/mediaelementplayer.css` at the top of the page. You will need to load `wagtail_podcast/js/jquery.min.js` before `wagtail_podcast/js/mediaelement-and-player.min.js` static files in the `wagtail_podcast/podcast_page.html` template.

To actually render the media-player add the following:

```
<audio class="mejs__player">
  <source src="{{ page.recording.file.url }}" type="audio/mpeg">
  Your browser cannot play this type of file.
</audio>
```

There is plenty of customisation available of both how the player looks and functions, see: [Media Element Player JS docs](#)

Do note that you can do a decent bit of customisation with the `style` html attribute.

4.1 Languages

Currently only these languages are fully supported:

1. English (Daniel F. Meyer)
2. Afrikaans (Daniel F. Meyer)

It would be super awesome if you translate it to your locale language and make a pull request so that everybody can enjoy your translations. You'll also get credit on this page.

To translate the app is super easy thanks to gettext and Django's builtin stuff. This will translate the user interface and admin interface.

```
$ cd wagtail_podcast
$ django-admin makemessages -l <your_locale>
```

Open the `wagtail_podcast/locale/<your_locale>LC_MESSAGES/django.po` file in your favourite text editor. Provide your translations and then run `django-admin compilemessages`. The translations should now automatically activate on server restart.

The language will default to what is set in `settings.py`; however, if a specific Wagtail user changes it then it will be what they set as their language or what language you serve the page in to the client. See the Django and Wagtail internationalisation documentation on this.

4.2 Caveats

1. You will need to manually add your language locale to the `PodcastParentPage` model and then `manage.py makemigrations` and `manage.py migrate` and then restart your server. If you don't do this the drop-down won't be available on the admin page and iTunes and other services will have discoverability issues.
2. I have not internationalised the urls yet; however, its not important as there is no url route that has anything except `/<YYYY>/<MM>` routing in it which is nice. This will need to be done when adding seasons.


```
class wagtail_podcast.models.PodcastPage(*args, **kwargs)
    Model that acts as a specific podcast and is a child of the PodcastParentPage object

exception DoesNotExist

exception MultipleObjectsReturned

blubrry_url
    Returns the blubrry url
        Return type str
        Returns Blubrry URL

clean()
    Validates that the audio file passed is indeed a valid MP3 file.

get_archives()
    Gets the archives relating to the podcast
        Parameters self –
        Return type list
        Returns List of dictionaries containing ‘date’ as key and a podcast as a value

get_context(request, *args, **kwargs)
    Returns the context for a given page. This overridden function adds the archives to the context.
        Parameters
            • request (HttpRequest) – Django request object
            • args – Django request *args
            • kwargs – Django request **kwargs
        Returns Django/Wagtail context object

get_feed_url
    Get the url of the feed that represents the rss/xml file of the podcast
```

Return type `str`

Returns URL of the rss/xml feed

get_podcast_url

Get the url of the parent podcast page

Return type `str`

Returns URL of the parent page

google_url

Returns the Google Play url

Return type `str`

Returns Google Play URL

itunes_url

Returns the itunes url

Return type `str`

Returns Itunes URL

save (**args*, ***kwargs*)

Saves the model and adds metadata like length of file in bytes for Itunes and other services and well as adds the duration to the model.

Parameters

- **args** – Not important
- **kwargs** – Not important

class `wagtail_podcast.models.PodcastParentPage` (**args*, ***kwargs*)

The model that acts as the base page for the podcast. All podcasts are children of this page and this page lists the various podcasts.

exception `DoesNotExist`

exception `MultipleObjectsReturned`

archiveMonth (*request*, ***kwargs*)

Routing for a specific year and month combination

Parameters

- **request** (`HttpRequest`) – Normal Django request object
- **kwargs** – Normal Django request kwargs

Return type `HttpResponse`

Returns Page displaying the podcasts from a specific month and year combo with pagination

archiveYear (*request*, ***kwargs*)

Returns the entries of a year

Parameters

- **request** (`HttpRequest`) – Django request object
- **kwargs** – Django request kwargs

Return type `HttpResponse`

Returns `HttpResponse`

get_archives()

Return the archives of the podcast (date-based)

Return type `list`

Returns List of dictionaries containing a key of 'date' and an archive to accompany it

get_context(request, *args, **kwargs)

Normal `get_context` used by Wagtail overridden to include to archives and podcasts

Parameters

- **request** (`HttpRequest`) –
- **args** – Django request args
- **kwargs** – Django request kwargs

Returns Context of the page

get_feed_url

Returns the url of the feed of the podcast. Used for subscribing to and submitting to services like iTunes

Return type `str`

Returns URL of the podcast

get_pagination(request, kwargs)

Paginates the podcast queryset. We return the last 5 podcasts by default.

Parameters

- **request** (`HttpRequest`) – Django request object
- **kwargs** – Django kwargs associated with the request

Return type `Paginator`

Returns Returns a pagination object

get_podcast_url

Return URL of a the podcast parent page

Return type `str`

Returns Url of the podcast parent page

get_podcasts

Returns all the podcast objects that are currently live

Return type `QuerySet`

Returns Returns a Queryset of all the podcast objects

podcastFeed(request, **kwargs)

Routing for the feed url for use by clients and services to access the rss/xml feed

Parameters

- **request** (`HttpRequest`) – Normal django request
- **kwargs** – Normal django request kwargs

Return type `HttpResponse`

Returns rss/xml file

CHAPTER 6

Proposed Roadmap

0.1 First release

0.1.5 (End of November) Have good test coverage

0.2 (1st week of December) Make template tags so that customising and using in own templates is easier Fully functional categories Seasons Automatic Ogg generation

0.3 (Mid-January) Video support with auto conversion (webm and mp4)

0.4 (End-January) Multiple podcasts per site Author filtering Tags and tag filtering Season filtering

0.5 (If demanded) Essentially a one click deploy version for people who want to host their own podcasts

CHAPTER 7

Changelog

0.1

Changes Initial Release

Breaking Changes n/a

Fixes n/a

CHAPTER 8

Thanks

This app wouldn't be possible without the following great projects and people:

1. Wagtail CMS
2. Django Web Framework
3. Mutagen
4. Media Element JS
5. Bootstrap
6. Django Social Share
7. Font Awesome

CHAPTER 9

Indices and tables

- `genindex`
- `modindex`
- `search`

W

`wagtail_podcast.models`, [11](#)

A

`archiveMonth()` (`wagtail_podcast.models.PodcastParentPage` method), 12
`archiveYear()` (`wagtail_podcast.models.PodcastParentPage` method), 12

B

`blubrry_url` (`wagtail_podcast.models.PodcastPage` attribute), 11

C

`clean()` (`wagtail_podcast.models.PodcastPage` method), 11

G

`get_archives()` (`wagtail_podcast.models.PodcastPage` method), 11
`get_archives()` (`wagtail_podcast.models.PodcastParentPage` method), 12
`get_context()` (`wagtail_podcast.models.PodcastPage` method), 11
`get_context()` (`wagtail_podcast.models.PodcastParentPage` method), 13
`get_feed_url` (`wagtail_podcast.models.PodcastPage` attribute), 11
`get_feed_url` (`wagtail_podcast.models.PodcastParentPage` attribute), 13
`get_pagination()` (`wagtail_podcast.models.PodcastParentPage` method), 13
`get_podcast_url` (`wagtail_podcast.models.PodcastPage` attribute), 12
`get_podcast_url` (`wagtail_podcast.models.PodcastParentPage` attribute), 13
`get_podcasts` (`wagtail_podcast.models.PodcastParentPage` attribute), 13
`google_url` (`wagtail_podcast.models.PodcastPage` attribute), 12

I

`itunes_url` (`wagtail_podcast.models.PodcastPage` attribute), 12

P

`podcastFeed()` (`wagtail_podcast.models.PodcastParentPage` method), 13
`PodcastPage` (class in `wagtail_podcast.models`), 11
`PodcastPage.DoesNotExist`, 11
`PodcastPage.MultipleObjectsReturned`, 11
`PodcastParentPage` (class in `wagtail_podcast.models`), 12
`PodcastParentPage.DoesNotExist`, 12
`PodcastParentPage.MultipleObjectsReturned`, 12

S

`save()` (`wagtail_podcast.models.PodcastPage` method), 12

W

`wagtail_podcast.models` (module), 11