



Sua aplicação não está segura

Live de Python # 163



picpay.me/dunossauro



apoia.se/livedepython



PIX



Ajude o projeto



Ademar Peixoto, Alex Lima, Alexandre Harano, Alexandre Santos, Alexandre Tsuno, Alysson Oliveira, Amaziles Carvalho, Andre Rodrigues, André Rocha, Arnaldo Turque, Bruno Oliveira, Caio Nascimento, César Almeida, César Moreira, Davi Ramos, David Kwast, Diego Guimarães, Dilenon Delfino, Douglas Bastos, Edgard Sampaio, Elias Soares, Érico Andrei, Eugenio Mazzini, Everton Alves, Fabio Barros, Fabio Castro, Fabrício Coelho, Flavkaze, Franklin Silva, Fábio Serrão, Gabriel Simonetto, Gabriel Soares, Gabriela Santiago, Geandreson Costa, Guilherme Felitti, Guilherme Marson, Guilherme Ostrock, Gustavo Chacon, Henrique Machado, Italo Silva, Johnny Tardin, Jonatas Leon, Jonatas Oliveira, Jorge Plautz, José Prado, João Lugão, João Schiavon, Juan Gutierrez, Jônatas Silva, Júlia Kastrup, Kaneson Alves, Leonardo Cruz, Leonardo Galani, Leonardo Mello, Lidianne Monteiro, Lorena Ribeiro, Lucas Barros, Lucas Ferreira, Lucas Mello, Lucas Mendes, Lucas Teixeira, Lucas Valino, Luiz Lima, Maiquel Leonel, Maiquel Leonel, Marcela Campos, Marcelo Rodrigues, Maria Clara, Melissa Mendonça, Moisés Andrade, Natan Cervinski, Nicolas Teodosio, Patric Lacouth, Patricia Minamizawa, Patrick Gomes, Paulo Tadei, Pedro Andrade, Pedro Pereira, Peterson Santos, Rafael Lino, Reinaldo Silva, Rodrigo Ferreira, Rodrigo Vaccari, Ronaldo Silva, Rubens Gianfaldoni, Sandro Mio, Silvio Xm, Thiago Araujo, Thiago Borges, Thiago Bueno, Tyrone Damasceno, Valdir Junior, Victor Geraldo, Vinícius Bastos, Vinícius Ferreira, Vítor Gomes, Wendel Rios, Wesley Mendes, Willian Lopes, Willian Lopes, Willian Rosa, Wilson Duarte



Obrigado você





1. Versionamento de libs

O que é? onde se esconde?

2. Checagem constante

Garantindo nossa segurança

3. Pacotes inseguros

Sim, eles existem

4. Vendoring

Sim, também temos esses casos

Vamos falar só sobre
o pypi



Disclaimer



Python 3.9.4

Release Date: April 4, 2021

This is the fourth maintenance release of Python 3.9

Python 3.9.4 is a hotfix release addressing an unintentional ABI incompatibility introduced in Python 3.9.3. **Upgrading is highly recommended to all users.** Details in [bpo-43710](#).

To reiterate, Python 3.9.3 was itself an expedited release due to its security content:

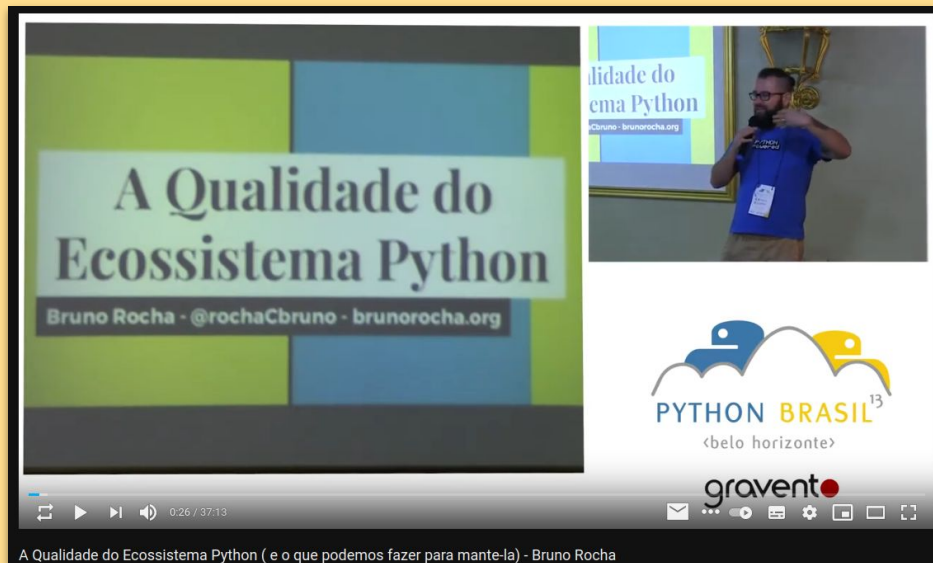
- [bpo-43631](#): high-severity CVE-2021-3449 and CVE-2021-3450 were published for OpenSSL, it's been upgraded to 1.1.1k in CI, and macOS and Windows installers.
- [bpo-42988](#): CVE-2021-3426: Remove the getfile feature of the pydoc module which could be abused to read arbitrary files on the disk (directory traversal vulnerability). Moreover, even source code of Python modules can contain sensitive data like passwords. Vulnerability reported by David Schwörer.
- [bpo-43285](#): ftplib no longer trusts the IP address value returned from the server in response to the PASV command by default. This prevents a malicious FTP server from using

<https://www.python.org/downloads/release/python-394/>



Motivador desta live





<https://youtu.be/niE53CSCAkc>



Continuação espiritual



Não entendeu, pergunte!



Aviso



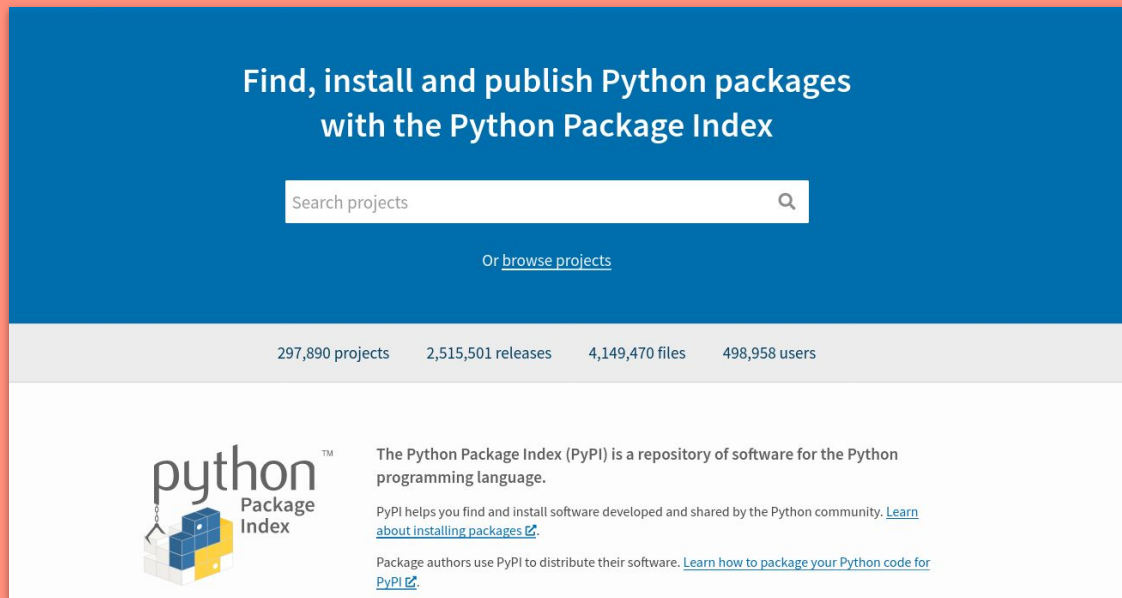
O que é, onde se
esconde?

Version
amento

Versionamento de bibliotecas



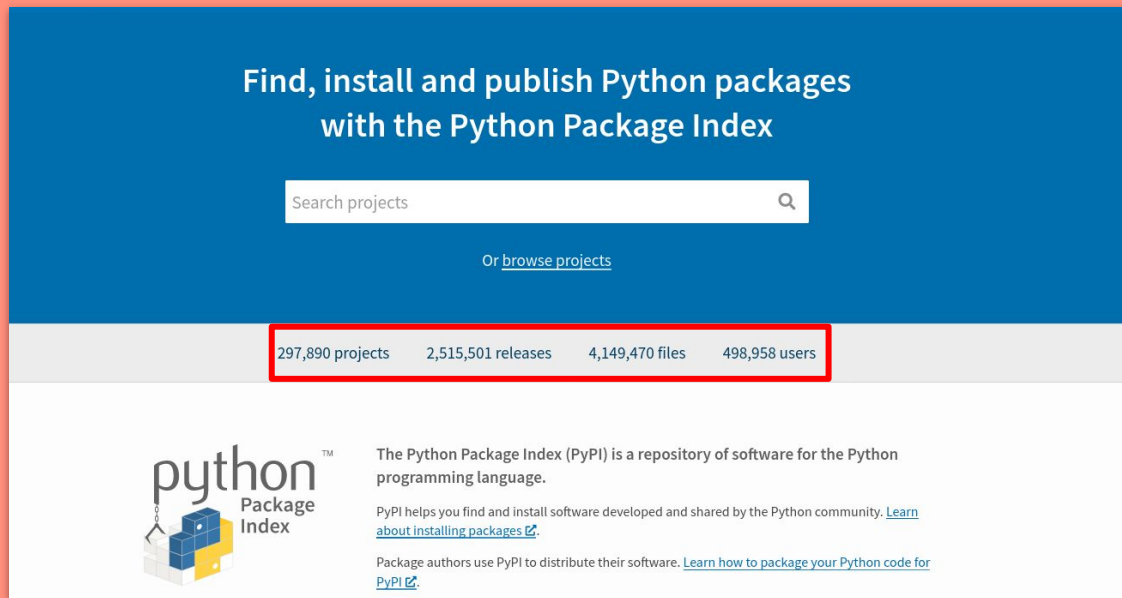
No ecossistema Python, as bibliotecas são "armazenadas" no pypi.org

A screenshot of the Python Package Index (PyPI) homepage. The header is blue with the text "Find, install and publish Python packages with the Python Package Index". Below this is a search bar with the placeholder text "Search projects" and a magnifying glass icon. Under the search bar, it says "Or [browse projects](#)". A statistics bar shows "297,890 projects", "2,515,501 releases", "4,149,470 files", and "498,958 users". The footer features the "python Package Index" logo and a description: "The Python Package Index (PyPI) is a repository of software for the Python programming language." It also includes links: "PyPI helps you find and install software developed and shared by the Python community. [Learn about installing packages](#)." and "Package authors use PyPI to distribute their software. [Learn how to package your Python code for PyPI](#)."

Versionamento de bibliotecas



No ecossistema Python, as bibliotecas são "armazenadas" no pypi.org



The screenshot shows the PyPI homepage with a blue header and a white footer. The header contains the text "Find, install and publish Python packages with the Python Package Index" and a search bar. Below the search bar is a link "Or browse projects". The footer contains the PyPI logo, a description of PyPI, and links to learn more about installing packages and packaging code for PyPI. A red box highlights the statistics section of the page.

297,890 projects	2,515,501 releases	4,149,470 files	498,958 users
------------------	--------------------	-----------------	---------------

python
Package
Index

The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. [Learn about installing packages](#).

Package authors use PyPI to distribute their software. [Learn how to package your Python code for PyPI](#).

Mas quem mantém o pypi?

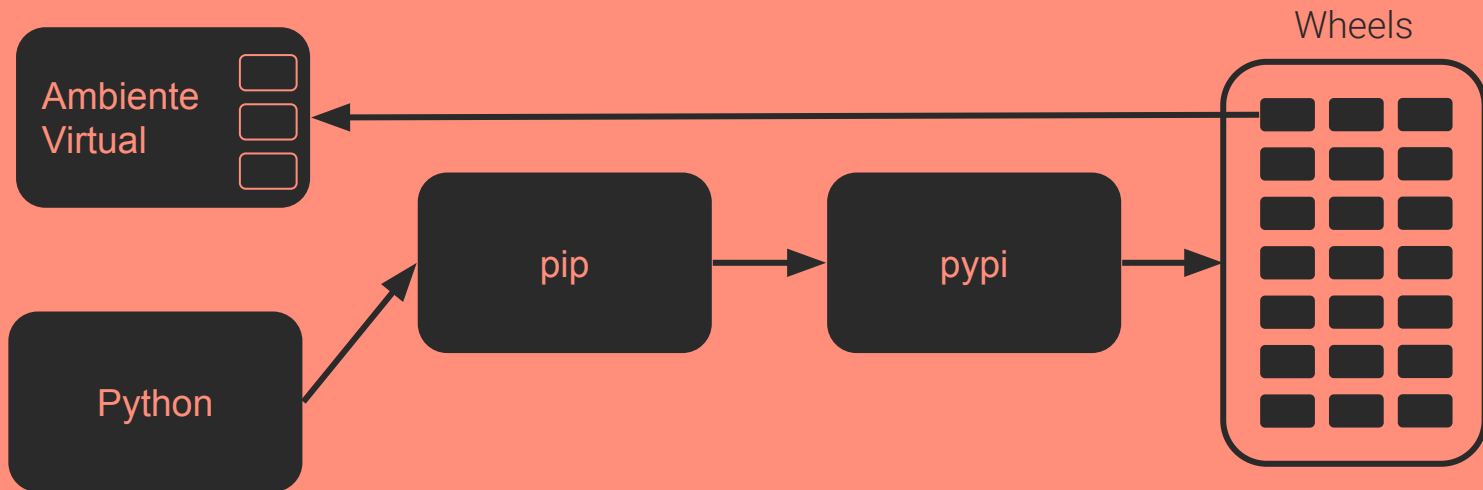


O **PyPI** [Python Package Index] é mantido pela **PyPA** [Python Package Authority].

A PyPA mantém muitos componentes importantes do ecossistema:

- pip
- setuptools
- virtualenv
- distutils
- wheel
- ...

O pip é a ferramenta mais tradicional para instalação de pacotes em python, mas o que acontece com um ``pip install ...``



Versionamento

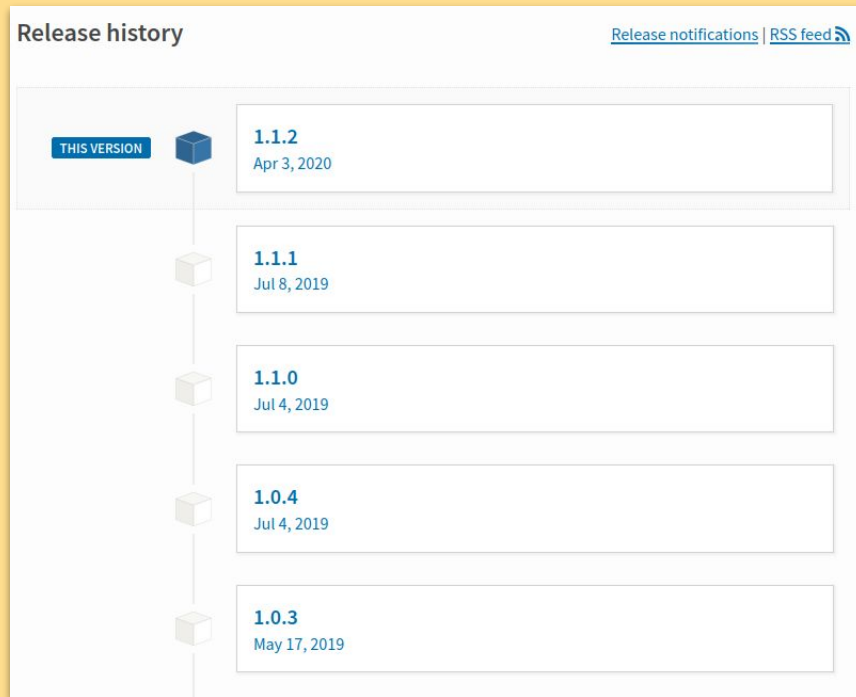


Se você acessar algum pacote no pypi vai se deparar com um histórico de releases.

Você pode instalar qualquer uma dessas versões:

`pip install biblioteca==1.0.4`

Release history [Release notifications](#) | [RSS feed](#)



The diagram shows a vertical timeline of releases for Flask. A blue cube icon labeled 'THIS VERSION' points to the 1.1.2 release. Below it are four white cube icons pointing to releases 1.1.1, 1.1.0, 1.0.4, and 1.0.3. Each release box contains the version number and the date.

Version	Date
1.1.2	Apr 3, 2020
1.1.1	Jul 8, 2019
1.1.0	Jul 4, 2019
1.0.4	Jul 4, 2019
1.0.3	May 17, 2019

<https://pypi.org/project/Flask/#history>

Como funciona o versionamento?



1.0.4

Major

Minor

Patch

Como funciona o versionamento?



1.0.4

Quebra de
compatibilidade

Major

Minor

Patch

Novas
funcionalidades

Correções de
bugs

Como funciona o versionamento?



1.0.4

Major

Minor

Patch

Bugs e atualizações de segurança
acontecem no patch

Checando e
automatizando

Checa
gem

De volta ao pip



Como fazemos normalmente?

```
pip install lib_x lib_y lib_z
```

```
pip freeze > requirements.txt
```

De volta ao pip



Como fazemos normalmente?

```
pip install lib_x lib_y lib_z
```

```
pip freeze > requirements.txt
```

Quando alguém já fez algo

```
git clone
```

```
pip install -r requirements.txt
```

pip list -o

poetry show -o



Mas como saber se algo está desatualizado?



Só pra não ficar só na teoria.



Bora pro código, vai



Quanto código com essa vulnerabilidade?



Flask==0.12.5

Pull requests Issues Marketplace Explore

Repositories	0
Code	48K
Commits	819
Issues	0
Discussions Beta	0
Packages	0
Marketplace	0
Topics	0
Wikis	2
Users	0

28,298 code results

elastic/apm-agent-python
tests/requirements/reqs-flask-0.12.txt

```
1 Flask>=0.12, <0.13
2 -r reqs-base.txt
```

● Text Showing the top two matches Last indexed on 29 Sep 2020

davidrossouw/cloud-run
simpson-api/requirements.txt

```
1 Flask==0.12
2 flask-restful==0.3.5
3 Flask-Cors
4 Flask-HTTPAuth
5 opencv-python
6 tensorflow==1.13.1
```

Como solucionar esse problema?



Existem frentes diferentes e regras diferentes em cada projeto/empresa:

- pre-commit
- Integração contínua
- pip-upgrader
- pyup
- ...

Como solucionar esse problema?



Existem frentes diferentes e regras diferentes em cada projeto/empresa:

- **pre-commit**
- Integração contínua
- pip-upgrader
- pyup
- ...

Como solucionar esse problema?



Existem frentes diferentes e regras diferentes em cada projeto/empresa:

- pre-commit
- Integração contínua
- **pip-upgrader**
- pyup
- ...

```
# pip install pip-upgrader  
pip-upgrade requirements.txt -p all --skip-package-installation
```

Como o pip pode nos ajudar com isso?

OPERADOR	DESCRIÇÃO	EXEMPLO
>	Versões maiores que	pacote>1.0.0
<	Versões menores que	pacote<1.0.0
<=	Versões menores ou iguais a	pacote<=1.0.0
>=	Versões maiores ou iguais a	pacote>=1.0.0
==	Versão igual a	pacote==1.0.0
!=	Versão diferente a	pacote!=1.0.0
~=	Versão compatível a	????
*	Qualquer versão de ...	???

Como o pip pode nos ajudar com isso?

OPERADOR	DESCRIÇÃO	EXEMPLO
~=	Versão compatível a	<ul style="list-style-type: none">● pacote~=1.0.1<ul style="list-style-type: none">○ 1.0.1, 1.0.2, 1.0.3, 1.0.4○ Nunca 1.1● pacote~=1.0<ul style="list-style-type: none">○ 1.0.1, 1.1.0, 1.2.0, 1.2.1○ Nunca 2.0
*	Qualquer versão de ...	<ul style="list-style-type: none">● pacote==3.1.*<ul style="list-style-type: none">○ 3.1.0, 3.1.1, 3.1.2

Combinação de operadores



Os operadores podem ser combinados, por exemplo:

- `pacote >= 3.0, < 4.0`
 - 3.1, 3.0.1, 3.0.10, 3.15.0
- `pacote >= 3.0, <= 4.0`
 - Pode ser o 4, mas nunca maior que 4.0.*

Como saber?

Inseg
uros

Como uma vulnerabilidade é noticiada?



Existe um banco de dados mantido pelo mitre.org onde ficam catalogados os CVE (Vulnerabilidades e exposições comuns)

<https://cve.mitre.org/>
(não esquecer de acessar)

Python 3.9.4

Release Date: April 4, 2021

This is the fourth maintenance release of Python 3.9

Python 3.9.4 is a hotfix release addressing an unintentional ABI incompatibility introduced in Python 3.9.3. **Upgrading is highly recommended to all users.** Details in [bpo-43710](#).

To reiterate, Python 3.9.3 was itself an expedited release due to its security content:

- [bpo-43631](#): high-severity CVE-2021-3449 and CVE-2021-3450 were published for OpenSSL, it's been upgraded to 1.1.1k in CI, and macOS and Windows installers.
- [bpo-42988](#): CVE-2021-3426: Remove the getfile feature of the pydoc module which could be abused to read arbitrary files on the disk (directory traversal vulnerability).
Moreover, even source code of Python modules can contain sensitive data like passwords. Vulnerability reported by David Schwörer.
- [bpo-43285](#): ftplib no longer trusts the IP address value returned from the server in response to the PASV command by default. This prevents a malicious FTP server from using the response to probe IPv4 address and port combinations on the client network. Code that requires the former vulnerable behavior may set a `trust_server_pasv_ipv4_address`



Motivador desta live



Como posso saber se isso é crítico?



Partiremos do princípio que uma vulnerabilidade é **sempre** crítica. Mas quão crítica?

O database do **Mitre** não tem essa informação. Porém o **NIST** (Instituto Nacional de Padrões e Tecnologia) mantém o **NVD** (Banco de dados Nacional de Vulnerabilidades) que contém mais detalhes sobre CVEs

<https://nvd.nist.gov/>

(não esquecer de acessar)

Quem notifica um CVE?



Existe um grupo de CNAs (Autoridade de Numeração de CVEs). CNAs podem ser grupos de pesquisa, empresas, institutos de segurança e etc...

https://cve.mitre.org/cve/request_id.html

(não esquecer de acessar)

Tá, mas o que eu faço com
isso?



Cheque **SEMPRE**



Safety



Nosso ecossistema é **lindo**, a Pyup mantém uma ferramenta chamada safety, que valida se nossos pacotes contêm CVEs.



<https://github.com/pyupio/safety>

pip install safety



Bora instalar



Safety



Fornece uma API de linha comando bem simples que nos permite checar os pacotes que contém vulnerabilidades documentadas

```
# pip install safety  
safety check -r requirements.txt --full-report
```

Automatizando



Existem frentes diferentes e regras diferentes em cada projeto/empresa:

- pre-commit
- Integração contínua
- dependa-bot (github)
- pipenv check

Automatizando



Existem frentes diferentes e regras diferentes em cada projeto/empresa:

- **pre-commit**
- Integração contínua
- dependabot (github)
- pipenv check

Sim, também
temos esse caso

Vendo
ring

Vendoring



Vendoring é uma tática para quando você não pode chamar o "pip" em produção. Você faz uma cópia da biblioteca no seu repositório.

Motivos:

- Em produção não acessa o pypa
- VPNs
- Auditoria de pacotes
- Pacotes diferentes para versões diferentes
- ...

Como fazer vendoring?

[<https://pip.pypa.io/en/latest/development/vendor-policy/#vendor-policy>]



Em python temos uma biblioteca chamada **vendoring** para fazer isso.

Ela faz o download das libs e das licenças da maneira correta.

```
pip install vendoring
```

Configurando o vendoring

Basicamente precisamos de uma estrutura para o vendoring instalar e usar as bibliotecas nos lugares certos.

```
[tool.vendoring]
destination = "pip/_vendor/"
requirements = "pip/_vendor/vendor.txt"
namespace = ""
protected-files = ["vendor.txt"]
patches-dir = "pip/_vendors/patches"
```

```
[tool.vendoring.transformations]
substitute = []
drop = []
```

```
[tool.vendoring.license.directories]
```

```
[tool.vendoring.license.fallback-urls]
```

```
[tool.vendoring.typing-stubs]
```

Configurando o vendoring

Basicamente precisamos de uma estrutura para o vendoring instalar e usar as bibliotecas nos lugares certos.

Aqui vamos montar um requirements por versão do python.

```
[tool.vendoring]
destination = "pip/ vendor/"
requirements = "pip/_vendor/vendor.txt"
namespace = ""
protected-files = ["vendor.txt"]
patches-dir = "pip/_vendors/patches"

[tool.vendoring.transformations]
substitute = []
drop = []

[tool.vendoring.license.directories]

[tool.vendoring.license.fallback-urls]

[tool.vendoring.typing-stubs]
```

vendor.txt



Arquivo onde você vai fazer o freeze das bibliotecas que serão baixadas para vendoring

```
requests==2.25.1; python_version >= "3.4"
```

Agora é só fazer o sync



```
$ vendoring sync  
Load configuration... Done!  
Clean existing libraries... Done!  
Add vendored libraries... Done!  
Fetch licenses... Done!  
Generate static-typing stubs... Done!
```

Agora é só fazer o sync



```
$ vendoring sync
```

```
Load configuration... Done!
```

```
Clean existing libraries
```

```
Add vendored libraries.
```

```
Fetch licenses... Done!
```

```
Generate static-typing
```

```
$ tree pip/_vendor
```

```
.
```

```
|___vendor.txt
```

```
|___requests.pyi
```

```
|___requests/
```


Estudo de caso



Bora dar uma olhada no poetry-core

<https://github.com/python-poetry/poetry-core>



picpay.me/dunossauro



apoia.se/livedepython



PIX



Ajude o projeto

