HMC HELMHOLTZ Metadata Collaboration

IAS-9 Materials Data Science & Informatics

JÜLICH Forschungszentrum

Streamlining Metadata Handling in Research Software Engineering

Mustafa Soylu[1] // Anton Pirogov[1] // Volker Hofmann[1] // Stefan Sandfeld[1]

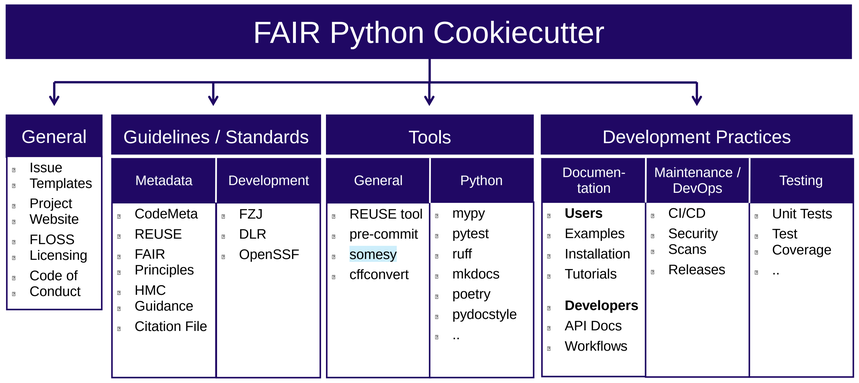
[1] Materials Data Science and Informatics (IAS-9), Forschungszentrum Jülich GmbH

FAIR PYTHON COOKIECUTTER

A Best-Practice Python Project Template

The majority of **software** in research is **written by** domain **scientists**, not experienced software engineers. However, **sustainable** and **FAIR software** development requires **more than just programming**. It requires substantial knowledge of **best practices** with respect to tools and processes for project management, collaboration, development and maintenance. The typical domain researcher has **insufficient time** to address such issues. The **fair-python-cookiecutter** is a project template targeting researchers and RSEs writing code in an academic environment. It helps implementing **best practices for software development and metadata.**

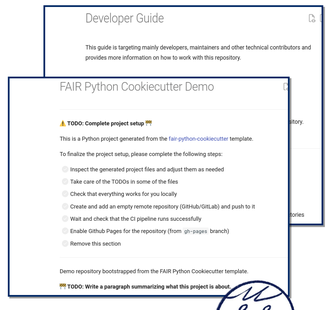
Tools and Best Practices in One Package



- A ready-to-use structure for modern Python development, testing, QA and packaging based on tools such as `poetry`, `pytest` and `pre-commit`
- GitHub + GitLab CI pipelines for development and deployment (e.g. PyPI)
- Project documentation website based on `mkdocs` and GitHub Pages
- Simplified management of relevant metadata using **`somesy`**
- Detailed documentation for setup and configuration to support quick adoption
- Implements recommendations by OpenSSF[1], DLR[2] and fair-software.eu[3]

Git Repo

Demo



Interested?

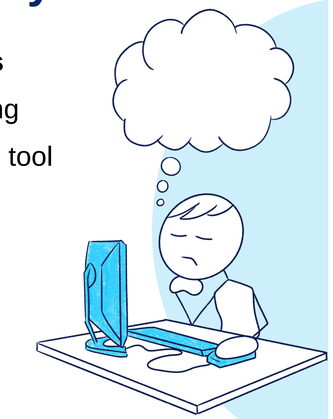
Get in touch!

[1] https://www.bestpractices.dev [2] https://rse.dlr.de [3] https://fair-software.eu

SOMESY

Software Metadata Synchronization

Modern research heavily relies on **software**, and in order to be FAIR it **needs rich and correct metadata**. Current metadata best practices include providing CITATION.cff and codemeta.json files, but software projects usually must use tool or language-specific files such as pyproject.toml or package.json that contain **similar metadata**. All these standards overlap, sometimes with misaligned meaning, creating **redundancy and ambiguity** between them. This makes metadata management an error-prone and tedious task.



Main Features of *somesy*

- Based on CITATION.cff version 1.2 metadata standard (with extensions)
- Supports **rich metadata** while avoiding needless duplication
- **Automates** the **synchronization** of software project metadata
- **Preserves other content and comments** in existing files like pyproject.toml
- **Reduces overhead** of maintaining metadata located in various files
- Provides a **pre-commit hook** → can check and fix issues before each commit
- Provides a **single source of truth** for common project metadata
- **Extensible** for support of other tool-specific formats and common standards

General Standards

CFF *CITATION*

JSON *CODEMETA*

Software Specific Metadata

Java, JS, julia, SETUPTOOLS, Poetry, MKDOCS

Git Repo

HELMHOLTZ