

Poetry

: <https://python-poetry.org/>

Install

```
curl -sSL  
https://raw.githubusercontent.com/python-poetry/poetry/master/get-poetry.py  
| python -
```

Setting

가 가

```
poetry config virtualenvs.in-project true
```

Docs

- [docker poetry](#)
- [poetry config](#)

poetry update

[pip](#) upgrade

```
pip install --upgrade pip
```

requirements.txt

```
cat requirements.txt|xargs poetry add
```

```
poetry config experimental.new-installer false
```

가

```
poetry add [package name]
```

apt-get install python3-venv

- <https://github.com/sdispater/poetry/issues/46>

```
% poetry init  
% poetry add `cat requirements.txt`
```

```
apt-get install python3-venv
```

```
poetry self:update --preview  
poetry env use python3.7
```

<https://www.itsupportwale.com/blog/how-to-upgrade-to-python-3-7-on-ubuntu-18-10/>

```
sudo apt-get install python3.7-venv
```

ERROR

httptools/parser/parser.c:4:10: fatal error: Python.h: No such file or directory

- <https://stackoverflow.com/questions/21530577/fatal-error-python-h-no-such-file-or-directory>

For apt (Ubuntu, Debian...):

```
sudo apt-get install python-dev # for python2.x installs  
sudo apt-get install python3-dev # for python3.x installs
```

- <https://poetry.eustace.io/docs/configuration/>

```
poetry config virtualenvs.in-project true
```

- <https://github.com/sdispater/poetry/issues/108>

ERROR

```
[ValueError]  
Setting virtualenvs.in-project does not exist
```

```
poetry config settings.virtualenvs.in-project true
```

Integrating Python Poetry with Docker

- <https://stackoverflow.com/questions/53835198/integrating-python-poetry-with-docker>

```
FROM python:3.6.6-alpine3.7  
  
ARG YOUR_ENV  
  
ENV YOUR_ENV=${YOUR_ENV} \  
    PYTHONFAULTHANDLER=1 \  
    PYTHONUNBUFFERED=1 \  
    PYTHONHASHSEED=random \  
    PIP_NO_CACHE_DIR=off \  
    PIP_DISABLE_PIP_VERSION_CHECK=on \  
    PIP_DEFAULT_TIMEOUT=100 \  
    POETRY_VERSION=0.12.11  
  
# System deps:  
RUN pip install "poetry==$POETRY_VERSION"  
  
# Copy only requirements to cache them in docker layer  
WORKDIR /code  
COPY poetry.lock pyproject.toml /code/  
  
# Project initialization:  
RUN poetry config settings.virtualenvs.create false \  
    && poetry install $(test "$YOUR_ENV" == production && echo "--no-dev") --  
no-interaction --no-ansi  
  
# Creating folders, and files for a project:  
COPY . /code
```

Error

- [python setup.py bdist_wheel did not run successfully](#)
-

Plugin Backlinks:

From:

<https://jace.link/> - **Various Ways**

Permanent link:

<https://jace.link/open/poetry>

Last update: **2022/07/18 12:16**

