
requests-forecast Documentation

Release 0.6.2

Jeff Triplett

July 20, 2014

1	requests-forecast	3
1.1	Requirements	3
1.2	Installation	3
1.3	Documentation	3
1.4	Quickstart	3
1.5	License	5
2	Installation	7
3	Usage	9
4	Contributing	11
4.1	Types of Contributions	11
4.2	Get Started!	12
4.3	Pull Request Guidelines	12
4.4	Tips	13
5	Credits	15
6	History	17
6.1	0.6.2 (2014-05-07)	17
6.2	0.6.1 (2014-05-07)	17
6.3	0.6.0 (2013-12-11)	17
6.4	0.5.0 (2013-12-10)	17
6.5	0.4.2 (2013-12-10)	17
6.6	0.4.1 (2013-11-05)	18
6.7	0.4.0 (2013-10-31)	18
6.8	0.3.0 (2013-08-11)	18
6.9	0.2.0 (2013-08-11)	18
6.10	0.1.0 (2013-07-27)	18
7	Indices and tables	19

Contents:

requests-forecast

For complete docs and API options see: <https://developer.forecast.io/docs/v2>

1.1 Requirements

- Python 2.6+
- requests 1.2.0+

1.2 Installation

To install requests-forecast, simply:

```
$ pip install requests-forecast
```

1.3 Documentation

- Development version: <http://requests-forecast.rtf.d.org>.

1.4 Quickstart

1.4.1 Creating a client

```
>>> from requests_forecast import Forecast
>>> forecast = Forecast(apikey='YOUR-API-KEY')
```

1.4.2 Getting the current conditions

```
>>> current = forecast.get_currently()
>>> print current['temperature']
58.9
```

1.4.3 Getting minutely conditions

```
>>> current = forecast.get_minutely()
>>> current['summary']
u'Mostly cloudy for the hour.'
>>> current['data'][0].keys()
[u'precipIntensity', u'time']
```

1.4.4 Getting hourly conditions

```
>>> current = forecast.get_hourly()
>>> current['temperature']
59.52
```

```
>>> forecast.get_hourly()['summary']
Mostly cloudy until tomorrow afternoon.
```

```
>>> forecast.get_hourly()['data'][0]['temperature']
>>> 59.52
```

1.4.5 Getting daily conditions

```
>>> forecast.get_daily()['summary']
u'Mixed precipitation off-and-on throughout the week; temperatures peaking at 70\xb0 on Sunday.'
>>> forecast.get_daily()['data'][0]['temperatureMax']
63.85
>>> forecast.get_daily()['data'][0]['temperatureMin']
35.05
```

1.4.6 Getting alerts

```
>>> forecast.alerts()
{
  "expires": 1366898400,
  "uri": "http://alerts.weather.gov/cap/wwacapget.php?x=KS124EFAC89CD0.FreezeWarning.124EFAD6F320KS.7
  "title": "Freeze Warning for Douglas, KS"
}
```

Example usage

```
from datetime import datetime
from requests_forecast import Forecast

forecast = Forecast(apikey='YOUR-API-KEY')
data = forecast.get(latitude=38.9717, longitude=-95.235)

>>> forecast.get_currently()['temperature']
58.9

data = forecast.get(latitude=38.9717, longitude=-95.235,
                    time=datetime(year=2013, month=12, day=29))
```



```
>>> forecast.get_currently(['temperature'])  
58.9
```

```
>>> forecast.get_currently(['temperature'])  
36.75
```

1.5 License

New BSD

Installation

At the command line:

```
$ easy_install requests-forecast
```

Or, if you have pip installed (which you should):

```
$ pip install requests-forecast
```

Usage

To use requests-forecast in a project:

```
from requests_forecast import Forecast
```

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at <https://gitlab.com/jeff.triplett/requests-forecast/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitLab issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitLab issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

4.1.4 Write Documentation

requests-forecast could always use more documentation, whether as part of the official requests-forecast docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://gitlab.com/jeff.triplett/requests-forecast/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *requests-forecast* for local development.

1. Fork the *requests-forecast* repo on GitLab.

2. Clone your fork locally:

```
$ git clone git@gitlab.com:your_name_here/requests-forecast.git
```

3. Install your local copy into a virtualenv. Assuming you have *virtualenvwrapper* installed, this is how you set up your fork for local development:

```
$ mkvirtualenv requests-forecast
$ cd requests-forecast/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass *flake8* and the tests, including testing other Python versions with *tox*:

```
$ flake8 requests-forecast tests
$ python setup.py test
$ tox
```

To get *flake8* and *tox*, just *pip* install them into your virtualenv.

6. Commit your changes and push your branch to GitLab:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitLab website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in *README.rst*.
3. The pull request should work for Python 2.6, 2.7, and 3.3, and for PyPy. Check https://travis-ci.org/jefftriplett/requests-forecast/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests/test_requests_forecast.py
```


Credits

Primary Authors:

- Jeff Triplett <jefftriplett>

Thanks to:

- katy lavallee <katylava>
- Adam Fast <adamfast>

6.1 0.6.2 (2014-05-07)

- Fix for timezones
- Fix for travis.ci timezone failures

6.2 0.6.1 (2014-05-07)

- Updated fixtures and tests.
- Added Python 3.4 support.

6.3 0.6.0 (2013-12-11)

- Refactored DataBlock to merge DataPoint and DataBlock.
- Simplified `__getattr__` and removed extra field processing.
- Fixed alert class.
- Removed `parse_timestamps` since it's not used.
- Changed decimal to float conversions.

6.4 0.5.0 (2013-12-10)

- The Decimal update should have been a minor update instead of a patch.

6.5 0.4.2 (2013-12-10)

- Added Decimal support. Thanks adamfast.

6.6 0.4.1 (2013-11-05)

- Fix a few documentation typos.

6.7 0.4.0 (2013-10-31)

- Fix `__title__` meta data.
- Fixed date tests.
- Updated dependencies.

6.8 0.3.0 (2013-08-11)

- Updated docs.

6.9 0.2.0 (2013-08-11)

- Fixed `TypeError` on `get_alerts` reported by `katylava`.

6.10 0.1.0 (2013-07-27)

- First release on PyPI.

Indices and tables

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