

A large, thick, orange swoosh that starts from the bottom left, curves upwards and to the right, then loops back down and to the left, framing the text.

# Sardana Controllers as Python Packages

Antonio Milán Otero

# Overview

- Cookiecutter
- Sardana controller template
- How to move old controllers
- Possibilities on CI/CD

# Cookiecutter

In their own words:

*A command-line utility that creates projects from **cookiecutters** (project templates), e.g. creating a Python package project from a Python package project template.*

Cookiecutter Documentation: <https://cookiecutter.readthedocs.io>

Minimal MAXIV Example: <https://github.com/amilan/cookiecutter-maxiv-minimal>

More elaborated project: <https://github.com/audreyr/cookiecutter-pypackage>

# Cookiecutter example: structure

cookiecutter-maxiv-minimal >

```
.
├── cookiecutter.json
├── {{cookiecutter.project_name}}
│   ├── .bumpversion.cfg
│   ├── conftest.py
│   ├── {{cookiecutter.package_name}}
│   │   ├── {{cookiecutter.package_name}}.py
│   │   └── __init__.py
│   └── docs
│       ├── conf.py
│       ├── index.rst
│       ├── make.bat
│       ├── Makefile
│       └── MAXIV_logo_rgb.png
├── .gitignore
├── LICENSE
├── README.rst
├── requirements.txt
├── setup.cfg
├── setup.py
├── tests
│   └── test_{{cookiecutter.package_name}}.py
└── README.md
```

## Cookiecutter example: cookiecutter.json

```
{  
  "project_name": "lib-maxiv-new_project",  
  "package_name": "package",  
  "project_description": "This is my new super cool project",  
  "author_name": "antmil",  
  "author_email": "antonio.milan_otero@maxiv.lu.se"  
}
```

# Cookiecutter example: test\_{{cookiecutter.package\_name}}.py

```
#!/usr/bin/env python
```

```
"""Test suite for {{cookiecutter.package_name}}."""
```

```
__author__ = '{{cookiecutter.author_name}}'
```

```
__docformat__ = 'restructuredtext'
```

```
def test_{{cookiecutter.package_name}}():
```

```
    """Some template test for the {{cookiecutter.package_name}} package."""
```

```
    pass
```

# Cookiecutter example: Usage

Once you have cookiecutter installed, you can clone the repository and run:

```
git clone git@github.com:amilan/cookiecutter-maxiv-minimal.git
```

```
cookiecutter cookiecutter-maxiv-minimal/
```

Or you can use it directly from the github repository:

```
cookiecutter gh:amilan/cookiecutter-maxiv-minimal
```

More detailed info: <https://cookiecutter.readthedocs.io/en/latest/usage.html>

# Sardana Controller template

Similar to the minimal.

Available here: <https://github.com/amilan/cookiecutter-sardana-ctrl>

Future additions:

- Choose different licenses
- Choose a tailored template for the type of controller to be developed
- Customized docs
- Etc ...



# Sardana Controllers: How to move an existing one

Instructions available in the wiki of the template project:

<https://github.com/amilan/cookiecutter-sardana-ctrl/wiki>

(as an exercise) Icepap controller and AlbaEM moved here:

<https://gitlab.maxiv.lu.se/antmil/sardana-controller-icepap>

<https://gitlab.maxiv.lu.se/antmil/sardana-ctrl-albaem>

# CI/CD: Use Case at MaxIV gitlab

The screenshot displays the GitLab CI/CD interface for the 'icepap-staging' project. The top navigation bar includes links for Projects, Groups, Activity, Milestones, and Snippets. The left sidebar shows the project structure, with 'CI / CD' and 'Pipelines' highlighted. The main content area shows the details of Pipeline #2722, which was triggered 1 day ago by Antoine Dupre and passed. The pipeline is titled 'Update .gitlab-ci.yml' and consists of 6 jobs from the 'testing' stage. The jobs are: trigger\_pipeline, build\_icepapcms, build\_osc, build\_pyicepap, testing\_el7\_x86..., and deploy\_icepapc... The pipeline status is 'passed'.

**icepap-staging**

Project  
Repository  
Issues  
Merge Requests

**CI / CD**

**Pipelines**

Jobs  
Schedules  
Charts

Operations  
Wiki  
Snippets  
Settings

KITS MAXIV > icepap-staging > Pipelines > #2722

**passed** Pipeline #2722 triggered 1 day ago by Antoine Dupre

### Update .gitlab-ci.yml

6 jobs from testing

latest

9d2cf2d1

**Pipeline** Jobs 6

**Trigger** **Build** **PublishRPM** **Deploy**

trigger\_pipeline  
build\_icepapcms  
build\_osc  
build\_pyicepap  
testing\_el7\_x86...  
deploy\_icepapc...

# Pros and Cons

## Pros:

- Better version handling
- Better CI/CD
- Enable the usage of Pipelines
- Documentation per Controller

## Cons:

- Hard to find existing controllers
- Catalog needed (it can be automatically updated via CI/CD)
- Existing controllers could stay in a single repository or moved to their own repository, but that will generate some work

Questions?