>>> network .toCode()

Network to Code

Intro To Poetry

Agenda

- What is poetry?
- Why use poetry?
- pyproject.toml
- poetry.lock
- First poetry project
- Updating Your Poetry Project

What is poetry?

What is poetry?

Poetry is a tool for dependency management and packaging in Python. It allows you to declare the libraries your project depends on and it will manage (install/update) them for you.

Dependency Tracking

- Declarative python package management
- Locking a snapshot in time
  - ansible = "^2.10.0"
  - Still builds at 2.10.0 even though 2.10.3 is out

```
$ poetry show --tree
requests-toolbelt 0.8.0 A utility belt for advanced users...
└── requests <3.0.0,>=2.0.1
    ├── certifi >=2017.4.17
    ├── chardet >=3.0.2,<3.1.0
    ├── idna >=2.5,<2.7
    └── urllib3 <1.23,>=1.21.1

$ poetry show --latest
pendulum 2.0.4   1.4.5 Python datetimes made easy.
django   1.11.11 2.0.3 A high-level Python Web framework ...
requests 2.18.4  2.18.4 Python HTTP for Humans.
```

Packaging & Publishing

- Built in packaging & publishing
- Simplifies releasing

```
$ poetry build

Building poetry (1.0.0)
- Building sdist
- Built poetry-1.0.0.tar.gz

- Building wheel
- Built poetry-1.0.0-py2.py3-none-any.whl
```

```
$ poetry publish

Publishing poetry (1.0.0) to PyPI

  - Uploading poetry-1.0.0.tar.gz 100%
  - Uploading poetry-1.0.0-py2.py3-none-any.whl 58%
```

Virtual Environment Management

- Obscures the need to understand managing virtual environments
- Allows ability to management multiple virtual environments per project
- Allows for management of system packages as well

Why use poetry?

Simplify Management

- Managing venvs with minimal overhead
- Declarative dependency tracking
- Repeatability

Collaboration

- Declarative dependency tracking/locking
- Reduces the chance of unpinned packages coming back to bite you
- Known good state

Dependency Mapping

- Locks based on version & hashes
- Not as fast as PIP but adds security tracking
- Tracking poetry.lock increase performance

pyproject.toml

Project/Dependency Management

- https://python-poetry.org/docs/pyproject/
- Declarative management
- Can be used with pip version 20
- Manage file with poetry command
  - Going direct to pyproject.toml can cause issues with lock file

Sample

```
[tool.poetry]
name = "awesome"

[tool.poetry.dependencies]
# These packages are mandatory and form the core of this package's distribution.
mandatory = "^1.0"

# A list of all of the optional dependencies, some of which are included in the
# below `extras`. They can be opted into by apps.
psycopg2 = { version = "^2.7", optional = true }
mysqlclient = { version = "^1.3", optional = true }

[tool.poetry.extras]
mysql = ["mysqlclient"]
pgsql = ["psycopg2"]
```

>>> network .toCode()

poetry.lock

poetry.lock

For your library, you may commit the poetry.lock file if you want to. This can help your team to always test against the same dependency versions. However, this lock file will not have any effect on other projects that depend on it. It only has an effect on the main project.

>>> network .toCode()

First poetry project

Initialize Poetry Project

- poetry init
  - Walks you through creating initial pyproject.toml
  - Does not perform initial install or venv creation

```
[pi@pi-k8s-controller:~/OfficeHours $ poetry init

This command will guide you through creating your pyproject.toml config.

[Package name [officehours]:
[Version [0.1.0]:
[Description []:  My fun office hours project
[Author [None, n to skip]:  Jeremy White <jeremy.white@networktocode.com>
[License []:  Apache-2.0
[Compatible Python versions [^3.7]:

[Would you like to define your main dependencies interactively? (yes/no) [yes]
You can specify a package in the following forms:
  - A single name (requests)
  - A name and a constraint (requests@^2.23.0)
  - A git url (git+https://github.com/python-poetry/poetry.git)
  - A git url with a revision (git+https://github.com/python-poetry/poetry.git#develop)
  - A file path (../my-package/my-package.whl)
  - A directory (../my-package/)
  - A url (https://example.com/packages/my-package-0.1.0.tar.gz)

[Search for package to add (or leave blank to continue): django@^3.0.0
Adding django@^3.0.0

[Add a package:

[Would you like to define your development dependencies interactively? (yes/no) [yes] no
Generated file

[tool.poetry]
name = "officehours"
version = "0.1.0"
description = "My fun office hours project"
authors = ["Jeremy White <jeremy.white@networktocode.com>"]
license = "Apache-2.0"

[tool.poetry.dependencies]
python = "^3.7"
django = "^3.0.0"

[tool.poetry.dev-dependencies]

[build-system]
requires = ["poetry-core>=1.0.0"]
build-backend = "poetry.core.masonry.api"


[Do you confirm generation? (yes/no) [yes]
[pi@pi-k8s-controller:~/OfficeHours $ ls -l
total 4
-rw-r--r-- 1 pi pi 366 Nov 13 15:21 pyproject.toml
pi@pi-k8s-controller:~/OfficeHours $
```

Generate poetry.lock & Install Packages

- poetry install
  - Resolves dependencies
  - Creates poetry.lock
  - Installs dependencies
  - Does not activate venv

```
[pi@pi-k8s-controller:~/OfficeHours $ ls -l
total 4
-rw-r--r-- 1 pi pi 366 Nov 13 15:21 pyproject.toml
[pi@pi-k8s-controller:~/OfficeHours $ poetry install
Creating virtualenv officehours-acv5MXjz-py3.7 in /home/pi/.cache/pypoetry/virtualenvs
Updating dependencies
Resolving dependencies... (2.9s)

Writing lock file

Package operations: 4 installs, 0 updates, 0 removals

  • Installing asgiref (3.3.1)
  • Installing pytz (2020.4)
  • Installing sqlparse (0.4.1)
  • Installing django (3.1.3)
[pi@pi-k8s-controller:~/OfficeHours $ ls -l
total 8
-rw-r--r-- 1 pi pi 2119 Nov 13 15:23 poetry.lock
-rw-r--r-- 1 pi pi  366 Nov 13 15:21 pyproject.toml
pi@pi-k8s-controller:~/OfficeHours $
```

Using Virtual Environment

- poetry shell
  - Activates venv

```
[pi@pi-k8s-controller:~/OfficeHours $ python
Python 2.7.16 (default, Oct 10 2019, 22:02:15)
[GCC 8.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
[>>> import django
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named django
[>>> exit()
[pi@pi-k8s-controller:~/OfficeHours $ poetry shell
Spawning shell within /home/pi/.cache/pypoetry/virtualenvs/officehours-acv5MXjz-py3.7
. /home/pi/.cache/pypoetry/virtualenvs/officehours-acv5MXjz-py3.7/bin/activate
pi@pi-k8s-controller:~/OfficeHours $ . /home/pi/.cache/pypoetry/virtualenvs/officehours-acv5MXjz-py3.7/bin/activate
[(officehours-acv5MXjz-py3.7) pi@pi-k8s-controller:~/OfficeHours $ python
Python 3.7.3 (default, Jul 25 2020, 13:03:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
[>>> import django
[>>>
```

DO NOT USE PIP

```
(officehours-acv5MXjz-py3.7) pi@pi-k8s-controller:~/OfficeHours $ pip install requests
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting requests
  Downloading requests-2.25.0-py2.py3-none-any.whl (61 kB)
     |████████████████████████████████| 61 kB 1.6 MB/s
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.2-py2.py3-none-any.whl (136 kB)
     |████████████████████████████████| 136 kB 6.1 MB/s
Collecting chardet<4,>=3.0.2
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133 kB)
     |████████████████████████████████| 133 kB 4.8 MB/s
Collecting certifi>=2017.4.17
  Downloading certifi-2020.11.8-py2.py3-none-any.whl (155 kB)
     |████████████████████████████████| 155 kB 6.5 MB/s
Collecting idna<3,>=2.5
  Downloading idna-2.10-py2.py3-none-any.whl (58 kB)
     |████████████████████████████████| 58 kB 2.0 MB/s
Installing collected packages: urllib3, chardet, certifi, idna, requests
Successfully installed certifi-2020.11.8 chardet-3.0.4 idna-2.10 requests-2.25.0 urllib3-1.26.2
(officehours-acv5MXjz-py3.7) pi@pi-k8s-controller:~/OfficeHours $ cat pyproject.toml
[tool.poetry]
name = "officehours"
version = "0.1.0"
description = "My fun office hours project"
authors = ["Jeremy White <jeremy.white@networktocode.com>"]
license = "Apache-2.0"

[tool.poetry.dependencies]
python = "^3.7"
django = "^3.0.0"

[tool.poetry.dev-dependencies]

[build-system]
requires = ["poetry-core>=1.0.0"]
build-backend = "poetry.core.masonry.api"
```

>>> network .toCode()

Updating Your Poetry Project

Adding Python Packages

- poetry add <python package>
  - Updates poetry.lock for new dependency
  - Adds to pyproject.toml

```
(officehours-acv5MXjz-py3.7) pi@pi-k8s-controller:~/OfficeHours $ poetry add flask@1.0

Updating dependencies
Resolving dependencies... (26.8s)

Writing lock file

Package operations: 6 installs, 0 updates, 0 removals

  • Installing markupsafe (1.1.1)
  • Installing click (7.1.2)
  • Installing itsdangerous (1.1.0)
  • Installing jinja2 (2.11.2)
  • Installing werkzeug (1.0.1)
  • Installing flask (1.0)
(officehours-acv5MXjz-py3.7) pi@pi-k8s-controller:~/OfficeHours $
```

Updating Pinned & Unpinned Libraries

- poetry add <python package>
- poetry update <python package>

```
[(officehours-acv5MXjz-py3.7) pi@pi-k8s-controller:~/OfficeHours $ poetry update flask
Updating dependencies
Resolving dependencies... (0.2s)

No dependencies to install or update
[(officehours-acv5MXjz-py3.7) pi@pi-k8s-controller:~/OfficeHours $
[(officehours-acv5MXjz-py3.7) pi@pi-k8s-controller:~/OfficeHours $
[(officehours-acv5MXjz-py3.7) pi@pi-k8s-controller:~/OfficeHours $
[(officehours-acv5MXjz-py3.7) pi@pi-k8s-controller:~/OfficeHours $
[(officehours-acv5MXjz-py3.7) pi@pi-k8s-controller:~/OfficeHours $
[(officehours-acv5MXjz-py3.7) pi@pi-k8s-controller:~/OfficeHours $ poetry add flask@1.1.1

Updating dependencies
Resolving dependencies... (0.4s)

Writing lock file

Package operations: 0 installs, 1 update, 0 removals

  • Updating flask (1.0 -> 1.1.1)
```