pypa

CONDA

Paketierung und Distribution für Python-Nutzer

Jannis Leidel, PyPA & conda

Leipzig Python User Group
11.07.2023

Paketierungsschnabeltier von https://monotreme.club/@dholth und gezeichnet von @eruditebaboon

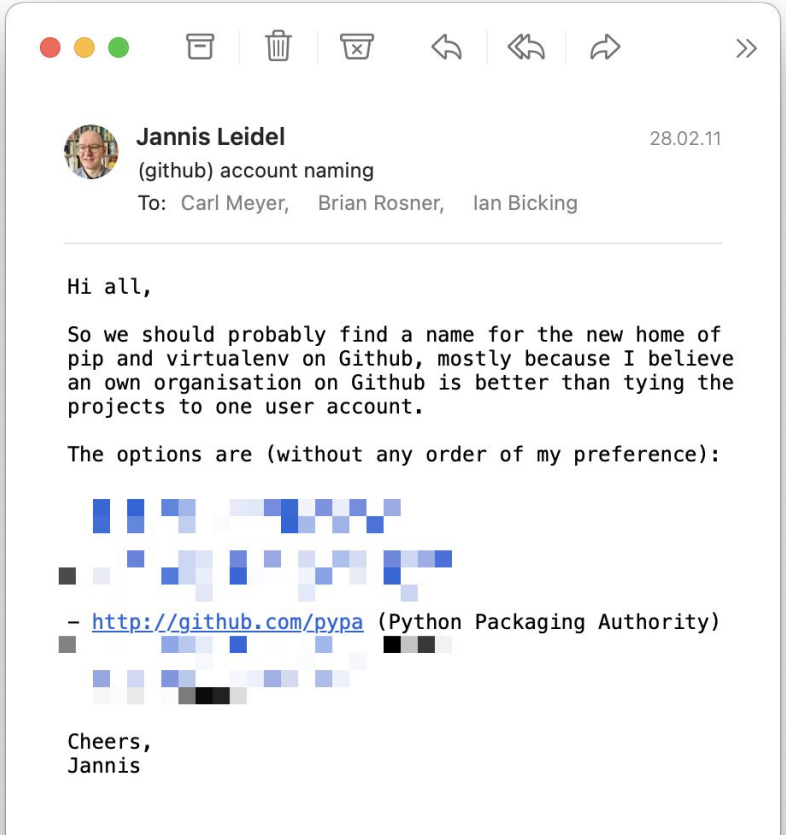
PyPA – Python Packaging Authority

Begann 2011 als Freiwilligengruppe, um die Wartung von pip und virtualenv zu übernehmen. Seit dem viele Kollaborationen mit CPython Team.

Folgt der PEP 609 als Governance-Richtlinie, ist hauptsächlich eine lose gekoppelte Gruppe von etwa 50 Projekten und etwa 100 Hauptentwicklern mit Schwerpunkt auf Dokumentation, Standards und Tools...

Baut oder vertreibt Python selbst nicht.

Wartet wichtige Pakete wie build, flit, hatch, manylinux, pip, pipenv, pipx, setuptools, virtualenv, wheel usw.



Jannis Leidel 28.02.11

(github) account naming

To: Carl Meyer, Brian Rosner, Ian Bicking

Hi all,

So we should probably find a name for the new home of pip and virtualenv on Github, mostly because I believe an own organisation on Github is better than tying the projects to one user account.

The options are (without any order of my preference):

- http://github.com/pypa (Python Packaging Authority)

Cheers,
Jannis

Was ist die Hauptaufgabe hinsichtlich der Paketierung von Python-Projekten?

Projekte weiterentwickeln und eine Community aufbauen, unter Verwendung von Spezifikationen und Standards.

Baut und kompiliert PyPA Python selbst?

Nein, die von PyPA gewarteten Projekte arbeiten mit verschiedenen Python-Ablegern zusammen.

Erstellt, verteilt oder installiert PyPA Python-Pakete?

Ja, viele Kernpakete, z. B. **twine** zum Hochladen von Paketen auf PyPI, mehrere (!) Installationsprogramme, High-Level-Tools wie **pip**.

Was versucht das Projekt allgemein zu erreichen?

Unterstützung für PyPA-Projekte, Standards fördern und mit anderen Communitys interagieren, wie der PSF Packaging Arbeitsgruppe, dem Python Steering Council, Distributoren, anderen Programmiersprachen usw.

Was versucht PyPA hinsichtlich der Paketierung und Distribution zu erreichen?

Förderung einer Community bestehend aus Projekten, die Community-Standards folgen. Mittlerweile viele, teilweise überlappende, Projekte.

Was sind einige der wichtigsten Schwachstellen des Projekts?

Wenig kohärente Strategie zur Weiterentwicklung außerhalb von Standards. Wachstumsschwierigkeiten, personell und technisch.

CONDA

Ein Open-Source-Paket- und Umgebungsmanager, der 2011/2012 für die sogenannte **Anaconda Distribution** erstellt wurde, um alle Abhängigkeiten zu verwalten, einschließlich Nicht-Python-Software.

Eine Community von etwa 150 Projekten, mehreren Unternehmen, die ein sprach-unabhängiges, plattformübergreifendes Paketmanagement-Ökosystem für Projekte jeder Größe und Komplexität unterstützen

Größter Kanal conda-forge: ~20k Pakete, ~5k Personen.

Conda-Community

Was ist die Hauptaufgabe der Conda-Community?

Der Hauptzweck besteht darin, einen Open-Source-, Multi-Plattform-, Sprach-agnostischen Paket- und Umgebungsmanager zur Verfügung zu stellen. conda ist in Python geschrieben, aber nicht darauf beschränkt.

Baut und kompiliert conda Python selbst?

Ja, verfügbar über verschiedene Kanäle, z.B. die kommerzielle **Anaconda Distribution** oder die **conda-forge** Community. Aber auch weitere.

Erstellt, verteilt oder installiert Python-Pakete?

Ja, conda-build cann wheel-Dateien herstellen und via pip und poetry Python-Pakete installieren. Viele Python-basiertePakete sind in verschiedenen Kanälen verfügbar.

Was versucht das Projekt allgemein zu erreichen?

Conda folgt Standards wie z.B. PEPs, wenn es notwendig und sinnvoll ist und wird zunehmend zu einer offenen Community. Es berücksichtigt den gesamten Abhängigkeitsbaum und verstehe auch Nicht-Python-Abhängigkeiten.

Was versucht PyPA hinsichtlich der Paketierung und Distribution zu erreichen?

Folgt den Standards und bewährten Praktiken von PyPA und PEPs, wenn möglich.

Was sind einige der wichtigsten Schwachstellen des Projekts?

Unklare Beziehung zwischen den PyPA- und Conda-Communities, unterschiedliche Schwerpunkte und leicht zu verwechselnde Abläufe.

Conda and PyPA (Auszug)

2011	PyPA start, Übergabe von pip/virtualenv	conda veröffentlicht
2012	Distutils2-Versuch schlägt fehl	Plattformübergreifende Distribution mit ~100 Paketen
2013	Wheel-Dateien, distribute/setuptools Merge	SAT Solver, binstar.org
2014	ensurepip in der Stdlib, PEP 440 Versionen,, packaging.python.org, get-pip.py	anaconda.org: zentrale Plattform mit Kanälen, conda-build
2015	PyPI-Hosting, neue “simple” API	conda-forge community gestartet, mit zugenannten “feedstocks”
2016	pypi.org, manylinux	noarch/universal-Pakete
2017	Mozilla Subvention für PyPI, pyproject.toml	Shell-Aktivierung erneuert

Conda and PyPA (Auszug)

2018	PyPI-Neustart	
2019	Open Technology Fund Projektgelder für PyPI	Virtuelle Pakete, neues .conda format, conda run, conda-press, mamba (C++ Variante von conda)
2020	PyPA Governance-Satzung	Conda Governance-Satzung, conda-forge Wachstum
2021	Finanzielle Unterstützung für PyPA via Python Software Foundation	Neuentwurf der Satzung, Metadaten werden signiert
2022	2-Faktor-Authorisierung für PyPI	Finanzielle Unterstützung für PyPA über NumFOCUS Stiftung, Plugins, neuer Solver auf Basis von mamba
2023	Python Packaging Strategie, Paket-Council?, PyPI organizations	Standards, conda.org, rattler (conda in Rust)

Interessante aktuelle PEPs und CEPs

- PEP 715 – Keine .egg-Dateien auf PyPI
- PEP 711 – PyBI Standard zur einfacheren Distributierung von Python-Interpretern
- PEP 710 – Prüfsummen von Paketen nach der Installation aufheben
- PEP 708 – Nachvollziehbarkeit von Paketen auf mehreren Indizes
- PEP 691 – JSON-API für PyPI
- PEP 668 – Schutz vor Veränderungen von Python-Installationen
- PEP 658 – Paketmetadaten direkt auf PyPI gehostet
- PEP 639 – Bessere Lizenzangaben

- CEP XX – Neues Format für die Paketspezifierung
- CEP XX – Optionale Abhängigkeiten
- CEP XX – Inkrementelle Updates für Kanalmetadaten
- CEP XX – Teilbare Conda-Umgebungen
- CEP 10 – Conda-Versionsunterstützung
- CEP 9 – Zeitplan für die Entfernung veralteter Funktionen in Conda
- CEP 8 – Conda Veröffentlichungszeitplan
- CEP 4 – Plugin-System für Conda
- CEP 3 – conda/mamba solver

Q & A


PackagingCon
2023
October 26-28, 2023
Speak at
PackagingCon
in Berlin
Submit Your Proposal

PackagingCon 2023

- 26.-28. Oktober 2023
- EUREF-Campus Berlin, nahe Südkreuz
- Fachvorträge zum Thema Paketierung und Distribution