



# Hatch



Python Rennes Meetup - 18 avril 2024



# Frédéric Collonval

## Développeur full-stack freelance @WebScIT

Mission actuelle @Datalayer

## Contributeur open-source

JupyterLab SSC rep 2023 – Jupyter distinguish contributor 2021

 <http://ow.ly/ZYuB30eITef>

 fcollonval



# Qu'est-ce que hatch

- Website : <https://hatch.pypa.io>
- « Un manager de projets Python moderne et extensible »
- License MIT
- Projet sous l'organisation GitHub Python Packaging Authority (pypa)
- Auteur : Ofek Lev
- Alternatives  
[pdm](#) (6,5k ), [poetry](#) (29,3k ), [setuptools](#) (2,3k ), [tox](#) (3,5k ),...



# Qu'est-ce que hatch

- « Un manager de projets Python moderne et extensible »
  - Build system
  - Environnements
  - Manage Python version
  - Analyse statique (via ruff)
  - Publication
  - Bump de versions
  - Création de projets — très rudimentaire

Focus de cette présentation



# L'exemple fil rouge

- Micro-service spécifié via [openAPI v3.0](#)
- Implémentation via [connexion](#)
- Objectifs
  - Package metadata depuis spécifications openAPI
  - Création d'un client Python pour le micro-service durant le build



# Packaging with hatch

- Configuration de cible ;  
sdist, wheel
- Par défaut, inclus tous les fichiers  
non-ignorés (respecte .gitignore)

## Options

- include/exclude
- artifacts; les fichiers générés au  
build

```
1 [build-system]
2 # Using git source is only needed because the package is not published
3 # It also means that it will not use the local code of hatch-openapi-version
4 requires = ["hatchling", "hatch-openapi-version @git+https://github.com/fcoll
5 build-backend = "hatchling.build"
6
7 [project]
8 name = "hatch-demo"
9 dynamic = ["version", "authors", "description", "license"]
10 readme = "README.md"
11 requires-python = ">=3.8"
12 keywords = []
13 > classifiers = [ ...
22 > dependencies = [ ...
27
28 > [project.optional-dependencies] ...
29 > [project.urls] ...
30
31 [tool.hatch.build.targets.sdist]
32 exclude = [ "hatch-openapi-version" ]
33
34 [tool.hatch.build.targets.wheel]
35 artifacts = [ "src/hatch_demo/openapi_client" ]
36
37 [tool.hatch.version]
38 source = "openapi"
39 path = "src/hatch_demo/spec/openapi.yaml"
40
41 [tool.hatch.build.hooks.version]
42 path = "src/hatch_demo/_version.py"
43 template = ""
44 # This file is auto-generated by Hatchling. As such, do not:
45 # - modify
46 # - track in version control e.g. be sure to add to .gitignore
47 __version__ = '{version}'
48
49 [tool.hatch.metadata.hooks.custom]
50
51 [tool.hatch.build.hooks.custom]
52
```



# Plugins

- Système de plugins basé sur pluggy (by pytest)

- Distribution via package Python avec `entry-points`

- Plugins disponibles

- Builder
- Build hook → build no-Python assets
- Metadata hook → read package metadata from non-python config
- Environment
- Environment collector
- Publisher
- Version source → handle version in non-python file
- Version scheme

```
33 [project.entry-points.hatch]  
34 openapi = "hatch_openapi_version"
```



# Version source

- Exemples
  - `hatch-vcs`
  - `hatch-nodejs-version` ; utilisé dans Jupyter
- Demo : `hatch_openapi_version.py`





# Metadata hook

- Exemple
  - `hatch-nodejs-version` ; utilisé dans Jupyter
- Demo : `Metadata custom hook`



# Build hook

- Exemple
  - `hatch-jupyter-builder`; utilisé dans Jupyter
- Demo : `Build custom hook`

