django-ical Documentation

Release 1.9.2

Ian Lewis

Jun 12, 2023

Contents

django-ical is a simple library/framework for creating ical feeds based in Django's syndication feed framework.

This documentation is modeled after the documentation for the syndication feed framework so you can think of it as a simple extension.

If you are familiar with the Django syndication feed framework you should be able to be able to use django-ical fairly quickly. It works the same way as the Django syndication framework but adds a few extension properties to support iCalendar feeds.

django-ical uses the icalendar library under the hood to generate iCalendar feeds.

Contents

1.1 The high-level framework

1.1.1 Overview

The high level iCal feed-generating is supplied by the `ICalFeed` class. To create a feed, write a `ICalFeed` class and point to an instance of it in your URLconf.

With RSS feeds, the items in the feed represent articles or simple web pages. The `ICalFeed` class represents an iCalendar calendar. Calendars contain items which are events.

1.1.2 Example

Let's look at a simple example. Here the item_start_datetime is a django-ical extension that supplies the start time of the event.

```python
from django_ical.views import ICalFeed
from examplecom.models import Event

class EventFeed(ICalFeed):
    """
    A simple event calender
    """
    product_id = '-//example.com//Example//EN'
    timezone = 'UTC'
    file_name = "event.ics"

    def items(self):
        return Event.objects.all().order_by('-start_datetime')

    def item_title(self, item):
        return item.title
```

```python
    def item_description(self, item):
        return item.description

    def item_start_datetime(self, item):
        return item.start_datetime
```

To connect a URL to this calendar, put an instance of the EventFeed object in your URLconf. For example:

```python
from django.conf.urls import patterns, url, include
from myproject.feeds import EventFeed

urlpatterns = patterns('',
    # ...
    (r'^latest/feed.ics$', EventFeed()),
    # ...
)
```

Example how recurrences are built using the django-recurrence package:

```python
from django_ical.utils import build_rrule_from_recurrences_rrule
from django_ical.views import ICalFeed
from examplecom.models import Event

class EventFeed(ICalFeed):
    """
    A simple event calender
    """
    # ...

    def item_rrule(self, item):
        """Adapt Event recurrence to Feed Entry rrule."""
        if item.recurrences:
            rules = []
            for rule in item.recurrences.rrules:
                rules.append(build_rrule_from_recurrences_rrule(rule))
            return rules

    def item_exrule(self, item):
        """Adapt Event recurrence to Feed Entry exrule."""
        if item.recurrences:
            rules = []
            for rule in item.recurrences.exrules:
                rules.append(build_rrule_from_recurrences_rrule(rule))
            return rules

    def item_rdate(self, item):
        """Adapt Event recurrence to Feed Entry rdate."""
        if item.recurrences:
            return item.recurrences.rdates

    def item_exdate(self, item):
        """Adapt Event recurrence to Feed Entry exdate."""
        if item.recurrences:
            return item.recurrences.exdates
```

Note that in django_ical.utils are also convienience methods to build rrules from scratch, from string

---

(serialized iCal) and `dateutil.rrule`.

1.1.3 File Downloads

The *file_name* parameter is an optional used as base name when generating the file. By default django-ical will not set the Content-Disposition header of the response. By setting the file_name parameter you can cause django_ical to set the Content-Disposition header and set the file name. In the example below, it will be called "event.ics".

```python
class EventFeed(ICalFeed):
    """
    A simple event calender
    """
    product_id = '-//example.com//Example//EN'
    timezone = 'UTC'
    file_name = "event.ics"


    # ...
```

The *file_name* parameter can be a method like other properties. Here we can set the file name to include the id of the object returned by *get_object()*.

```python
class EventFeed(ICalFeed):
    """
    A simple event calender
    """
    product_id = '-//example.com//Example//EN'
    timezone = 'UTC'

    def file_name(self, obj):
        return "feed_%s.ics" % obj.id


    # ...
```

1.1.4 Alarms

Alarms must be *icalendar.Alarm* objects, a list is expected as the return value for *item_valarm*.

```python
from icalendar import Alarm
from datetime import timedelta

def item_valarm(self, item):
    valarm = Alarm()
    valarm.add('action', 'display')
    valarm.add('description', 'Your message text')
    valarm.add('trigger', timedelta(days=-1))
    return [valarm]
```

1.1.5 Tasks (Todos)

It is also possible to generate representations of tasks (or deadlines, todos) which are represented in iCal with the dedicated *VTODO* component instead of the usual *VEVENT*.

To do so, you can use a specific method to determine which type of component a given item should be translated as:

```python
from django_ical.views import ICalFeed
from examplecom.models import Deadline

class EventFeed(ICalFeed):
    """
    A simple event calender with tasks
    """
    product_id = '-//example.com//Example//EN'
    timezone = 'UTC'
    file_name = "event.ics"

    def items(self):
        return Deadline.objects.all().order_by('-due_datetime')

    def item_component_type(self):
        return 'todo' # could also be 'event', which is the default

    def item_title(self, item):
        return item.title

    def item_description(self, item):
        return item.description

    def item_due_datetime(self, item):
        return item.due_datetime
```

1.1.6 Property Reference and Extensions

django-ical adds a number of extensions to the base syndication framework in order to support iCalendar feeds and ignores many fields used in RSS feeds. Here is a table of all of the fields that django-ical supports.

Property name	iCalendar field name	Description
product_id	'PRODID'_	The calendar product ID
timezone	X-WR-TIMEZONE	The calendar timezone
title	X-WR-CALNAME	The calendar name/title
description	X-WR-CALDESC	The calendar description
method	'METHOD'_	The calendar method such as meeting requests.
item_guid	UID	The event's unique id. This id should be *globally* unique so you should add an
item_title	SUMMARY	The event name/title
item_description	DESCRIPTION	The event description
item_link	URL	The event url
item_class	'CLASS'_	The event class (e.g. PUBLIC, PRIVATE, CONFIDENTIAL)
item_created	CREATED	The event create time
item_updateddate	LAST-MODIFIED	The event modified time
item_start_datetime	DTSTART	The event start time
item_end_datetime	DTEND	The event end time
item_location	LOCATION	The event location
item_geolocation	GEO	The latitude and longitude of the event. The value returned by this property sho
item_transparency	TRANSP	The event transparency. Defines whether the event shows up in busy searches. (
item_organizer	ORGANIZER	The event organizer. Expected to be a vCalAddress object. See iCalendar docu
item_attendee	ATTENDEE	The event attendees. Expected to be a list of vCalAddress objects. See iCalend
item_rrule	RRULE	The recurrence rule for repeating events. See iCalendar documentation or tests

Table 1 – continued from pre

Property name	iCalendar field name	Description
item_rdate	RDATE	The recurring dates/times for a repeating event. See iCalendar documentation o
item_exdate	EXDATE	The dates/times for exceptions of a recurring event. See iCalendar documentati
item_valarm	VALARM	Alarms for the event, must be a list of Alarm objects. See iCalendar documenta
item_status	STATUS	The status of an event. Can be CONFIRMED, CANCELLED or TENTATIVE.
item_completed	COMPLETED	The date a task was completed.
item_percent_complete	PERCENT-COMPLETE	A number from 0 to 100 indication the completion of the task.
item_priority	PRIORITY	An integer from 0 to 9. 0 means undefined. 1 means highest priority.
item_due	DUE	The date a task is due.
item_categories	CATEGORIES	A list of strings, each being a category of the task.
calscale	CALSCALE	Not yet documented.
method	**'METHOD'_**	Not yet documented.
prodid	**'PRODID'_**	Not yet documented.
version	VERSION	Not yet documented.
attach	ATTACH	Not yet documented.
class	**'CLASS'_**	Not yet documented.
comment	COMMENT	Not yet documented.
resources	RESOURCES	Not yet documented.
duration	DURATION	Not yet documented.
freebusy	FREEBUSY	Not yet documented.
tzid	TZID	Not yet documented.
tzname	TZNAME	Not yet documented.
tzoffsetfrom	TZOFFSETFROM	Not yet documented.
tzoffsetto	TZOFFSETTO	Not yet documented.
tzurl	TZURL	Not yet documented.
contact	CONTACT	Not yet documented.
recurrence_id	**'RECURRENCE_ID'_**	Not yet documented.
related_to	**'RELATED_TO'_**	Not yet documented.
action	ACTION	Not yet documented.
repeat	REPEAT	Not yet documented.
trigger	TRIGGER	Not yet documented.
sequence	SEQUENCE	Not yet documented.
request_status	**'REQUEST_STATUS'_**	Not yet documented.

Note: django-ical does not use the `link` property required by the Django syndication framework.

1.2 The low-level framework

Behind the scenes, the high-level iCalendar framework uses a lower-level framework for generating feeds' ical data. This framework lives in a single module: `django_ical.feedgenerator`.

You use this framework on your own, for lower-level feed generation. You can also create custom feed generator subclasses for use with the feed_type option.

See: The syndication feed framework: Specifying the type of feed

1.3 API Reference

Release 1.9

Date Jun 12, 2023

1.3.1 django_ical.feedgenerator

1.3.2 django_ical.views

1.3.3 django_ical.utils

CHAPTER 2

Indices and tables

- genindex
- modindex
- search