

Evolution of dependency management

Nikita Grishko

> whoami

- Software Engineer at Flo Health Inc.
- Love to code
- And a very lazy person



Dependency management
can become tedious...



Community!

- `pyproject.toml`: the future of Python packaging
(<https://testandcode.com/52>)
- Pipenv: The Future of Python Dependency Management
(<https://www.youtube.com/watch?v=GBQAKldqgZs>)
- Python Application Dependency Management in 2018
(<https://hynek.me/articles/python-app-deps-2018/>)
- cargo (Rust), composer (PHP), yarn (JS), vgo (Go), ...
- pip-tools, pipenv, flit,  poetry

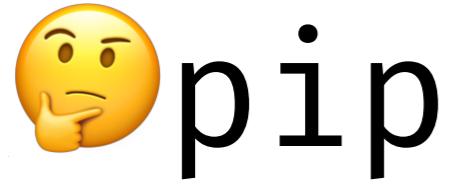
Why not pip?

☺ pip

- 4 April 2011
- pip is a recursive acronym for "Pip Installs Packages"
- Python 2.7.9 and later, and Python 3.4 and later include pip by default

pip

- pip install requests ≥ 1.0
- pip install -U -r requirements.txt
- pip uninstall requests
- pip freeze
- pip list --outdated



- pip install -U -r requirements.txt



pip



2. ng@ip-10-88-1-56: ~/virtualenvs/tmp-d14b838adc96c6b (zsh)

```
$ bat requirements.txt
```

| | |
|---|-------------------------------|
| | File: requirements.txt |
| 1 | Django |
| 2 | requests |

```
$ █
```



pip

● ○ ●

2. ng@ip-10-88-1-56: ~/virtualenvs/tmp-d14b838adc96c6b (zsh)

```
$ bat requirements.txt
```

| | |
|--|-------------------------------|
| | File: requirements.txt |
|--|-------------------------------|

| | |
|---|-----------------|
| 1 | Django=2.1.5 |
| 2 | requests=2.21.0 |

```
$ █
```

😢 pip

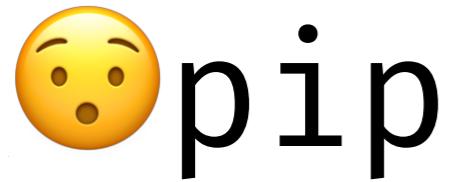
```
2. ng@ip-10-88-1-56: ~/virtualenvs/tmp-d14b838adc96c6b (zsh)
```

```
$ pip freeze
certifi==2018.11.29
chardet==3.0.4
Django==2.1.5
idna==2.8
pytz==2018.9
requests==2.21.0
urllib3==1.24.1
```

```
$ |
```

😎 pip

- pip install -U -r requirements.in
- pip freeze > requirements.txt
- pip install -U -r requirements.txt



- requirements.in + requirements.txt
- requirements-dev.in + requirements-dev.txt



```
2. ng@ip-10-88-1-56: ~/virtualenvs/tmp-57a09ffc04879f4 (zsh)
$ bat requirements.txt

File: requirements.txt
1 Django=1.4.0
2 django-debug-informer=0.4.0

$ █
```



2. ng@ip-10-88-1-56: ~/virtualenvs/tmp-57a09ffc04879f4 (zsh)

```
$ pip install -U -r requirements.txt
Collecting Django==1.4.0 (from -r requirements.txt (line 1))
Collecting django-debug-informer==0.4.0 (from -r requirements.txt (line 2))
  Using cached https://files.pythonhosted.org/packages/9b/3e/7f46e7681e418ab3fb497b1d0149ff7cae74269f12f8f17860d5b479564a/django_debug_informer-0.4.0-py2.py3-none-any.whl
Requirement already satisfied, skipping upgrade: pip≥1.10 in ./lib/python3.7/site-packages (from django-debug-informer==0.4.0->r requirements.txt (line 2)) (19.0.2)
django-debug-informer 0.4.0 has requirement django<3.0, ≥2.0, but you'll have django 1.4 which is incompatible.
Installing collected packages: Django, django-debug-informer
Successfully installed Django-1.4 django-debug-informer-0.4.0
```

```
$ echo $?
```

```
0
```

```
$ █
```

pip

```
2. ng@ip-10-88-1-56: ~/virtualenvs/tmp-57a09ffc04879f4 (zsh)  
$ pip check  
django-debug-informer 0.4.0 has requirement django<3.0, ≥2.0, but you  
have django 1.4.
```

```
$ echo $?  
1
```

```
$
```



pip

● ○ ●

2. ng@ip-10-88-1-56: ~/virtualenvs/tmp-cb9ba8d088fcc3f (zsh)

```
$ pip freeze
certifi=2018.11.29
chardet=3.0.4
idna=2.8
requests=2.21.0
urllib3=1.24.1
```

```
$ █
```



pip

● ○ ●

2. ng@ip-10-88-1-56: ~/virtualenvs/tmp-cb9ba8d088fcc3f (zsh)

```
$ pip show requests
```

Name: requests

Version: 2.21.0

Summary: Python HTTP for Humans.

Home-page: http://python-requests.org

Author: Kenneth Reitz

Author-email: me@kennethreitz.org

License: Apache 2.0

Location: /Users/ng/.virtualenvs/tmp-cb9ba8d088fcc3f/lib/python3.7/site-packages

Requires: idna, chardet, urllib3, certifi

Required-by:

```
$ █
```



pip

● ○ ●

2. noglob pip uninstall requests (python3.7)

```
$ pip uninstall requests
```

```
Uninstalling requests-2.21.0:
```

```
  Would remove:
```

```
    /Users/ng/.virtualenvs/tmp-cb9ba8d088fcc3f/lib/python3.7/site-packages/requests-2.21.0.dist-info/*
```

```
    /Users/ng/.virtualenvs/tmp-cb9ba8d088fcc3f/lib/python3.7/site-packages/requests/*
```

```
Proceed (y/n)? █
```



2. ng@ip-10-88-1-56: ~/virtualenvs/tmp-cb9ba8d088fcc3f (zsh)

```
$ pip uninstall requests
Uninstalling requests-2.21.0:
  Would remove:
    /Users/ng/.virtualenvs/tmp-cb9ba8d088fcc3f/lib/python3.7/site-packages/requests-2.21.0.dist-info/*
    /Users/ng/.virtualenvs/tmp-cb9ba8d088fcc3f/lib/python3.7/site-packages/requests/*
Proceed (y/n)? y
Successfully uninstalled requests-2.21.0
```

\$ █



pip

- pip show
- pip show --outdated

But why?

Dependency resolver

Dependency management
can become tedious...

Tooling

- pyenv
- pip
- virtualenv + virtualenv-wrapper
- zest.releaser
- twine
- ...

 Why not pip?

- Dependency resolver
- Tooling

We have options!



pip-tools

- <https://github.com/jazzband/pip-tools>
- 10 September 2012
- 3k+ stars on GitHub
- A set of tools to keep your pinned Python dependencies fresh
- pip-tools = pip-compile + pip-sync



pip-tools

```
2. ng@ip-10-88-1-56: ~/virtualenvs/tmp-52750e84d444d5c (zsh)
```

```
$ bat requirements.in
```

```
File: requirements.in
```

```
1 Django=1.4.0
2 django-debug-informer=0.4.0
```

```
$ pip-compile --output-file requirements.txt requirements.in █
```

☺️pip-tools

```
● ● ● 2. ng@ip-10-88-1-56: ~/virtualenvs/tmp-52750e84d444d5c (zsh)
.13, 1.10, 1.10, 1.10.1, 1.10.1, 1.10.2, 1.10.2, 1.10.3, 1.10.3, 1.10
.4, 1.10.4, 1.10.5, 1.10.5, 1.10.6, 1.10.6, 1.10.7, 1.10.7, 1.10.8, 1
.10.8, 1.11, 1.11, 1.11.1, 1.11.1, 1.11.2, 1.11.2, 1.11.3, 1.11.3, 1.
11.4, 1.11.4, 1.11.5, 1.11.5, 1.11.6, 1.11.6, 1.11.7, 1.11.7, 1.11.8,
1.11.8, 1.11.9, 1.11.9, 1.11.10, 1.11.10, 1.11.11, 1.11.11, 1.11.12,
1.11.12, 1.11.13, 1.11.13, 1.11.14, 1.11.14, 1.11.15, 1.11.15, 1.11.
16, 1.11.16, 1.11.17, 1.11.17, 1.11.18, 1.11.18, 1.11.20, 1.11.20, 2.
0, 2.0, 2.0.1, 2.0.1, 2.0.2, 2.0.2, 2.0.3, 2.0.3, 2.0.4, 2.0.4, 2.0.5
, 2.0.5, 2.0.6, 2.0.6, 2.0.7, 2.0.7, 2.0.8, 2.0.8, 2.0.9, 2.0.9, 2.0.
10, 2.0.10, 2.0.12, 2.0.12, 2.0.13, 2.0.13, 2.1, 2.1, 2.1.1, 2.1.1, 2
.1.2, 2.1.2, 2.1.3, 2.1.3, 2.1.4, 2.1.4, 2.1.5, 2.1.5, 2.1.7, 2.1.7
Skipped pre-versions: 1.8a1, 1.8b1, 1.8b2, 1.8rc1, 1.9a1, 1.9b1, 1.9r
c1, 1.9rc2, 1.10a1, 1.10a1, 1.10b1, 1.10b1, 1.10rc1, 1.10rc1, 1.11a1,
1.11b1, 1.11rc1, 1.11rc1, 2.0a1, 2.0b1, 2.0rc1, 2.1a1, 2.1b1, 2.1rc1
, 2.2a1, 2.2a1, 2.2b1, 2.2b1
There are incompatible versions in the resolved dependencies.

$ echo $?
2

$ |
```

☺ pip-tools

```
● ○ ● 2. ng@ip-10-88-1-56: ~/virtualenvs/tmp-52750e84d444d5c (zsh)
```

```
$ bat requirements.in
```

| | |
|---|-------------------------------|
| | File: requirements.in |
| 1 | Django ≥ 1.4.0, <3.0.0 |
| 2 | django-debug-informer ≥ 0.4.0 |

```
$ pip-compile --output-file requirements.txt requirements.in
```

pip-tools

```
2. ng@ip-10-88-1-56: ~/virtualenvs/tmp-52750e84d444d5c (zsh)
$ pip-compile --output-file requirements.txt requirements.in
#
# This file is autogenerated by pip-compile
# To update, run:
#
#     pip-compile --output-file requirements.txt requirements.in
#
django-debug-informer=0.4.0
django=2.1.7
pytz=2018.9          # via django

$ pip-sync
```



pip-tools

```
2. ng@ip-10-88-1-56: ~/virtualenvs/tmp-52add59a3697e7f (zsh)
$ bat requirements.in

+-----+
| File: requirements.in |
+-----+
| 1 | oslo.utils=1.4.0 |
+-----+
$ |
```

😭 pip-tools

```
2. ng@ip-10-88-1-56: ~/virtualenvs/tmp-52add59a3697e7f (zsh)
$ pip-compile --output-file requirements.txt requirements.in
Could not find a version that matches pbr!=0.7, $\neq$ 2.1.0,<1.0, $\geq$ 0.6, $\geq$ 2
.0.0
Tried: 0.5.2.5.g5b3e942, 0.5.0, 0.5.1, 0.5.2, 0.5.4, 0.5.5, 0.5.6, 0.
5.7, 0.5.8, 0.5.10, 0.5.11, 0.5.12, 0.5.13, 0.5.14, 0.5.15, 0.5.16, 0
.5.17, 0.5.18, 0.5.19, 0.5.20, 0.5.21, 0.5.22, 0.5.23, 0.6, 0.7.0, 0.
8.0, 0.8.1, 0.8.2, 0.9.0, 0.9.0, 0.10.0, 0.10.0, 0.10.1, 0.10.1, 0.10
.2, 0.10.2, 0.10.3, 0.10.3, 0.10.4, 0.10.4, 0.10.5, 0.10.5, 0.10.6, 0
.10.6, 0.10.7, 0.10.7, 0.10.8, 0.10.8, 0.11.0, 0.11.0, 0.11.1, 0.11.1
, 1.0.0, 1.0.0, 1.0.1, 1.0.1, 1.1.0, 1.1.0, 1.1.1, 1.1.1, 1.2.0, 1.2.
0, 1.3.0, 1.3.0, 1.4.0, 1.4.0, 1.5.0, 1.5.0, 1.6.0, 1.6.0, 1.7.0, 1.7
.0, 1.8.0, 1.8.0, 1.8.1, 1.8.1, 1.9.0, 1.9.0, 1.9.1, 1.9.1, 1.10.0, 1
.10.0, 2.0.0, 2.0.0, 2.1.0, 2.1.0, 3.0.0, 3.0.0, 3.0.1, 3.0.1, 3.1.0,
3.1.0, 3.1.1, 3.1.1, 4.0.0, 4.0.0, 4.0.1, 4.0.1, 4.0.1, 4.0.2, 4.0.2, 4.0.3
, 4.0.3, 4.0.4, 4.0.4, 4.1.0, 4.1.0, 4.1.1, 4.1.1, 4.1.1, 4.2.0, 4.2.0, 4.3.
0, 4.3.0, 5.0.0, 5.0.0, 5.1.0, 5.1.0, 5.1.1, 5.1.1, 5.1.1, 5.1.2, 5.1.2
There are incompatible versions in the resolved dependencies.

$ █
```

😢 pip-tools

- oslo.utils (1.4.0)
 - pbr ($\geq 0.6, \neq 0.7, < 1.0$) \leftarrow pbr=0.11.1
 - oslo.i18n ($\geq 1.3.0$) \leftarrow oslo.i18n=3.20.0

😢 pip-tools

- oslo.utils (1.4.0)
 - pbr ($\geq 0.6, \neq 0.7, < 1.0$) \leftarrow pbr=0.11.1
- oslo.i18n ($\geq 1.3.0$) \leftarrow oslo.i18n=3.20.0
 - pbr ($\neq 2.1.0, \geq 2.0.0$) \leftarrow incompatible with pbr=0.11.1

Why not pip-tools?

- pip-tools has a very simple dependency resolver
- pip-tools is slow, especially in big projects where you have a lot of deps
- pip-tools can't verify that your lock file is outdated
- pip-tools can't be used with various versions of Python in one project because of conditional deps (pkg; python_version<'3.3'')
- pip-tools is only about dependency resolution



pipenv

- <https://github.com/pypa/pipenv>
- 19 January 2017
- 15k+ stars on GitHub
- Started by Kenneth Reitz, now is a part of PyPA and officially recommended tool
- Python Development Workflow for Humans

pipenv

- <https://github.com/pypa/pipfile>
- `Pipfile` is a replacement for the existing standard pip's `requirements.txt` file
- `Pipfile` **designed for** projects which are **services**, not libraries

☺️ pipenv

```
2. ng@ip-10-88-1-56: ~/virtualenvs/tmp-64761d6072d5020 (zsh)
$ bat Pipfile

File: Pipfile

1 [[source]]
2 url = "https://pypi.python.org/simple"
3 name = "pypi"
4
5 [requires]
6 python_version = "3.7"
7
8 [packages]
9 requests = { version = "*", index = "pypi" }
10
11 [dev-packages]
12 pytest = "*"

$ █
```

☺️ pipenv

```
● ● ● 2. ng@ip-10-88-1-56: /var/folders/h0/bzt71j_j2vz49p5vx53jp6yr0000gn/T/tmp.Mxpz5Eyf (zsh)
```

```
$ pipenv lock
Locking [dev-packages] dependencies...
✓ Success!
Locking [packages] dependencies...
✓ Success!
Updated Pipfile.lock (f60c2d)!
```

```
$ ls -l
total 16
-rw-r--r-- 1 ng  staff   184 Feb 14 22:58 Pipfile
-rw-r--r-- 1 ng  staff  3963 Feb 14 22:59 Pipfile.lock

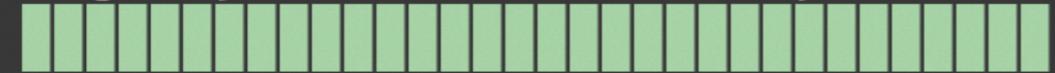
$ █
```

☺️ pipenv

```
● ● ● 2. ng@ip-10-88-1-56: /var/folders/h0/bzt71j_j2vz49p5vx53jp6yr0000gn/T/tmp.Mxpz5Eyf (zsh)
```

```
$ pipenv sync
```

```
Installing dependencies from Pipfile.lock (f60c2d)...
```



```
5/5 - 00:00:02
```

```
To activate this project's virtualenv, run pipenv shell.
```

```
Alternatively, run a command inside the virtualenv with pipenv run.
```

```
All dependencies are now up-to-date!
```

```
$ pipenv graph
```

```
requests=2.21.0
```

- certifi [required: ≥2017.4.17, installed: 2018.11.29]
- chardet [required: ≥3.0.2,<3.1.0, installed: 3.0.4]
- idna [required: ≥2.5,<2.9, installed: 2.8]
- urllib3 [required: ≥1.21.1,<1.25, installed: 1.24.1]

```
$ █
```

🤩 pipenv

- Automatically install required Pythons, if pyenv is available
- Automatically creates a virtualenv in a standard location
- Automatically verifies that `Pipfile.lock` is outdated
- Automatically loads ` `.env` files, if they exist

🤩 pipenv

- Checks for security vulnerabilities
(`pipenv check`)
- Can display currently-installed dependency graph information
(`pipenv graph`)
- Can check Python version
(`pipenv install --deploy`)

Is everything so good?



pipenv

```
● ● ● 2. ng@ip-10-88-1-56: /var/folders/h0/bzt71j_j2vz49p5vx53jp6yr0000gn/T/tmp.Mxpz5Eyf (zsh)
```

```
$ pipenv install oslo.utils=1.4.0
```



The logo for pipenv, featuring the word "pipenv" in a large, bold, black sans-serif font. A small orange circle is positioned to the left of the letter "p".

● ● 2. ng@ip-10-88-1-56: /var/folders/h0/bzt71j_j2vz49p5vx53jp6yr0000gn/T/tmp.Mxpz5Eyf (zsh)

```
$ pipenv install oslo.utils==1.4.0
Installing oslo.utils=1.4.0...
Adding oslo.utils to Pipfile's [packages]...
✓ Installation Succeeded
Pipfile.lock not found, creating...
Locking [dev-packages] dependencies...
Locking [packages] dependencies...
✗ Locking Failed!
[pipenv.exceptions.ResolutionFailure]:      File "/usr/local/Cellar/pipenv/2018.11.26_1/libexec/lib/python3.7/site-packages/pipenv/resolver.py", line 69, in resolve
[pipenv.exceptions.ResolutionFailure]:          req_dir=requirements_dir
[pipenv.exceptions.ResolutionFailure]:      File "/usr/local/Cellar/pipenv/2018.11.26_1/libexec/lib/python3.7/site-packages/pipenv/utils.py", line 726, in resolve_deps
[pipenv.exceptions.ResolutionFailure]:          req_dir=req_dir,
[pipenv.exceptions.ResolutionFailure]:      File "/usr/local/Cellar/pipenv/2018.11.26_1/libexec/lib/python3.7/site-packages/pipenv/utils.py", line 480, in actually_resolve_deps
[pipenv.exceptions.ResolutionFailure]:          resolved_tree = resolver.resolve()
```



pipenv

- pipenv uses pip-tools for dependency resolution
- pipenv uses safety to check vulnerabilities
- pipenv uses...

😢 pipenv

```
● ● ● 2. ng@ip-10-88-1-56: /var/folders/h0/bzt71j_j2vz49p5vx53jp6yr0000gn/T/tmp.Mxpz5Eyf (zsh)
$ pipenv install --dev black
Installing black...
Adding black to Pipfile's [dev-packages]...
✓ Installation Succeeded
Pipfile.lock not found, creating...
Locking [dev-packages] dependencies...
✗ Locking Failed!
[pipenv.exceptions.ResolutionFailure]:      File "/usr/local/Cellar/pipenv/2018.11.26_1/libexec/lib/python3.7/site-packages/pipenv/resolver.py", line 69, in resolve
[pipenv.exceptions.ResolutionFailure]:          req_dir=requirements_dir
[pipenv.exceptions.ResolutionFailure]:      File "/usr/local/Cellar/pipenv/2018.11.26_1/libexec/lib/python3.7/site-packages/pipenv/utils.py", line 726, in resolve_deps
[pipenv.exceptions.ResolutionFailure]:          req_dir=req_dir,
[pipenv.exceptions.ResolutionFailure]:      File "/usr/local/Cellar/pipenv/2018.11.26_1/libexec/lib/python3.7/site-packages/pipenv/utils.py", line 480, in actually_resolve_deps
[pipenv.exceptions.ResolutionFailure]:          resolved_tree = resolver.resolve()
[pipenv.exceptions.ResolutionFailure]:      File "/usr/local/Cellar/pipenv/2018.11.26_1/libexec/lib/python3.7/site-packages/pipenv/resolver.py", line 102, in resolve
[pipenv.exceptions.ResolutionFailure]:      self._lock_packages()
[pipenv.exceptions.ResolutionFailure]:      File "/usr/local/Cellar/pipenv/2018.11.26_1/libexec/lib/python3.7/site-packages/pipenv/resolver.py", line 110, in _lock_packages
[pipenv.exceptions.ResolutionFailure]:          self._lock_package(package)
```



pipenv

```
● ● ● 2. ng@ip-10-88-1-56: /var/folders/h0/bzt71j_j2vz49p5vx53jp6yr0000gn/T/tmp.Mxpz5Eyf (zsh)
```

```
$ pipenv install --dev --pre black  
Installing black...  
Adding black to Pipfile's [dev-packages]...
```

```
✓ Installation Succeeded
```

```
Pipfile.lock not found, creating...
```

```
Locking [dev-packages] dependencies...
```

```
✓ Success!
```

```
Locking [packages] dependencies...
```

```
Updated Pipfile.lock (3ffe2)!
```

```
Installing dependencies from Pipfile.lock (3ffe2)...
```



```
5/5 - 00:00:01
```

```
To activate this project's virtualenv, run pipenv shell.
```

```
Alternatively, run a command inside the virtualenv with pipenv run.
```

```
$ █
```



pipenv

● ● ● 2. ng@ip-10-88-1-56: /var/folders/h0/bzt71j_j2vz49p5vx53jp6yr0000gn/T/tmp.Mxpz5Eyf (zsh)

```
$ bat Pipfile
```

File: **Pipfile**

```
1 [[source]]
2 url = "https://pypi.python.org/simple"
3 name = "pypi"
4
5 [requires]
6 python_version = "3.7"
7
8 [packages]
9
10 [dev-packages]
11 black = "*"
12
13 [pipenv]
14 allow_prereleases = true
```

```
$ |
```



- <https://github.com/pypa/pipenv/issues/1760>



pipenv + setup.py

- <https://pipenv.readthedocs.io/en/latest/advanced/#pipfile-vs-setuppy>
- You can use only one version of Python in your project
- You need to define your dependencies in `Pipfile` and in setup.py too
- `Pipfile` **designed for** projects which are **services**, not libraries



pipenv + flit

- <https://github.com/takluyver/flit>
- 600+ stars on GitHub
- Simplified packaging of Python modules and a setup.py replacement

Why not pipenv?

- pipenv has a very simple dependency resolver (same as pip-tools)
- pipenv is very slow in dependency resolution!
- pipenv can't be used with various versions of Python due to limitations of Pipfile
- pipenv is hard to use if your project is a library



poetry





- <https://github.com/sdispater/poetry>
- 20 February 2018
- 3k+ stars on GitHub
- Started by Sébastien Eustace
- Inspired by cargo (Rust) and composer (PHP)
- Python dependency management and packaging made easy

poetry

- poetry is a tool to handle dependency installation as well as building and packaging of Python packages. It only needs one file to do all of that: the new, standardized `pyproject.toml`
- <https://www.python.org/dev/peps/pep-0518/>
- <https://www.python.org/dev/peps/pep-0517/>



poetry

● ● ● 2. ng@ip-10-88-1-56: /var/folders/h0/bzt71j_j2vz49p5vx53jp6yr0000gn/T/tmp.0dmGHm9v (zsh)

```
$ bat pyproject.toml
```

File: **pyproject.toml**

```
1 [tool.poetry]
2   name = "bypycon2019"
3   version = "0.1.0"
4   description = ""
5   authors = ["Nikita Grishko <gr1n@protonmail.com>"]
6
7   [tool.poetry.dependencies]
8     python = "^3.7"
9     django = "≥2.0.0"
10
11  [tool.poetry.dev-dependencies]
12
13  [build-system]
14    requires = ["poetry≥0.12"]
15    build-backend = "poetry.masonry.api"
```



poetry

```
● ● ● 2. ng@ip-10-88-1-56: /var/folders/h0/bzt71j_j2vz49p5vx53jp6yr0000gn/T/tmp.0dmGHm9v (zsh)
```

```
$ poetry lock
Updating dependencies
Resolving dependencies ... (6.0s)
```

```
Writing lock file
```

```
$ ls -l
total 16
-rw-r--r--  1 ng  staff  1970 Feb 14 23:22 poetry.lock
-rw-r--r--  1 ng  staff   297 Feb 14 23:22 pyproject.toml
```

```
$ █
```



poetry

```
● ● ● 2. ng@ip-10-88-1-56: /var/folders/h0/bzt71j_j2vz49p5vx53jp6yr0000gn/T/tmp.0dmGHm9v (zsh)
```

```
$ poetry install
Installing dependencies from lock file
```

```
Package operations: 2 installs, 0 updates, 0 removals
```

- Installing pytz (2018.9)
- Installing django (2.1.7)

```
$ poetry show
django 2.1.7 A high-level Python Web framework that encourages ...
pytz    2018.9 World timezone definitions, modern and historical
```

```
$ poetry show --tree
django 2.1.7 A high-level Python Web framework that encourages rapid
development and clean, pragmatic design.
└── pytz *
```

```
$ █
```

poetry

- Automatically creates a `virtualenv` in a standard location
- Can display currently-installed dependency graph information
(`poetry show --tree`)
- Can display outdated dependency information
(`poetry show --outdated`)
- Can build and upload your package to PyPI
(`poetry build && poetry upload`)



poetry

```
● ● ● 2. ng@ip-10-88-1-56: /var/folders/h0/bzt71j_j2vz49p5vx53jp6yr0000gn/T/tmp.0dmGHm9v (zsh)
```

```
$ poetry add oslo.utils=1.4.0
```

⭐️ poetry

```
● ● ● 2. ng@ip-10-88-1-56: /var/folders/h0/bzt71j_j2vz49p5vx53jp6yr0000gn/T/tmp.0dmGHm9v (zsh)
```

```
$ poetry add oslo.utils=1.4.0
```

Updating dependencies

Resolving dependencies ... (2.1s)

Package operations: 8 installs, 0 updates, 0 removals

Writing lock file

- Installing pytz (2018.9)
- Installing babel (2.6.0)
- Installing pbr (0.11.1)
- Installing iso8601 (0.1.12)
- Installing netaddr (0.7.19)
- Installing netifaces (0.10.9)
- Installing oslo.i18n (2.1.0)
- Installing oslo.utils (1.4.0)

```
$ █
```

This is possible thanks
to the efficient
dependency resolver at
the heart of Poetry

poetry

- oslo.utils (1.4.0)
 - pbr ($\geq 0.6, \neq 0.7, < 1.0$) \leftarrow pbr=0.11.1
 - oslo.i18n ($\geq 1.3.0$) \leftarrow oslo.i18n=3.20.0
 - pbr ($\neq 2.1.0, \geq 2.0.0$) \leftarrow incompatible with pbr=0.11.1



poetry

- oslo.utils (1.4.0)
 - pbr ($\geq 0.6, \neq 0.7, < 1.0$) \leftarrow pbr=0.11.1
 - oslo.i18n ($\geq 1.3.0$) \leftarrow oslo.i18n=2.1.0
 - pbr ($\geq 0.11, < 2.0$) \leftarrow compatible with pbr=0.11.1



Sébastien Eustace

@SDisPater

Читать



If you value your sanity, don't try to make a dependency resolver, especially for Python.

Перевести твит

19:09 - 7 июн. 2018 г.

5 ретвитов 26 отметок «Нравится»



2

5

26

✉

poetry = pip + virtualenv
+ virtualenv-wrapper +
zest.releaser + twine +
dependency resolver

Why not poetry?

- Lack of integrations with services like PyUp, Heroku, etc.
- Bugs...

Conclusion

- If you are a brave person try to use something new
- To my mind, **poetry** is the best choice now and we're going to use it at Flo

Thanks !

<https://github.com/Gr1N>

😱 pythonloc

- <https://github.com/cs01/pythonloc>
- <https://www.python.org/dev/peps/pep-0582/>