
Joyous Documentation

Release 1.4

Linuxsoftware

Feb 17, 2021

Contents:

1	Installing	3
1.1	Compatibility	3
2	Tutorials	5
2.1	Out of the box	5
2.2	Keeping things simple	9
2.3	A few easy changes	11
3	Usage guide	21
3.1	Calendar	21
3.2	Events	23
3.3	Groups	26
3.4	Settings	26
3.5	View Restrictions	28
3.6	iCal	28
3.7	Integration	29
4	Reference	31
4.1	Models	31
4.2	Template Tags	51
4.3	Formats	52
4.4	Utils	52
4.5	Admin User Interface	56
4.6	Fields	56
4.7	Middleware	56
4.8	Holidays	56
4.9	CSS	57
4.10	Templates	60
5	Release notes	67
5.1	Joyous 1.4.0 release notes	67
5.2	Joyous 1.3.1 release notes	68
5.3	Joyous 1.2.0 release notes	69
5.4	Joyous 1.1.0 release notes	70
5.5	Joyous 1.0.1 release notes	71
5.6	Joyous 1.0.0 release notes	71
5.7	Joyous 0.9.5 release notes	72

5.8	Joyous 0.9.4 release notes	73
5.9	Joyous 0.9.3 release notes	74
5.10	Joyous 0.9.2 release notes	75
5.11	Joyous 0.9.1 release notes	76
5.12	Joyous 0.9.0 release notes	77
5.13	Joyous 0.8.3 release notes	81
5.14	Joyous 0.8.2 release notes	82
5.15	Joyous 0.8.1 release notes	83
6	Indices and tables	87
	Python Module Index	89
	Index	91

Joyous is a reusable calendar application for Wagtail. *May all your events be joyous.*

The easiest way to install Joyous is from the [Python Package Index](#).

Install the package.

```
$ pip install ls.joyous
```

Add `ls.joyous` and `wagtail.contrib.modeladmin` to your `INSTALLED_APPS`.

```
INSTALLED_APPS = [  
    ...  
    'ls.joyous',  
    'wagtail.contrib.modeladmin',  
    ...  
]
```

Make sure `USE_TZ` is set to `True`

```
USE_TZ = True
```

The blocks `content`, `extra_css` and `extra_js` are required in the `base.html` template for the Joyous templates to work. A Wagtail project based upon the [default template](#) will have these.

Run migrations and collectstatic.

```
$ ./manage.py migrate  
$ ./manage.py collectstatic --no-input
```

1.1 Compatibility

Joyous version 1.4.0 is known to work with the following versions of Wagtail, Django and Python.

Wagtail	Django	Python
2.10.2	3.1.6	3.7.1
2.10.2	3.1.6	3.8.7
2.10.2	3.1.6	3.9.1
2.11.3	3.1.6	3.7.1
2.11.3	3.1.6	3.8.7
2.11.3	3.1.6	3.9.1
2.12	3.1.6	3.7.1
2.12	3.1.6	3.8.7
2.12	3.1.6	3.9.1

Other versions may work - YMMV.

Step by step instructions on using Joyous.

2.1 Out of the box

This tutorial covers what it is like to install a Joyous calendar, with no configuration or customisations, on a fresh Wagtail site.

0. You need to have setup a Wagtail site. Follow the Wagtail instructions at [Getting started](#) if you haven't already. I will assume your Wagtail site is called `mysite`, replace this with your actual site name.
1. Install Joyous and its dependencies:

```
$ pip install ls.joyous
```

2. Add `ls.joyous` and `wagtail.contrib.modeladmin` to your `INSTALLED_APPS` setting in `mysite/mysite/settings/base.py`.

```
INSTALLED_APPS = [  
    ...  
    'ls.joyous',  
    'wagtail.contrib.modeladmin',  
    ...  
]
```

3. Run the Django `manage.py` commands for setting up a new application to your site.

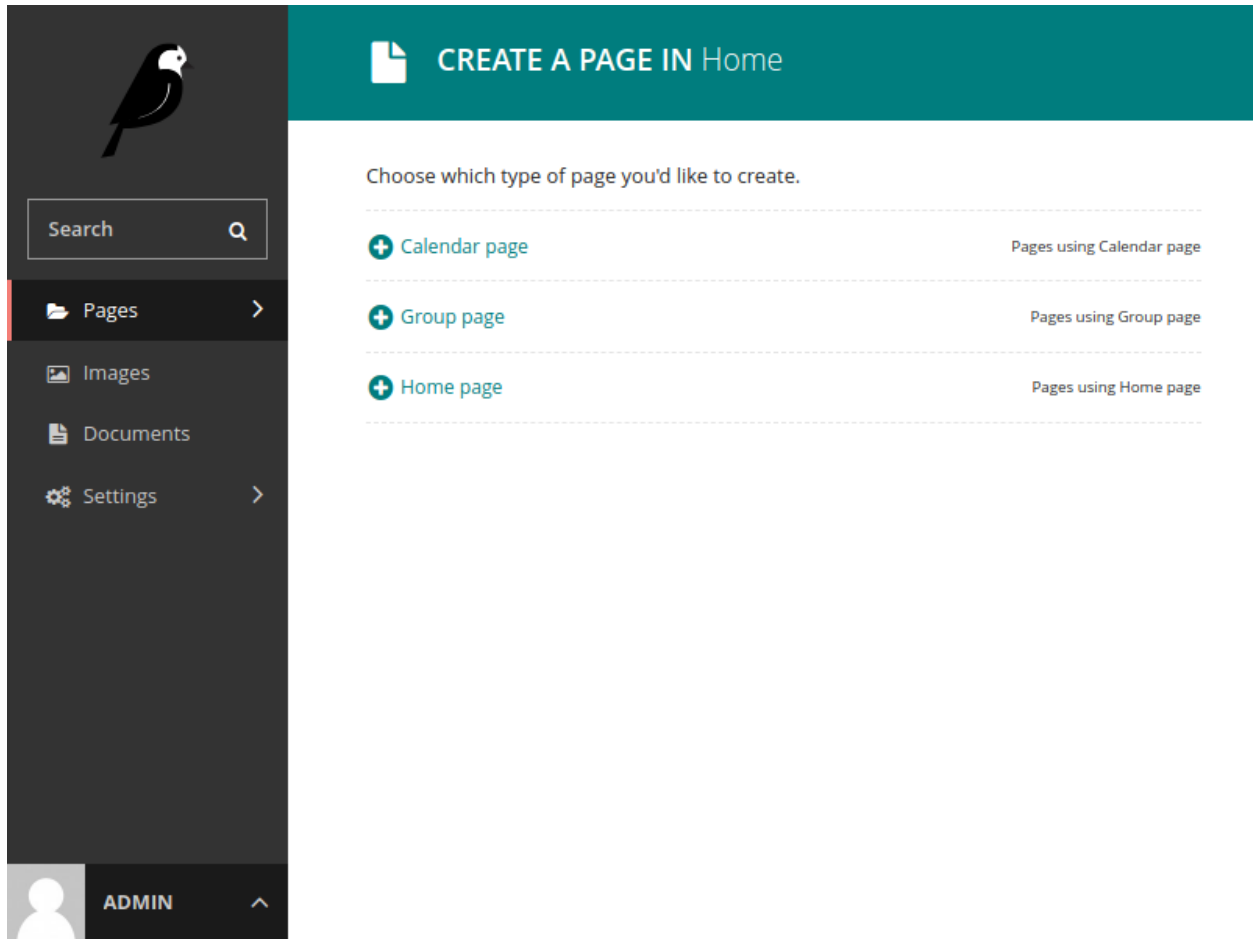
```
$ ./manage.py migrate  
$ ./manage.py collectstatic --no-input
```

4. Now you have added the `ls.joyous` application to your project, start your server.

```
$ ./manage.py runserver
```

And, point your web browser at `http://localhost:8000/admin/`.

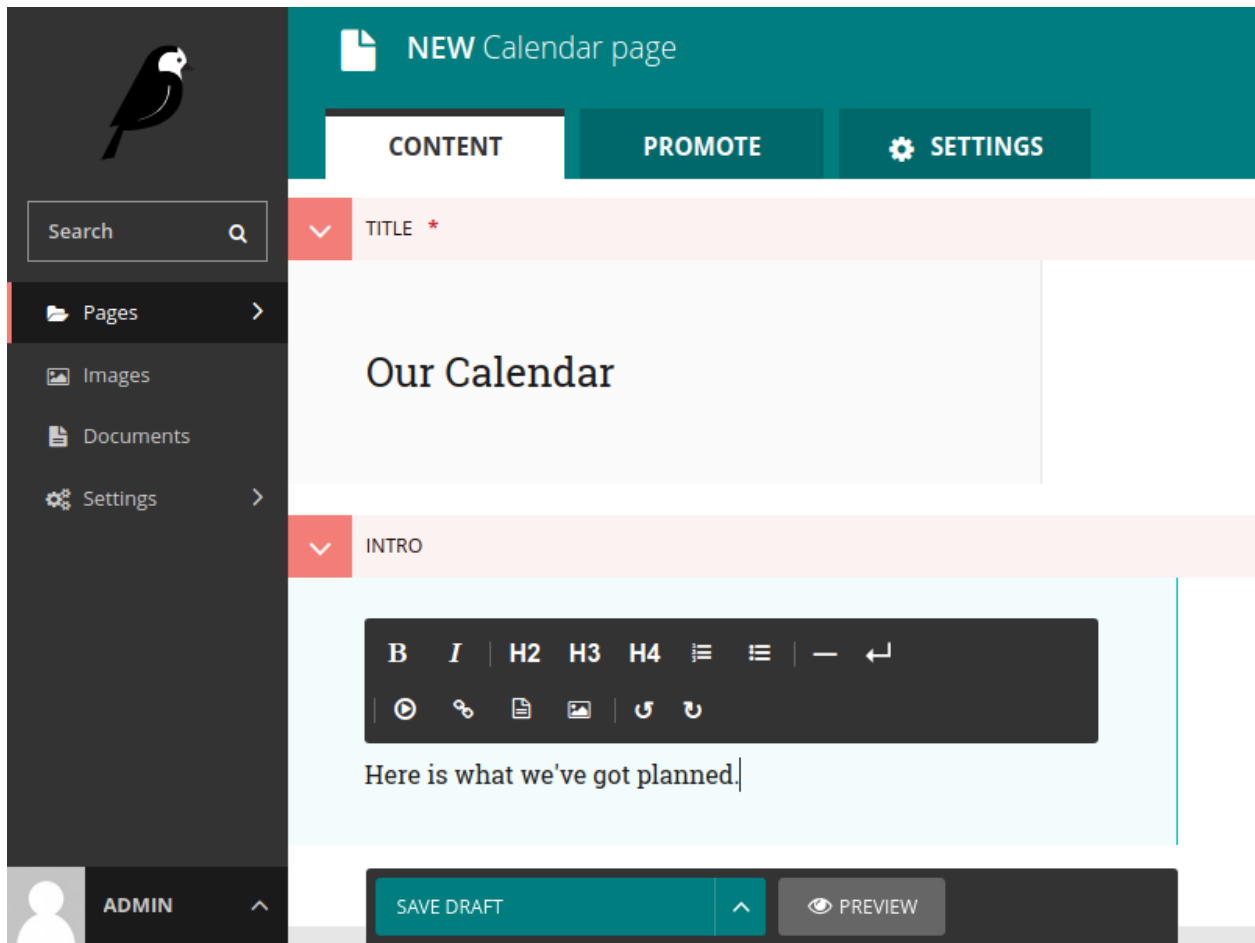
5. Select Home | Add child page, and you will see that the `Calendar page` is a new possibility.



6. Go ahead and choose `Calendar page`. Add a `Title` and maybe some `Intro` (introductory text).
7. Publish this. Select `View live`, and there is our new Joyous calendar.

Note: There can only be one `CalendarPage` per Wagtail site, so if you wish to repeat step 5 you will have to delete this one first. (See *SpecificCalendarPage* if you do want multiple calendars.)


8. To add an event to the calendar, add it as a child-page. You have the choice of four types (We can simplify this - covered in *Keeping things simple*).
 - Event page
 - Multiday event page
 - Multiday recurring event page
 - Recurring event page



Our Calendar

Here is what we've got planned.

[List View](#) | [Weekly View](#) | [This Month](#)

≤ March ≥ ≤ 2019 ≥						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						


[Export iCal](#)

9. Choose Event page. Add a Title, Date and some Details. The Time zone will default to your current time zone which is set in your Wagtail Account settings.
10. Publish your event. View the calendar again. Your event will be displayed on the date you set for it.

Our Calendar

Here is what we've got planned.

[List View](#) | [Weekly View](#) | [This Month](#)

≤ March ≥ ≤ 2019 ≥						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11 Calligraphy Lesson	12	13	14	15	16
17	18	19	20	21	22	23 
24	25	26	27	28	29	30

The calendar can also be displayed in a weekly view.

And as a list of upcoming events.

Probably you will want to customise the calendars and events on your site to suit your audience, but hopefully this tutorial has given you a useful introduction to Joyous.

2.2 Keeping things simple

Timezones? Views? Groups? Multiday-recurring events? All you wanted was a simple calendar of local one-off events! This tutorial will show you how to hide unneeded features to give your users a more streamlined interface.

In this example we will create a calendar for the gigs our band is playing.

0. Install Wagtail and Joyous. (See the *previous tutorial* for details on that.)
1. Create an application to hold the customisations we are going to make.


```
$ ./manage.py startapp gigs
```

2. Add this application to your INSTALLED_APPS setting in `mysite/mysite/settings/base.py`

Our Calendar

Here is what we've got planned.

[List View](#) | [This Week](#) | [Monthly View](#)

≤ Week 11 ≥							≤ 2019 ≥						
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
10 Mar	11 Mar Calligraphy Lesson	12 Mar	13 Mar	14 Mar	15 Mar	16 Mar							
													

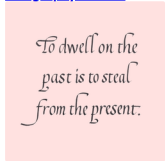
Our Calendar

Here is what we've got planned.

[Past Events](#) | [Weekly View](#) | [Monthly View](#)

Upcoming Events

[Calligraphy Lesson](#)



March 11, 2019

Beautiful handwriting is something we all admire. In a typing age, however, it is fast going the way of the Dodo. Recently, as a sort of backlash calligraphy has become a popular pastime across the world.

[Event Cal](#)



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

3. Edit `gigs/models.py` to have

```
from ls.joyous.models import (MultidayEventPage, RecurringEventPage,
                              MultidayRecurringEventPage, removeContentPanels)

# Hide unwanted event types
MultidayEventPage.is_creatable = False
RecurringEventPage.is_creatable = False
MultidayRecurringEventPage.is_creatable = False

# Hide unwanted content
removeContentPanels(["category", "tz", "group_page", "website"])
```

4. Start your server

```
$ ./manage.py runserver
```

And, point your web browser at `http://localhost:8000/admin/`.

5. Select Home | Add child page, and add a Calendar page. (Or if you already have, select Edit for it.)
6. Give the calendar a suitable Title etc.
7. Select the Settings tab. Under VIEW OPTIONS | View choices untick the List View and Weekly View options.
8. Publish this.
9. Add a child-page to your calendar. Notice how there is no need to select the event type, it goes straight to creating a new simple Event page.
10. Add a Title, Date and some Details. Notice how the user is not prompted for the fields Category, TZ, Group page, or Website due to the call to `removeContentPanels` in `gigs/models.py`.
11. Publish your event. View the calendar again. Your event will be displayed on the date you set for it.

Notice that the links for List View and Weekly View are not shown.

Hopefully your users will find this interface easy to use.

2.3 A few easy changes

It is time to customize Joyous to suit your needs. This tutorial will show some easy changes to make.

0. I will assume you already have Joyous installed and a calendar added. (See the *first tutorial* for details on that.)
1. Create an application to hold the customisations we are going to make. (If you already have one, then you can use that and skip this step.)

```
$ ./manage.py startapp myevents
```

Home

NEW Calendar page

CONTENT PROMOTE SETTINGS

Search

Pages

Images

Documents

Settings

SCHEDULED PUBLISHING

Go live date/time: Expiry date/time:

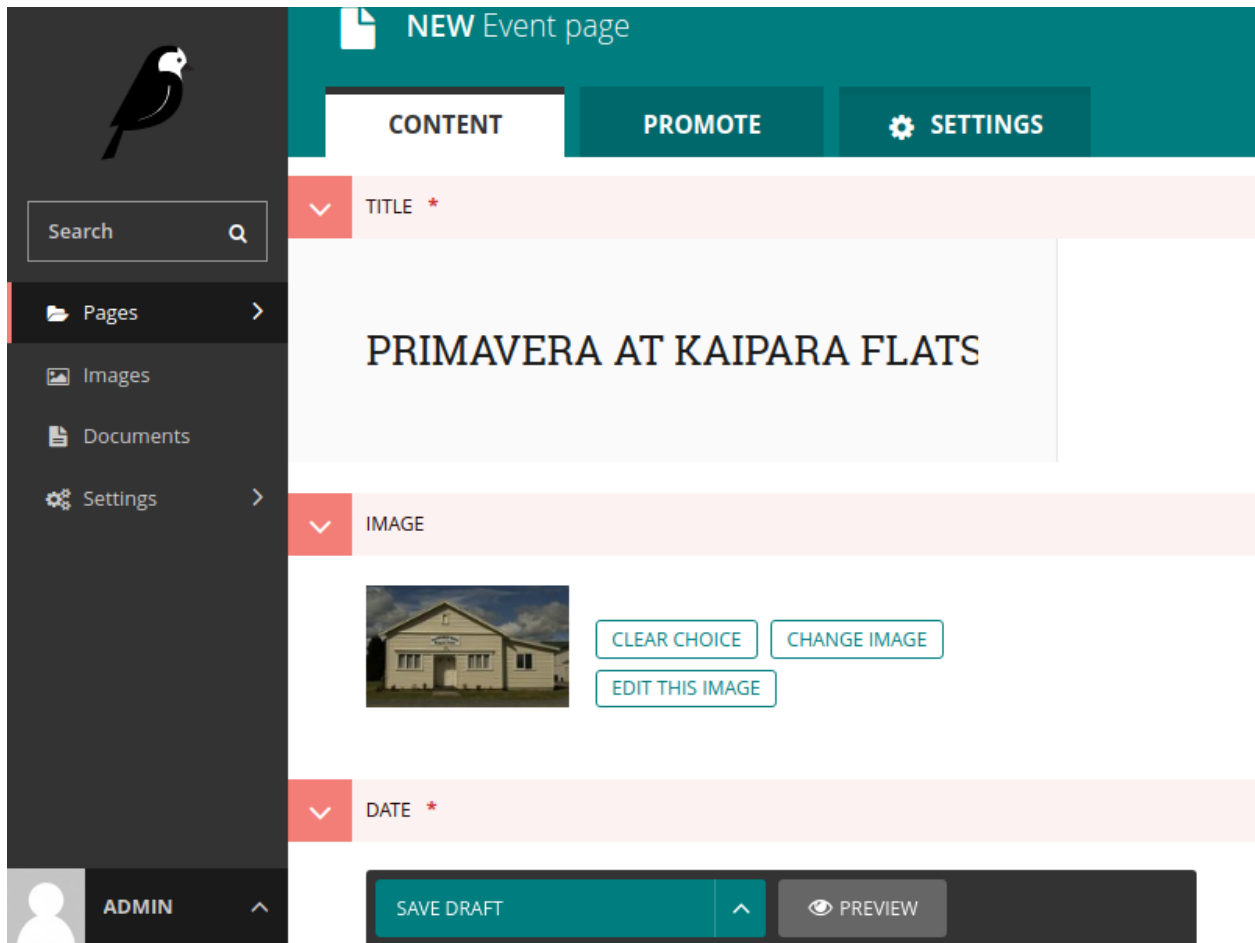
VIEW OPTIONS

View choices: List View Weekly View Monthly View

Default Monthly View

SAVE DRAFT PREVIEW

ADMIN



Gig Calendar

[This Month](#)

< November >			< 2019 >			
Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2 7:30 p.m. PRIMAVER AT KAIPARA FLATS RANFURLY ±
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23 
24	25	26	27	28	29	30

Add this application to your `INSTALLED_APPS` setting in `mysite/mysite/settings/base.py`

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

2.3.1 Settings

2. To start with let's add some settings. These can go in `mysite/mysite/settings/base.py`.

```
JOYOUS_THEME_CSS = "/static/joyous/css/joyous_forest_theme.css"
JOYOUS_HOLIDAYS = "Scotland"
JOYOUS_DATE_FORMAT = "%l %S %\o%\f F X"
JOYOUS_DATE_SHORT_FORMAT = "%j M X"
JOYOUS_TIME_FORMAT = "%f%q"
JOYOUS_TIME_INPUT = "%12"
```

- a. `JOYOUS_THEME_CSS` includes a theme CSS file into each Joyous page. You could do the same thing by overriding the Joyous templates (which we will look at later), but this is easier. We've chosen the forest theme file for a green palette.
- b. `JOYOUS_HOLIDAYS` adds holiday names onto our calendar. We've chosen the holidays of "Scotland". For the holidays of a state, province or region add it in square brackets after the country name, e.g. "Canada [NS] ". Multiple countries and regions can be listed separated by commas or by

using `*` as a wildcard. See `python-holidays` for all the countries and regions that are supported.

- c. `JOYOUS_DATE_FORMAT` for dates like “Monday 11th of March”.
 - d. `JOYOUS_DATE_SHORT_FORMAT` for short dates like “11 Mar”.
 - e. `JOYOUS_TIME_FORMAT` for times like “9am”.
 - f. `JOYOUS_TIME_INPUT` allows the editor to enter times in the 12 hour format, e.g. 9am.
3. The following Django settings are also important. Make sure they are set to values that are correct for you.

```
USE_TZ = True
TIME_ZONE = "Europe/London"
USE_I18N = True
USE_L10N = True
LANGUAGE_CODE = 'en-uk'
```

- a. Joyous uses timezone-aware datetimes, so `USE_TZ` must be set to `True`. If it is not you will get an error like `localtime() cannot be applied to a naive datetime when trying to view a calendar`.
 - b. `TIME_ZONE` sets the default timezone that Django uses. Wagtail also allows an editor to [change their time zone](#) for the Wagtail admin interface using the Account Settings | Current time zone panel.
 - c. `USE_I18N` turns on the Django translation system. If you only ever want to display English you could set it to `False`, but you might as well set it to `True` in case you ever want to display your website in another language.
 - d. `USE_L10N` enables Django’s localized formatting of numbers and dates. `JOYOUS_DATE_FORMAT`, `JOYOUS_DATE_SHORT_FORMAT`, `JOYOUS_TIME_FORMAT`, and `JOYOUS_FIRST_DAY_OF_WEEK` override Django’s formatting, but if they were not set then Joyous dates and times would be formatted according to the current locale. See your `django/conf/locale` directory to find these format files. If you want, you can create your own [custom format](#).
 - e. `LANGUAGE_CODE` sets the default locale that Django will use.
4. Start your server

```
$ ./manage.py runserver
```

And, have a look at your calendar and events.

2.3.2 Templates

Now, say you don’t want to display the download links, and want bigger images on your event pages. You can do this by [overriding](#) the Joyous templates. And, you can use [template inheritance](#) to override just the particular *blocks* you want to change. E.g. just override the footer.

- 5. Create a `templates/joyous` directory in your app.
- 6. Add the following files to your this directory. This will replace the download links in the footers of the calendars and events.

```
calendar_base.html
```


```
{% extends "joyous/calendar_base.html" %}
{% block cal_footer %}{% endblock cal_footer %}
```

```
calendar_list_upcoming.html
```

Our Calendar

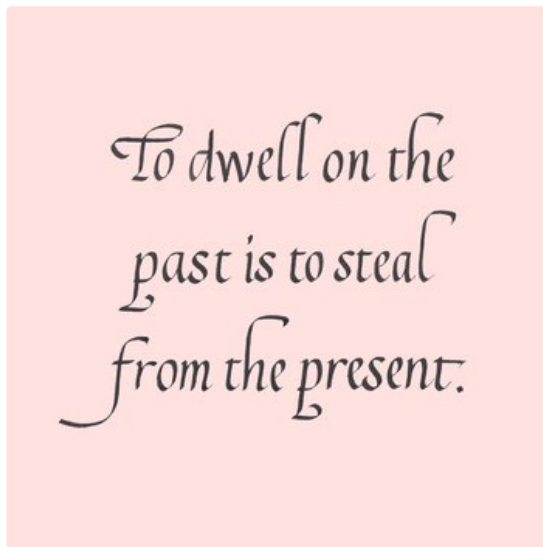
Here is what we have got planned.

[List View](#) | [Weekly View](#) | [This Month](#)

< November >							< 2019 >	
Sun	Mon	Tue	Wed	Thu	Fri	Sat		
					1	2		
3	4	5	6	7	8	9		
10	11	12	13	14	15	16		
17	18	19	20	21	22	23		
24	25	26	27	28	29	30	St. Andrew's Day	

Calligraphy Lesson

This event has finished.



Monday 11th of March

Beautiful handwriting is something we all admire. In a typing age, however, it is fast going the way of the Dodo. Recently, as a sort of backlash calligraphy has become a popular pastime across the world.

[Google Calendar](#) [Export iCal](#)



```
{% extends "joyous/calendar_list_upcoming.html" %}
{% block cal_footer %}{% endblock cal_footer %}
```

event_base.html

```
{% extends "joyous/event_base.html" %}
{% block event_footer %}{% endblock event_footer %}
```

7. Edit `event_base.html` and override the `event_image` block for a larger image. (Don't forget that you need to load `wagtailimages_tags` to use the `image` tag.)

event_base.html

```
{% extends "joyous/event_base.html" %}
{% load wagtailimages_tags %}

{% block event_footer %}{% endblock event_footer %}

{% block event_image %}
  {% if page.image %}
    <div class="joy-img">
      {% image page.image width=500 class="joy-img__img" %}
    </div>
  {% endif %}
{% endblock event_image %}
```

8. Also `postponement_page_from.html` has its own definition of `event_image` (because it displays the image of the original event not the postponement) so, for completeness, add an override for that too.

postponement_page_from.html

```
{% extends "joyous/postponement_page_from.html" %}
{% load wagtailimages_tags %}

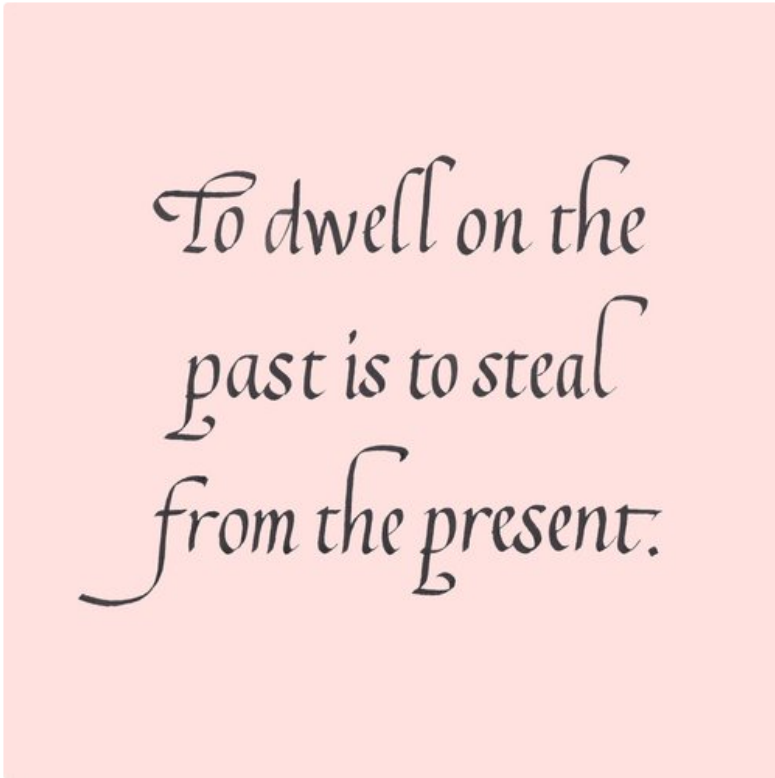
{% block event_image %}
  {% if overrides.image %}
    <div class="joy-img">
      {% image overrides.image width=500 class="joy-img__img" %}
    </div>
  {% endif %}
{% endblock event_image %}
```

9. Have another look at your calendar and events. Notice how the export links are gone and the images are larger.

I hope that this tutorial was useful.

Calligraphy Lesson

This event has finished.



Monday 11th of March

Beautiful handwriting is something we all admire. In a typing age, however, it is fast going the way of the Dodo. But, as a sort of backlash calligraphy has become a popular pastime across the world.



3.1 Calendar

The CalendarPage is the heart of the Joyous application.

3.1.1 Views

Users can display the calendar in a Monthly, Weekly, or List View. (The default view is set in the Settings tab.)

The first day of the week used, Sunday or Monday, depends upon your Django `format localization` or `FIRST_DAY_OF_WEEK` setting. This can be over-ridden using the `JOYOUS_FIRST_DAY_OF_WEEK` setting.

There are actually multiple list views

- All upcoming events
- All past events
- and a “secret” all events of a day - unless there’s only one - in which case it just redirects straight to that event

3.1.2 URLs

How a calendar is displayed or exported depends upon the path and query string of the URL used to access it. Consider a calendar with the slug `/events/`:

<code>/events/</code>	Default view of the calendar - set as a per-calendar property.
<code>/events/month/</code>	Monthly view.
<code>/events/week/</code>	Weekly view.
<code>/events/day/</code>	Day list view.
<code>/events/upcoming/</code>	List of upcoming events.
<code>/events/past/</code>	List of past events.
<code>/events/?view=list</code>	Specified (list weekly monthly) view of the calendar.
<code>/events/2017/</code>	Default view of the calendar for 2017
<code>/events/2017/?view=weekly</code>	Specified view for 2017.
<code>/events/2018/Apr/</code>	Monthly view for April 2018.
<code>/events/2018/5/</code>	Monthly view for May 2018.
<code>/events/2018/W2/</code>	Weekly view for Week 2 of 2018.
<code>/events/2018/6/18/</code>	Day list view for the 18th of June 2018.
<code>/events/?format=ical</code>	Export as an <i>iCal</i> file.
<code>/events/?format=rss</code>	Export as a RSS feed.

3.1.3 Models

CalendarPage

A *CalendarPage* displays all the events in the same Wagtail Site as itself.

If that isn't what you want, then have a look at *GeneralCalendarPage* or *SpecificCalendarPage*.

GeneralCalendarPage

Displays all the events in the database ignoring site boundaries. See *GeneralCalendarPage*.

GeneralCalendarPage is disabled by default. Use `GeneralCalendarPage.is_creatable = True` to enable it.

SpecificCalendarPage

Displays only those events which are children of itself. See *SpecificCalendarPage*.

SpecificCalendarPage is disabled by default. Use `SpecificCalendarPage.is_creatable = True` to enable it.

Derive your own

If you would like some other kind of event selection you can derive your own version of *CalendarPage*.

Have a look at the source-code of the *CalendarPages* classes if you would like some other kind of event selection. The methods `_getEventsOnDay`, `_getEventsByDay`, `_getEventsByWeek`, `_getUpcomingEvents`, and `_getPastEvents` determine what events are displayed. The methods `_getEventFromUid` and `_getAllEvents` are for import and export.

3.1.4 Holidays

Holidays are a property of the *CalendarPage*.

If the `JOYOUS_HOLIDAYS` setting is set then it is used to select holidays from `python-holidays`. But it is also possible to add other holiday sources (e.g. from `workalendar` or just a simple dict) via `register`. And to add individual days via `add`.

For example:

```
from datetime import date
from workalendar.america import Ontario

CalendarPage.holidays.register(Ontario())
CalendarPage.holidays.add(date(2019,4,29), "HAPPY HAPPY")
```

It would also be possible to derieve different Calendar models and give them different sets of holidays. Holidays for *CalendarPage* are determined programmatically, but a derieved Calendar model could choose to change this, e.g. store the holidays in the database so that different pages of the same model could have different holidays.

3.2 Events

Events are the building blocks of the Joyous application.

When creating a new event the user is prompted which type of event to create

- *SimpleEventPage*
- *MultidayEventPage*
- *RecurringEventPage*
- *MultidayRecurringEventPage*

All the types of events share some fields in common.

- Title
- UID
- Category
- Image
- Start time
- Finish time
- Time zone
- Group page
- Details
- Location

The start and finish times are optional. If left blank start is considered as the beginning-of-the-day, and finish is considered as the end-of-the-day.

The times and dates of an event are in a certain time zone. This will default to the user's current time zone which is set in their Wagtail Account Settings.

3.2.1 One Off Events

SimpleEventPage

A simple event is one that occurs just once on some date. See *SimpleEventPage*.

MultidayEventPage

A multiday event is one that spans several days. Unlike the simple event it has a start date and a finish date. See *MultidayEventPage*.

An example might be a boat cruise which departs on a certain date and then returns 4 days later.

What about

A party that starts on 2018-12-31 at 9pm and finishes on 2019-01-01 at 2am

This party spans 2 days, so the most accurate way of recording it would be with a *MultidayEventPage*.

But you might not consider the party as *really* being on January 1. (After all who is going to turn up after midnight?) So you could record it as a *SimpleEventPage*, but leave the `time_` field blank, or enter it as 24:59:59. It is technically not as accurate, but it is up to you.

3.2.2 Recurring Events

RecurringEventPage

A recurring event has multiple occurrences repeating by a certain rule. See *RecurringEventPage*.

The recurrence rule is somewhat based upon RFC 5545 RRules. Events can occur daily, weekly, monthly, or yearly intervals, by weekday, monthday, and month. See *Recurrence*.

A *RecurringEventPage* can have child pages of its own which specify exceptions to the rule. See *CancellationPage*, *PostponementPage*, and *ExtraInfoPage* below for more details.

MultidayRecurringEventPage

This is like a recurring event, but each recurrence may last for multiple days. See *MultidayRecurringEventPage*.

ExtraInfoPage

An *ExtraInfoPage* holds some extra details for a certain occurrence of a recurring event.

CancellationPage

A *CancellationPage* removes a certain occurrence of a recurring event. If given a `cancellation_title` this will appear in place of the occurrence, but if not the occurrence is just quietly removed.

PostponementPage

A *PostponementPage* both removes an occurrence and adds a replacement event. It is a bit like a combined *CancellationPage* and *SimpleEventPage* in one.

Postponement pages have two views. View the new details at the regular path (e.g. `/events/bbq/2019-04-10-postponement/`) and the details of the cancellation by via the `/from/` view (e.g. `/events/bbq/2019-04-10-postponement/from/`).

When a Postponement is not a postponement

The *PostponementPage* was named with the intention that it would be used for when an occurrence of a recurring event had to be postponed until a later time. But it could also be used to move the occurrence to start at an earlier time, finish at a different time, or change some other field.

If you would like to change the name, you can do so by putting the following bit of code in your application's `models.py` or `wagtail_hooks.py`.

```
from ls.joyous.models import PostponementPage

PostponementPage._meta.verbose_name = "event change"
PostponementPage._meta.verbose_name_plural = "event changes"
PostponementPage.slugName = "change"
```

RescheduleMultidayEventPage

The multiday version of the *PostponementPage*. See *RescheduleMultidayEventPage*.

ExtCancellationPage

An *ExtCancellationPage* removes all occurrences of a recurring event from the `cancelled_from_date` until the optional `cancelled_to_date`. If the to date is not given, then the event is cancelled “until further notice”. If given a `cancellation_title` this will appear in place of the occurrence, but if not the occurrence is just quietly removed.

Note: A *ExtCancellationPage* is exported in iCalendar format as a series of EXDATES. If the to date is not given or is far in the future there could be hundreds of EXDATES. However Google Calendar will refuse to import an event with more than about 90 EXDATES. Investigation of this issue is still in progress.

ClosedForHolidaysPage

A *ClosedForHolidaysPage* removes any occurrences of a recurring event if they are on a holiday. All holidays or a specific list of holidays can apply. If given a `cancellation_title` this will appear in place of the occurrence, but if not the occurrence is just quietly removed.

Note: A *ClosedForHolidaysPage* is also exported in iCalendar format as a series of EXDATES. The same problem trying to import these into Google Calendar as above applies.

3.3 Groups

Your organisation might be made up of several groups which can have their own events. Or you might want to group events together for another reason.

The template tag `group_upcoming_events` displays the events that are coming up for a group.

Adding the event as a child of a group automatically assigns the event to that group, or you can set the `group_page` field on the event.

Note: It is not expected that you will both add an event as the child of a group and set the `group_page` field. But if you do then: if it is the same group, the event will only show up in the group once; if they are different groups, the event will show up in both groups, but will only show itself as belonging to the group that it is a child of.

3.3.1 Models

GroupPage

A simple default page model for a group. It just has a title and a rich-text content area.

The template for GroupPage includes the `group_upcoming_events` template tag.

3.4 Settings

3.4.1 JOYOUS_DATE_FORMAT

Default: Falls back to Django date formatting. (See [Format localization](#), `DATE_FORMAT`)

Format of dates, if different from the Django standard. Uses the same options as `date-and-time-formatting-specifiers`, plus 'X' which only gives the year if it is not the current year.

Added in *version 0.9.1*. Use `JOYOUS_DATE_FORMAT = "l jS \\o\\f F X"` for formatting as it was previously.

3.4.2 JOYOUS_DATE_SHORT_FORMAT

Default: Falls back to Django date formatting. (See [Format localization](#), `SHORT_DATE_FORMAT`)

Short format of dates, if different from the Django standard. Uses the same options as `JOYOUS_DATE_FORMAT`.

Added in *version 0.9.1*. Use `JOYOUS_DATE_SHORT_FORMAT = "j F Y"` for formatting as it was previously.

3.4.3 JOYOUS_EVENTS_PER_PAGE

Default: 25

Page limit for a list of events.

Added in *version 0.8.1*.

3.4.4 JOYOUS_FIRST_DAY_OF_WEEK

Default: Falls back to Django date formatting. (See [Format localization](#), `FIRST_DAY_OF_WEEK`)

The first day of the week, 0=Sunday or 1=Monday.

Added in *version 0.9.5*.

3.4.5 JOYOUS_GROUP_MODEL

Default: `"joyous.GroupPage"`

To swap out the *Groups* model.

3.4.6 JOYOUS_GROUP_SELECTABLE

Default: `False`

Enable group selection? `False` or `True`.

3.4.7 JOYOUS_HOLIDAYS

Default: `"` (Empty string)

Observed holidays using [python-holidays](#). Specified as string of countries [with regions in square brackets] separated by commas. e.g. `"NZ[WTL,Nelson],AU[*],Northern Ireland"`.

See *Holidays*.

3.4.8 JOYOUS_RSS_FEED_IMAGE

Default: `"/static/joyous/img/logo.png"`

This is the image that is displayed on RSS for your channel.

3.4.9 JOYOUS_THEME_CSS

Default: `"` (Empty string)

The path of a theme CSS file to include. Joyous CSS does not push colour or font choices. But there are theme CSS files available which you can optionally choose to import using this setting.

Available themes:

- `joyous_coast_theme.css`: Greys and gold.
- `joyous_forest_theme.css`: Greens.
- `joyous_stellar_theme.css`: A dark background theme.

Added in *version 0.9.0*. Use `JOYOUS_THEME_CSS = "/static/joyous/css/joyous_coast_theme.css"` to continue with the previous default appearance.

3.4.10 JOYOUS_TIME_FORMAT

Default: Falls back to Django time formatting. (See [Format localization](#), `TIME_FORMAT`)

Format of times, if different from the Django standard. Uses the same options as [date-and-time-formatting-specifiers](#), plus 'q' which gives am or pm in lowercase.

Added in *version 0.9.1*. Use `JOYOUS_TIME_FORMAT = "fq"` for formatting as it was previously.

3.4.11 JOYOUS_TIME_INPUT

Default: "24"

Prompt for 12 or 24 hour times.

3.4.12 JOYOUS_UPCOMING_INCLUDES_STARTED

Default: False

If this is set to `True` then the list of upcoming events will also include events that have already started but have not yet finished.

Added in *version 0.9.5*.

3.5 View Restrictions

Just like other Wagtail pages Joyous event pages can be made [private](#).

When an event is private and you do not have authority to view it, it will not show up on the calendar or in any list of events. But, if it is a Cancellation or Postponement it will still have an effect upon its parent `RecurringEvent`. E.g. If an event that occurs weekly is postponed for this week, BUT the postponement is hidden from you, then you will see there is not an occurrence of that event this week, but you will not see why. [I can't imagine why anyone would want to hide the postponement, but not the recurring event - but if they do, this is the design decision Joyous implements.]

For password protected events: "authority to view" means you have viewed that page by entering the correct password once in this login-session already. Joyous is not going to prompt for passwords itself.

When exporting if the user has publish rights (includes the rights to change the view restriction) to the event then it will be exported, otherwise it'll be based upon their view rights.

3.6 iCal

3.6.1 Import

iCalendar is the common calendar file format. Google, Meetup, Facebook, Outlook, Thunderbird, etc can generate iCalendar (.ics) files for a whole calendar of events or just one event.

Importing events from an iCalendar file is done via the settings tab of editing a calendar page. The events will be imported as the children of this page.

Steps for iCalendar import:

- Pick a calendar page, or create a new one.
- Select the settings tab

- Choose you iCalendar file
- Select “Save Draft” and events will be imported as draft pages
- or “Publish” and events will be imported as published pages

Note: Before Joyous imports an event it checks if that event already exists. (the UID property is used for matching) If it already does then it will only be updated as a new revision if this is a newer version (The last-modified or timestamp property is used.) This avoids duplicates and unnecessary revisions

Joyous converts events from the iCalendar file into simple, multiday or recurring event pages as appropriate.

3.6.2 Export

Steps for iCalendar export:

Joyous can export a iCalendar file with a whole calendar of event, or just one event.

To export a single event:

- view the event
- click the “Export ICal” link

To export a whole calendar:

- view the calendar
- click the “Export ICal” link

3.7 Integration

3.7.1 GoogleCalendar

Subscribe

You can subscribe to a Joyous calendar in GoogleCalendar using *iCal*.

To do this:

1. Log in to your GoogleCalendar
2. Click “Other calendars +” (Add other calendars)
3. Choose “From URL”
4. Enter the url of your Joyous calendar with “?format=ical” on the end, e.g. <http://demo.linuxsoftware.nz/joyous/?format=ical>
5. Change any other settings you like (e.g. the calendar name)
6. Exit settings by clicking “<-” up the top-left.

GoogleCalendar will now display your Joyous events and will sync with changes that are made.

Import

You can also export from GoogleCalendar and import in to a Joyous calendar using *iCal*.

To do this:

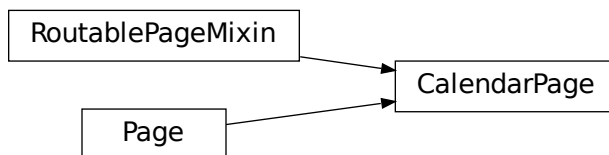
1. Log in to your GoogleCalendar
2. Choose Settings | Import & export | Export
3. Log in to your Joyous calendar
4. Pick a calendar page, or create a new one.
5. Select the Settings tab
6. Choose the file that you downloaded from Google (you do not need to unzip it).
7. Select “Publish” and events will be imported as published pages.

Joyous will now display your GoogleCalendar events, but you will have to manually repeat the process to re-sync them

4.1 Models

4.1.1 Calendar

CalendarPage



```
class ls.joyous.models.CalendarPage(*args, **kwargs)
```

```
  Bases: wagtail.contrib.rutable_page.models.RutablePageMixin, wagtail.core.models.Page
```

CalendarPage displays all the events which are in the same site.

holidays

The holidays to be displayed by this calendar.

intro

Introductory text.

view_choices

What types of calendar views the user can select.

default_view

The default calendar view to display to the user.

routeDefault (*request*, *year=None*)

Route a request to the default calendar view.

routeByMonthAbbr (*request*, *year*, *monthAbbr*)

Route a request with a month abbreviation to the monthly view.

serveMonth (*request*, *year=None*, *month=None*)

Monthly calendar view.

serveWeek (*request*, *year=None*, *week=None*)

Weekly calendar view.

serveDay (*request*, *year=None*, *month=None*, *dom=None*)

The events of the day list view.

serveUpcoming (*request*)

Upcoming events list view.

servePast (*request*)

Past events list view.

serveMiniMonth (*request*, *year=None*, *month=None*)

Serve data for the MiniMonth template tag.

classmethod can_create_at (*parent*)

Checks if this page type can be created as a subpage under a parent page instance.

classmethod _allowAnotherAt (*parent*)

You can only create one of these pages per site.

classmethod peers ()

Return others of the same concrete type.

_getEventsOnDay (*request*, *day*)

Return all the events in this site for a given day.

_getEventsByDay (*request*, *firstDay*, *lastDay*)

Return the events in this site for the dates given, grouped by day.

_getEventsByWeek (*request*, *year*, *month*)

Return the events in this site for the given month grouped by week.

_getUpcomingEvents (*request*)

Return the upcoming events in this site.

_getPastEvents (*request*)

Return the past events in this site.

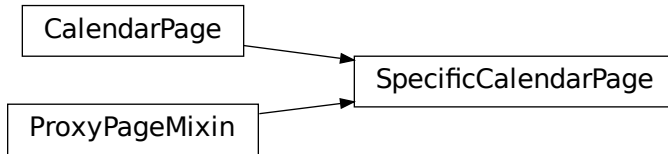
_getEventFromUid (*request*, *uid*)

Try and find an event with the given UID in this site.

_getAllEvents (*request*)

Return all the events in this site.

SpecificCalendarPage



```
class ls.joyous.models.SpecificCalendarPage (*args, **kwargs)
```

```
Bases: ls.joyous.utils.mixins.ProxyPageMixin, ls.joyous.models.calendar.CalendarPage
```

SpecificCalendarPage displays only the events which are its children

```
classmethod _allowAnotherAt (parent)
```

Don't limit creation.

```
_getEventsByDay (request, firstDay, lastDay)
```

Return my child events for the dates given, grouped by day.

```
_getEventsByWeek (request, year, month)
```

Return my child events for the given month grouped by week.

```
_getUpcomingEvents (request)
```

Return my upcoming child events.

```
_getPastEvents (request)
```

Return my past child events.

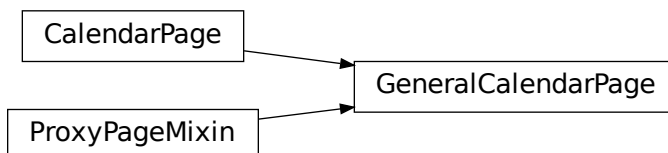
```
_getEventFromUid (request, uid)
```

Try and find a child event with the given UID.

```
_getAllEvents (request)
```

Return all my child events.

GeneralCalendarPage



```
class ls.joyous.models.GeneralCalendarPage (*args, **kwargs)
    Bases: ls.joyous.utils.mixins.ProxyPageMixin, ls.joyous.models.calendar.CalendarPage
```

GeneralCalendarPage displays all the events no matter where they are

```
classmethod _allowAnotherAt (parent)
```

You can only create one of these pages.

```
_getEventsByDay (request, firstDay, lastDay)
```

Return all events for the dates given, grouped by day.

```
_getEventsByWeek (request, year, month)
```

Return all events for the given month grouped by week.

```
_getUpcomingEvents (request)
```

Return all the upcoming events.

```
_getPastEvents (request)
```

Return all the past events.

```
_getEventFromUid (request, uid)
```

Try and find an event with the given UID.

```
_getAllEvents (request)
```

Return all the events.

4.1.2 Events

Get Events API

```
ls.joyous.models.getAllEventsByDay (request, fromDate, toDate, *, home=None, holidays=None)
```

Return all the events (under home if given) for the dates given, grouped by day.

Parameters

- **request** – Django request object
- **fromDate** – starting date (inclusive)
- **toDate** – finish date (inclusive)
- **home** – only include events that are under this page (if given)
- **holidays** – the holidays that are celebrated for these dates

Return type list of `EventsOnDay` objects

```
ls.joyous.models.getAllEventsByWeek (request, year, month, *, home=None, holidays=None)
```

Return all the events (under home if given) for the given month, grouped by week.

Parameters

- **request** – Django request object
- **year** (*int*) – the year
- **month** (*int*) – the month
- **home** – only include events that are under this page (if given)
- **holidays** – the holidays that are celebrated for these dates

Returns a list of sublists (one for each week) each of 7 elements which are either None for days outside of the month, or the events on the day.

Return type list of lists of None or `EventsOnDay` objects

`ls.joyous.models.getAllUpcomingEvents` (*request*, *, *home=None*, *holidays=None*)
Return all the upcoming events (under home if given).

Parameters

- **request** – Django request object
- **home** – only include events that are under this page (if given)
- **holidays** – holidays that may affect these events

Return type list of `ThisEvents`

`ls.joyous.models.getAllPastEvents` (*request*, *, *home=None*, *holidays=None*)
Return all the past events (under home if given).

Parameters

- **request** – Django request object
- **home** – only include events that are under this page (if given)
- **holidays** – holidays that may affect these events

Return type list of the `ThisEvents`

`ls.joyous.models.getGroupUpcomingEvents` (*request*, *group*, *holidays=None*)
Return all the upcoming events that are assigned to the specified group.

Parameters

- **request** – Django request object
- **group** – for this group page
- **holidays** – holidays that may affect these events

Return type list of `ThisEvents`

`ls.joyous.models.getEventFromUid` (*request*, *uid*)
Get the event by its UID (raises `PermissionDenied` if we have no authority, `ObjectDoesNotExist` if it is not found).

Parameters

- **request** – Django request object
- **uid** – iCal unique identifier

Return type event page

`ls.joyous.models.getAllEvents` (*request*, *, *home=None*, *holidays=None*)
Return all the events (under home if given).

Parameters

- **request** – Django request object
- **home** – only include events that are under this page (if given)
- **holidays** – holidays that may affect these events

Return type list of event pages

EventsOnDay

class `ls.joyous.models.EventsOnDay` (*date*, *holiday=None*, *days_events=None*, *continuing_events=None*)

The events that occur on a certain day. Both events that start on that day and events that are still continuing.

date

holiday

days_events
The events that start on this day

Return type list of the namedtuple `ThisEvent` (title, page, url)

continuing_events
The events that are still continuing on this day

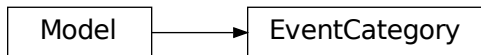
Return type list of the namedtuple `ThisEvent` (title, page, url)

all_events
All the events that occur on this day, `days_events` + `continuing_events`.

preview
A short description of some of the events on this day (limited to 100 characters).

weekday
The weekday abbreviation for this days (e.g. “mon”).

EventCategory



class `ls.joyous.models.EventCategory` (**args*, ***kwargs*)

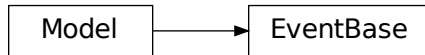
Bases: `django.db.models.base.Model`

The category type of an event.

code
A short 4 character code for the category.

name
The category name.

EventBase



```
class ls.joyous.models.EventBase (*args, **kwargs)
```

```
Bases: django.db.models.base.Model
```

uid

A unique identifier for the event, used for iCal import/export.

category

What type of event is this?

image

A banner image for the event.

time_from

The time the event starts (optional).

time_to

The time the event finishes (optional).

tz

The time zone for the event. No, sorry, you can't set different time zones for `time_from` and `time_to`.

group_page

A page chosen to link a group of events together.

details

Free text for whatever else you want to say about the event.

location

Where the event will occur. If `wagtailmaps` is installed `MapFieldPanel` will be used, but this is not a requirement.

website

A website location for the event.

group

The group this event belongs to. Adding the event as a child of a group automatically assigns the event to that group.

_current_datetime_from

The datetime this event will start or did start in the local timezone, or None if it is finished.

_future_datetime_from

The datetime this event next starts in the local timezone, or None if in the past.

_past_datetime_from

The datetime this event previously started in the local timezone, or None if it never did.

_first_datetime_from

The datetime this event first started in the local time zone, or None if it never did.

status

The current status of the event (started, finished or pending).

status_text

A text description of the current status of the event.

at

A string describing what time the event starts (in the local time zone).

classmethod `_removeContentPanels` (*args)

Remove the panels and so hide the fields named.

isAuthorized (*request*)

Is the user authorized for the requested action with this event?

get_context (*request*, *args, **kwargs)

_getLocalWhen (*date_from*, *date_to=None*)

Returns a string describing when the event occurs (in the local time zone).

_getFromTime (*atDate=None*)

Time that the event starts (in the local time zone) for the given date.

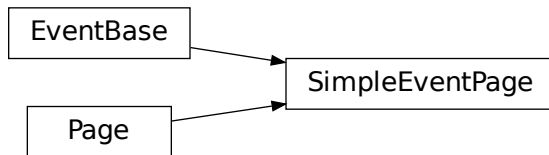
_getFromDt ()

Datetime that the event starts (in the local time zone).

_getToDt ()

Datetime that the event ends (in the local time zone).

SimpleEventPage



```

class ls.joyous.models.SimpleEventPage(id, path, depth, numchild, translation_key, locale, title, draft_title, slug, content_type, live, has_unpublished_changes, url_path, owner, seo_title, show_in_menus, search_description, go_live_at, expire_at, expired, locked, locked_at, locked_by, first_published_at, last_published_at, latest_revision_created_at, live_revision, alias_of, page_ptr, uid, category, image, time_from, time_to, tz, group_page, details, location, website, date)
    
```

Bases: `ls.joyous.models.event_base.EventBase`, `wagtail.core.models.Page`

date

The date that the event occurs on.

when

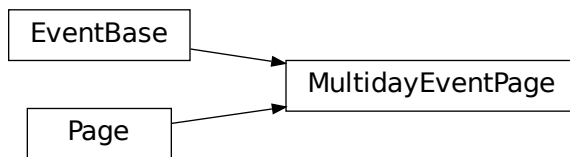
A string describing when the event occurs (in the local time zone).

`_getFromTime` (*atDate=None*)
Time that the event starts (in the local time zone).

`_getFromDt` ()
Datetime that the event starts (in the local time zone).

`_getToDt` ()
Datetime that the event ends (in the local time zone).

MultidayEventPage



class `ls.joyous.models.MultidayEventPage` (*id, path, depth, numchild, translation_key, locale, title, draft_title, slug, content_type, live, has_unpublished_changes, url_path, owner, seo_title, show_in_menus, search_description, go_live_at, expire_at, expired, locked, locked_at, locked_by, first_published_at, last_published_at, latest_revision_created_at, live_revision, alias_of, page_ptr, uid, category, image, time_from, time_to, tz, group_page, details, location, website, date_from, date_to*)

Bases: `ls.joyous.models.event_base.EventBase`, `wagtail.core.models.Page`

`date_from`
The date the event starts.

`date_to`
The date the event finishes.

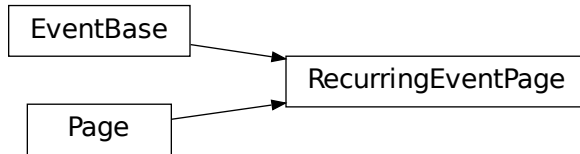
`when`
A string describing when the event occurs (in the local time zone).

`_getFromTime` (*atDate=None*)
Time that the event starts (in the local time zone).

`_getFromDt` ()
Datetime that the event starts (in the local time zone).

`_getToDt` ()
Datetime that the event ends (in the local time zone).

RecurringEventPage



```
class ls.joyous.models.RecurringEventPage(id, path, depth, numchild, translation_key, locale, title, draft_title, slug, content_type, live, has_unpublished_changes, url_path, owner, seo_title, show_in_menus, search_description, go_live_at, expire_at, expired, locked, locked_at, locked_by, first_published_at, last_published_at, latest_revision_created_at, live_revision, alias_of, page_ptr, uid, category, image, time_from, time_to, tz, group_page, details, location, website, repeat, num_days)
```

Bases: `ls.joyous.models.event_base.EventBase`, `wagtail.core.models.Page`

repeat

The recurrence rule of when the event occurs.

num_days

The number of days an occurrence lasts for.

next_date

Date when this event is next scheduled to occur in the local time zone (Does not include postponements, but does exclude cancellations)

_current_datetime_from

The datetime this event will start or did start in the local timezone, or None if it is finished.

_future_datetime_from

The datetime this event next starts in the local timezone, or None if in the past.

prev_date

Date when this event last occurred in the local time zone (Does not include postponements, but does exclude cancellations)

_past_datetime_from

The datetime this event previously started in the local time zone, or None if it never did.

_first_datetime_from

The datetime this event first started in the local time zone, or None if it never did.

status

The current status of the event (started, finished or pending).

status_text

A text description of the current status of the event.

when

A string describing when the event occurs (in the local time zone).

__getFromTime (*atDate=None*)

What was the time of this event? Due to time zones that depends what day we are talking about. If no day is given, assume today.

Parameters *atDate* – day in local timezone for which we want the *time_from*

Return type *time_from* in local timezone

__futureExceptions (*request*)

Returns all future extra info, cancellations and postponements created for this recurring event

__nextOn (*request*)

Formatted date/time of when this event (including any postponements) will next be on

__occursOn (*myDate*)

Returns true iff an occurrence of this event starts on this date (given in the event's own timezone).

(Does not include postponements, but does exclude cancellations.)

__getMyFirstDatetimeFrom ()

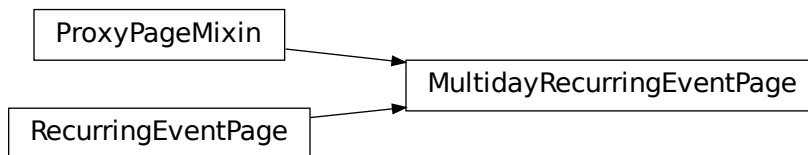
The datetime this event first started, or None if it never did.

__getMyFirstDatetimeTo (*myFirstDt=False*)

The datetime this event first finished, or None if it never did.

__getMyNextDate ()

Date when this event is next scheduled to occur in my time zone (Does not include postponements, but does exclude cancellations)

MultidayRecurringEventPage

```
class ls.joyous.models.MultidayRecurringEventPage (*args, **kwargs)
```

```
Bases: ls.joyous.utils.mixins.ProxyPageMixin, ls.joyous.models.recurring_events.RecurringEventPage
```

A proxy of RecurringEventPage that exposes the hidden *num_days* field.

EventExceptionBase



```
class ls.joyous.models.EventExceptionBase (*args, **kwargs)
```

```
Bases: django.db.models.base.Model
```

overrides

The recurring event that we are updating. overrides is also the parent (the published version of parent), but the parent is not set until the child is saved and added.

num_days

Shortcut for overrides.num_days.

time_from

Shortcut for overrides.time_from.

time_to

Shortcut for overrides.time_to.

tz

Shortcut for overrides.tz.

group

Shortcut for overrides.group.

category

Shortcut for overrides.category.

image

Shortcut for overrides.image.

location

Shortcut for overrides.location.

website

Shortcut for overrides.website.

at

A string describing what time the event starts (in the local time zone).

overrides_repeat

The recurrence rule of the event being overridden.

get_context (*request*, *args, **kwargs)

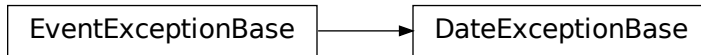
isAuthorized (*request*)

Is the user authorized for the requested action with this event?

_copyFieldsFromParent (*parent*)

Copy across field values from the recurring event parent.

DateExceptionBase



```

class ls.joyous.models.DateExceptionBase (*args, **kwargs)
  Bases: ls.joyous.models.recurring_events.EventExceptionBase

  except_date
    For this date.

  local_title
    Localised version of the human-readable title of the page.

  when
    A string describing when the event occurs (in the local time zone).

  full_clean (*args, **kwargs)
    Apply fixups that need to happen before per-field validation occurs. Sets the page's title.

  _getLocalWhen (date_from, num_days=1)
    Returns a string describing when the event occurs (in the local time zone).

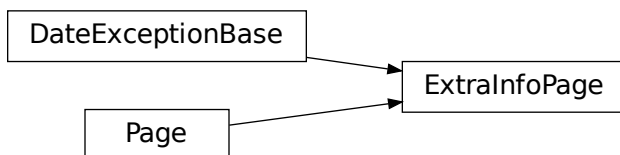
  _getFromTime (atDate=None)
    Time that the event starts (in the local time zone).

  _getFromDt ()
    Datetime that the event starts (in the local time zone).

  _getToDt ()
    Datetime that the event ends (in the local time zone).

  _copyFieldsFromParent (parent)
    Copy across field values from the recurring event parent.
  
```

ExtraInfoPage



```
class ls.joyous.models.ExtraInfoPage(id, path, depth, numchild, translation_key, locale, title, draft_title, slug, content_type, live, has_unpublished_changes, url_path, owner, seo_title, show_in_menus, search_description, go_live_at, expire_at, expired, locked, locked_at, locked_by, first_published_at, last_published_at, latest_revision_created_at, live_revision, alias_of, page_ptr, overrides, except_date, extra_title, extra_information)
```

Bases: `ls.joyous.models.recurring_events.DateExceptionBase`, `wagtail.core.models.Page`

extra_title

A more specific title for this occurrence (optional).

extra_information

Information just for this date.

details

Shortcut for `overrides.details`.

status

The current status of the event (started, finished or pending).

status_text

A text description of the current status of the event.

_current_datetime_from

The datetime this event will start or did start in the local timezone, or None if it is finished.

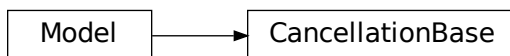
_future_datetime_from

The datetime this event next starts in the local timezone, or None if in the past.

_past_datetime_from

The datetime this event previously started in the local timezone, or None if it never did.

CancellationBase



```
class ls.joyous.models.CancellationBase(*args, **kwargs)
```

Bases: `django.db.models.base.Model`

cancellation_title

Show in place of cancelled event (Leave empty to show nothing).

cancellation_details

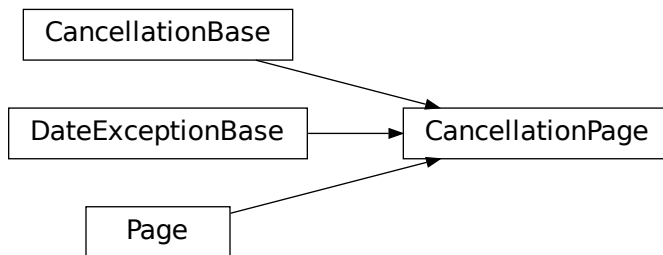
Why was the event cancelled?

status

The current status of the event

status_text

A text description of the current status of the event.

CancellationPage

```

class ls.joyous.models.CancellationPage(id, path, depth, numchild, translation_key, locale, title, draft_title, slug, content_type, live, has_unpublished_changes, url_path, owner, seo_title, show_in_menus, search_description, go_live_at, expire_at, expired, locked, locked_at, locked_by, first_published_at, last_published_at, latest_revision_created_at, live_revision, alias_of, page_ptr, overrides, except_date, cancellation_title, cancellation_details)
  
```

Bases: `ls.joyous.models.recurring_events.CancellationBase`, `ls.joyous.models.recurring_events.DateExceptionBase`, `wagtail.core.models.Page`

`_current_datetime_from`

The datetime the event that was cancelled would start in the local timezone, or None if it would have finished by now.

`_future_datetime_from`

The datetime the event that was cancelled would start in the local timezone, or None if that is in the past.

`_past_datetime_from`

The datetime of the event that was cancelled in the local timezone, or None if it never would have.

`getCancellationUrl` (*request=None*)

The URL to a page describing the cancellation for Postponements and plain Cancellations.

`cancellation_url`

The URL to a page describing the cancellation for Postponements and plain Cancellations.

RescheduleEventBase



class `ls.joyous.models.RescheduleEventBase` (**args, **kwargs*)

Bases: `ls.joyous.models.event_base.EventBase`

This class exists just so that the class attributes defined here are picked up by Django before the properties from `EventExceptionBase`, as well as before `EventBase`.

num_days

The number of days an occurrence lasts for.

tz

Shortcut for `overrides.tz`.

group

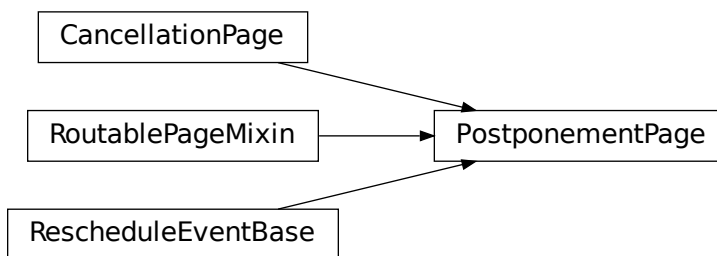
Shortcut for `overrides.group`.

uid

Shortcut for `overrides.uid`.

get_context (*request, *args, **kwargs*)

PostponementPage



```

class ls.joyous.models.PostponementPage(id, path, depth, numchild, translation_key, locale, title, draft_title, slug, content_type, live, has_unpublished_changes, url_path, owner, seo_title, show_in_menus, search_description, go_live_at, expire_at, expired, locked, locked_at, locked_by, first_published_at, last_published_at, latest_revision_created_at, live_revision, alias_of, page_ptr, overrides, except_date, cancellation_title, cancellation_details, cancellation_page_ptr, category, image, time_from, time_to, details, location, website, num_days, postponement_title, date)

Bases:      wagtail.contrib.routable_page.models.RoutablePageMixin,      ls.joyous.models.recurring_events.RescheduleEventBase,      ls.joyous.models.recurring_events.CancellationPage

postponement_title
    The title for the postponed event.

date
    The date that the event was postponed to.

status
    The current status of the event (started, finished or pending).

when
    A string describing when the postponement occurs (in the local time zone).

postponed_from_when
    A string describing when the event was postponed from (in the local time zone).

what
    May return a 'postponed' or 'rescheduled' string depending what the start and finish time of the event has been changed to.

postponed_from
    Date that the event was postponed from (in the local time zone).

postponed_to
    Date that the event was postponed to (in the local time zone).

at
    A string describing what time the event starts (in the local time zone).

serveCancellation (request)

_getFromTime (atDate=None)
    Time that the postponement starts (in the local time zone).

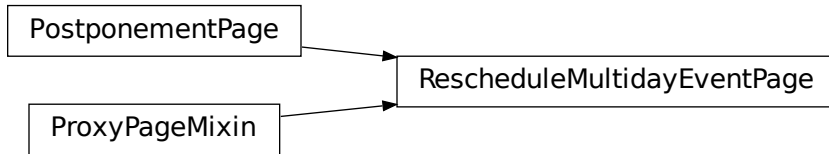
_getFromDt ()
    Datetime that the postponement starts (in the local time zone).

_getToDt ()
    Datetime that the postponement ends (in the local time zone).

_copyFieldsFromParent (parent)
    Copy across field values from the recurring event parent.

```

RescheduleMultidayEventPage

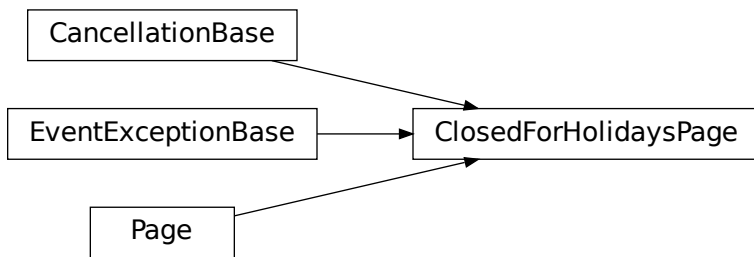


```

class ls.joyous.models.RescheduleMultidayEventPage(*args, **kwargs)
    Bases: ls.joyous.utils.mixins.ProxyPageMixin, ls.joyous.models.recurring_events.PostponementPage
    
```

A proxy of PostponementPage that exposes the hidden num_days field.

ClosedForHolidaysPage



```

class ls.joyous.models.ClosedForHolidaysPage(id, path, depth, numchild, translation_key,
    locale, title, draft_title, slug, content_type, live, has_unpublished_changes,
    url_path, owner, seo_title, show_in_menus, search_description, go_live_at, expire_at,
    expired, locked, locked_at, locked_by, first_published_at, last_published_at,
    latest_revision_created_at, live_revision, alias_of, page_ptr, overrides,
    cancellation_title, cancellation_details, all_holidays)
    
```

Bases: ls.joyous.models.recurring_events.CancellationBase, ls.joyous.models.recurring_events.EventExceptionBase, wagtail.core.models.Page

all_holidays

Closed for all holidays?

closed_for

Or, closed for these holidays

local_title

Localised version of the human-readable title of the page.

when

A string describing when the event occurs (in the local time zone).

closed

The list of holidays we are closed for, or “ALL” if we are closed for all holidays.

__current_datetime_from

The datetime the event that was cancelled would start in the local timezone, or None if it would have finished by now.

__future_datetime_from

The datetime the event that was cancelled would start in the local timezone, or None if that is in the past.

__past_datetime_from

The datetime of the event that was cancelled in the local timezone, or None if it never would have.

__closed_for_dates

Return all the dates which we are closed for in the local timezone

classmethod can_create_at (*parent*)

Checks if this page type can be created as a subpage under a parent page instance.

__getMyDates ()

Return all the dates which we are closed for in the event timezone

__getFromTime (*atDate=None*)

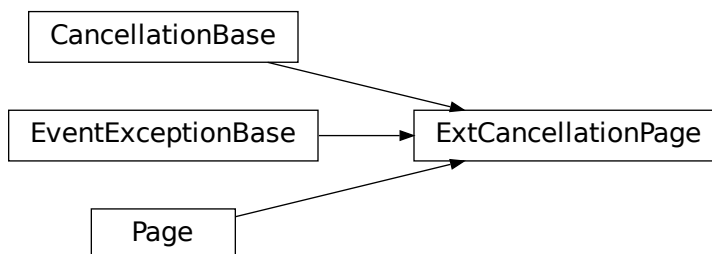
Time that the event starts (in the local time zone).

__cacheClosedSet ()

Cache the set of names of the holidays we are closed for. This needs to be re-called if the page’s fields change.

__closedOn (*myDate*)

Are we closed for holidays on myDate?

ExtCancellationPage

class `ls.joyous.models.ExtCancellationPage` (*id, path, depth, numchild, translation_key, locale, title, draft_title, slug, content_type, live, has_unpublished_changes, url_path, owner, seo_title, show_in_menus, search_description, go_live_at, expire_at, expired, locked, locked_at, locked_by, first_published_at, last_published_at, latest_revision_created_at, live_revision, alias_of, page_ptr, overrides, cancellation_title, cancellation_details, cancelled_from_date, cancelled_to_date*)

Bases: `ls.joyous.models.recurring_events.CancellationBase`, `ls.joyous.models.recurring_events.EventExceptionBase`, `wagtail.core.models.Page`

cancelled_from_date

Cancelled from this date

cancelled_to_date

Cancelled to this date (Leave empty for “until further notice”)

local_title

Localised version of the human-readable title of the page.

until_when

A string describing how long we are cancelled for

when

A string describing when the event occurs (in the local time zone).

__current_datetime_from

The datetime the event that was cancelled would start in the local timezone, or None if it would have finished by now.

__future_datetime_from

The datetime the event that was cancelled would start in the local timezone, or None if that is in the past.

__past_datetime_from

The datetime of the event that was cancelled in the local timezone, or None if it never would have.

full_clean (**args, **kwargs*)

Apply fixups that need to happen before per-field validation occurs. Sets the page’s title.

__getMyDates (*fromDate=datetime.date(1, 1, 1), toDate=datetime.date(3000, 12, 31)*)

Return all the dates which we are cancelled for in the event timezone. Only if the event occurs on those dates. Limited by fromDate and toDate if given.

__getMyRawDates (*fromDate=datetime.date(1, 1, 1), toDate=datetime.date(3000, 12, 31)*)

Return all the dates which we are cancelled for in the event timezone. Without considering if the event occurs on those dates. Limited by fromDate and toDate if given.

__getFromTime (*atDate=None*)

Time that the event starts (in the local time zone).

__closedOn (*myDate*)

Are we closed on myDate?

4.1.3 Groups

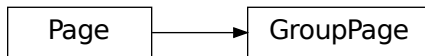
`ls.joyous.models.get_group_model_string()`

Get the dotted `app.Model` name for the group model as a string. Useful for developers that need to refer to the group model, such as in foreign keys, but the model itself is not required.

```
ls.joyous.models.get_group_model()
```

Get the group model from the `JOYOUS_GROUP_MODEL` setting. Useful for developers that need the group model. Defaults to the standard `ls.joyous.models.GroupPage` model if no custom model is defined.

GroupPage



```
class ls.joyous.models.GroupPage(id, path, depth, numchild, translation_key, locale, title, draft_title, slug, content_type, live, has_unpublished_changes, url_path, owner, seo_title, show_in_menus, search_description, go_live_at, expire_at, expired, locked, locked_at, locked_by, first_published_at, last_published_at, latest_revision_created_at, live_revision, alias_of, page_ptr, content)
```

Bases: `wagtail.core.models.Page`

content

An area of text for whatever you like

4.2 Template Tags

4.2.1 Tags

```
ls.joyous.templatetags.joyous_tags.events_this_week(context)
```

Displays a week's worth of events. Starts week with Monday, unless today is Sunday.

```
ls.joyous.templatetags.joyous_tags.minicalendar(context)
```

Displays a little ajax version of the calendar.

```
ls.joyous.templatetags.joyous_tags.all_upcoming_events(context)
```

Displays a list of all upcoming events.

```
ls.joyous.templatetags.joyous_tags.subsite_upcoming_events(context)
```

Displays a list of all upcoming events in this site.

```
ls.joyous.templatetags.joyous_tags.group_upcoming_events(context, group=None)
```

Displays a list of all upcoming events that are assigned to a specific group. If the group is not specified it is assumed to be the current page.

```
ls.joyous.templatetags.joyous_tags.future_exceptions(context, rrevent=None)
```

Displays a list of all the future exceptions (extra info, cancellations and postponements) for a recurring event. If the recurring event is not specified it is assumed to be the current page.

```
ls.joyous.templatetags.joyous_tags.next_on(context, rrevent=None)
```

Displays when the next occurrence of a recurring event will be. If the recurring event is not specified it is assumed to be the current page.

`ls.joyous.templatetags.joyous_tags.location_gmap` (*context, location*)
Display a link to Google maps iff we are using WagtailGMaps

4.2.2 Filters

`ls.joyous.templatetags.joyous_tags.time_display` (*time*)
format the time in a readable way

`ls.joyous.templatetags.joyous_tags.at_time_display` (*time*)
format as being “at” some time

`ls.joyous.templatetags.joyous_tags.date_display` (*date*)
format the date in a readable way

4.3 Formats

4.3.1 iCal

class `ls.joyous.formats.ICalHandler`
Serve and load iCalendar files

4.3.2 Google

class `ls.joyous.formats.GoogleCalendarHandler`
Redirect to a new Google Calendar event

4.3.3 RSS

class `ls.joyous.formats.RssHandler`
Serve a RSS Feed

4.3.4 Null

class `ls.joyous.formats.NullHandler`
Serve nothing (when nothing else matches)

4.4 Utils

4.4.1 Many Things

`ls.joyous.utils.manythings.hrJoin` (*items*)
Joins items together in a human readable string e.g. ‘wind, ice and fire’

`ls.joyous.utils.manythings.toOrdinal` (*n*)
Returns the ordinal name of a number e.g. ‘first’

`ls.joyous.utils.manythings.toTheOrdinal` (*n, inTitleCase=True*)
Returns the definite article with the ordinal name of a number e.g. ‘the second’ Becomes important for languages with multiple definite articles (e.g. French)

4.4.2 Mix-ins

class `ls.joyous.utils.mixins.ProxyPageMixin` (**args*, ***kwargs*)

Adding this mixin allows inheritance without needing a new table. The proxy model has its own content type which allows it to be selected as a separate page type in the Wagtail admin interface. No change is made to the manager, but `peers()` will return a queryset of others of the same type. See also <https://github.com/wagtail/wagtail/pull/1736>

classmethod `peers` ()

Return others of the same type

4.4.3 Names

`ls.joyous.utils.names.MONDAY_TO_SUNDAY` = ('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday')
Names of days of the week, from Monday to Sunday

`ls.joyous.utils.names.MONTH_ABBRS` = ('', 'Jan.', 'Feb.', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December')
Abbreviations of the months, with Jan at index 1

`ls.joyous.utils.names.MONTH_NAMES` = ('', 'January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December')
Names of the months, with January at index 1

`ls.joyous.utils.names.MON_TO_SUN` = ('Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun')
Abbreviations of days of the week, from Mon to Sun

`ls.joyous.utils.names.SUNDAY_TO_SATURDAY` = ('Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday')
Names of days of the week, from Sunday to Saturday

`ls.joyous.utils.names.SUN_TO_SAT` = ('Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat')
Abbreviations of days of the week, from Sun to Sat

`ls.joyous.utils.names.WEEKDAY_ABBRS` = ('Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun')
Abbreviations of days of the week, from Mon to Sun

`ls.joyous.utils.names.WEEKDAY_NAMES` = ('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday')
Names of days of the week, from Monday to Sunday

`ls.joyous.utils.names.WEEKDAY_NAMES_PLURAL` = ('Mondays', 'Tuesdays', 'Wednesdays', 'Thursdays', 'Fridays', 'Saturdays', 'Sundays')
The names of days of the week in plural, Mondays to Sundays

`ls.joyous.utils.names.WRAPPED_MONTH_NAMES` = ('December', 'January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December', 'January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December')
Names of the months, December, January, ... December, January

class `ls.joyous.utils.names._Names`

Holds internationalized names and only translates them at the last possible moment, when they are accessed.

4.4.4 Recurrence

class `ls.joyous.utils.recurrence.Recurrence` (**args*, ***kwargs*)

Implementation of the recurrence rules somewhat based upon [RFC 5545](#) RRules, implemented using `dateutil.rule`.

Does not support timezones ... and probably never will. Does not support a frequency of by-hour, by-minute or by-second.

byeaster

An offset from Easter Sunday.

bymonth

The months where the recurrence will be applied.

bymonthday

The month days where the recurrence will be applied.

bysetpos

The nth occurrence of the rule inside the frequency period.

byweekday

The weekdays where the recurrence will be applied. In RFC5545 this is called BYDAY, but is renamed by dateutil to avoid ambiguity.

byweekno

The week numbers to apply the recurrence to.

byyearday

The year days to apply the recurrence to.

count

Limit on the number of occurrences.

dtstart

The recurrence start date.

freq

How often the recurrence repeats. (0,1,2,3)

frequency

How often the recurrence repeats. ("YEARLY", "MONTHLY", "WEEKLY", "DAILY")

getCount ()

How many occurrences will be generated.

interval

The interval between each freq iteration.

until

The last occurrence in the rule is the greatest date that is less than or equal to the value specified in the until parameter.

wkst

The week start day. The default week start is got from calendar.firstweekday() which Joyous sets based on the Django FIRST_DAY_OF_WEEK format.

class `ls.joyous.utils.recurrence.Weekday` (*wkday, n=None*)

Represents a day of the week, for every occurrence of the week or for a specific week in the period. e.g. The first Friday of the month.

4.4.5 Tell time

`ls.joyous.utils.telltime._timeFormat` (*when, formatStr*)

Format a single time, e.g. 10am

`ls.joyous.utils.telltime.dateFormat` (*when*)

Format the date when, e.g. Friday 14th of April 2011

Uses the format given by JOYOUS_DATE_FORMAT if that is set, or otherwise the standard Django date format.

`ls.joyous.utils.telltime.dateShortFormat` (*when*)

Short version of the date when, e.g. 14 April 2017

Uses the format given by JOYOUS_DATE_SHORT_FORMAT if that is set, or otherwise the standard Django date format.

```
ls.joyous.utils.telldtime.getAwareDatetime (date, time, tz, timeDefault=datetime.time(23,
                                          59, 59, 999999))
```

Get a datetime in the given timezone from date and optionally time. If time is not given it will default to timeDefault if that is given or if not then to the end of the day.

```
ls.joyous.utils.telldtime.getLocalDate (*args, **kwargs)
```

Get the date in the local timezone from date and optionally time

```
ls.joyous.utils.telldtime.getLocalDateAndTime (date, time, *args, **kwargs)
```

Get the date and time in the local timezone from date and optionally time

```
ls.joyous.utils.telldtime.getLocalDatetime (date, time, tz=None, timeDe-
                                          fault=datetime.time(23, 59, 59, 999999))
```

Get a datetime in the local timezone from date and optionally time

```
ls.joyous.utils.telldtime.getLocalTime (date, time, *args, **kwargs)
```

Get the time in the local timezone from date and time

```
ls.joyous.utils.telldtime.getLocalTimeAtDate (atDate, time, *args, **kwargs)
```

Get the time at a certain date in the local timezone

```
ls.joyous.utils.telldtime.getTimeFrom (time_from)
```

Return time_from if it is set, otherwise return the start of the day

```
ls.joyous.utils.telldtime.getTimeTo (time_to)
```

Return time_to if it is set, otherwise return the end of the day

```
ls.joyous.utils.telldtime.timeFormat (time_from, time_to=None, prefix="", infix=None)
```

Format the times time_from and optionally time_to, e.g. 10am

Uses the format given by JOYOUS_TIME_FORMAT if that is set, or otherwise the standard Django time format.

```
ls.joyous.utils.telldtime.todayUtc ()
```

The current date in the UTC timezone

4.4.6 Weeks

Sets “Monday” or “Sunday” as the first day of the week depending upon the Django FIRST_DAY_OF_WEEK format.

```
ls.joyous.utils.weeks.gregorian_to_week_date (date_value)
```

year, week and day for the given Gregorian calendar date

```
ls.joyous.utils.weeks.num_weeks_in_year (sweek_year)
```

Get the number of weeks in this year

```
ls.joyous.utils.weeks.week_info (sweek_year, sweek_week)
```

Give all the info we need from one calculation (first_day, last_day, prev_year_num_weeks, year_num_weeks)

```
ls.joyous.utils.weeks.week_of_month (date_value)
```

Returns a 0-starting index of which week in the month this date is

```
ls.joyous.utils.weeks.weekday_abbr = ('Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat')
```

Abbreviations of the days of the week

```
ls.joyous.utils.weeks.weekday_name = ('Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday')
```

Names of the days of the week

4.5 Admin User Interface

4.5.1 Edit Handlers

class `ls.joyous.edit_handlers.TZDatePanel` (*field_name*, *args, **kwargs)

Will display the timezone of the date if it is not the current TZ

class `ls.joyous.edit_handlers.ExceptionDatePanel` (*field_name*, *args, **kwargs)

Used to select from the dates of the recurrence

class `ls.joyous.edit_handlers.TimePanel` (*field_name*, *args, **kwargs)

Used to select time using either a 12 or 24 hour time widget

class `ls.joyous.edit_handlers.ConcealedPanel` (*children*, *heading*, *classname=""*,
help_text="")

A panel that can be hidden

4.5.2 Widgets

class `ls.joyous.widgets.Time12hrInput` (*attrs=None*)

Display and Edit time fields in a 12hr format

class `ls.joyous.widgets.RecurrenceWidget` (*attrs=None*)

Widget for entering the rule of a recurrence

class `ls.joyous.widgets.ExceptionDateInput` (*attrs=None*, *format=None*)

Display and Edit the dates which are the exceptions to a recurrence rule

4.6 Fields

class `ls.joyous.fields.RecurrenceField` (*args, **kwargs)

DB Field for recurrences

class `ls.joyous.fields.MultipleSelectField` (*args, **kwargs)

Field with multiple *static* choices (not via m2m)

From <https://gist.github.com/kottenator/9a50e4207cff15c03f8e> by Rostyslav Bryzgunov

Value is stored in DB as comma-separated values Default widget is `forms.CheckboxSelectMultiple` Python value: list of values

4.7 Middleware

class `ls.joyous.middleware.UserTimeZoneMiddleware` (*get_response=None*)

If there is a Wagtail user with their time zone set then activate that time zone for all pages (not just the Wagtail Admin).

4.8 Holidays

class `ls.joyous.holidays.Holidays` (*holidaySetting='JOYOUS_HOLIDAYS'*)

Defines what holidays are celebrated on what dates.

- add** (*date*, *value*)
Add a holiday to an individual date.
- get** (*date*)
Get all the holidays that are celebrated on this date.
- names** ()
Get a list of all the holiday names, sorted by month-day.
- register** (*src*)
Register a new source of holiday data.

4.9 CSS

All Joyous CSS classes begin with the prefix `joy-` and try to follow the BEM protocol of `block__element--modifier`.

4.9.1 Blocks

```
.joy-overlay  
.joy-popup  
.joy-content  
.joy-title  
.joy-cal  
.joy-cal-list  
.joy-minical  
.joy-this-week  
.joy-view-choices  
.joy-days-events  
.joy-export  
.joy-ev-item  
.joy-field  
.joy-img-link  
.joy-img  
.joy-ev-who  
.joy-ev-when  
.joy-ev-from-when  
.joy-ev-to-when  
.joy-ev-next-on  
.joy-ev-where  
.joy-ev-website  
.joy-ev-details  
.joy-ev-extra  
.joy-ev-status  
.joy-ev-related  
.joy-grp-list  
.joy-pg  
.joy-rr
```

4.9.2 Elements

```
.joy-title__link
.joy-popup__outer
.joy-popup__content
.joy-popup__close
.joy-view-choices__link
.joy-cal__head
.joy-cal__headings
.joy-cal__heading
.joy-cal__period-heading
.joy-cal__period-ctrl
.joy-cal__month-name
.joy-cal__week-name
.joy-cal__year-number
.joy-cal__weekdays
.joy-cal__weekday
.joy-cal__body
.joy-cal__week
.joy-cal__no-day
.joy-cal__day
.joy-cal__read-more
.joy-cal__day-title
.joy-cal__date
.joy-cal__holiday-name
.joy-minical__head
.joy-minical__headings
.joy-minical__heading
.joy-minical__month-heading
.joy-minical__month-name
.joy-minical__year-number
.joy-minical__weekdays
.joy-minical__weekday
.joy-minical__body
.joy-minical__week
.joy-minical__day
.joy-minical__no-day
.joy-minical__day-title
.joy-minical__date
.joy-this-week__title
.joy-this-week__events
.joy-this-week__day
.joy-this-week__day-title
.joy-this-week__date
.joy-this-week__weekday
.joy-this-week__holiday-name
.joy-this-week__days-events
.joy-this-week__cal
.joy-this-week__cal-link
.joy-days-events__event
.joy-days-events__event-time
.joy-days-events__event-title
.joy-days-events__event-text
.joy-img-link__img
.joy-img__img
.joy-ev-who__link
.joy-ev-to-when__link
.joy-ev-where__map-link
.joy-ev-website__link
```

(continues on next page)

(continued from previous page)

```
.joy-ev-related__item
.joy-ev-related__link
.joy-ev-related__overrides
.joy-pg__page
.joy-pg__ctrl
.joy-pg__num
.joy-rr__advanced-repeat
.joy-rr__advanced-monthly-repeat
.joy-rr__advanced-weekly-repeat
.joy-rr__advanced-yearly-repeat
.joy-rr__day-choice
.joy-rr__days-input
.joy-rr__short-input
.joy-rr__months-input
.joy-rr__freq-choice
.joy-rr__half-field
.joy-rr__double-field
.joy-rr__ical-value
.joy-rr__interval-num
.joy-rr__interval-units-days
.joy-rr__interval-units-months
.joy-rr__interval-units-weeks
.joy-rr__interval-units-years
.joy-rr__months
.joy-rr__ord-choice
.joy-rr__primary
.joy-rr__secondary
.joy-rr__show-advanced
.joy-rr__show-advanced-cbx
.joy-rr__simple-repeat
.joy-rr__start-date
.joy-rr__until-date
.joy-rr__weekdays
```

4.9.3 Modifiers

```
.joy-title--page
.joy-title--list
.joy-title--item
.joy-cal--weekly
.joy-cal--monthly
.joy-cal__weekday--sun
.joy-cal__day--today
.joy-cal__day--yesterday
.joy-cal__day--lastweek
.joy-cal__day--first
.joy-cal__day--last
.joy-cal__day--has-events
.joy-cal__date--holiday
.joy-cal__date--sun
.joy-minical__weekday--sun
.joy-minical__day--today
.joy-minical__date--holiday
.joy-minical__date--sun
.joy-minical__date--event-link
```

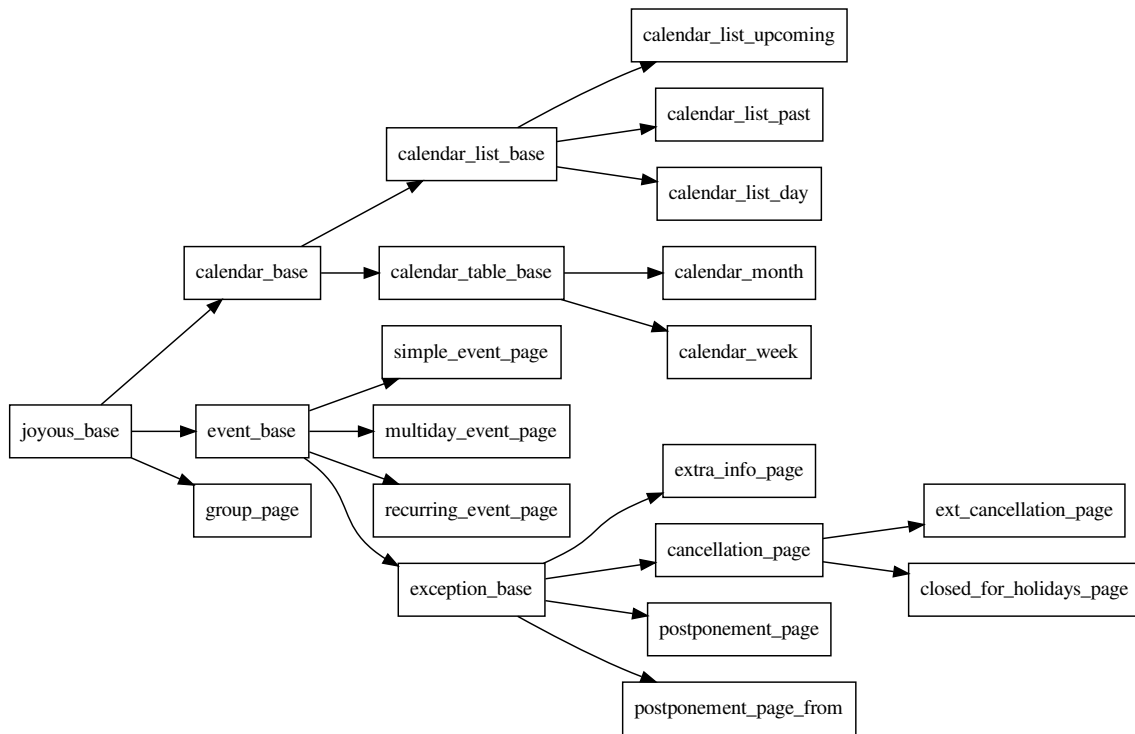
(continues on next page)

(continued from previous page)

```
.joy-minical__date--event
.joy-this-week__day--in-past
.joy-this-week__day--today
.joy-days-events__event--continues
.joy-cal-list--upcoming
.joy-cal-list--past
.joy-cal-list--day
.joy-pg__ctrl--disabled
.joy-pg__num--active
```

4.10 Templates

4.10.1 Hierarchy



4.10.2 Templates

`joyous/joyous_base.html`

Extends : `base.html`

The blocks **content**, **extra_css** and **extra_js** are required in the `base.html` template for the Joyous templates to work. A Wagtail project based upon the [default template](#) will have these.

Blocks

- **body_class**
- **extra_css**
- **content**
- **extra_js**

joyous/calendar_base.html

Extends : joyous/joyous_base.html

Blocks

- **content**
 - **cal_all**
 - * **cal_intro**
 - * **view_choices**
 - * **cal_subtitle**
 - * **cal_events**
 - * **cal_footer**

joyous/calendar_list_base.html

Extends : joyous/calendar_base.html

Blocks

- **cal_events**
 - **cal_view_class**
 - **event_item**
 - **events_pagination**

Includes

- joyous/includes/event_item.html

joyous/calendar_list_upcoming.html

Extends : joyous/calendar_list_base.html

Blocks

- `view_choices`
- `cal_view_class`
- `cal_subtitle`
- `cal_footer`

Includes

- `joyous/includes/events_view_choices.html`

`joyous/calendar_list_past.html`

Extends : `joyous/calendar_list_base.html`

Blocks

- `view_choices`
- `cal_view_class`
- `cal_subtitle`

Includes

- `joyous/includes/events_view_choices.html`

`joyous/calendar_list_day.html`

Extends : `joyous/calendar_list_base.html`

Blocks

- `view_choices`
- `cal_view_class`
- `cal_subtitle`

Includes

- `joyous/includes/event_item.html`

`joyous/calendar_table_base.html`

Extends : `joyous/calendar_base.html`

Blocks

- **cal_events**
 - **cal_view_class**
 - **cal_thead**
 - * **cal_heading**
 - * **cal_weekday**
 - **cal_tbody**
 - * **cal_week**
 - **cal_day**
 - **cal_day_title**
 - **days_events**
- **extra_js**

Includes

- joyous/includes/events_view_choices.html
- joyous/includes/joyjq.html

joyous/calendar_month.html

Extends : joyous/calendar_table_base.html

Blocks

- **view_choices**
- **cal_view_class**
- **cal_heading**
- **cal_day_title**

Includes

- joyous/includes/events_view_choices.html

joyous/calendar_week.html

Extends : joyous/calendar_table_base.html

Blocks

- `view_choices`
- `cal_view_class`
- `cal_heading`
- `cal_day_title`

Includes

- `joyous/includes/events_view_choices.html`

`joyous/event_base.html`

Extends : `joyous/joyous_base.html`

Blocks

- **content**
 - `event_status`
 - `event_title`
 - `event_image`
 - `event_who`
 - `event_when`
 - `event_where`
 - `event_details`
 - `event_footer`

Includes

- `joyous/includes/who.html`
- `joyous/includes/when.html`
- `joyous/includes/where.html`

`joyous/simple_event_page.html`

Extends : `joyous/event_base.html`

`joyous/multiday_event_page.html`

Extends : `joyous/event_base.html`

[joyous/recurring_event_page.html](#)

Extends : joyous/event_base.html

Blocks

- **event_footer**

[joyous/exception_base.html](#)

Extends : joyous/event_base.html

Blocks

- **event_footer**

[joyous/extra_info_page.html](#)

Extends : joyous/exception_base.html

Blocks

- **event_title**
- **event_details**

[joyous/cancellation_page.html](#)

Extends : joyous/exception_base.html

Blocks

- **event_title**
- **event_details**

[joyous/postponement_page.html](#)

Extends : joyous/exception_base.html

Blocks

- **event_title**
- **event_details**

joyous/postponement_page_from.html

Extends : joyous/exception_base.html

Blocks

- **event_status**
- **event_title**
- **event_when**
- **event_image**
- **event_details**

joyous/ext_cancellation_page.html

Extends : joyous/cancellation_page.html

joyous/closed_for_holidays_page.html

Extends : joyous/cancellation_page.html

joyous/group_page.html

Extends : joyous/joyous_base.html

Blocks

- **content**

5.1 Joyous 1.4.0 release notes

- *What's new*
 - *Translations*
 - *Compatibility with Wagtail 2.11*
 - *Compatibility with Wagtail 2.12*
 - *Deprecated*
 - *Bug fixes*

5.1.1 What's new

Translations

Thanks to Tomas Walch for the new Swedish translation.

Compatibility with Wagtail 2.11

Now that 2.11 is the new LTS for Wagtail replace `request.site` with `Site.find_for_request(request)` removing the need for `wagtail.core.middleware.SiteMiddleware`. Thanks to Tomas Walch for this.

Compatibility with Wagtail 2.12

Deprecated

The `JOYOUS_DEFEND_FORMS` setting is deprecated. It still works, but is no longer documented.

Bug fixes

- FormDefender should cope with `base_form_class = None`
- Do not assimilate subclasses
- Fix for `_gregorian_to_ssw` where date is in last week of prev year
- Tests should cope with year missing from title of exceptions
- Stop using deprecated `is_ajax` function

5.2 Joyous 1.3.1 release notes

- *What's new*
 - *Compatibility with wagtail-modeltranslations*
 - *Compatibility with Wagtail 2.10*
 - *Bug fixes*
- *Upgrade considerations*

5.2.1 What's new

Compatibility with wagtail-modeltranslations

This release adds new setting `JOYOUS_DEFEND_FORMS`. If this is set to `True` then the Joyous page models will not allow their `base_form_class` to be replaced. Instead they will assimilate the newly assigned form class. This setting was added to work around a bug in the `wagtail-modeltranslations` app. The bug was fixed in `wagtail-modeltranslations` version 0.10.14. Upgrading to v0.10.14 or later is the preferred solution, rather than enabling `JOYOUS_DEFEND_FORMS`.

Init functions have been added to anything inheriting from `models.Model` to avoid `modeltranslation` `patch_constructor` breaking initialization of pages which have multiple inheritance.

Compatibility with Wagtail 2.10

Use the Django Admin version of JQuery

Bug fixes

- Times returned from `getLocalDateAndTime` should be naive so that they are compatible with `time.min` and `time.max`.

5.2.2 Upgrade considerations

5.3 Joyous 1.2.0 release notes

- *What's new*
 - *ExtCancellationPage*
 - *ClosedForHolidaysPage*
 - *Holidays*
 - *iCal*
 - *RSS*
 - *Other features*
 - *Bug fixes*
 - *Source code restructure*

5.3.1 What's new

ExtCancellationPage

Extended cancellations apply to all occurrences of a recurring event from the `cancelled_from_date` until the optional `cancelled_to_date`. If the to date is not given, then the event is cancelled “until further notice”. Named ExtCancellation pages are listed just like named Cancellations.

ClosedForHolidaysPage

A recurring event page can now have a closed for holidays page added to it. This indicates that event does not occur on holidays. All holidays or a specific list of holidays can apply. Named ClosedForHolidays pages are listed just like named Cancellations.

Holidays

- Joyous holidays can now be added together.
- The setting that is parsed for Holidays can be changed from `JOYOUS_HOLIDAYS`.
- Holidays can be passed to API functions and `RecurringEvent` to ensure everyone is using the same definition of holidays.

iCal

- When an event is closed for extended cancellations or on holidays these are exported as EXDATES.
- NOTE: Google Calendar will refuse to import an event with more than about 90 EXDATES. Investigation of this issue is still in progress.

RSS

- Named ExtCancellation pages and named ClosedForHolidays pages are listed just like named Cancellations.

Other features

- There is a new DateExceptionBase class that CancellationPage, PostponementPage and ExtraInfoPage inherit from.
- There is a new CancellationBase class that CancellationPage, PostponementPage, ExtCancellationPage, and ClosedForHolidaysPage inherit from.
- New getLocalTimeAtDate function.
- RecurringEventPage._getMyFirstDatetimeTo() now takes an optional myFirstDt argument to save from calculating this again.

Bug fixes

- When _getFromTime takes an atDate it is in the local time zone.
- Fix _getToDt to add on num_days
- Record no authority to update an exception as an iCal load failure
- Fix ExceptionDateInput to take account of JOYOUS_FIRST_DAY_OF_WEEK

Source code restructure

models/events.py was getting too large so has been broken up. Users of Joyous should not notice any difference.

5.4 Joyous 1.1.0 release notes

- *What's new*
 - *Translations*
 - *iCal*
 - *Other features*
 - *Bug fixes*

5.4.1 What's new

Translations

- Thanks to Yuki Kikuchi for the new Japanese translation.
- Thanks to Pandevmonium for the new Italian translation.

iCal

- Import whole zip files containing iCal files

Other features

- Provide defaults for otherwise undefined template variables

Bug fixes

- Return a 404 for invalid day dates
- Fix removeContentPanels to work with postponements
- Handle duplicate UIDs in a iCal import
- Include Cancellations (or Postponements) with the cancellation_title filled in in the RSS feed

5.5 Joyous 1.0.1 release notes

- *What's new*
 - CSS
 - *Bug fixes*
- *Upgrade considerations*

5.5.1 What's new

CSS

Added a new class `.joy-content` to the content block of all templates. This matches other Joyous classes in having the `joy-` prefix. The old `.content` class has been kept for backwards compatibility.

Bug fixes

- Return exception `datetime_from` properties (`_current_datetime_from`, `_future_datetime_from`, `_past_datetime_from`) in the local time zone (as per the documentation).
- When prepopulating a cancellation keep the next date in the event's timezone

5.5.2 Upgrade considerations

5.6 Joyous 1.0.0 release notes

- *What's new*
- *Compatibility*

This is a milestone release. Following [semantic versioning](#) this indicates Joyous now has a stable public API, and no incompatible changes to it will be made unless a 2.0 version is released.

5.6.1 What's new

Not much has changed since [version 0.9.5](#), but this release marks:

- 5 years since the beginnings of Joyous in my first events app based upon [Wagtaildemo](#)
- 2 years since the first commit of Joyous as a reusable calendar app
- 367 commits since then
- contributions from 5 [people](#)
- 100% test coverage
- 20100 lines of code (45% in tests)

5.6.2 Compatibility

Joyous version 1.0.0 is known to work with the following versions of Wagtail, Django and Python.

Wagtail	Django	Python
2.6.3	2.2.10	3.6.7
2.6.3	2.2.10	3.7.1
2.6.3	2.2.10	3.8.0
2.7.1	2.2.10	3.6.7
2.7.1	2.2.10	3.7.1
2.7.1	2.2.10	3.8.0
2.8	3.0.3	3.6.7
2.8	3.0.3	3.7.1
2.8	3.0.3	3.8.0

Other versions may work - YMMV.

5.7 Joyous 0.9.5 release notes

- *What's new*
 - *Option to include started events in the upcoming list*
 - *Wagtail 2.8*
 - *Other features*
 - *Bug fixes*

- *Upgrade considerations*

5.7.1 What's new

Option to include started events in the upcoming list

This release adds new setting `JOYOUS_UPCOMING_INCLUDES_STARTED`. If this is set to `True` then the list of upcoming events will also include events that have already started but have not yet finished.

New filters `current` for upcoming and started events and `future` for just upcoming events have been added to `EventQuerySet`. The old filter `upcoming` now just delegates to the appropriate one depending upon `JOYOUS_UPCOMING_INCLUDES_STARTED`.

New event properties `_current_datetime_from` and `_future_datetime_from` are used by the filters and for sorting the list of events. The old property `_upcoming_datetime_from` has been removed.

Wagtail 2.8

Support for Wagtail 2.8 (and Django 3.0).

Other features

- New `JOYOUS_FIRST_DAY_OF_WEEK` setting that will override Django's definition of the first day of the week if set.
- Remove unneeded `rss.ExtraInfoEntry.setImage()`
- Use naive datetimes in `PostponementPage.what()`
- Dynamically copy postponements field values (should work for derieved classes too)

Bug fixes

- Subtract `page.num_days` for continuuing events starting before `fromDate`
- Force a call to `_fetch_all` in `EventQuerySet.count()`

5.7.2 Upgrade considerations

- The old property `_upcoming_datetime_from` has been removed.

5.8 Joyous 0.9.4 release notes

- *What's new*
 - *List "named" Cancellations*
 - *Other features*

- *Bug fixes*
- *Upgrade considerations*

5.8.1 What's new

List “named” Cancellations

A Cancellation (or Postponement) with the `cancellation_title` filled in will be shown in the list of upcoming or past events at the date the event was cancelled. This matches how cancellations are already displayed in the monthly and weekly calendar views.

Other features

- Don't build unnecessary holiday objects

Bug fixes

- Fix weekly urls at the end of the year
- Use None to mark default byweekday and bymonth
- Count PermissionDenied errors in import as failures

5.8.2 Upgrade considerations

5.9 Joyous 0.9.3 release notes

- *What's new*
 - *Translations*
 - *CSS*
 - *Pytest*
 - *Documentation*
 - *Other features*
 - *Bug fixes*
- *Upgrade considerations*

5.9.1 What's new

Translations

Thanks to Alexandre Marinho for the new Brazilian Portuguese translation.

CSS

New modifier `.joy-cal__day--has-events` added to calendar days that have events. Thanks to JorneVL for requesting this.

Pytest

Option to run tests in parallel using pytest. Runs about twice as fast.

Documentation

Added a new tutorial - "A few easy changes".

Other features

- Accept `*` as a wildcard for countries in `JOYOUS_HOLIDAYS`.
- Add the Time12hr JS all in one place.
- Move category, image, location and website property shortcuts from `ExtraInfoPage` to `EventExceptionBase`.
- Move the block `event_image` from `exception_base.html` to `postponement_page_from.html`

Bug fixes

5.9.2 Upgrade considerations

5.10 Joyous 0.9.2 release notes

- *What's new*
 - *RSS*
 - *jQuery*
 - *Other features*
 - *Bug fixes*
- *Upgrade considerations*

5.10.1 What's new

RSS

An RSS Feed of the Upcoming Events is now available. Via a link at the bottom of the `/calendar/upcoming/` page, or by adding `?format=rss` to any calendar page URL.

jQuery

The Calendar pages and MiniCalendar tag have had an undocumented dependency on jQuery being loaded somewhere on the pages that they are on (e.g. in base.html). I have changed this so now they will load jQuery 3.2.1 as joyJQ via noConflict. This is done in the included template joyous/includes/joyjq.html.

```
{% load static %}
<script src="{% static 'wagtailadmin/js/vendor/jquery-3.2.1.min.js' %}"></
<script>
<script>joyJQ = $.noConflict(true);</script>
```

Just [override](#) this template if you don't like how it works. An empty file here will restore the previous behaviour. The Calendar and MiniCalendar will prefer using joyJQ for jQuery, but if that does not exist will fall back to \$.

The Admin RecurrenceWidget also has a dependency upon jQuery. But as jQuery is always available on Admin pages this is not a problem. The RecurrenceWidget will prefer using joyJQ for jQuery, but if that does not exist will fall back to \$, and if that does not exist will fall back to django.jQuery.

Other features

- Rename timeTo as getTimeTo, timeFrom as getTimeFrom
- MultidayRecurringEventPage now uses the template recurring_event_page.html
- RescheduleMultidayEventPage now uses the template postponement_page.html
- Improvements to documentation
- Better unit test coverage

Bug fixes

- Return [] if getGroupUpcomingEvents is called without a group
- Return [] if future_exceptions tag is called without a recurring event
- Use gettext in at_time_display filter
- dateShortFormat will use Django's SHORT_DATE_FORMAT if JOYOUS_DATE_SHORT_FORMAT is not set

5.10.2 Upgrade considerations

5.11 Joyous 0.9.1 release notes

- *What's new*
 - *Date and Time formats*
 - *Other features*
 - *Bug fixes*
- *Upgrade considerations*

5.11.1 What's new

Date and Time formats

New settings `JOYOUS_DATE_FORMAT`, `JOYOUS_DATE_SHORT_FORMAT` and `JOYOUS_TIME_FORMAT` now allow custom formatting of `dateFormat`, `dateShortFormat` (formerly known as `dateFormatYMD`) and `timeFormat`. (See *Settings* for details.) If a setting is not set then Joyous will default to using the standard Django formatting.

Use the following settings to keep the date and time formats as they were previously.

```
JOYOUS_DATE_FORMAT = "l jS \\o\\f F X"
JOYOUS_DATE_SHORT_FORMAT = "j F Y"
JOYOUS_TIME_FORMAT = "fq"
```

Other features

- Better unit test coverage.

Bug fixes

- In `_add12hrFormats` fix which locales have 12hr time formats added
- Documentation fixes
- Fix the CSS class of `exception_base` and `postponement_page` event images

5.11.2 Upgrade considerations

5.12 Joyous 0.9.0 release notes

- *What's new*
 - *CSS*
 - *Holidays*
 - *Other features*
 - *Bug fixes*
- *Upgrade considerations*
 - *CSS*

5.12.1 What's new

CSS

`joyous.css` has been stripped down to the bare structural basics that will display elements in appropriate places, but without pushing colour or font choices. New theme CSS files are available which the user can optionally choose to import if they wish. Added a new setting `JOYOUS_THEME_CSS` to make it easy to include a theme CSS file without needing to override the `joyous_base.html` file.

Add the following to your settings file (e.g. `mysite/mysite/settings/base.py`) to continue with the previous default appearance.

```
JOYOUS_THEME_CSS = "/static/joyous/css/joyous_coast_theme.css"
```

Available themes:

- `joyous_coast_theme.css`: Greys and gold.
- `joyous_forest_theme.css`: Greens.
- `joyous_stellar_theme.css`: A dark background theme.

All Joyous CSS classes now begin with the prefix `joy-` and try to follow the BEM protocol of `block__element--modifier`.

Holidays

Holidays are now a property of the `CalendarPage`. This means it is possible to derieve different Calendar models and give them different sets of holidays. Holidays for `CalendarPage` are still determined programmatically, but a derieved Calendar model could choose to change this, e.g. store the holidays in the database so that different pages of the same model could have different holidays.

The `JOYOUS_HOLIDAYS` setting still works the same - selecting holidays from `python-holidays`. But now, additional holiday sources (e.g. from `workalendar` or just a simple dict) can be registered via `register`. As can individual days via `add`.

Other features

- Restructured templates to use more inheritance and less inclusion.

Bug fixes

- Fix bug when `num_days` differs from event to occurrence
- Make sure we have translations for the names used in exception page titles. Always generate the title and slug in English, then translate in the property `local_title`.

5.12.2 Upgrade considerations

CSS

There has been substantial changes to the CSS selectors. This table shows some of them.

<code>#calendar-overlay</code>	<code>.joy-overlay</code>
<code>.day-popup-outer</code>	<code>.joy-popup__outer</code>
<code>.day-popup</code>	<code>.joy-popup__content</code>
<code>.day-popup .close</code>	<code>.joy-popup__close</code>
<code>.events-view</code>	<code>.joy-view-choices</code>
<code>.events-view .together</code>	<code>.joy-view-choices__link</code>
<code>table.calendar</code>	<code>.joy-cal</code>
<code>.calendar thead</code>	<code>.joy-cal__head</code>
<code>.calendar thead tr.heading th</code>	<code>.joy-cal__heading</code>

Continued on next page

Table 1 – continued from previous page

.calendar thead tr th	.joy-cal__weekday
.calendar .month-heading	.joy-cal__period-heading
.calendar .month-heading a	.joy-cal__period-ctrl
.calendar .month-heading .month-name	.joy-cal__month-name
.calendar .week-heading	.joy-cal__period-heading
.calendar .week-heading a	.joy-cal__period-ctrl
.calendar .week-heading .week-name	.joy-cal__week-name
.calendar .year-heading	.joy-cal__period-heading
.calendar .year-heading a	.joy-cal__period-ctrl
.calendar .year-heading .year-number	.joy-cal__year-number
.calendar tbody	.joy-cal__body
.calendar tbody tr	.joy-cal__week
.calendar tbody tr td	.joy-cal__day
.calendar tbody tr td.noday	.joy-cal__no-day
.calendar tbody tr td.today	.joy-cal__day-today
.calendar tbody tr td.yesterday	.joy-cal__day-yesterday
.calendar tbody tr td.lastweek	.joy-cal__day-lastweek
.calendar th.sun	.joy-cal__weekday-sun
.calendar .sun h4	.joy-cal__date-sun
.calendar .holiday h4	.joy-cal__date-holiday
.calendar .day-title	.joy-cal__day-title
.calendar .day-title .holiday-name	.joy-cal__holiday-name
.calendar td.day a.read-more	.joy-cal__read-more
.minicalendar	.joy-minical
.minicalendar thead	.joy-minical__head
.minicalendar thead tr.heading th	.joy-minical__heading
.minicalendar thead .month-heading	.joy-minical__month-heading
.minicalendar thead .month-name	.joy-minical__month-name
.minicalendar thead .year-number	.joy-minical__year-number
.minicalendar thead .month-nav	.joy-minical__month-nav
a.minicalPrev	.joy-minical-prev
a.minicalNext	.joy-minical-next
.minicalendar thead tr th	.joy-minical__weekday
.minicalendar tbody tr td	.joy-minical__day
.calendar tbody tr td.noday	.joy-minical__no-day
.minicalendar tbody tr td .day-title	.joy-minical__day-title
.minicalendar tbody tr td.today .day-title	.joy-minical__day-title-today
.minicalendar .holiday	.joy-minical__date-holiday,
.minicalendar .sun	.joy-minical__date-sun
.days-events .event	.joy-days-events__event
.days-events .event-continues	.joy-days-events__event-continues
.days-events .event-time	.joy-days-events__event-time
.days-events .event-title	.joy-days-events__event-title
.upcoming-events	.joy-cal-list
.past-events	.joy-cal-list
.events-on-day	.joy-cal-list
.event-item	.joy-ev-item
.event-title	.joy-title-item
.event-image	.joy-img-link
.event-image	.joy-img

Continued on next page

Table 1 – continued from previous page

.event-group-page	.joy-ev-who
.event-when	.joy-ev-when
.event-postponed-from	.joy-ev-from-when
.event-postponed-to	.joy-ev-to-when
.event-next-on	.joy-ev-next-on
.event-location	.joy-ev-where
.event-location .gmap	.joy-ev-where__map-link
.event-website	.joy-ev-website
.event-website a	.joy-ev-website__link
.event-extra-info	.joy-ev-extra
.event-status	.joy-ev-status
.event-export	.joy-export
.upcoming-events	.joy-grp-list
.event-item-title	.joy-title-item
.event-index	.joy-ev-related
.event-index .item	.joy-ev-related__item
.event-index a	.joy-ev-related__link
.event-index a strong	.joy-ev-related__overrides strong
.events-this-week	.joy-this-week
.events-this-week h3	.joy-this-week__title
.events-this-week .events	.joy-this-week__events
.events-this-week .day	.joy-this-week__day
.events-this-week .day.in-past	.joy-this-week__day-in-past
.events-this-week .day.today	.joy-this-week__day-today
.events-this-week .event-day	.joy-this-week__day-title
.events-this-week .event-day h4	.joy-this-week__weekday
.events-this-week .calendar-link	.joy-this-week__cal-link
.events-this-week .days-events .holiday	.joy-this-week__holiday-name
.events-this-week .days-events	.joy-this-week__days-events
.events-this-week .days-events .event	.joy-days-events__event
.events-pagination	.joy-pg
.events-pagination li	.joy-pg__page
.events-pagination li > a	.joy-pg__ctrl
.events-pagination li.disabled > a	.joy-pg__ctrl-disabled
.events-pagination li > a	.joy-pg__num
.events-pagination li > span	.joy-pg__num
.events-pagination li.active > a	.joy-pg__num-active
.events-pagination li.active > span	.joy-pg__num-active
.event-details	.joy-ev-details
.events-on-day-detailed	—
.upcoming-events-detailed	—
.together	—
.content-inner	—
.calendar-options	—
.ev-recurrence-fields	.joy-rr
.ev-advanced-monthly-repeat	.joy-rr__advanced-monthly-repeat
.ev-advanced-repeat	.joy-rr__advanced-repeat
.ev-advanced-weekly-repeat	.joy-rr__advanced-weekly-repeat
.ev-advanced-yearly-repeat	.joy-rr__advanced-yearly-repeat
.ev-day-choice	.joy-rr__day-choice

Continued on next page

Table 1 – continued from previous page

.ev-days-input	.joy-rr__days-input
.ev-short-input	.joy-rr__short-input
.ev-months-input	.joy-rr__months-input
.ev-freq-choice	.joy-rr__freq-choice
.ev-half-field	.joy-rr__half-field
.ev-double-field	.joy-rr__double-field
.ev-ical-value	.joy-rr__ical-value
.ev-interval-num	.joy-rr__interval-num
.ev-interval-units-days	.joy-rr__interval-units-days
.ev-interval-units-months	.joy-rr__interval-units-months
.ev-interval-units-weeks	.joy-rr__interval-units-weeks
.ev-interval-units-years	.joy-rr__interval-units-years
.ev-months	.joy-rr__months
.ev-ord-choice	.joy-rr__ord-choice
.ev-primary	.joy-rr__primary
.ev-secondary	.joy-rr__secondary
.ev-show-advanced	.joy-rr__show-advanced
.ev-show-advanced-cbx	.joy-rr__show-advanced-cbx
.ev-simple-repeat	.joy-rr__simple-repeat
.ev-start-date	.joy-rr__start-date
.ev-until-date	.joy-rr__until-date
.ev-weekdays	.joy-rr__weekdays

5.13 Joyous 0.8.3 release notes

- *What's new*
 - *Translations*
 - *iCal*
 - *Other features*
 - *Bug fixes*
- *Upgrade considerations*

5.13.1 What's new

Translations

- Thanks to @Jean-Zombie for the new German translation.

iCal

- Add support for X-ALT-DESC

Other features

- **Thanks to @Jean-Zombie for suggestions on making Calendar templates easier to customize:**
 - Templates now have more block definitions available to override.
 - serve functions now return a `TemplateResponse`.
 - also, the `Calendar._getExtraContext` function can be overridden to add context.
- Raise `PermissionDenied` from `getEventFromUid` if we don't have authority.

Bug fixes

- Use `toTheOrdinal` where appropriate

5.13.2 Upgrade considerations

5.14 Joyous 0.8.2 release notes

- *What's new*
 - *Wagtail 2.5*
 - *Postponement Page*
 - *Other features*
 - *Bug fixes*
- *Upgrade considerations*
 - *ThisEvent*
 - *auth.0011_update_proxy_permissions*

5.14.1 What's new

Wagtail 2.5

- Support for Wagtail 2.5 (and Django 2.2).

Postponement Page

- Show cancellation details for postponement page in `/from/` sub-view.

Other features

- Include version number in calendar table tag.
- Add django admin interface.
- Restructured templates to use more inheritance and less inclusion.

- Refactor ByDayIterables using EventsByDayList.

Bug fixes

- 1.4 macro isn't working with ReadTheDocs, so hardcoded it.
- Fix display of Cancellations

5.14.2 Upgrade considerations

ThisEvent

The namedtuple `ThisEvent` now has this definition

```
ThisEvent = namedtuple("ThisEvent", "title page url")
```

The extra field, `url`, is used to customise the url for an event. This is used to link to the `/from/` sub-view of Postponement Pages.

When unpacking the tuple you will have to include this field too, e.g.

```
{% for title, page, url in evod.days_events %}
```

auth.0011_update_proxy_permissions

You might see an error running the `auth.0011_update_proxy_permissions` migration something like...

```
django.db.utils.IntegrityError: duplicate key value violates unique
↳constraint "idx_18141_auth_permission_content_type_id_01ab375a_uniq"
DETAIL: Key (content_type_id, codename)=(42, add_specificcalendarpage)
↳already exists
```

This is a [known issue](#). There is a fix coming for this in Django, hopefully in 2.2.1. Until then the migration can fairly safely be faked when the error occurs.

```
$ ./manage.py migrate --fake auth 0011_update_proxy_permissions
$ ./manage.py migrate
```

5.15 Joyous 0.8.1 release notes

- *What's new*
 - *Documentation*
 - *Internationalisation*
 - *CSS*
 - *base.html*
 - *iCal*

- *Other features*
- *Bug fixes*

5.15.1 What's new

Documentation

You are reading it.

Internationalisation

Joyous now supports multiple languages. Thanks to Dylan Ferris for the work on the internationalization of Joyous and creating the French translation.

Heureux heureux, joie joie.

CSS

`.page-index .item-heading` has been changed to `.page-index strong`, and `.event-index .item-heading` has been changed to `.event-index strong`

base.html

Joyous no longer needs a customised `base.html` with “`global_css`” and “`background`” blocks. The `joyous.css` is included in the “`extra_css`” block, and the HTML required for popups on the calendar pages are generated dynamically by Javascript.

iCal

- Export RRULE UNTIL dates as UTC datetimes / use timezone upon import
- Define timezones transitions until 2038 if no other end date is available
- Add an option to convert UTC timezone events into local time
- Use X-WR-TIMEZONE if given as the current timezone during an import
- `MultidayRecurringEventPage` import/export
- `RescheduleMultidayEventPage` import/export

Other features

- Add `JOYOUS_EVENTS_PER_PAGE` setting
- Switch to “rescheduled from” not “postponed from” if the postponement is earlier than the original event
- Create `RescheduleMultidayEventPage` exception type for `MultidayRecurringEventPage`

Bug fixes

- Handle creating a postponement for a finished recurring event
- Postponements can be created after the until date so ignore that for working out the status of a recurring event
- A datetime may have a tzinfo of None
- Fix double inclusion of group events
- Exceptions of MultidayRecurringEventPage now display the number of days
- Non-reversible reg-exp portion: ‘(?i)’ exception (Dylan Ferris)

CHAPTER 6

Indices and tables

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

I

- `ls.joyous.edit_handlers`, 56
- `ls.joyous.fields`, 56
- `ls.joyous.formats`, 52
- `ls.joyous.holidays`, 56
- `ls.joyous.middleware`, 56
- `ls.joyous.models`, 50
- `ls.joyous.templatetags.joyous_tags`, 51
- `ls.joyous.utils.manythings`, 52
- `ls.joyous.utils.mixins`, 53
- `ls.joyous.utils.names`, 53
- `ls.joyous.utils.recurrence`, 53
- `ls.joyous.utils.telltime`, 54
- `ls.joyous.utils.weeks`, 55
- `ls.joyous.widgets`, 56

Symbols

- `_Names` (class in *ls.joyous.utils.names*), 53
- `_allowAnotherAt()` (*ls.joyous.models.CalendarPage* class method), 32
- `_allowAnotherAt()` (*ls.joyous.models.GeneralCalendarPage* class method), 34
- `_allowAnotherAt()` (*ls.joyous.models.SpecificCalendarPage* class method), 33
- `_cacheClosedSet()` (*ls.joyous.models.ClosedForHolidaysPage* method), 49
- `_closedOn()` (*ls.joyous.models.ClosedForHolidaysPage* method), 49
- `_closedOn()` (*ls.joyous.models.ExtCancellationPage* method), 50
- `_closed_for_dates` (*ls.joyous.models.ClosedForHolidaysPage* attribute), 49
- `_copyFieldsFromParent()` (*ls.joyous.models.DateExceptionBase* method), 43
- `_copyFieldsFromParent()` (*ls.joyous.models.EventExceptionBase* method), 42
- `_copyFieldsFromParent()` (*ls.joyous.models.PostponementPage* method), 47
- `_current_datetime_from` (*ls.joyous.models.CancellationPage* attribute), 45
- `_current_datetime_from` (*ls.joyous.models.ClosedForHolidaysPage* attribute), 49
- `_current_datetime_from` (*ls.joyous.models.EventBase* attribute), 37
- `_current_datetime_from` (*ls.joyous.models.ExtCancellationPage* attribute), 50
- `_current_datetime_from` (*ls.joyous.models.ExtraInfoPage* attribute), 44
- `_current_datetime_from` (*ls.joyous.models.RecurringEventPage* attribute), 40
- `_getAllEvents()` (*ls.joyous.models.CalendarPage* method), 32
- `_getAllEvents()` (*ls.joyous.models.GeneralCalendarPage* method), 34
- `_getAllEvents()` (*ls.joyous.models.SpecificCalendarPage* method), 33
- `_getEventFromUid()` (*ls.joyous.models.ExtCancellationPage* attribute), 50
- `_current_datetime_from` (*ls.joyous.models.ExtraInfoPage* attribute), 44
- `_current_datetime_from` (*ls.joyous.models.RecurringEventPage* attribute), 40
- `_first_datetime_from` (*ls.joyous.models.EventBase* attribute), 37
- `_first_datetime_from` (*ls.joyous.models.RecurringEventPage* attribute), 40
- `_futureExceptions()` (*ls.joyous.models.RecurringEventPage* method), 41
- `_future_datetime_from` (*ls.joyous.models.CancellationPage* attribute), 45
- `_future_datetime_from` (*ls.joyous.models.ClosedForHolidaysPage* attribute), 49
- `_future_datetime_from` (*ls.joyous.models.EventBase* attribute), 37
- `_future_datetime_from` (*ls.joyous.models.ExtCancellationPage* attribute), 50
- `_future_datetime_from` (*ls.joyous.models.ExtraInfoPage* attribute), 44
- `_future_datetime_from` (*ls.joyous.models.RecurringEventPage* attribute), 40

(*ls.joyous.models.CalendarPage method*), 32
 _getEventFromUid () (*ls.joyous.models.GeneralCalendarPage method*), 34
 _getEventFromUid () (*ls.joyous.models.SpecificCalendarPage method*), 33
 _getEventsByDay () (*ls.joyous.models.CalendarPage method*), 32
 _getEventsByDay () (*ls.joyous.models.GeneralCalendarPage method*), 34
 _getEventsByDay () (*ls.joyous.models.SpecificCalendarPage method*), 33
 _getEventsByWeek () (*ls.joyous.models.CalendarPage method*), 32
 _getEventsByWeek () (*ls.joyous.models.GeneralCalendarPage method*), 34
 _getEventsByWeek () (*ls.joyous.models.SpecificCalendarPage method*), 33
 _getEventsOnDay () (*ls.joyous.models.CalendarPage method*), 32
 _getFromDt () (*ls.joyous.models.DateExceptionBase method*), 43
 _getFromDt () (*ls.joyous.models.EventBase method*), 38
 _getFromDt () (*ls.joyous.models.MultidayEventPage method*), 39
 _getFromDt () (*ls.joyous.models.PostponementPage method*), 47
 _getFromDt () (*ls.joyous.models.SimpleEventPage method*), 39
 _getFromTime () (*ls.joyous.models.ClosedForHolidaysPage method*), 49
 _getFromTime () (*ls.joyous.models.DateExceptionBase method*), 43
 _getFromTime () (*ls.joyous.models.EventBase method*), 38
 _getFromTime () (*ls.joyous.models.ExtCancellationPage method*), 50
 _getFromTime () (*ls.joyous.models.MultidayEventPage method*), 39
 _getFromTime () (*ls.joyous.models.PostponementPage method*), 47
 _getFromTime () (*ls.joyous.models.RecurringEventPage method*), 41
 _getFromTime () (*ls.joyous.models.SimpleEventPage method*), 39
 (*ls.joyous.models.CalendarPage method*), 39
 _getLocalWhen () (*ls.joyous.models.DateExceptionBase method*), 43
 _getLocalWhen () (*ls.joyous.models.EventBase method*), 38
 _getMyDates () (*ls.joyous.models.ClosedForHolidaysPage method*), 49
 _getMyDates () (*ls.joyous.models.ExtCancellationPage method*), 50
 _getMyFirstDatetimeFrom () (*ls.joyous.models.RecurringEventPage method*), 41
 _getMyFirstDatetimeTo () (*ls.joyous.models.RecurringEventPage method*), 41
 _getMyNextDate () (*ls.joyous.models.RecurringEventPage method*), 41
 _getMyRawDates () (*ls.joyous.models.ExtCancellationPage method*), 50
 _getPastEvents () (*ls.joyous.models.CalendarPage method*), 32
 _getPastEvents () (*ls.joyous.models.GeneralCalendarPage method*), 34
 _getPastEvents () (*ls.joyous.models.SpecificCalendarPage method*), 33
 _getToDt () (*ls.joyous.models.DateExceptionBase method*), 43
 _getToDt () (*ls.joyous.models.EventBase method*), 38
 _getToDt () (*ls.joyous.models.MultidayEventPage method*), 39
 _getToDt () (*ls.joyous.models.PostponementPage method*), 47
 _getToDt () (*ls.joyous.models.SimpleEventPage method*), 39
 _getUpcomingEvents () (*ls.joyous.models.CalendarPage method*), 32
 _getUpcomingEvents () (*ls.joyous.models.GeneralCalendarPage method*), 34
 _getUpcomingEvents () (*ls.joyous.models.SpecificCalendarPage method*), 33
 _nextOn () (*ls.joyous.models.RecurringEventPage method*), 41
 _occursOn () (*ls.joyous.models.RecurringEventPage method*), 41
 _past_datetime_from (*ls.joyous.models.CancellationPage attribute*), 45
 _past_datetime_from (*ls.joyous.models.ClosedForHolidaysPage attribute*), 49
 _past_datetime_from

- (*ls.joyous.models.EventBase* attribute), 37
- `_past_datetime_from` (*ls.joyous.models.ExtCancellationPage* attribute), 50
- `_past_datetime_from` (*ls.joyous.models.ExtraInfoPage* attribute), 44
- `_past_datetime_from` (*ls.joyous.models.RecurringEventPage* attribute), 40
- `_removeContentPanels()` (*ls.joyous.models.EventBase* class method), 38
- `_timeFormat()` (in module *ls.joyous.utils.telltime*), 54
- ## A
- `add()` (*ls.joyous.holidays.Holidays* method), 56
- `all_events` (*ls.joyous.models.EventsOnDay* attribute), 36
- `all_holidays` (*ls.joyous.models.ClosedForHolidaysPage* attribute), 48
- `all_upcoming_events()` (in module *ls.joyous.templatetags.joyous_tags*), 51
- `at` (*ls.joyous.models.EventBase* attribute), 38
- `at` (*ls.joyous.models.EventExceptionBase* attribute), 42
- `at` (*ls.joyous.models.PostponementPage* attribute), 47
- `at_time_display()` (in module *ls.joyous.templatetags.joyous_tags*), 52
- ## B
- `byeaster` (*ls.joyous.utils.recurrence.Recurrence* attribute), 53
- `bymonth` (*ls.joyous.utils.recurrence.Recurrence* attribute), 53
- `bymonthday` (*ls.joyous.utils.recurrence.Recurrence* attribute), 54
- `bysetpos` (*ls.joyous.utils.recurrence.Recurrence* attribute), 54
- `byweekday` (*ls.joyous.utils.recurrence.Recurrence* attribute), 54
- `byweekno` (*ls.joyous.utils.recurrence.Recurrence* attribute), 54
- `byyearday` (*ls.joyous.utils.recurrence.Recurrence* attribute), 54
- ## C
- `CalendarPage` (class in *ls.joyous.models*), 31
- `can_create_at()` (*ls.joyous.models.CalendarPage* class method), 32
- `can_create_at()` (*ls.joyous.models.ClosedForHolidaysPage* class method), 49
- `cancellation_details` (*ls.joyous.models.CancellationBase* attribute), 44
- `cancellation_title` (*ls.joyous.models.CancellationBase* attribute), 44
- `cancellation_url` (*ls.joyous.models.CancellationPage* attribute), 45
- `CancellationBase` (class in *ls.joyous.models*), 44
- `CancellationPage` (class in *ls.joyous.models*), 45
- `cancelled_from_date` (*ls.joyous.models.ExtCancellationPage* attribute), 50
- `cancelled_to_date` (*ls.joyous.models.ExtCancellationPage* attribute), 50
- `category` (*ls.joyous.models.EventBase* attribute), 37
- `category` (*ls.joyous.models.EventExceptionBase* attribute), 42
- `closed` (*ls.joyous.models.ClosedForHolidaysPage* attribute), 49
- `closed_for` (*ls.joyous.models.ClosedForHolidaysPage* attribute), 48
- `ClosedForHolidaysPage` (class in *ls.joyous.models*), 48
- `code` (*ls.joyous.models.EventCategory* attribute), 36
- `ConcealedPanel` (class in *ls.joyous.edit_handlers*), 56
- `content` (*ls.joyous.models.GroupPage* attribute), 51
- `continuing_events` (*ls.joyous.models.EventsOnDay* attribute), 36
- `count` (*ls.joyous.utils.recurrence.Recurrence* attribute), 54
- ## D
- `date` (*ls.joyous.models.EventsOnDay* attribute), 36
- `date` (*ls.joyous.models.PostponementPage* attribute), 47
- `date` (*ls.joyous.models.SimpleEventPage* attribute), 38
- `date_display()` (in module *ls.joyous.templatetags.joyous_tags*), 52
- `date_from` (*ls.joyous.models.MultidayEventPage* attribute), 39
- `date_to` (*ls.joyous.models.MultidayEventPage* attribute), 39
- `DateExceptionBase` (class in *ls.joyous.models*), 43
- `dateFormat()` (in module *ls.joyous.utils.telltime*), 54
- `dateShortFormat()` (in module *ls.joyous.utils.telltime*), 54
- `days_events` (*ls.joyous.models.EventsOnDay* attribute), 36
- `default_view` (*ls.joyous.models.CalendarPage* attribute), 31
- `details` (*ls.joyous.models.EventBase* attribute), 37
- `details` (*ls.joyous.models.ExtraInfoPage* attribute), 44
- `dtstart` (*ls.joyous.utils.recurrence.Recurrence* attribute), 54

E

EventBase (class in *ls.joyous.models*), 37
 EventCategory (class in *ls.joyous.models*), 36
 EventExceptionBase (class in *ls.joyous.models*), 42
 events_this_week () (in module *ls.joyous.templatetags.joyous_tags*), 51
 EventsOnDay (class in *ls.joyous.models*), 36
 except_date (*ls.joyous.models.DateExceptionBase* attribute), 43
 ExceptionDateInput (class in *ls.joyous.widgets*), 56
 ExceptionDatePanel (class in *ls.joyous.edit_handlers*), 56
 ExtCancellationPage (class in *ls.joyous.models*), 49
 extra_information (*ls.joyous.models.ExtraInfoPage* attribute), 44
 extra_title (*ls.joyous.models.ExtraInfoPage* attribute), 44
 ExtraInfoPage (class in *ls.joyous.models*), 43

F

freq (*ls.joyous.utils.recurrence.Recurrence* attribute), 54
 frequency (*ls.joyous.utils.recurrence.Recurrence* attribute), 54
 full_clean () (*ls.joyous.models.DateExceptionBase* method), 43
 full_clean () (*ls.joyous.models.ExtCancellationPage* method), 50
 future_exceptions () (in module *ls.joyous.templatetags.joyous_tags*), 51

G

GeneralCalendarPage (class in *ls.joyous.models*), 33
 get () (*ls.joyous.holidays.Holidays* method), 57
 get_context () (*ls.joyous.models.EventBase* method), 38
 get_context () (*ls.joyous.models.EventExceptionBase* method), 42
 get_context () (*ls.joyous.models.RescheduleEventBase* method), 46
 get_group_model () (in module *ls.joyous.models*), 50
 get_group_model_string () (in module *ls.joyous.models*), 50
 getAllEvents () (in module *ls.joyous.models*), 35
 getAllEventsByDay () (in module *ls.joyous.models*), 34
 getAllEventsByWeek () (in module *ls.joyous.models*), 34

getAllPastEvents () (in module *ls.joyous.models*), 35
 getAllUpcomingEvents () (in module *ls.joyous.models*), 35
 getAwareDatetime () (in module *ls.joyous.utils.telltime*), 55
 getCancellationUrl () (*ls.joyous.models.CancellationPage* method), 45
 getCount () (*ls.joyous.utils.recurrence.Recurrence* method), 54
 getEventFromUid () (in module *ls.joyous.models*), 35
 getGroupUpcomingEvents () (in module *ls.joyous.models*), 35
 getLocalDate () (in module *ls.joyous.utils.telltime*), 55
 getLocalDateAndTime () (in module *ls.joyous.utils.telltime*), 55
 getLocalDatetime () (in module *ls.joyous.utils.telltime*), 55
 getLocalTime () (in module *ls.joyous.utils.telltime*), 55
 getLocalTimeAtDate () (in module *ls.joyous.utils.telltime*), 55
 getTimeFrom () (in module *ls.joyous.utils.telltime*), 55
 getTimeTo () (in module *ls.joyous.utils.telltime*), 55
 GoogleCalendarHandler (class in *ls.joyous.formats*), 52
 gregorian_to_week_date () (in module *ls.joyous.utils.weeks*), 55
 group (*ls.joyous.models.EventBase* attribute), 37
 group (*ls.joyous.models.EventExceptionBase* attribute), 42
 group (*ls.joyous.models.RescheduleEventBase* attribute), 46
 group_page (*ls.joyous.models.EventBase* attribute), 37
 group_upcoming_events () (in module *ls.joyous.templatetags.joyous_tags*), 51
 GroupPage (class in *ls.joyous.models*), 51

H

holiday (*ls.joyous.models.EventsOnDay* attribute), 36
 Holidays (class in *ls.joyous.holidays*), 56
 holidays (*ls.joyous.models.CalendarPage* attribute), 31
 hrJoin () (in module *ls.joyous.utils.manythings*), 52

I

ICALHandler (class in *ls.joyous.formats*), 52
 image (*ls.joyous.models.EventBase* attribute), 37
 image (*ls.joyous.models.EventExceptionBase* attribute), 42

interval (*ls.joyous.utils.recurrence.Recurrence attribute*), 54
 intro (*ls.joyous.models.CalendarPage attribute*), 31
 isAuthorized() (*ls.joyous.models.EventBase method*), 38
 isAuthorized() (*ls.joyous.models.EventExceptionBase method*), 42
 ls.joyous.utils.manythings (*module*), 52
 ls.joyous.utils.mixins (*module*), 53
 ls.joyous.utils.names (*module*), 53
 ls.joyous.utils.recurrence (*module*), 53
 ls.joyous.utils.telltime (*module*), 54
 ls.joyous.utils.weeks (*module*), 55
 ls.joyous.widgets (*module*), 56

J

JOYOUS_DATE_FORMAT setting, 26
 JOYOUS_DATE_SHORT_FORMAT setting, 26
 JOYOUS_EVENTS_PER_PAGE setting, 26
 JOYOUS_FIRST_DAY_OF_WEEK setting, 26
 JOYOUS_GROUP_MODEL setting, 27
 JOYOUS_GROUP_SELECTABLE setting, 27
 JOYOUS_HOLIDAYS setting, 27
 JOYOUS_RSS_FEED_IMAGE setting, 27
 JOYOUS_THEME_CSS setting, 27
 JOYOUS_TIME_FORMAT setting, 27
 JOYOUS_TIME_INPUT setting, 28
 JOYOUS_UPCOMING_INCLUDES_STARTED setting, 28

L

local_title (*ls.joyous.models.ClosedForHolidaysPage attribute*), 49
 local_title (*ls.joyous.models.DateExceptionBase attribute*), 43
 local_title (*ls.joyous.models.ExtCancellationPage attribute*), 50
 location (*ls.joyous.models.EventBase attribute*), 37
 location (*ls.joyous.models.EventExceptionBase attribute*), 42
 location_gmap() (*in module ls.joyous.templatetags.joyous_tags*), 51
 ls.joyous.edit_handlers (*module*), 56
 ls.joyous.fields (*module*), 56
 ls.joyous.formats (*module*), 52
 ls.joyous.holidays (*module*), 56
 ls.joyous.middleware (*module*), 56
 ls.joyous.models (*module*), 31, 34, 50
 ls.joyous.templatetags.joyous_tags (*module*), 51

M

minicalendar() (*in module ls.joyous.templatetags.joyous_tags*), 51
 MON_TO_SUN (*in module ls.joyous.utils.names*), 53
 MONDAY_TO_SUNDAY (*in module ls.joyous.utils.names*), 53
 MONTH_ABBRS (*in module ls.joyous.utils.names*), 53
 MONTH_NAMES (*in module ls.joyous.utils.names*), 53
 MultidayEventPage (*class in ls.joyous.models*), 39
 MultidayRecurringEventPage (*class in ls.joyous.models*), 41
 MultipleSelectField (*class in ls.joyous.fields*), 56

N

name (*ls.joyous.models.EventCategory attribute*), 36
 names() (*ls.joyous.holidays.Holidays method*), 57
 next_date (*ls.joyous.models.RecurringEventPage attribute*), 40
 next_on() (*in module ls.joyous.templatetags.joyous_tags*), 51
 NullHandler (*class in ls.joyous.formats*), 52
 num_days (*ls.joyous.models.EventExceptionBase attribute*), 42
 num_days (*ls.joyous.models.RecurringEventPage attribute*), 40
 num_days (*ls.joyous.models.RescheduleEventBase attribute*), 46
 num_weeks_in_year() (*in module ls.joyous.utils.weeks*), 55

O

overrides (*ls.joyous.models.EventExceptionBase attribute*), 42
 overrides_repeat (*ls.joyous.models.EventExceptionBase attribute*), 42

P

peers() (*ls.joyous.models.CalendarPage class method*), 32
 peers() (*ls.joyous.utils.mixins.ProxyPageMixin class method*), 53
 postponed_from (*ls.joyous.models.PostponementPage attribute*), 47
 postponed_from_when (*ls.joyous.models.PostponementPage attribute*), 47

postponed_to (*ls.joyous.models.PostponementPage attribute*), 47
 postponement_title (*ls.joyous.models.PostponementPage attribute*), 47
 PostponementPage (*class in ls.joyous.models*), 46
 prev_date (*ls.joyous.models.RecurringEventPage attribute*), 40
 preview (*ls.joyous.models.EventsOnDay attribute*), 36
 ProxyPageMixin (*class in ls.joyous.utils.mixins*), 53

R

Recurrence (*class in ls.joyous.utils.recurrence*), 53
 RecurrenceField (*class in ls.joyous.fields*), 56
 RecurrenceWidget (*class in ls.joyous.widgets*), 56
 RecurringEventPage (*class in ls.joyous.models*), 40
 register() (*ls.joyous.holidays.Holidays method*), 57
 repeat (*ls.joyous.models.RecurringEventPage attribute*), 40
 RescheduleEventBase (*class in ls.joyous.models*), 46
 RescheduleMultidayEventPage (*class in ls.joyous.models*), 48
 routeByMonthAbbr() (*ls.joyous.models.CalendarPage method*), 32
 routeDefault() (*ls.joyous.models.CalendarPage method*), 32
 RssHandler (*class in ls.joyous.formats*), 52

S

serveCancellation() (*ls.joyous.models.PostponementPage method*), 47
 serveDay() (*ls.joyous.models.CalendarPage method*), 32
 serveMiniMonth() (*ls.joyous.models.CalendarPage method*), 32
 serveMonth() (*ls.joyous.models.CalendarPage method*), 32
 servePast() (*ls.joyous.models.CalendarPage method*), 32
 serveUpcoming() (*ls.joyous.models.CalendarPage method*), 32
 serveWeek() (*ls.joyous.models.CalendarPage method*), 32
 setting
 JOYOUS_DATE_FORMAT, 26
 JOYOUS_DATE_SHORT_FORMAT, 26
 JOYOUS_EVENTS_PER_PAGE, 26
 JOYOUS_FIRST_DAY_OF_WEEK, 26
 JOYOUS_GROUP_MODEL, 27
 JOYOUS_GROUP_SELECTABLE, 27
 JOYOUS_HOLIDAYS, 27

 JOYOUS_RSS_FEED_IMAGE, 27
 JOYOUS_THEME_CSS, 27
 JOYOUS_TIME_FORMAT, 27
 JOYOUS_TIME_INPUT, 28
 JOYOUS_UPCOMING_INCLUDES_STARTED, 28
 SimpleEventPage (*class in ls.joyous.models*), 38
 SpecificCalendarPage (*class in ls.joyous.models*), 33
 status (*ls.joyous.models.CancellationBase attribute*), 44
 status (*ls.joyous.models.EventBase attribute*), 37
 status (*ls.joyous.models.ExtraInfoPage attribute*), 44
 status (*ls.joyous.models.PostponementPage attribute*), 47
 status (*ls.joyous.models.RecurringEventPage attribute*), 40
 status_text (*ls.joyous.models.CancellationBase attribute*), 44
 status_text (*ls.joyous.models.EventBase attribute*), 38
 status_text (*ls.joyous.models.ExtraInfoPage attribute*), 44
 status_text (*ls.joyous.models.RecurringEventPage attribute*), 40
 subsite_upcoming_events() (*in module ls.joyous.templatetags.joyous_tags*), 51
 SUN_TO_SAT (*in module ls.joyous.utils.names*), 53
 SUNDAY_TO_SATURDAY (*in module ls.joyous.utils.names*), 53

T

Time12hrInput (*class in ls.joyous.widgets*), 56
 time_display() (*in module ls.joyous.templatetags.joyous_tags*), 52
 time_from (*ls.joyous.models.EventBase attribute*), 37
 time_from (*ls.joyous.models.EventExceptionBase attribute*), 42
 time_to (*ls.joyous.models.EventBase attribute*), 37
 time_to (*ls.joyous.models.EventExceptionBase attribute*), 42
 timeFormat() (*in module ls.joyous.utils.telltime*), 55
 TimePanel (*class in ls.joyous.edit_handlers*), 56
 todayUtc() (*in module ls.joyous.utils.telltime*), 55
 toOrdinal() (*in module ls.joyous.utils.manythings*), 52
 toTheOrdinal() (*in module ls.joyous.utils.manythings*), 52
 tz (*ls.joyous.models.EventBase attribute*), 37
 tz (*ls.joyous.models.EventExceptionBase attribute*), 42
 tz (*ls.joyous.models.RescheduleEventBase attribute*), 46
 TZDatePanel (*class in ls.joyous.edit_handlers*), 56

U

uid (*ls.joyous.models.EventBase attribute*), 37

uid (*ls.joyous.models.RescheduleEventBase* attribute),
 46
 until (*ls.joyous.utils.recurrence.Recurrence* attribute),
 54
 until_when (*ls.joyous.models.ExtCancellationPage*
 attribute), 50
 UserTimezoneMiddleware (class in
 ls.joyous.middleware), 56

V

view_choices (*ls.joyous.models.CalendarPage* at-
 tribute), 31

W

website (*ls.joyous.models.EventBase* attribute), 37
 website (*ls.joyous.models.EventExceptionBase* at-
 tribute), 42
 week_info() (in module *ls.joyous.utils.weeks*), 55
 week_of_month() (in module *ls.joyous.utils.weeks*),
 55
 Weekday (class in *ls.joyous.utils.recurrence*), 54
 weekday (*ls.joyous.models.EventsOnDay* attribute), 36
 weekday_abbrev (in module *ls.joyous.utils.weeks*), 55
 WEEKDAY_ABBRS (in module *ls.joyous.utils.names*), 53
 weekday_name (in module *ls.joyous.utils.weeks*), 55
 WEEKDAY_NAMES (in module *ls.joyous.utils.names*), 53
 WEEKDAY_NAMES_PLURAL (in module
 ls.joyous.utils.names), 53
 what (*ls.joyous.models.PostponementPage* attribute), 47
 when (*ls.joyous.models.ClosedForHolidaysPage* at-
 tribute), 49
 when (*ls.joyous.models.DateExceptionBase* attribute),
 43
 when (*ls.joyous.models.ExtCancellationPage* attribute),
 50
 when (*ls.joyous.models.MultidayEventPage* attribute),
 39
 when (*ls.joyous.models.PostponementPage* attribute), 47
 when (*ls.joyous.models.RecurringEventPage* attribute),
 40
 when (*ls.joyous.models.SimpleEventPage* attribute), 38
 wkst (*ls.joyous.utils.recurrence.Recurrence* attribute),
 54
 WRAPPED_MONTH_NAMES (in module
 ls.joyous.utils.names), 53