

---

# **Wagtail Event Calendar Documentation**

**Daniel F. Meyer**

**Sep 01, 2019**



**CONTENTS:**

<b>1</b>	<b>Getting Started</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	Usage . . . . .	3
1.3	Customising . . . . .	4
1.4	Caveats . . . . .	4
<b>2</b>	<b>Syncing</b>	<b>5</b>
<b>3</b>	<b>Customising</b>	<b>7</b>
<b>4</b>	<b>Internationalisation</b>	<b>9</b>
4.1	Languages . . . . .	9
4.2	Caveats . . . . .	9
<b>5</b>	<b>API Documentation</b>	<b>11</b>
5.1	FullCalendar API . . . . .	11
5.2	Models . . . . .	11
<b>6</b>	<b>Proposed Roadmap</b>	<b>15</b>
<b>7</b>	<b>Changelog</b>	<b>17</b>
<b>8</b>	<b>Thanks</b>	<b>19</b>
<b>9</b>	<b>Indices and tables</b>	<b>21</b>
	<b>Python Module Index</b>	<b>23</b>
	<b>Index</b>	<b>25</b>



This is a very basic calendar app that integrates with Wagtail. It allows a person to create events that will be shown on the website via fullcalendar and also generates an ical file.

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\\_eventcalendar git](#) to make pull requests or log issues etc. Documentation is at readthedocs.io: [wagtail\\_eventcalendar documentation](#)



## GETTING STARTED

### 1.1 Installation

To install run `pip install wagtail_eventcalendar`

It should automatically install the dependencies; however, if it doesn't then you will need to install them manually with: `pip install mutagen wagtailgmaps icalendar pytz django-social-share`

Remember to add `wagtail_eventcalendar` (along with the others mentioned) to your installed apps in `settings.py` i.e.

```
INSTALLED_APPS = [  
    ...  
    'wagtail_eventcalendar',  
    'wagtail.contrib.modeladmin',  
    'wagtailgmaps',  
    'django_social_share',  
    'wagtail.contrib.routable_page',  
]
```

Requirements:

```
python3  
icalendar  
wagtail  
django  
django-social-share  
pytz  
wagtailgmaps
```

You will need to follow the wagtailgmaps installation instructions to get things to work: <https://github.com/springload/wagtailgmaps/>

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 Calendar Root Page to your page hierarchy and fill in the details as necessary.

To create a new calendar entry just click the tab on the left handside labelled calendar and add Calendar Events there. You can also add, delete and manage your categories in the panel.

To sync with your devices see: [Syncing](#)

To get an ical file for either a calendar, a calendar filtered by category or even just a specific event, append `/ical/` to the url.

## 1.3 Customising

Honestly I give a really basic thing. No menus, no title bar, just some bootstrap cards, 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](#)

## 1.4 Caveats

1. I haven't implemented nested categories yet. I will eventually, but not just yet.
2. It only exports to ical for now. I will hopefully add support for caldav and gcal as well at some point.
3. Recurring events are currently not supported. This is really high on my list, but I haven't figured out a really great way to do it whilst keeping everything simple and manageable.



**SYNCING**

There isn't any real syncing implemented. If you can subscribe to an internet icalendar then you can sync. This is well supported on iOS and OSX, as well as pretty much everything in the Linux world. You will need an app on Android. Outlook has an extension available too.

You cannot add events to the calendar from a device. Only receive. Most of the time when a user clicks the <calendar add> button then it will give them the ical file. How each client handles that is up to the client. Most will just add the calendar entry to the device (all devices I know of including Android and Microsoft out of the box) but will not support syncing with them.

If you have very many clients accessing the calendar for syncing purposes it can have performance ramifications. This is because ical files are always dynamically generated on each and every request to them. You may be able to use Varnish to cache them for a few minutes at a time thus bringing requests actually hitting the servers down to a very negligible amount.

There is at present no *gcal and caldav syncing support*. If I do ever introduce it, it will be one way only i.e. devices cannot add to a calendar, only the admin interface can. Two way syncing is out of the scope of this project.

All calendars are at present public. I have not implemented any kind of permissions to them.



## 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_eventcalendar/css/fullcalendar.min.css` at the top of the page. You will need to load `wagtail_eventcalendar/js/jquery.min.js` and `wagtail_eventcalendar/js/moment.min.js` before `wagtail_podcast/js/fullcalendar.min.js` static files in the `wagtail_eventcalendar/event_calendar.html` and `wagtail_eventcalendar/event_calendar_category.html` templates.

To actually render the calendar add the following:

```
<div id='calendar'></div>
<script type="text/javascript">
    $(document).ready(function () {
        $('#calendar').fullCalendar({
            header: {center: 'month,agendaWeek,upcoming'},
            timeFormat: 'H(:mm)',
            nowIndicator: true,
            timezone: 'Africa/Johannesburg', //Change as necessary
            eventColor: "#f8f9fa", //Change as necessary
            eventTextColor: "#9c27b0", //Change as necessary
            businessHours: [{
                dow: [1, 2, 3, 4, 5],
                start: '08:00',
                end: '13:00'
            }, //change as necessary
            {
                dow: [7],
                start: '10:00',
                end: '11:00'
            }
        ],
        views: {
            upcoming: {
                type: 'list',
                duration: {months: 2},
                buttonText: 'Upcoming', //translate as necessary
                validRange: function (nowDate) {
                    return {
                        start: nowDate,
                        end: nowDate.clone().add(1, 'months')
                    };
                }
            }
        }
    })
}
```

(continues on next page)

(continued from previous page)

```
        },
        // put your options and callbacks here
        events: '{{ page.get_url }}events/'
    })
  });
</script>
```

There is plenty of customisation available of both how the calendar looks and functions, see: [FullCalendar JS docs](#) . In the above snippet of code you'll see that i added come comment that give you a decent starting point.

## INTERNATIONALISATION

### 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.

#### 4.1.1 Admin and General Website

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_eventcalendar/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.

#### 4.1.2 Calendar

The calendar is also rather easy to translate. in the `templates\wagtail_eventcalendar\event_calendar.html` and `templates\wagtail_eventcalendar\event_calendar_category.html` templates just look at the commented but about translating where it says 'Upcoming'.

To translate the other aspects of the calendar see: [locale](#). Make sure to load this javascript file after you load `fullcalendar.min.js`. Also remember to change the locale setting.

### 4.2 Caveats

1. I have not internationalised the urls yet; however, its on my list. Just not high on my list of changes.



## API DOCUMENTATION

### 5.1 FullCalendar API

To get events we return JSON. The api is super simple, but perfectly functional.

To get the events you need three components in your GET request namely `start`, `end` and `timezone` so a URL would look something like this `https://example.com/calendar/events?start=<start date and time>&end=<end date and time>&timezone='<timezone>'`

This API is made for simple GET requests only and is based on what FullCalendar uses.

### 5.2 Models

```
class wagtail_eventcalendar.models.Category (*args, **kwargs)
    Category to which an event belongs

exception DoesNotExist

exception MultipleObjectsReturned

clean()
    Ensures that there are no circular references when it comes to nested categories.

description
    Optional description of a specific category.

name
    Required. Name of the category

parent
    Optional. Not used currently. Planned for later as part of nested categories.

save (*args, **kwargs)
    Overrides Page model save method. Handles the slug

slug
    Required. Not used. Planned for later. Make it the same as the name for now.

class wagtail_eventcalendar.models.CategoryEventPage (*args, **kwargs)
    Internally used model. Ignore.

exception DoesNotExist

exception MultipleObjectsReturned
```

```
class wagtail_eventcalendar.models.EventCalPage(*args, **kwargs)
    Calendar entry/ an even base model.

    exception DoesNotExist

    exception MultipleObjectsReturned

    categories
        Optional category that a specific calendar entry may belong to

    clean()
        Checks that the end date and time occurs after the start date and date

    description
        Required. Description of the event/calendar entry

    end_dt
        Required. End datetime of the event/calender entry. Must be after start_dt else it raises a Validation Error

    property get_categories
        Gets all the event categories.

        Return type QuerySet

        Returns Queryset containing the Categories objects

    property get_status_text
        Shows the status text of a calender entry/event

        Return type Union[str, bool]

        Returns Str if the event is finished, or begun but not yet completed else false.

    icalView(request, *args, **kwargs)
        Route that returns an ical file for a specific event.

        Parameters

- request (HttpRequest) – Django HttpRequest
- args – Normal request args
- kwargs (dict) – Normal request kwargs

Return type HttpResponse

        Returns ical file as part of HttpResponse with only the details of a specific event

    image
        Optional image to associate with a calendar entry. Only really useful for the website

    location
        Optional location information

    problem_status
        Optional true/false indicating whether there is an issue with an event. It is important to both the ical files and the website

    problem_text
        Optional text that describes what is wrong. Used in conjunction with problem_status. Requires problem_status = true to work at all.

    save(*args, **kwargs)
        Overloads the save method of the Page model. It applies the default image to a calendar entry/event if it doesn't already have one.
```



**start\_dt**

Required. Start datetime of the event/calendar entry

**class** wagtail\_eventcalendar.models.EventCalendar(\*args, \*\*kwargs)

Base calendar class which actually displays the calendar.

**exception** DoesNotExist

**exception** MultipleObjectsReturned

**default\_image**

Default image to be used for calendar entries

**description**

Short description of the calendar.

**getEvents**(request)

Route that returns the events. Is accessed by fullcalendar as the api to call

**Parameters** request (HttpRequest) – Django request

**Return type** HttpResponse

**Returns** JSON of the events and their details

**getEventsByCategory**(request, \*\*kwargs)

Gets the events for a specific category for a specific timeframe. Is accessed by fullcalendar.js

**Parameters**

- **request** (HttpRequest) – Django request
- **kwargs** (dict) – Django request kwargs

**Return type** HttpResponse

**Returns** HttpResponse

**property** get\_categories

Gets the calendar categories that currently exist

**Return type** QuerySet

**Returns** Returns a Queryset of Category objects

**property** get\_url

Gets the url of the calendar page

**Return type** str

**Returns** Url of the calendar page

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

Route that produces the ical files requested by clients.

**Parameters**

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

**Return type** HttpResponse

**Returns** HttpResponse containing an ical file

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

Route that produces the ical files requested by clients, but filtered for a specific category

**Parameters**

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

**Return type** `HttpResponse`

**Returns** `HttpResponse` containing an ical file

**viewByCategory** (*request*, \*\*kwargs)

View calendar by a specific category

**Parameters**

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

**Return type** `HttpResponse`

**Returns** `HttpResponse` that shows a calendar filtered by a category

## PROPOSED ROADMAP

**0.1** First release

**0.1.5 (End of November 2018)** Have good test coverage

**0.2 (1st week of December 2018)** Make template tags so that customising and using in own templates is far easier

**0.3 (Mid-January 2019)** Add recurring events functionality

**0.4 (End-January 2019)** Making the urls i18n compliant

**0.5 (End-February 2019)** Implementing permission controls so only certain users can view calendars and entries

**0.6 (July 2019)** Implement sync to caldav and gcal



## CHANGELOG

### 0.1

**Changes** Initial Release

**Breaking Changes** n/a

**Fixes** n/a



## THANKS

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

1. [Wagtail CMS](#)
2. [Django Web Framework](#)
3. [iCalendar](#)
4. [FullCalendar JS](#)
5. [Wagtailgmaps](#)
6. [Bootstrap](#)
7. [Django Social Share](#)
8. [Font Awesome](#)





## INDICES AND TABLES

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



## PYTHON MODULE INDEX

### W

wagtail\_eventcalendar.models, [11](#)



## INDEX

### C

`categories` (*wagtail\_eventcalendar.models.EventCalendar* attribute), 12  
`Category` (class in *wagtail\_eventcalendar.models*), 11  
`Category.DoesNotExist`, 11  
`Category.MultipleObjectsReturned`, 11  
`CategoryEventPage` (class in *wagtail\_eventcalendar.models*), 11  
`CategoryEventPage.DoesNotExist`, 11  
`CategoryEventPage.MultipleObjectsReturned`, 11  
`clean()` (*wagtail\_eventcalendar.models.Category* method), 11  
`clean()` (*wagtail\_eventcalendar.models.EventCalendar* method), 12

### D

`default_image` (*wagtail\_eventcalendar.models.EventCalendar* attribute), 13  
`description` (*wagtail\_eventcalendar.models.Category* attribute), 11  
`description` (*wagtail\_eventcalendar.models.EventCalendar* attribute), 13  
`description` (*wagtail\_eventcalendar.models.EventCalendar* attribute), 12

### E

`end_dt` (*wagtail\_eventcalendar.models.EventCalendar* attribute), 12  
`EventCalendar` (class in *wagtail\_eventcalendar.models*), 13  
`EventCalendar.DoesNotExist`, 13  
`EventCalendar.MultipleObjectsReturned`, 13  
`EventCalendarPage` (class in *wagtail\_eventcalendar.models*), 11  
`EventCalendarPage.DoesNotExist`, 12  
`EventCalendarPage.MultipleObjectsReturned`, 12

### G

`get_categories()` (*wagtail\_eventcalendar.models.EventCalendar* property), 13  
`get_categories()` (*wagtail\_eventcalendar.models.EventCalendarPage* property), 12  
`get_status_text()` (*wagtail\_eventcalendar.models.EventCalendarPage* property), 12  
`get_url()` (*wagtail\_eventcalendar.models.EventCalendar* property), 13  
`get_events()` (*wagtail\_eventcalendar.models.EventCalendar* method), 13  
`get_events_by_category()` (*wagtail\_eventcalendar.models.EventCalendar* method), 13

### I

`icalView()` (*wagtail\_eventcalendar.models.EventCalendar* method), 13  
`icalView()` (*wagtail\_eventcalendar.models.EventCalendarPage* method), 12  
`icalViewCategory()` (*wagtail\_eventcalendar.models.EventCalendar* method), 13  
`image` (*wagtail\_eventcalendar.models.EventCalendarPage* attribute), 12

### L

`location` (*wagtail\_eventcalendar.models.EventCalendarPage* attribute), 12

### N

`name` (*wagtail\_eventcalendar.models.Category* attribute), 11

### P

`parent` (*wagtail\_eventcalendar.models.Category* attribute), 11

`problem_status` (wagtail\_eventcalendar.models.EventCalPage attribute), [12](#)  
`problem_text` (wagtail\_eventcalendar.models.EventCalPage attribute), [12](#)

## S

`save()` (wagtail\_eventcalendar.models.Category method), [11](#)  
`save()` (wagtail\_eventcalendar.models.EventCalPage method), [12](#)  
`slug` (wagtail\_eventcalendar.models.Category attribute), [11](#)  
`start_dt` (wagtail\_eventcalendar.models.EventCalPage attribute), [12](#)

## V

`viewByCategory()` (wagtail\_eventcalendar.models.EventCalendar method), [14](#)

## W

`wagtail_eventcalendar.models` (module), [11](#)