Beyond Pip: Pipenv and Poetry

Reindert-Jan Ekker

@rjekker | https://nl.linkedin.com/in/rjekker

Overview

Pip and venv are standard tools

Alternatives

- pipenv
- poetry

Why?

- Using one tool instead of two
- Deterministic builds
- Security (hashes)
- More features

```
# dependencies for my project
# (specified through requirements.txt or pyproject.toml)



package_a==1.0.0
package_b==2.3.4
```

Dependency Management 1

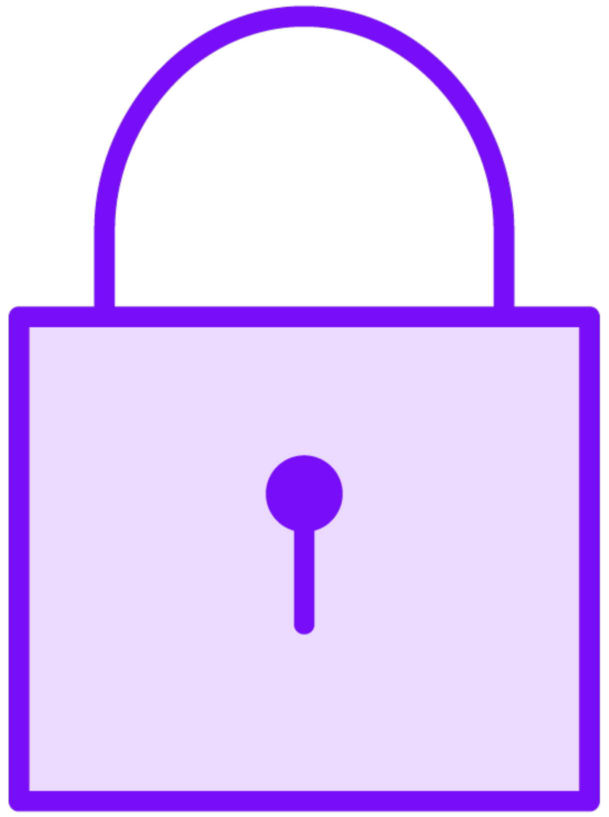
Non-deterministic: versions of sub-dependencies may vary

```
# dependencies for my project

package_a==1.0.0

sub_dependency_a1==2.1.3

sub_dependency_a2=4.5.6

package_b==2.3.4

sub_dependeny_b1==6.7.8

# etc..
```

Dependency Management 2

Greater chance of conflicts
Over-specified: now I need to manage all these versions

Solution: Locking



Top-level dependency declarations

- We actually care about these
- Don’t have to be pinned

Installed, tested versions

- Lock file
- Managed by tool
- Pinned versions
- Hashes

Deterministic builds

pipenv

Focused on applications

All versions are pinned

Pipfile for top-level dependencies

Pipfile.lock locks versions, hashes

poetry

Libraries and applications

Custom dependency resolver

Building and distributing packages

Pyproject.toml for top-level dep.

poetry.lock locks versions, hashes

Demo

Pipenv introduction

Demo

Pipenv workflow

Demo

Pipenv: keeping packages up to date

Pipenv Commands: Install, Uninstall

```
# Install packages
# This updates Pipfile, installs the package in the venv and
# Updates the lockfile
pipenv install requests
pipenv install requests~=2.28   # Supports version specifiers
pipenv install -d black         # Development dependency

# Install all dependencies from the Pipfile
pipenv install

# Remove package from venv, Pipfile and lockfile
# Does not uninstall dependencies!
pipenv uninstall
```

Pipenv Commands: Updating

```
# Install packages from lockfile in your environment
pipenv sync
pipenv sync -d   # Install development dependencies


# Update: update packages and dependencies to latest version
pipenv update
pipenv update requests

# Check which packages have security updates
pipenv check
```

Pipenv Commands: Getting Info

```
# Show installed packages
pipenv graph


# Get help
pipenv -h
pipenv install -h
pipenv update -h
```

Pipenv Commands: Running

```
# Run commands inside venv
pipenv run python my_script.py
pipenv run black my_pkg


# Start a shell with an active venv
# Make sure to end with ”exit” instead of “deactivate”
pipenv shell
```

Demo

Poetry introduction

Demo

Poetry workflow

Demo

Running commands with poetry

Demo

Poetry: groups and extras

Poetry Commands: Create Project, Add, Remove

```
# Create a new project skeleton
poetry new my_project

# Initialize a myproject.toml in an existing project
poetry init

# Add a dependency. Updates pyproject.toml and lockfile, installs package
poetry add requests
poetry add requests@1.2.3
poetry add requests@^2.1

# Remove a dependency. Updates pyproject.toml and lockfile
# Uninstalls package and its dependencies
poetry remove requests
```

Poetry Commands: Install, Update

```
# Install dependencies from pyproject.toml and update the lockfile
# This will not remove packages
poetry install

# Make sure environment matches lockfile
# Remove packages if necessary
poetry install --sync

# Update packages, within the version specifiers from pyproject.toml
poetry update
poetry update requests
```

Poetry Commands: Getting Info

```
# Show installed packages
poetry show
poetry show -tree
poetry show requests
```

```
# Get help
poetry help
poetry help install
poetry help update
```

Poetry Commands: Running

```
# Run commands inside venv
poetry run python my_script.py
poetry run black my_pkg


# Start a shell with an active venv
# Make sure to end with ”exit” instead of “deactivate”
poetry shell
```

Resources and Final Thoughts

https://pipenv.pypa.io

https://python-poetry.org

https://github.com/jazzband/pip-tools/

https://pdm.fming.dev

Summary

Alternatives to pip/venv

- Using one tool instead of two
- Deterministic builds
- Security (hashes)

pipenv

- Aimed at applications
- Pipfile, Pipfile.lock

poetry

- Libraries or applications
- pyproject.toml, poetry.lock