PEP 541 – Package Index Name Retention

Author: Łukasz Langa <lukasz at python.org>
BDFL-Delegate: Mark Mangoba <mmangoba at python.org>
Discussions-To: Distutils-SIG list
Status: Final
Type: Process
Topic: Packaging
Created: 12-Jan-2017
Post-History:
Resolution: Distutils-SIG message

Abstract

This PEP proposes an extension to the Terms of Use [1] of the Package Index [2], clarifying expectations of package owners regarding ownership of a package name on the Package Index, specifically with regards to conflict resolution.

Existing package repositories such as CPAN [3], NPM [4], and GitHub [5] will be investigated as prior art in this field.

Rationale

Given that package names on the Index are sharing a single flat namespace, a unique name is a finite resource. The growing age of the Package Index causes a constant rise of situations of conflict between the current use of the name and a different suggested use of the same name.

This document aims to provide general guidelines for solving the most typical cases of such conflicts.

Approval Process

As the application of this policy has potential legal ramifications for the Python Software Foundation, the approval process used is more formal than that used for most PEPs.

Rather than accepting the PEP directly, the assigned BDFL-Delegate will instead recommend its acceptance to the PSF's Packaging Working Group. After consultation with the PSF's General Counsel, adoption of the policy will then be subject to a formal vote within the working group.

This formal approval process will be used for both initial adoption of the policy, and for adoption of any future amendments.

Specification

The main idea behind this document is that the Package Index serves the community. Every user is invited to upload content to the Package Index under the Terms of Use, understanding that it is at the sole risk of the user.

While the Package Index is not a backup service, the maintainers of the Package Index do their best to keep that content accessible indefinitely in its published form. However, in certain edge cases the greater community's needs might overweigh the individual's expectation of ownership of a package name.

The use cases covered by this document are:

- Abandoned projects:
  - continued maintenance by a different set of users; or
  - removal from the Index for use with a different project.
- Active projects:
  - resolving disputes over a name.
- Invalid projects:
  - projects subject to a claim of intellectual property infringement.

The proposed extension to the Terms of Use, as expressed in the Implementation section, will be published as a separate document on the Package Index, linked next to existing Terms of Use in the front page footer.

Implementation

Reachability

The user of the Package Index is solely responsible for being reachable by the Package Index maintainers for matters concerning projects that the user owns. In every case where contacting the user is necessary, the maintainers will try to do so at least three times, using the following means of contact:

- the e-mail address on file in the user's profile on the Package Index;
- the e-mail address listed in the Author field for a given project uploaded to the Index; and
- any e-mail addresses found in the given project's documentation on the Index or on the listed Home Page.

The maintainers stop trying to reach the user after six weeks.

Abandoned projects

A project is considered *abandoned* when ALL of the following are met:

- owner not reachable (see Reachability above);
- no releases within the past twelve months; and
- no activity from the owner on the project's home page (or no home page listed).

All other projects are considered *active*.

Continued maintenance of an abandoned project

If a candidate appears willing to continue maintenance on an *abandoned* project, ownership of the name is transferred when ALL of the following are met:

- the project has been determined *abandoned* by the rules described above;

- the candidate is able to demonstrate their own failed attempts to contact the existing owner;
- the candidate is able to demonstrate improvements made on the candidate's own fork of the project;
- the candidate is able to demonstrate why a fork under a different name is not an acceptable workaround; and
- the maintainers of the Package Index don't have any additional reservations.

Under no circumstances will a name be reassigned against the wishes of a reachable owner.

Removal of an abandoned project

Projects are never removed from the Package Index solely on the basis of abandonment. Artifacts uploaded to the Package Index hold inherent historical value.

An *abandoned* project can be transferred to a new owner for purposes of reusing the name when ALL of the following are met:

- the project has been determined *abandoned* by the rules described above;
- the candidate is able to demonstrate their own failed attempts to contact the existing owner;
- the candidate is able to demonstrate that the project suggested to reuse the name already exists and meets notability requirements;
- the candidate is able to demonstrate why a fork under a different name is not an acceptable workaround;
- download statistics on the Package Index for the existing package indicate project is not being used; and
- the maintainers of the Package Index don't have any additional reservations.

Name conflict resolution for active projects

The maintainers of the Package Index are not arbiters in disputes around *active* projects. There are many possible scenarios here, a non-exclusive list describing some real-world examples is presented below. None of the following qualify for package name ownership transfer:

1. User A and User B share project X. After some time they part ways and each of them wants to continue the project under name X.
2. User A owns a project X outside the Package Index. User B creates a package under the name X on the Index. After some time, User A wants to publish project X on the Index but realizes name is taken. This is true even if User A's project X gains notability and the User B's project X is not notable.
3. User A publishes project X to the Package Index. After some time User B proposes bug fixes to the project but no new release is published by User A. This is true even if User A agrees to publish a new version and later doesn't, even if User B's changes are merged to the source code repository for project X.

Again, the list above is not exclusive. The maintainers of the Package Index recommend users to get in touch with each other and solve the issue by respectful communication (see the PSF Code of Conduct [6]).

Invalid projects

A project published on the Package Index meeting ANY of the following is considered invalid and will be removed from the Index:

- project does not conform to Terms of Use;
- project is malware (designed to exploit or harm systems or users directly, to facilitate command-and-control attacks, or perform data exfiltration);
- project is spam (designed to advertise or solicit goods or services);
- project contains illegal content;
- project violates copyright, trademarks, patents, or licenses;
- project is name squatting (package has no functionality or is empty);
- project name, description, or content violates the Code of Conduct;
- project uses obfuscation to hide or mask functionality; or
- project is abusing the Package Index for purposes it was not intended.

The Package Index maintainers pre-emptively declare certain package names as unavailable for security reasons.

Intellectual property policy

It is the policy of Python Software Foundation and the Package Index maintainers to be appropriately responsive to claims of intellectual property infringement by third parties. It is not the policy of the Python Software Foundation nor the Package Index maintainers to pre-screen uploaded packages for any type of intellectual property infringement.

Possibly-infringing packages should be reported to legal@python.org and counsel to the Python Software Foundation will determine an appropriate response. A package can be removed or transferred to a new owner at the sole discretion of the Python Software Foundation to address a claim of infringement.

A project published on the Package Index meeting ANY of the following may be considered infringing and subject to removal from the Index or transferral to a new owner:

- project contains unlicensed copyrighted material from a third party, and is subject to a properly made claim under the DMCA;
- project uses a third party's trademark in a way not covered by nominal or fair use guidelines;
- project clearly implicates a patented system or process, and is the subject of a complaint; or
- project is subject to an active lawsuit.

In the event of a complaint for intellectual property infringement, a copy of the complaint will be sent to the package owner. In some cases, action may be taken by the Package Index maintainers before the owner responds.

The role of the Python Software Foundation

The Python Software Foundation [7] is the non-profit legal entity that provides the Package Index as a community service.

The Package Index maintainers can escalate issues covered by this document for resolution by the Packaging Workgroup if the matter is not clear enough. Some decisions *require* additional judgement by the Board, especially in cases of Code of

Conduct violations or legal claims. Recommendations made by the Board are sent to the Packaging Workgroup [8] for review.

The Packaging Workgroup has the final say in any disputes covered by this document and can decide to reassign or remove a project from the Package Index after careful consideration even when not all requirements listed here are met.

How to request a name transfer

If you want to take over an existing project name on PyPI, these are the steps to follow:

1. Try to contact the current owner(s) directly: email them and open an issue if you can find a related repository. The processes described here are meant as a last resort if the owner cannot be contacted.
2. Check the criteria above to see when a transfer is allowed. In particular, the criteria for reusing a name for a different project are more stringent than for continuing maintenance of the same project - although it's not easy to get a name transferred in either case.
3. Search the PyPI Support issues to see if anyone else is already requesting the same name.
4. If all the criteria are met to transfer ownership of the name, open a new issue to request it, detailing why you believe each relevant criterion is satisfied.

Prior art

NPM contains a separate section linked from the front page called Package Name Disputes. It is described as a "living document", as of January 2017 its contents might be summarized as follows:

- package name squatting is prohibited;
- users wanting to reuse a project name are required to contact the existing author, with cc to support@npmjs.com;
- all contact must conform to the NPM Code of Conduct;
- in case of no resolution after a few weeks, npm inc. holds the right to the final decision in the matter.

CPAN lets any user upload modules with the same name. PAUSE, a related index, only lists modules uploaded by the primary maintainer or listed co-maintainers. CPAN documentation doesn't address disputes otherwise.

GitHub's terms of service contain an exhaustive list of behavior not meeting general conditions of use. While not codified anywhere, GitHub does agree for users to reclaim abandoned account names by archiving the abandoned account and letting the other user or organization rename their account. This is done on a case-by-case basis.

Rejected Proposals

The original approach was to hope for the best and solve issues as they arise without written policy. This is not sustainable. The lack of generally available guidelines in writing on package name conflict resolution is causing unnecessary tensions. From the perspective of users, decisions made by the Package Index maintainers without written guidelines may appear arbitrary. From the perspective of the Package Index maintainers, solving name conflicts is a stressful task due to risk of unintentional harm due to lack of defined policy.

References

[1] Terms of Use of the Python Package Index (https://pypi.org/policy/terms-of-use/)

[2] The Python Package Index (https://pypi.org/)

[3] The Comprehensive Perl Archive Network (http://www.cpan.org/)

[4] Node Package Manager (https://www.npmjs.com/package/left-pad)

[5] GitHub (https://github.com/)

[6] Python Community Code of Conduct (https://www.python.org/psf/codeofconduct/)

[7]

Python Software Foundation (https://www.python.org/psf/)

[8]

Python Packaging Working Group (https://wiki.python.org/psf/PackagingWG/)

Copyright

This document has been placed in the public domain.

Acknowledgements

The many participants of the Distutils and Catalog SIGs for their ideas over the years.

Source: https://github.com/python/peps/blob/main/peps/pep-0541.rst

Last modified: 2023-09-09 17:39:29 GMT