DateRangeParser Documentation

Release 0.6

Robin Wilson

December 10, 2015

Contents

CHAPTER 1

Introduction

DateRangeParser is a Python module which makes it easy to parse human-style date ranges like “4-8th May” or “Wed 19th June - Thurs 30th August 2014”. It is very simple, consisting of one *parse* method which does the parsing.

This module is released under the GNU Lesser General Public License - see the COPYING and COPYING.LESSER files in the source directory for details.

CHAPTER 2

Installation

DateRangeParser can be installed from the Python Package Index by running:

```
pip install daterangeparser
```

DateRangeParser is built upon PyParsing, and requires the PyParsing module to be installed for it to work correctly. The command above should install PyParsing if it is not currently installed.

CHAPTER 3

Quickstart

Using DateRangeParser is very easy - simply follow the steps below.

1. Import the parse function:

```
from daterangeparser import parse
```

2. Run the parse function on a string:

```
start, end = parse("3rd May-18th July 2014")
```

3. Use the results somehow:

```
print "Start = %s" % start
print "End = %s" % end
print "Days between start and end = %i" % (end-start).days
```

4. That's it! Simple, isn't it.

3.1 What formats will this work with?

Most date range formats that use specific dates (rather than 'tomorrow' or 'last wednesday') should work fine. The parser works for date ranges and single dates (in this case returning a start date and *None* for the end date), and ignores any time details that are in the strings.

Examples that are known to work include:

- 27th-29th June 2010
- January 10th - 11th
- 30 May to 9th Aug
- 3rd Jan 1980 – 2nd Jan 2013
- Wed 23 Jan -> Sat 16 February 2013
- Tuesday 29 May - Sat 2 June 2012
- From 1 to 9 Jul
- 14th July 1988
- July 14th 1988
- 1988 14th July

- Nov 18th - 23rd Dec
- 07:00 Tue 7th June - 17th July 3:30pm
- Jan - Mar (gives dates from the first day in the first month to the last day in the second month, taking into account leap years)

More details are available in the function documentation below.

3.2 Function Documentation

daterangeparser.**parse**(*text*, *allow_implicit=True*)

Parses a date range string and returns the start and end as datetimes.

Accepted formats:

This parsing routine works with date ranges and single dates, and should work with a wide variety of human-style string formats, including:

•27th-29th June 2010

•30 May to 9th Aug

•3rd Jan 1980 - 2nd Jan 2013

•Wed 23 Jan - Sat 16 February 2013

•Tuesday 29 May -> Sat 2 June 2012

•From 27th to 29th March 1999

•1–9 Jul

•14th July 1988

•23rd October 7:30pm

•From 07:30 18th Nov to 17:00 24th Nov

Notes:

•If an error encountered while parsing the date range then a

pyparsing.ParseException will be raised. - If no year is specified then the current year is used. - All day names are ignored, so there is no checking to see whether, for example, the 23rd Jan 2013 is actually a Wednesday. - All times are ignored, assuming they are placed either before or after each date, otherwise they will cause an error. - The separators that are allows as part of the date range are *to*, *until*, -, – and ->, plus the unicode em and en dashes. - Other punctuation, such as commas, is ignored.

Parameters

- **text** – The string to parse
- **allow_implicit** – If implicit dates are allowed. For example,

string 'May' by default treated as range from May, 1st to May, 31th. Setting allow_implicit to False helps avoid it.

Returns A tuple `(start, end)` where each element is a datetime object.

If the string only defines a single date then the tuple is `(date, None)`. All times in the datetime objects are set to 00:00 as this function only parses dates.

CHAPTER 4

Release Notes

4.1 1.2

Now works on both Python 2 and Python 3

Modified so that the start month is used if the end month doesn't exist (so something like "Jan 10th-11th" will now work)

4.2 1.1.1

Modified so that "July" now produces a range from the 1st to the 31st July.

Added ranges based only on years, so "2013" produces a range from 1st Jan 2013 to the 31st Dec 2013, and similarly for "1995-2010".

4.3 1.0

Added ability to parse dates with no days specified - they will default to the first or last day of the month.

For example:

"Sep 2011 - Nov 2013" will produce 01/09/2011 to 30/11/2013

This also works with single dates:

"July" will produce 01/07/XXXX (where XXXX is the current year)

Fixed minor bugs, and bumped to stable release of version 1.0

4.4 0.6

First release with main functionality.

P