

---

# **radar Documentation**

***Release 0.3***

**Artur Barseghyan <[artur.barseghyan@gmail.com](mailto:artur.barseghyan@gmail.com)>**

December 10, 2013



---

# Contents

---



Random date generation.



---

# Prerequisites

---

- Python 2.6.8+, 2.7.+, 3.3.+





---

# Installation

---

Install latest stable version from PyPI:

```
$ pip install radar
```



---

# Usage and examples

---

## 3.1 Basic usage

```
>>> import radar
>>> radar.random_datetime()
datetime.datetime(2013, 5, 24, 16, 54, 52)
```

## 3.2 Specify date range

You may pass `datetime.datetime` or `datetime.date` objects:

```
>>> import datetime
>>> import radar
>>> radar.random_date(
>>>     start = datetime.datetime(year=2000, month=5, day=24),
>>>     stop = datetime.datetime(year=2013, month=5, day=24)
>>> )
datetime.date(2012, 12, 31)
```

You may also pass strings:

```
>>> radar.random_datetime(start='2012-05-24T00:00:00', stop='2013-05-24T23:59:59')
datetime.datetime(2013, 4, 18, 17, 54, 6)
```

## 3.3 Generate random time

```
>>> radar.random_time(start='2012-01-01T00:00:00', stop='2012-01-01T23:59:59')
datetime.time(11, 33, 59)
```

## 3.4 Advanced usage

When strings are passed, by default `radar` uses `python-dateutil` package to parse dates. Date parser of the `dateutil` package is quite heavy, although is extremely smart. As an alternative, `radar` comes with own parser

`radar.utils.parse`, which is much lighter (about 5 times faster compared to `dateutil`).

Using built-in parser:

```
>>> radar.random_datetime(start='2012-05-24T00:00:00', stop='2013-05-24T23:59:59', parse=radar.utils
datetime.datetime(2012, 11, 10, 15, 43, 40)
```

Built-in parser parses the dates using formats specified in `radar.defaults.FORMATS`:

```
>>> start = radar.utils.parse('2012-01-01')
datetime.datetime(2012, 1, 1, 0, 0)
>>> stop = radar.utils.parse('2013-01-01')
datetime.datetime(2013, 1, 1, 0, 0)
```

If you want to add more formats, define your own formats and feed them to built-in parser:

```
>>> MY_FORMATS = (
>>>     ("%d-%m-%YT%H:%M:%S", True),
>>>     ("%d-%m-%Y", False)
>>> )
>>>
>>> def my_parse(timestamp):
>>>     return radar.utils.parse(timestamp, formats=MY_FORMATS)
>>>
>>> radar.random_datetime(start='24-05-2012T00:00:00', stop='24-05-2013T23:59:59', parse=my_parse)
datetime.datetime(2012, 11, 10, 15, 43, 40)
```

## 3.5 General notes

If you expect to have really weird date formats when generating random dates from strings, you might want to consider installing wonderful *python-dateutil* package.

When generating thousands of objects (using `dateutil` or built-in parser), you're advised to pass date ranges as `datetime.datetime` or `datetime.date` objects, rather than passing strings (parsing costs time).

A good example:

```
>>> start = radar.utils.parse('2000-01-01')
>>> stop = radar.utils.parse('2013-12-31')
>>> for i in xrange(1000000):
>>>     radar.random_datetime(start=start, stop=stop)
```

See <https://bitbucket.org/barseghyanartur/radar/src> (example) directory for benchmarks and more examples.

---

# Documentation

---

## 4.1 radar Package

`radar.__init__.random_datetime` (*start=None, stop=None, parse=<function parse at 0x3516ed8>*)

Generates a random `datetime.datetime` or `datetime.date` object from ranges given.

### Parameters

- **start** (*mixed*) – Can be either a `datetime.datetime`, `datetime.date` or a `str`. Defaults to 1970-01-01.
- **end** (*mixed*) – Can be either a `datetime.datetime`, `datetime.date` or a `str`. Defaults to `datetime.datetime.now`.
- **parse** (*func*) – Parser function used to parse the date formats when `start` or `stop` arguments are strings.

### Return `datetime.datetime`

`radar.__init__.random_date` (*start=None, stop=None, parse=<function parse at 0x3516ed8>*)

Generates a random `datetime.date` object from ranges given.

### Parameters

- **start** (*mixed*) – Can be either a `datetime.datetime`, `datetime.date` or a `str`. Defaults to 1970-01-01.
- **end** (*mixed*) – Can be either a `datetime.datetime`, `datetime.date` or a `str`. Defaults to `datetime.datetime.now`.
- **parse** (*func*) – Parser function used to parse the date formats when `start` or `stop` arguments are strings.

### Return `datetime.date`

`radar.__init__.random_time` (*start=None, stop=None, parse=<function parse at 0x3516ed8>*)

Generates a random `datetime.time` object from ranges given.

### Parameters

- **start** (*mixed*) – Can be either a `datetime.datetime`, `datetime.date` or a `str`. Defaults to 1970-01-01.

- **end** (*mixed*) – Can be either a `datetime.datetime`, `datetime.date` or a `str`. Defaults to `datetime.datetime.now`.
- **parse** (*func*) – Parser function used to parse the date formats when `start` or `stop` arguments are strings.

Return `datetime.time`

## 4.2 exceptions Module

**exception** `radar.exceptions.UnrecognisedDateFormat`

Bases: `exceptions.ValueError`

Unrecognised date format.

**exception** `radar.exceptions.InvalidDateRange`

Bases: `exceptions.ValueError`

Invalid date range.

## 4.3 utils Module

`radar.utils.parse(timestamp, formats=None)`

Parse the given datetime according to the format given.

### Parameters

- **timestamp** (*str*) –
- **formats** (*list*) – List of formats.

Return `datetime.datetime`

### Example

```
>>> [("%Y-%m-%dT%H:%M:%S", True), ("%Y-%m-%d", False)]
```

`radar.utils.gettext(s)`

---

# License

---

GPL 2.0/LGPL 2.1





---

# Support

---

For any issues contact me at the e-mail given in the *Author* section.



---

# Author

---

Artur Barseghyan <[artur.barseghyan@gmail.com](mailto:artur.barseghyan@gmail.com)>



---

# Indices and tables

---

- *genindex*
- *modindex*
- *search*



---

# Python Module Index

---

## r

radar.\_\_init\_\_, ??  
radar.exceptions, ??  
radar.utils, ??