django-ical Documentation

Release 1.3

Ian Lewis

November 15, 2015

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.

Requirements:

- Django >= 1.3
- icalendar >= 2.0.1

Contents:

CHAPTER 1

The high-level framework

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.

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.

```
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

    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:

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

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

```
    # ...
)
```

1.2 File Downloads

The *file_name* parameter is an optional used as base name when generating the file. By default django_ics 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".

```
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()*.

```
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

    # ...
```

1.3 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 name/title
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 @<domain_name> to your id.
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_updated	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 should be a two-tuple containing the latitude and longitude as float values. semicolon. Ex: *(37.386013, -122.082932)*
item_transparency	TRANSP	The event transparency. Defines whether the event shows up in busy searches. (e.g. OPAQUE, TRANSPARENT)
item_organizer	ORGANIZER	The event organizer. Expected to be a vCalAddress object. See iCalendar documentation or tests to know how to build them.

Note:

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

CHAPTER 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

CHAPTER 3

API Reference

Release 1.3

Date November 15, 2015

3.1 django_ical.feedgenerator

iCalendar feed generation library – used for generating iCalendar feeds.

Sample usage:

```
>>> from django_ical import feedgenerator
>>> from datetime import datetime
>>> feed = feedgenerator.ICal20Feed(
...     title=u"My Events",
...     link=u"http://www.example.com/events.ical",
...     description=u"A iCalendar feed of my events.",
...     language=u"en",
... )
>>> feed.add_item(
...     title="Hello",
...     link=u"http://www.example.com/test/",
...     description="Testing."
...     start_datetime=datetime(2012, 5, 6, 10, 00),
...     end_datetime=datetime(2012, 5, 6, 12, 00),
... )
>>> fp = open('test.ical', 'w')
>>> feed.write(fp, 'utf-8')
>>> fp.close()
```

For definitions of the iCalendar format see: http://www.ietf.org/rfc/rfc2445.txt

class `django_ical.feedgenerator.`**`ICal20Feed`**(*title*, *link*, *description*, *language=None*, *author_email=None*, *author_name=None*, *author_link=None*, *subtitle=None*, *categories=None*, *feed_url=None*, *feed_copyright=None*, *feed_guid=None*, *ttl=None*, ***kwargs*)

iCalendar 2.0 Feed implementation.

`write`(*outfile*, *encoding*)

Writes the feed to the specified file in the specified encoding.

write_items(*calendar*)
Write all events to the calendar

django_ical.feedgenerator.**DefaultFeed**
alias of *ICal20Feed*

3.2 django_ical.views

Views for generating ical feeds.

class django_ical.views.**ICalFeed**
iCalendar Feed

Existing Django syndication feeds

Title X-WR-CALNAME

Description X-WR-CALDESC

Item_guid UID

Item_title SUMMARY

Item_description DESCRIPTION

Item_link URL

Extension fields

Method METHOD

Timezone X-WR-TIMEZONE

Item_class CLASS

Item_timestamp DTSTAMP

Item_created CREATED

Item_modified LAST-MODIFIED

Item_start_datetime DTSTART

Item_end_datetime DTEND

Item_transparency TRANSP

feed_type
alias of ICal20Feed

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

d

D

F

I

W