

Publish Your First Python Package

with Poetry

Behzad Samadi

January 15, 2023

What is Poetry?

- **Managing dependencies**

What is Poetry?

- **Managing dependencies**
- **Creating virtual environments**

What is Poetry?

- **Managing dependencies**
- **Creating virtual environments**
- **Packaging**

Installing Poetry

```
# Install pipx
```

```
python3 -m pip install --user pipx
```

```
python3 -m pipx ensurepath
```

```
# Install poetry
```

```
pipx install poetry
```

Creating a Python Project

```
$ poetry new simple-calculator
```

```
Created package simple_calculator in simple-calculator
```

```
$ tree ./simple-calculator/
```

```
./simple-calculator/
```

```
  pyproject.toml
```

```
  README.md
```

```
  simple_calculator
```

```
    __init__.py
```

```
  tests
```

```
    __init__.py
```

2 directories, 4 files

Python Project Structure

`pyproject.toml` **contains the metadata and dependencies of the project.**

```
[tool.poetry]
name = "poetry-demo"
version = "0.1.0"
description = ""
authors = ["Sébastien Eustace <sebastien@eustace.io>"]
readme = "README.md"
packages = [{include = "poetry_demo"}]

[tool.poetry.dependencies]
python = "^3.7"
```

- **The** `simple_calculator` **folder contains the source code.**

Python Project Structure

- **The `simple_calculator` folder contains the source code.**
- **The `tests` folder contains the tests.**

- **Adding a new dependency:**

```
poetry add requests
```

Python Project Dependencies

- **Adding a new dependency:**

```
poetry add requests
```

- **Installing dependencies**

```
poetry install
```

- Running the project

```
poetry run
```

Running Your Code

- **Running the project**

```
poetry run
```

- **Running a script**

```
poetry run simple_calculator/power.py
```

Running Your Code

- **Running the project**

```
poetry run
```

- **Running a script**

```
poetry run simple_calculator/power.py
```

- **Running pytest**

```
poetry run pytest
```

Building and Publishing Your Package

- **Building the project generates both sdist (source) and wheel (compiled) formats.**

```
poetry build
```

Building and Publishing Your Package

- **Building the project generates both sdist (source) and wheel (compiled) formats.**

```
poetry build
```

- **You can then publish your package on PyPi**

```
poetry publish
```

Publishing Your Package to Test PyPI

- To publish a package on PyPI, you need to create an account on <https://pypi.org>.

Publishing Your Package to Test PyPI

- To publish a package on PyPI, you need to create an account on <https://pypi.org>.
- Before publishing on the main repository, it is better to test your package on Test PyPI.

Publishing Your Package to Test PyPI

- To publish a package on PyPI, you need to create an account on <https://pypi.org>.
- Before publishing on the main repository, it is better to test your package on Test PyPI.
- The process is similar. You need to create an account on Test PyPI and get a token.

```
poetry config repositories.testpypi \
    https://test.pypi.org/legacy/
poetry config pypi-token.test-pypi <your-token>
poetry publish -r test-pypi
```

- **Increase the version (patch, minor, major, ..., prerelease)**

```
poetry version patch
```

Publishing Your Package to PyPI

- **Increase the version (patch, minor, major, ..., prerelease)**

```
poetry version patch
```

- **Publish!**

```
poetry config pypi-token.pypi <your-token>
```

```
poetry publish
```