<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parsed Formats</td>
<td>3</td>
</tr>
<tr>
<td>1.1 Dates</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Times</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Time Zones</td>
<td>3</td>
</tr>
<tr>
<td>Where it Differs From ISO 8601</td>
<td>5</td>
</tr>
<tr>
<td>Installation</td>
<td>7</td>
</tr>
<tr>
<td>API</td>
<td>9</td>
</tr>
<tr>
<td>Authors</td>
<td>11</td>
</tr>
<tr>
<td>Index</td>
<td>13</td>
</tr>
</tbody>
</table>
This module parses the most common forms of ISO 8601 date strings (e.g. 2007-01-14T20:34:22+00:00) into datetime objects.

```python
>>> import iso8601
>>> iso8601.parse_date("2007-01-25T12:00:00Z")
datetime.datetime(2007, 1, 25, 12, 0, tzinfo=<iso8601.Utc>)
```

See the LICENSE file for the license this package is released under.

If you want more full featured parsing look at:

- https://arrow.readthedocs.io - arrow
- https://pendulum.eustace.io - pendulum
- https://labix.org/python-dateutil - python-dateutil
- https://docs.python.org/3/library/datetime.html#datetime.datetime.fromisoformat - Yes, Python 3 has built in parsing too!
You can parse full date + times, or just the date. In both cases a datetime instance is returned but with missing times defaulting to 0, and missing days / months defaulting to 1.

1.1 Dates

- YYYY-MM-DD
- YYYYYMDD
- YYYY-MM (defaults to 1 for the day)
- YYYY (defaults to 1 for month and day)

1.2 Times

- hh:mm:ss.nn
- hhmss.nn
- hh:mm (defaults to 0 for seconds)
- hhmm (defaults to 0 for seconds)
- hh (defaults to 0 for minutes and seconds)

1.3 Time Zones

- Nothing, will use the default timezone given (which in turn defaults to UTC).
- Z (UTC)
- +/-hh:mm
• +/-hhmm
• +/-hh
Where it Differs From ISO 8601

Known differences from the ISO 8601 spec:

- You can use a " " (space) instead of T for separating date from time.
- Days and months without a leading 0 (2 vs 02) will be parsed.
- If time zone information is omitted the default time zone given is used (which in turn defaults to UTC). Use a default of None to yield naive datetime instances.
CHAPTER 3

Installation

To install simply use pip:

```
pip install iso8601
```
iso8601.parse_date(datestring: str, default_timezone: Optional[datetime.timezone] = datetime.timezone.utc) → datetime.datetime

Parses ISO 8601 dates into datetime objects

The timezone is parsed from the date string. However it is quite common to have dates without a timezone (not strictly correct). In this case the default timezone specified in default_timezone is used. This is UTC by default.

Parameters

- **datestring** – The date to parse as a string
- **default_timezone** – A datetime tzinfo instance to use when no timezone is specified in the datestring. If this is set to None then a naive datetime object is returned.

Returns A datetime.datetime instance

Raises ParseError when there is a problem parsing the date or constructing the datetime instance.

exception iso8601.ParseError

Raised when there is a problem parsing a date string
CHAPTER 5

Authors

Currently active or previously active committers:

• Michael Twomey
• Julien Danjou
P

parse_date() (in module iso8601), 9
ParseError, 9