
Documentation httpclient_{logging}

Release 1.0

owner

Feb 19, 2024

Contents

1	Installation	2
2	Usage	2
3	Registered sitecustomize-entrypoint	2
4	Compatibility	2
5	Notable dependencies	2
6	Changelog	3
6.1	Unreleased (YYYY-MM-DD)	3
6.2	1.0 (2023-03-30)	3
7	Security Policy	3
7.1	Supported Versions	3
7.2	Reporting a Vulnerability	3
8	MIT License	4
9	Contributors	4
10	How to contribute	4
10.1	Development environment setup	4
10.2	Issues and feature requests	5
11	Contributor Covenant Code of Conduct	5
11.1	Our Pledge	5
11.2	Our Standards	5
11.3	Enforcement Responsibilities	6
11.4	Scope	6
11.5	Enforcement	6
11.6	Enforcement Guidelines	6
11.7	Attribution	7
12	httpclient_logging	7
12.1	Submodules	7
12.2	Package Contents	9
13	Indices and tables	9
	Python Module Index	10

1 Installation

Install via pip:

```
> bin/pip install httpclient-logging
```

Or add to your poetry-based project:

```
> poetry add httpclient-logging
```

2 Usage

The only thing left to do for you is the create a `.env` in the root of your project.

3 Registered sitecustomize-entrypoint

The `httpclient_logging.entrypoint`-function is registered as a `sitecustomize-entrypoint` in our `pyproject.toml`:

```
[tool.poetry.plugins]
[tool.poetry.plugins."sitecustomize"]
httpclient_logging = "httpclient_logging:entrypoint"
```

Sitecustomize and all its registered entrypoints will be executed at the start of *every* python-process. For more information, please see [sitecustomize-entrypoints](#)

4 Compatibility

`httpclient-logging` works on Python 3.8+, including PyPy3. Tested until Python 3.11,

5 Notable dependencies

- [sitecustomize-entrypoints](#)

6 Changelog

All notable changes to this project will be documented in this file.

6.1 Unreleased (YYYY-MM-DD)

- Remove `httpclient_logging.patch.cancel-function`, since this is now available in `sitecustomize-entrpoints>=1.1.0`
- Depend on `sitecustomize-entrpoints>=1.1.0`
-
- Remove range-pinning `python = ">=3.8.0,<4.0"`, only specify bottom-version `">=3.8.0"`
- Remove range-pinned dependencies `tox`, `nox` that require us to pin `python = "<4.0"`

6.2 1.0 (2023-03-30)

- Refactored and renamed entrypoint into `autoread_dotenv.entrypoint`.
- Add boilerplate-files to comply with Github's *Community Standards*
- Test releasing via `poetry-release`.
- Convert docs from restructured text to markdown.
- Update project-description in `pyproject.toml`. [WouterVH]
- Add `readthedocs.yaml`.
- Package created by [Wouter Vanden Hove wouter@libranet.eu]

7 Security Policy

7.1 Supported Versions

Use this section to tell people about which versions of your project are currently being supported with security updates.

Version Supported	————	—————	0.x	:white_check_mark:	1.0.x	:white_check_mark:	
---------------------	------	-------	-----	--------------------	-------	--------------------	--

7.2 Reporting a Vulnerability

This project follows a 90 day disclosure timeline.

To report a security issue, please email security@libranet.eu with

- a description of the issue
- the steps you took to create the issue,
- affected versions
- and if known, mitigations for the issue

Our team will acknowledge receiving your email within 3 working days.

8 MIT License

Copyright (c) 2023 [Libranet.eu](https://libranet.eu).

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

9 Contributors

Special thanks for all the people who have helped on this project so far.

Append your name if you have contributed to this package. We use anti-chronological ordering (oldest on top).

- [Wouter Vanden Hove <wouter@libranet.eu>](mailto:wouter@libranet.eu)

10 How to contribute

When contributing to this repository, please first discuss the change you wish to make via issue, email, or any other method with the owners of this repository before making a change.

Please note we have a [code of conduct](#), please follow it in all your interactions with the project.

10.1 Development environment setup

Proceed to describe how to setup local development environment. e.g:

To set up a development environment, please follow these steps:

1. Clone the repo

```
git clone https://github.com/libranet/httpclient-logging
```

2. Run make install

```
make install
```

10.2 Issues and feature requests

You've found a bug in the source code, a mistake in the documentation or maybe you'd like a new feature? Take a look at [GitHub Discussions](#) to see if it's already being discussed.

You can help us by [submitting an issue on GitHub](#). Before you create an issue, make sure to search the issue archive – your issue may have already been addressed.

Please try to create bug reports that are:

- *Reproducible*. Include steps to reproduce the problem.
- *Specific*. Include as much detail as possible: which version, what environment, etc.
- *Unique*. Do not duplicate existing opened issues.
- *Scoped to a Single Bug*. One bug per report.

Even better: Submit a pull request with a fix or new feature!

How to submit a Pull Request

1. Search our repository for open or closed [Pull Requests](#) that relate to your submission. You don't want to duplicate effort.
2. Fork the project
3. Create your feature branch (`git checkout -b feat/amazing_feature`)
4. Commit your changes (`git commit -m 'feat: add amazing_feature'`) `httpclient`-logging uses [conventional commits](#), so please follow the specification in your commit messages.
5. Push to the branch (`git push origin feat/amazing_feature`)
6. [Open a Pull Request](#)

11 Contributor Covenant Code of Conduct

11.1 Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

11.2 Our Standards

Examples of behavior that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people
- Being respectful of differing opinions, viewpoints, and experiences
- Giving and gracefully accepting constructive feedback
- Accepting responsibility and apologizing to those affected by our mistakes, and learning from the experience
- Focusing on what is best not just for us as individuals, but for the overall community

Examples of unacceptable behavior include:

- The use of sexualized language or imagery, and sexual attention or advances of any kind

- Trolling, insulting or derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or email address, without their explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

11.3 Enforcement Responsibilities

Community leaders are responsible for clarifying and enforcing our standards of acceptable behavior and will take appropriate and fair corrective action in response to any behavior that they deem inappropriate, threatening, offensive, or harmful.

Community leaders have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

11.4 Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is officially representing the community in public spaces. Examples of representing our community include using an official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

11.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported to the community leaders responsible for enforcement at security@libranet.eu. All complaints will be reviewed and investigated promptly and fairly.

All community leaders are obligated to respect the privacy and security of the reporter of any incident.

11.6 Enforcement Guidelines

Community leaders will follow these Community Impact Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

1. Correction

Community Impact: Use of inappropriate language or other behavior deemed unprofessional or unwelcome in the community.

Consequence: A private, written warning from community leaders, providing clarity around the nature of the violation and an explanation of why the behavior was inappropriate. A public apology may be requested.

2. Warning

Community Impact: A violation through a single incident or series of actions.

Consequence: A warning with consequences for continued behavior. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.

3. Temporary Ban

Community Impact: A serious violation of community standards, including sustained inappropriate behavior.

Consequence: A temporary ban from any sort of interaction or public communication with the community for a specified period of time. No public or private interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, is allowed during this period. Violating these terms may lead to a permanent ban.

4. Permanent Ban

Community Impact: Demonstrating a pattern of violation of community standards, including sustained inappropriate behavior, harassment of an individual, or aggression toward or disparagement of classes of individuals.

Consequence: A permanent ban from any sort of public interaction within the community.

11.7 Attribution

This Code of Conduct is adapted from the [Contributor Covenant](https://www.contributor-covenant.org/version/2/0/code_of_conduct.html), version 2.0, available at https://www.contributor-covenant.org/version/2/0/code_of_conduct.html.

Community Impact Guidelines were inspired by [Mozilla's code of conduct enforcement ladder](#).

For answers to common questions about this code of conduct, see the FAQ at <https://www.contributor-covenant.org/faq>. Translations are available at <https://www.contributor-covenant.org/translations>.

12 httpclient_logging

`httpclient_logging.__init__.`

12.1 Submodules

`httpclient_logging.patch`

`httpclient_logging.patch.`

Module Contents

Functions

<code>set_httpclient_debuglevel([debug_level])</code>	if http-debuglevel > 0, debug messages in the
<code>patch_httpclient_print()</code>	Patch the print-function used in http.client to use a call to log.debug() instead.
<code>unpatch_httpclient_print()</code>	Unpatch the print-function used in http.client.
<code>configure()</code>	Configure the http.client.HTTPConnection-class
<code>undo()</code>	Undo the configured steps.

Attributes

<code>pre_patched_value</code>
<code>log</code>

`httpclient_logging.patch.pre_patched_value`

`httpclient_logging.patch.log`

`httpclient_logging.patch.set_httpclient_debuglevel(debug_level=None)`

if http-debuglevel > 0, debug messages in the http.client.HTTPConnection-class will be printed to STDOUT.

Return type

None

`httpclient_logging.patch.patch_httpclient_print()`

Patch the print-function used in http.client to use a call to log.debug() instead.

Return type

None

`httpclient_logging.patch.unpatch_httpclient_print()`

Unpatch the print-function used in http.client.

Return type

None

`httpclient_logging.patch.configure()`

Configure the http.client.HTTPConnection-class

Configure this class to use the debuglevel from an environment-variable DEBU-
GLEVEL_HTTPCONNECTION and to use a logger instead of a print-statements to output to standard
output.

Return type

None

`httpclient_logging.patch.undo()`

Undo the configured steps.

Return type

None

12.2 Package Contents

```
httpclient_logging.__version__ = '1.0'
```

```
httpclient_logging.__copyright__ = 'Copyright 2023 Libranet - MIT License.'
```

13 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Python Module Index

h

`httpclient_logging`, [7](#)

`httpclient_logging.patch`, [7](#)

Index

Symbols

`__copyright__` (*in module `httpclient_logging`*), 9

`__version__` (*in module `httpclient_logging`*), 9

C

`configure()` (*in module `httpclient_logging.patch`*), 8

H

`httpclient_logging`

module, 7

`httpclient_logging.patch`

module, 7

L

`log` (*in module `httpclient_logging.patch`*), 8

M

module

`httpclient_logging`, 7

`httpclient_logging.patch`, 7

P

`patch_httpclient_print()` (*in module `httpclient_logging.patch`*), 8

`pre_patched_value` (*in module `httpclient_logging.patch`*), 8

S

`set_httpclient_debuglevel()` (*in module `httpclient_logging.patch`*), 8

U

`undo()` (*in module `httpclient_logging.patch`*), 8

`unpatch_httpclient_print()` (*in module `httpclient_logging.patch`*), 8