pytest-html

Dave Hunt

Oct 25, 2022

CONTENTS:

pytest-html is a plugin for pytest that generates a HTML report for test results.

CHAPTER

ONE

INSTALLATION

1.1 Requirements

pytest-html will work with Python >=3.6 or PyPy3.

1.2 Installing pytest-html

To install pytest-html using pip:

```
$ pip install pytest-html
```

To install from source:

```
$ pip install -e .
```

CHAPTER

TWO

USER GUIDE

2.1 ANSI codes

Note that ANSI code support depends on the ansi2html package. Due to the use of a less permissive license, this package is not included as a dependency. If you have this package installed, then ANSI codes will be converted to HTML in your report.

2.2 Creating a self-contained report

In order to respect the Content Security Policy (CSP), several assets such as CSS and images are stored separately by default. You can alternatively create a self-contained report, which can be more convenient when sharing your results. This can be done in the following way:

```
$ pytest --html=report.html --self-contained-html
```

Images added as files or links are going to be linked as external resources, meaning that the standalone report HTML file may not display these images as expected.

The plugin will issue a warning when adding files or links to the standalone report.

2.3 Enhancing reports

2.3.1 Appearance

Custom CSS (Cascasding Style Sheets) can be passed on the command line using the `--css` option. These will be applied in the order specified, and can be used to change the appearance of the report.

```
$ pytest --html=report.html --css=highcontrast.css --css=accessible.css
```

2.3.2 Report Title

By default the report title will be the filename of the report, you can edit it by using the `pytest_html_report_title` hook:

```
def pytest_html_report_title(report):
    report.title = "My very own title!"
```

2.3.3 Environment

The *Environment* section is provided by the pytest-metadata plugin, and can be accessed via the `pytest_configure` and `pytest_sessionfinish` hooks:

To modify the *Environment* section **before** tests are run, use `pytest_configure`:

```
def pytest_configure(config):
    config._metadata["foo"] = "bar"
```

To modify the *Environment* section **after** tests are run, use `pytest_sessionfinish`:

```
import pytest


@pytest.hookimpl(tryfirst=True)
def pytest_sessionfinish(session, exitstatus):
    session.config._metadata["foo"] = "bar"
```

Note that in the above example @pytest.hookimpl(tryfirst=True) is important, as this ensures that a best effort attempt is made to run your `pytest_sessionfinish` **before** any other plugins (including `pytest-html` and `pytest-metadata`) run theirs. If this line is omitted, then the *Environment* table will **not** be updated since the `pytest_sessionfinish` of the plugins will execute first, and thus not pick up your change.

The generated table will be sorted alphabetically unless the metadata is a `collections.OrderedDict`.

It is possible to redact variables from the environment table. Redacted variables will have their names displayed, but their values grayed out. This can be achieved by setting `environment_table_redact_list` in your INI configuration file (e.g.: `pytest.ini`). `environment_table_redact_list` is a `linelist` of regexes. Any environment table variable that matches a regex in this list has its value redacted.

For example, the following will redact all environment table variables that match the regexes `^foo$`, `.*redact.*`, or `bar`:

```
[pytest]
environment_table_redact_list = ^foo$
    .*redact.*
    bar
```

2.3.4 Additional summary information

You can edit the *Summary* section by using the `pytest_html_results_summary` hook:

```
from py.xml import html


def pytest_html_results_summary(prefix, summary, postfix):
    prefix.extend([html.p("foo: bar")])
```

2.3.5 Extra content

You can add details to the HTML report by creating an 'extra' list on the report object. Here are the types of extra content that can be added:

Type	Example
Raw HTML	`extra.html('<div>Additional HTML</div>')`
JSON	`extra.json({'name': 'pytest'})`
Plain text	`extra.text('Add some simple Text')`
URL	`extra.url('http://www.example.com/')`
Image	`extra.image(image, mime_type='image/gif', extension='gif')`
Image	`extra.image('/path/to/file.png')`
Image	`extra.image('http://some_image.png')`

Note: When adding an image from file, the path can be either absolute or relative.

Note: When using `--self-contained-html`, images added as files or links may not work as expected, see section *Creating a self-contained report* for more info.

There are also convenient types for several image formats:

Image format	Example
PNG	`extra.png(image)`
JPEG	`extra.jpg(image)`
SVG	`extra.svg(image)`

The following example adds the various types of extras using a `pytest_runtest_makereport` hook, which can be implemented in a plugin or conftest.py file:

```
import pytest


@pytest.hookimpl(hookwrapper=True)
def pytest_runtest_makereport(item, call):
    pytest_html = item.config.pluginmanager.getplugin("html")
    outcome = yield
    report = outcome.get_result()
    extra = getattr(report, "extra", [])
    if report.when == "call":
        # always add url to report
        extra.append(pytest_html.extras.url("http://www.example.com/"))
        xfail = hasattr(report, "wasxfail")
```

(continues on next page)

(continued from previous page)

```
        if (report.skipped and xfail) or (report.failed and not xfail):
            # only add additional html on failure
            extra.append(pytest_html.extras.html("<div>Additional HTML</div>"))
        report.extra = extra
```

You can also specify the `name` argument for all types other than `html` which will change the title of the created hyper link:

```
extra.append(pytest_html.extras.text("some string", name="Different title"))
```

It is also possible to use the fixture `extra` to add content directly in a test function without implementing hooks. These will generally end up before any extras added by plugins.

```
from pytest_html import extras


def test_extra(extra):
    extra.append(extras.text("some string"))
```

2.3.6 Modifying the results table

You can modify the columns of the report by implementing custom hooks for the header and rows. The following example `conftest.py` adds a description column with the test function docstring, adds a sortable time column, and removes the links column:

```
from datetime import datetime
from py.xml import html
import pytest


def pytest_html_results_table_header(cells):
    cells.insert(2, html.th("Description"))
    cells.insert(1, html.th("Time", class_="sortable time", col="time"))
    cells.pop()


def pytest_html_results_table_row(report, cells):
    cells.insert(2, html.td(report.description))
    cells.insert(1, html.td(datetime.utcnow(), class_="col-time"))
    cells.pop()


@pytest.hookimpl(hookwrapper=True)
def pytest_runtest_makereport(item, call):
    outcome = yield
    report = outcome.get_result()
    report.description = str(item.function.__doc__)
```

You can also remove results by implementing the `pytest_html_results_table_row` hook and removing all cells. The following example removes all passed results from the report:

```
def pytest_html_results_table_row(report, cells):
    if report.passed:
        del cells[:]
```

The log output and additional HTML can be modified by implementing the `pytest_html_results_html` hook. The following example replaces all additional HTML and log output with a notice that the log is empty:

```
from py.xml import html


def pytest_html_results_table_html(report, data):
    if report.passed:
        del data[:]
        data.append(html.div("No log output captured.", class_="empty log"))
```

2.4 Display options

2.4.1 Auto Collapsing Table Rows

By default, all rows in the **Results** table will be expanded except those that have `Passed`.

This behavior can be customized either with a query parameter: `?collapsed=Passed,XFailed,Skipped` or by setting the `render_collapsed` in a configuration file (pytest.ini, setup.cfg, etc).

```
[pytest]
render_collapsed = True
```

NOTE: Setting `render_collapsed` will, unlike the query parameter, affect all statuses.

2.4.2 Controlling Test Result Visibility Via Query Params

By default, all tests are visible, regardless of their results. It is possible to control which tests are visible on page load by passing the `visible` query parameter. To use this parameter, please pass a comma separated list of test results you wish to be visible. For example, passing `?visible=passed,skipped` will show only those tests in the report that have outcome `passed` or `skipped`.

Note that this match is case insensitive, so passing `PASSED` and `passed` has the same effect.

The following query parameters may be passed:

- `passed`
- `skipped`
- `failed`
- `error`
- `xfailed`
- `xpassed`
- `rerun`

2.4.3 Formatting the Duration Column

The formatting of the timestamp used in the `Durations` column can be modified by setting `duration_formatter` on the `report` attribute. All time.strftime formatting directives are supported. In addition, it is possible to supply `%f` to get duration milliseconds. If this value is not set, the values in the `Durations` column are displayed in `%S.%f` format where `%S` is the total number of seconds a test ran for.

Below is an example of a `conftest.py` file setting `duration_formatter`:

```
import pytest


@pytest.hookimpl(hookwrapper=True)
def pytest_runtest_makereport(item, call):
    outcome = yield
    report = outcome.get_result()
    setattr(report, "duration_formatter", "%H:%M:%S.%f")
```

NOTE: Milliseconds are always displayed with a precision of 2

CHAPTER

THREE

API REFERENCE

This is a reference to the plugin API.

3.1 Hooks

This plugin exposes the following hooks:

pytest_html.hooks.**pytest_html_report_title**(*report*)

Called before adding the title to the report

pytest_html.hooks.**pytest_html_results_summary**(*prefix*, *summary*, *postfix*)

Called before adding the summary section to the report

pytest_html.hooks.**pytest_html_results_table_header**(*cells*)

Called after building results table header.

pytest_html.hooks.**pytest_html_results_table_html**(*report*, *data*)

Called after building results table additional HTML.

pytest_html.hooks.**pytest_html_results_table_row**(*report*, *cells*)

Called after building results table row.

CHAPTER

FOUR

DEVELOPMENT

To contribute to *pytest-html* you can use Pipenv to manage a python virtual environment and pre-commit to help you with styling and formatting.

To setup the virtual environment and pre-commit, run:

```
$ pipenv install --dev
$ pipenv run pre-commit install
```

If you're not using Pipenv, run the following to install pre-commit:

```
$ pip install pre-commit
$ pre-commit install
```

4.1 Automated Testing

All pull requests and merges are tested in GitHub Actions which are defined inside `.github` folder.

To retrigger CI to run again for a pull request, you either use dropdown option, close and reopen pull-request or to just update the branch containing it.

You can do this with *git commit –allow-empty*

4.2 Running Tests

4.2.1 Python

You will need Tox installed to run the tests against the supported Python versions. If you're using Pipenv it will be installed for you.

With Pipenv, run:

```
$ pipenv run tox
```

Otherwise, to install and run, do:

```
$ pip install tox
$ tox
```

4.2.2 JavaScript

You will need npm installed to run the JavaScript tests. Internally, we use Grunt and QUnit to run the tests.

Once npm is installed, you can install all needed dependencies by running:

```
$ npm install
```

Run the following to execute the tests:

```
$ npm test
```

4.3 Documentation

Documentation is hosted on Read the Docs, and is written in RST. Remember to add any new files to the `toctree` section in `index.rst`.

To build your documentation, run:

```
$ tox -e docs
```

You can then run a local webserver to verify your changes compiled correctly.

4.4 SASS/SCSS/CSS

You will need npm installed to compile the CSS, which is generated via SASS/SCSS.

Once npm is installed, you can install all needed dependencies by running:

```
$ npm install
```

Run the following to generate the CSS:

```
$ npm run build:css
```

4.5 Releasing a new version

Follow these steps to release a new version of the project:

1. Update your local master with the upstream master (`git pull --rebase upstream master`)
2. Create a new branch
3. Update the changelog with the new version, today's date, and all changes/new features
4. Commit and push the new branch and then create a new pull request
5. Wait for tests and reviews and then merge the branch
6. Once merged, update your local master again (`git pull --rebase upstream master`)
7. Tag the release with the new release version (`git tag v<new tag>`)
8. Push the tag (`git push upstream --tags`)

9. Done. Check Github Actions for release progress.

CHAPTER

FIVE

CHANGELOG

Versions follow Semantic Versioning (`<major>.<minor>.<patch>`).

5.1 Version History

5.1.1 3.2.0 (2022-10-25)

- Explicitly add py.xml dependency.
  - Thanks to @smartEBL for the PR
- Implement the `visible` URL query parameter to control visibility of test results on page load. (#399)
  - Thanks to @TheCorp for reporting and @gnikonorov for the fix
- Make the report tab title reflect the report name. (#412)
  - Thanks to @gnikonorov for the PR
- Implement `environment_table_redact_list` to allow for redaction of environment table values. (#233)
  - Thanks to @fenchu for reporting and @gnikonorov for the PR

5.1.2 3.1.1 (2020-12-13)

- Fix issue with reporting of missing CSS files. (#388)
  - Thanks to @prakhargurunani for reporting and fixing!

5.1.3 3.1.0 (2020-12-2)

- Stop attaching test reruns to final test report entries (#374)
  - Thanks to @VladimirPodolyan for reporting and @gnikonorov for the fix
- Allow for report duration formatting (#376)
  - Thanks to @brettnolan for reporting and @gnikonorov for the fix

5.1.4 3.0.0 (2020-10-28)

- Respect `--capture=no`, `--show-capture=no`, and `-s` pytest flags (#171)
  - Thanks to @bigunyak for reporting and @gnikonorov for the fix
- Make the `Results` table `Links` column sortable (#242)
  - Thanks to @vashirov for reporting and @gnikonorov for the fix
- Fix issue with missing image or video in extras. (#265 and pytest-selenium#237)
  - Thanks to @p00j4 and @anothermattbrown for reporting and @christiansandberg and @superdodd and @dhalperi for the fix
- Fix attribute name for compatibility with `pytest-xdist` 2. (#305)
  - Thanks to @Zac-HD for the fix
- Post process HTML generation to allow teardown to appear in the HTML output. (#131)
  - Thanks to @iwanb for reporting and @csm10495 for the fix

5.1.5 2.1.1 (2020-03-18)

- Fix issue with funcargs causing failures. (#282)
  - Thanks to @ssbarnea for reporting and @christiansandberg for the fix

5.1.6 2.1.0 (2020-03-09)

- Added support for MP4 video format. (#260)
  - Thanks to @ExaltedBagel for the PR
- Added support for sorting metadata by key. (#245)
  - Thanks to @ssbarnea for reporting and @ExaltedBagel for the fix
- Added support for rendering reports collapsed (#239)
  - Thanks to @Wramberg for suggesting this enhancement
- Added *extra* fixture (#269)
  - Thanks to @christiansandberg for the PR
- Added ability to change report title using hook (#270)
  - Thanks to @werdeil for the PR

5.1.7 2.0.1 (2019-10-05)

- Properly check for presence of CSS file. (#246)
  - Thanks to @wanam for reporting, and @krzysztof-pawlik-gat for the fix
- Added support for UTF-8 display. (#244)
  - Thanks to @Izhu666 for the PR
- Fix initial sort on column. (#247)

  - Thanks to @wanam for reporting and fixing

5.1.8 2.0.0 (2019-09-09)

- Drop support for Python 2.7. We will continue to accept patches to `1.22.x` for the time being.
  - Thanks to @hugovk for the PR

5.1.9 1.22.0 (2019-08-06)

- Refactor assets naming to be more readable and OS safe.
  - This solves multiple reported issues, mainly from Windows users.
  - Thanks to @franz-95 and @Uil2Liv for reporting and testing fixes.
- Add line break to log section of the report.
  - Thanks to @borntyping for reporting and fixing!

5.1.10 1.21.1 (2019-06-19)

- Fix issue with assets filenames being too long.
  - Thanks to @D3X for reporting and providing a fix

5.1.11 1.21.0 (2019-06-17)

- Allow opening generated html report in browser (@ssbarnea)
- Handle when report title is stored as an environment variable (@BeyondEvil)
- Change assets naming method (@SunInJuly)

5.1.12 1.20.0 (2019-01-14)

- Tests running with Pytest 4.0 and Python 3.7
- Stop filtering out falsy environment values (#175)
  - Thanks to @jknotts for reporting the issue and to @crazymerlyn for providing a fix
- Removed extraneous space from anchor tag (@chardbury)
- Always define __version__ even if get_distribution() fails (@nicoddemus)
- Refactor css config code (@crazymerlyn)

5.1.13 1.19.0 (2018-06-01)

- Allow collapsed outcomes to be configured by using a query parameter
  - Thanks to @Formartha for suggesting this enhancement and to @jacebrowning for providing a patch

5.1.14 1.18.0 (2018-05-22)

- Preserve the order if metadata is `OrderedDict`
  - Thanks to @jacebrowning for suggesting this enhancement and providing a patch

5.1.15 1.17.0 (2018-04-05)

- Add support for custom CSS (#116)
  - Thanks to @APshenkin for reporting the issue and to @i-am-david-fernandez for providing a fix
- Report collection errors (#148)
  - Thanks to @Formartha for reporting the issue
- Add hook for modifying summary section (#109)
  - Thanks to @shreyashah for reporting the issue and to @j19sch for providing a fix
- Add filename to report as heading
  - Thanks to @j19sch for the PR

5.1.16 1.16.1 (2018-01-04)

- Fix for including screenshots on Windows (#124)
  - Thanks to @ngavrish for reporting the issue and to @pinkie1378 for providing a fix

5.1.17 1.16.0 (2017-09-19)

- Improve rendering of collections in metadata (@rasmuspeders1)

5.1.18 1.15.2 (2017-08-15)

- Always decode byte string in extra text
  - Thanks to @ch-t for reporting the issue and providing a fix

5.1.19 1.15.1 (2017-06-12)

- Fix pytest dependency to 3.0 or later
  - Thanks to @silvana-i for reporting the issue and to @nicoddemus for providing a fix

5.1.20 1.15.0 (2017-06-09)

- Fix encoding issue in longrepr values
  - Thanks to @tomga for reporting the issue and providing a fix
- Add ability to specify images as file or URL
  - Thanks to @BeyondEvil for the PR

5.1.21 1.14.2 (2017-03-10)

- Always encode content for data URI
  - Thanks to @micheletest and @BeyondEvil for reporting the issue and confirming the fix

5.1.22 1.14.1 (2017-02-28)

- Present metadata without additional formatting to avoid issues due to unpredictable content types

5.1.23 1.14.0 (2017-02-27)

- Add hooks for modifying the test results table
- Replace environment section with values from pytest-metadata
- Fix encoding for asset files
- Escape contents of log sections

5.1.24 1.13.0 (2016-12-19)

- Disable ANSI codes support by default due to dependency on ansi2html package with less permissive licensing

5.1.25 1.12.0 (2016-11-30)

- Add support for JPG and SVG images (@bhzunami)
- Add version number and PyPI link to report header (@denisra)

5.1.26 1.11.1 (2016-11-25)

- Fix title of checkbox disappearing when unchecked (@vashirov)

5.1.27 1.11.0 (2016-11-08)

- Add support for ANSI codes in logs (@premkarat)

5.1.28 1.10.1 (2016-09-23)

- Fix corrupt image asset files
- Remove image links from self-contained report
- Fix issue with unexpected passes not being reported in pytest 3.0

5.1.29 1.10.0 (2016-08-09)

- Hide filter checkboxes when JavaScript is disabled (@RibeiroAna)
- Removed rerun outcome unless the plugin is active (@RibeiroAna)
- Introduce `--self-contained-html` option to store CSS and assets inline (@RibeiroAna)
- Save images, text, and JSON extras as files in an assets directory (@RibeiroAna)
- Use an external CSS file (@RibeiroAna)
- Set initial sort order in the HTML (@RibeiroAna)
- Allow visibility of extra details to be toggled (@leitzler)

5.1.30 1.9.0 (2016-07-04)

- Split pytest_sessionfinish into generate and save methods (@karandesai-96)
- Show tests rerun by pytest-rerunfailures plugin (@RibeiroAna)
- Added a feature to filter tests by outcome (@RibeiroAna)

5.1.31 1.8.1 (2016-05-24)

- Include captured output for passing tests

5.1.32 1.8.0 (2016-02-24)

- Remove duplication from the environment section
- Dropped support for Python 3.2
- Indicated setup and teardown in report
- Fixed colour of errors in report

5.1.33 1.7 (2015-10-19)

- Fixed INTERNALERROR when an xdist worker crashes (@The-Compiler)
- Added report sections including stdout and stderr to log

5.1.34 1.6 (2015-09-08)

- Fixed environment details when using pytest-xdist

5.1.35 1.5.1 (2015-08-18)

- Made environment fixture session scoped to avoid repeating content

5.1.36 1.5 (2015-08-18)

- Replaced custom hook for setting environemnt section with a fixture

5.1.37 1.4 (2015-08-12)

- Dropped support for pytest 2.6
- Fixed unencodable strings for Python 3 (@The-Compiler)

5.1.38 1.3.2 (2015-07-27)

- Prevented additional row if log has no content or there is no extra HTML

5.1.39 1.3.1 (2015-05-26)

- Fixed encoding issue in Python 3

5.1.40 1.3 (2015-05-26)

- Show extra content regardless of test result
- Added support for extra content in JSON format

5.1.41 1.2 (2015-05-20)

- Changed default sort order to test result (@The-Compiler)

5.1.42 1.1 (2015-05-08)

- Added Python 3 support

5.1.43 1.0 (2015-04-20)

- Initial release

PYTHON MODULE INDEX

p

INDEX

M

P