

---

# **nose-htmloutput**

*Release 0.6.0*

**Apr 14, 2017**



---

## Contents

---

<b>1</b>	<b>Overview</b>	<b>3</b>
1.1	nose-htmloutput . . . . .	3
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>Reference</b>	<b>9</b>
4.1	nose_htmloutput . . . . .	9
<b>5</b>	<b>Contributing</b>	<b>11</b>
5.1	Bug reports . . . . .	11
5.2	Documentation improvements . . . . .	11
5.3	Feature requests and feedback . . . . .	11
5.4	Development . . . . .	12
<b>6</b>	<b>Authors</b>	<b>13</b>
<b>7</b>	<b>Changelog</b>	<b>15</b>
7.1	0.1.0 (2014-01-01) . . . . .	15
7.2	0.3.0 (?) . . . . .	15
7.3	0.5.0 (2014-10-04) . . . . .	15
7.4	0.6.0 (2015-09-06) . . . . .	15
<b>8</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>



Contents:



### nose-htmlooutput

Nose plugin that generates a nice html test report.

- Free software: BSD license

### Installation

```
pip install nose-htmlooutput
```

### Usage

- |                         |  |
|-------------------------|--|
| <b>--with-html</b>      | Enable plugin HtmlOutput: Output test results as pretty html.<br>[NOSE_WITH_HTML]            |
| <b>--html-file=FILE</b> | Path to html file to store the report in. Default is nosetests.html in the working directory |

### Development

To run the all tests run:

```
tox
```

### Example





## CHAPTER 2

---

### Installation

---

At the command line:

```
pip install nose-httmloutput
```



## CHAPTER 3

---

### Usage

---

To use nose-htloutput in a project:

```
import nose_htloutput
```



## nose\_htmloutput

**class** `nose_htmloutput.HtmlOutput`

Output test results as pretty html.

**addError** (*test, err, capt=None*)

Add error output to Xunit report.

**addFailure** (*test, err, capt=None*)

Add failure output to Xunit report.

**configure** (*options, config*)

Configures the xunit plugin.

**options** (*parser, env*)

Sets additional command line options.

**report** (*stream*)

Writes an Xunit-formatted XML file

The file includes a report of test errors and failures.

`nose_htmloutput.exc_message` (*exc\_info*)

Return the exception's message.

`nose_htmloutput.nice_classname` (*obj*)

Returns a nice name for class object or class instance.

```
>>> nice_classname(Exception())
'...Exception'
>>> nice_classname(Exception)
'...Exception'
```



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

### 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.

### Documentation improvements

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

### Feature requests and feedback

The best way to send feedback is to file an issue at <https://github.com/ionelmc/nose-htlmloutput/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 :)

## Development

To set up *nose-htmloutput* for local development:

1. Fork *nose-htmloutput* on GitHub.
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/nose-htmloutput.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.

## 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*.

## 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 ...



## CHAPTER 6

---

### Authors

---

- Ionel Cristian Mărieș - <http://blog.ionelmc.ro>
- Zack Brunson - <https://github.com/zbrunson>



### **0.1.0 (2014-01-01)**

- First release on PyPI.

### **0.3.0 (?)**

- Fixed exception handling on Python 3.

### **0.5.0 (2014-10-04)**

- Made addError handle exceptions the same way as addFailure.

### **0.6.0 (2015-09-06)**

- Added Error column.
- Changed Fail/Error columns to be colored if there is a failure/error.



## CHAPTER 8

---

### Indices and tables

---

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



**n**

nose\_htmloutput, 9





## A

`addError()` (`nose_htmloutput.HtmlOutput` method), 9  
`addFailure()` (`nose_htmloutput.HtmlOutput` method), 9

## C

`configure()` (`nose_htmloutput.HtmlOutput` method), 9

## E

`exc_message()` (in module `nose_htmloutput`), 9

## H

`HtmlOutput` (class in `nose_htmloutput`), 9

## N

`nice_classname()` (in module `nose_htmloutput`), 9  
`nose_htmloutput` (module), 9

## O

`options()` (`nose_htmloutput.HtmlOutput` method), 9

## R

`report()` (`nose_htmloutput.HtmlOutput` method), 9