

# パッケージを配ろう

Atsushi Odagiri

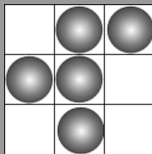
2024-04-25

# Outline

- 1 パッケージを配ろう
- 2 パッケージを配るということ
- 3 パッケージを配るための PEP
- 4 実践 package repository
- 5 参考文献

# お前誰よ

- Atsushi Odagiri
- Open Collector
- Python は 1.5 くらいのころから



## ケンオール

# パッケージエコシステム

- 作る
- 配る
- 使う

# パッケージを配るということ

- 広く一般に向けて配る
- 狭い範囲で限られた利用のために配る

# 広く一般に向けて pypi で配る

- PyPA ツールのデフォルト
- `tween` でアップロード
- `pip` がダウンロードしてインストール

# 狭い範囲で限られた利用のために配る

- マイクロサービスのそれぞれで使うようなライブラリ
- 特殊なパッチをあてたローカルフレーバーライブラリ

# 狭い範囲で配る

- 社内ネットワークや VPN の中で
- k8s や vpc の中で
- 範囲内の IP アドレスにだけ
- 認証をつけたい



# パッケージを配るための PEP

- PEP 458 – Secure PyPI downloads with signed repository metadata
- PEP 480 – Surviving a Compromise of PyPI: End-to-end signing of packages
- PEP 503 – Simple Repository API
- PEP 592 – Adding “Yank” Support to the Simple API
- PEP 629 – Versioning PyPI’s Simple API
- PEP 658 – Serve Distribution Metadata in the Simple Repository API
- PEP 691 – JSON-based Simple API for Python Package Indexes
- PEP 700 – Additional Fields for the Simple API for Package Indexes
- PEP 714 – Rename dist-info-metadata in the Simple API

# Simple Repository

# Metadata

- wheel の中にある Metadata

# JSON

- やることが多い!

# PyPI の Simple Repository

- <https://pypi.org/simple/> とても大きいのでアクセス注意！

pip で simple repository を使う

# httplib.server でのお手軽 repository

- ダウンロードできるリンクがあればいいので http モジュールでサーバーを起動するだけ
- wheel ファイルのあるディレクトリで実行

```
python3 -m pip download pyramid
```

```
python3 -m http.server
```

```
pip install pyramid -f http://localhost:8000/ --no-index
```

## Directory listing for /

- [hupper-1.12.1-py3-none-any.whl](#)
- [PasteDeploy-3.1.0-py3-none-any.whl](#)
- [plaster-1.1.2-py2.py3-none-any.whl](#)
- [plaster\\_pastedeploy-1.0.1-py2.py3-none-any.whl](#)
- [pyramid-2.0.2-py3-none-any.whl](#)
- [setuptools-69.5.1-py3-none-any.whl](#)
- [translationstring-1.4-py2.py3-none-any.whl](#)
- [venusian-3.1.0-py3-none-any.whl](#)
- [WebOb-1.8.7-py2.py3-none-any.whl](#)
- [zope.deprecation-5.0-py3-none-any.whl](#)
- [zope.interface-6.3-cp311-cp311-manylinux\\_2\\_5\\_x86\\_64.manylinux1\\_x86\\_64.manylinux\\_2\\_17\\_x86\\_64.manylinux2014\\_x86\\_64.whl](#)

# wsgi app

- web アプリケーションにする
- DB などを使わず起動するだけで使える



# wheel ファイルを探し出す

- wheel ファイルのファイル名は形式が決まっている  
`{distribution}-{version}(-{build tag})?-{python tag}-{abi tag}-{platform tag}.whl`.

# pypi version

- 今回は v1.0 の範囲でやってみます
  - PEP691 v1.0
  - PEP714 v1.1

# metadata

- METADATA を wheel から取り出す

```
def get_metadata(whl: pathlib.Path):  
    parts = whl.name.split("-")  
    dist_name, version = parts[0], parts[1]  
    metadata_path = f"{dist_name}-{version}.dist-info/METADATA"  
    with zipfile.ZipFile(whl) as zf:  
        with zf.open(metadata_path) as metadata:  
            return metadata.read()
```

# json に対応

- project list
- project detail

# project list

- v1.0 は name のみ

```
Project = TypedDict("Project", {"name": str})
ProjectList = TypedDict(
    "ProjectList",
    {
        "meta": Meta,
        "projects": list[Project],
    },
)
```

# project detail

```
ProjectDetail = TypedDict(  
    "ProjectDetail",  
    {  
        "name": str,  
        "files": list[ProjectFile],  
        "meta": Meta,  
    },  
)
```

# project file

- すごく情報量が増えた

```
ProjectFile = TypedDict(  
    "ProjectFile",  
    {  
        "filename": str,  
        "url": str,  
        "hashes": dict[str, str],  
        "requires-python": NotRequired[str],  
        "dist-info-metadata": NotRequired[str],  
        "gpg-sig": NotRequired[bool],  
        "yanked": NotRequired[bool],  
    },  
)
```

# pip から使う

- project list 呼ばれてないかも？



# find-links vs index-url

```
$ pip install pyramid --index-url=http://localhost:8000/
```

# The Update Framework

- TUF

# 参考文献

- PyPA Simple Repository API, <https://packaging.python.org/en/latest/specifications/simple-repository-api/>
- The Update Framework, <https://theupdateframework.io/>