

---

# Poetaster

*Release 0.1.1*

September 21, 2016



<b>1</b>	<b>Overview</b>	<b>1</b>
1.1	Installation . . . . .	1
1.2	Documentation . . . . .	1
1.3	Development . . . . .	1
<b>2</b>	<b>Installation</b>	<b>3</b>
<b>3</b>	<b>Usage</b>	<b>5</b>
<b>4</b>	<b>Reference</b>	<b>7</b>
4.1	poetaster . . . . .	7
<b>5</b>	<b>Contributing</b>	<b>9</b>
5.1	Bug reports . . . . .	9
5.2	Documentation improvements . . . . .	9
5.3	Feature requests and feedback . . . . .	9
5.4	Development . . . . .	9
<b>6</b>	<b>Authors</b>	<b>11</b>
<b>7</b>	<b>Changelog</b>	<b>13</b>
7.1	0.1.1 (2016-09-21) . . . . .	13
7.2	0.1.0dev1 (2016-07-27) . . . . .	13
<b>8</b>	<b>Indices and tables</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>



---

## Overview

---

docs	
tests	
package	

Poetry tricks and tools for found – and generated – poetry.

- Free software: BSD license

## 1.1 Installation

```
pip install poetaster
```

## 1.2 Documentation

<https://python-poetaster.readthedocs.io/>

## 1.3 Development

To run the all tests run:

```
tox
```

Note, to combine the coverage data from all the tox environments run:

Windows	set PYTEST_ADDOPTS=--cov-append tox
Other	PYTEST_ADDOPTS=--cov-append tox



---

## Installation

---

At the command line:

```
pip install poetaster
```





---

### Usage

---

To use Poetaster in a project:

```
import poetaster
```



---

**Reference**

---

**4.1 poetaster**



---

## Contributing

---

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

### 5.1 Bug reports

When [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.

### 5.2 Documentation improvements

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

### 5.3 Feature requests and feedback

The best way to send feedback is to file an issue at <https://github.com/jkahn/python-poetaster/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 code contributions are welcome :)

### 5.4 Development

To set up *python-poetaster* for local development:

1. Fork [python-poetaster](#) (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/python-poetaster.git
```

3. Create a branch for local development:

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

Now you can make your changes locally.

4. When you're done making changes, run all the checks, doc builder and spell checker with `tox` one command:

```
tox
```

5. Commit your changes and push your branch to GitHub:

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

6. Submit a pull request through the GitHub website.

### 5.4.1 Pull Request Guidelines

If you need some code review or feedback while you're developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`)<sup>1</sup>.
2. Update documentation when there's new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

### 5.4.2 Tips

To run a subset of tests:

```
tox -e envname -- py.test -k test_myfeature
```

To run all the test environments in *parallel* (you need to `pip install detox`):

```
detox
```

---

<sup>1</sup> If you don't have all the necessary python versions available locally you can rely on Travis - it will [run the tests](#) for each change you add in the pull request.  
It will be slower though ...

---

**Authors**

---

- Jeremy G. Kahn - <http://trochee.net/>





---

## Changelog

---

### 7.1 0.1.1 (2016-09-21)

- First release with functional haiku extraction.

### 7.2 0.1.0dev1 (2016-07-27)

- First release on PyPI.



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



**p**

poetaster, [7](#)



## P

poetaster (module), [7](#)