



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 893240



DELIVERABLE

Project Acronym: REScoop VPP

Grant Agreement number: 893240

Project Title: Smart Building Ecosystem for Energy Communities

D6.2 Coding Guidelines and Documentation

Revision: 1

Authors:

Jan Pecinovsky (EnergielD)

Max Helskens (EnergielD)

Peter Armstrong (Carbon Coop)

Jess Fairbarn (Carbon Coop)

Emmanuel Charon (Enercoop)

REVISION HISTORY AND STATEMENT OF ORIGINALITY

Revision History

Revision	Date	Author	Organization	Description
Draft	28/05/2021	JP	EnergyID	First Draft
Final version	12/06/2021	JP	EnergyID	Final version
Final revision	14/06/2021	MN	SNAP	Final revision

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Executive Summary

This deliverable is a brief reference, documenting the websites and repositories that contain the coding guidelines and documentation of the REScoopVPP-tools.

These code and documentation repositories target 3 distinct audiences:

1. The developers and maintainers of the tools
2. The parties responsible for installing COFY-boxes on location
3. Third parties, interested in using and or adapting the tools

There are 3 documentation and code web pages: COFY-docs, a wiki-style documentation page for COFY-box and COFY-cloud documentation and installation guidelines; the REScoopVPP Gitlab, a code repository for all software blocks running on COFY-box and COFY-cloud; and Enda, the Python machine learning project for energy portfolio forecasting.

Table of Contents

COFY Docs	4
REScoopVPP Gitlab	5
Enda	6

COFY Docs

The COFY-docs website, a wiki page found at <https://docs.cofybox.io>, serves as the *homepage* for all documentation. It contains the architecture and documentation for the COFY-boxes' firmware and software blocks; architecture, data exchange formats and deployment instructions for the COFY-cloud backend system; and installation instructions for end users and technicians that install COFY-boxes in the field.

This wiki page is frequently updated. The swagger documentation for the COFY-cloud API will also be linked on this page.

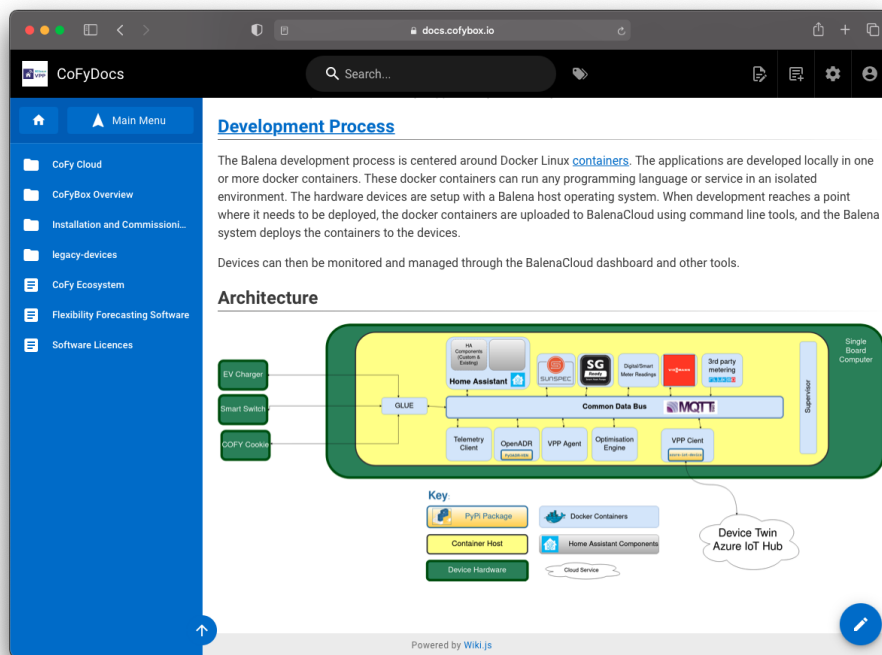


figure 1: Screenshot of the COFY-Docs website

REScoopVPP Gitlab

The REScoopVPP project page on Gitlab, found at <https://gitlab.com/rescoopvpp>, hosts the code repositories of the COFY-boxes' software blocks and the COFY-cloud's *Data Processing Engines*. Next to code versioning, Gitlab is also used as a tool for issue and bug tracking by the development team.

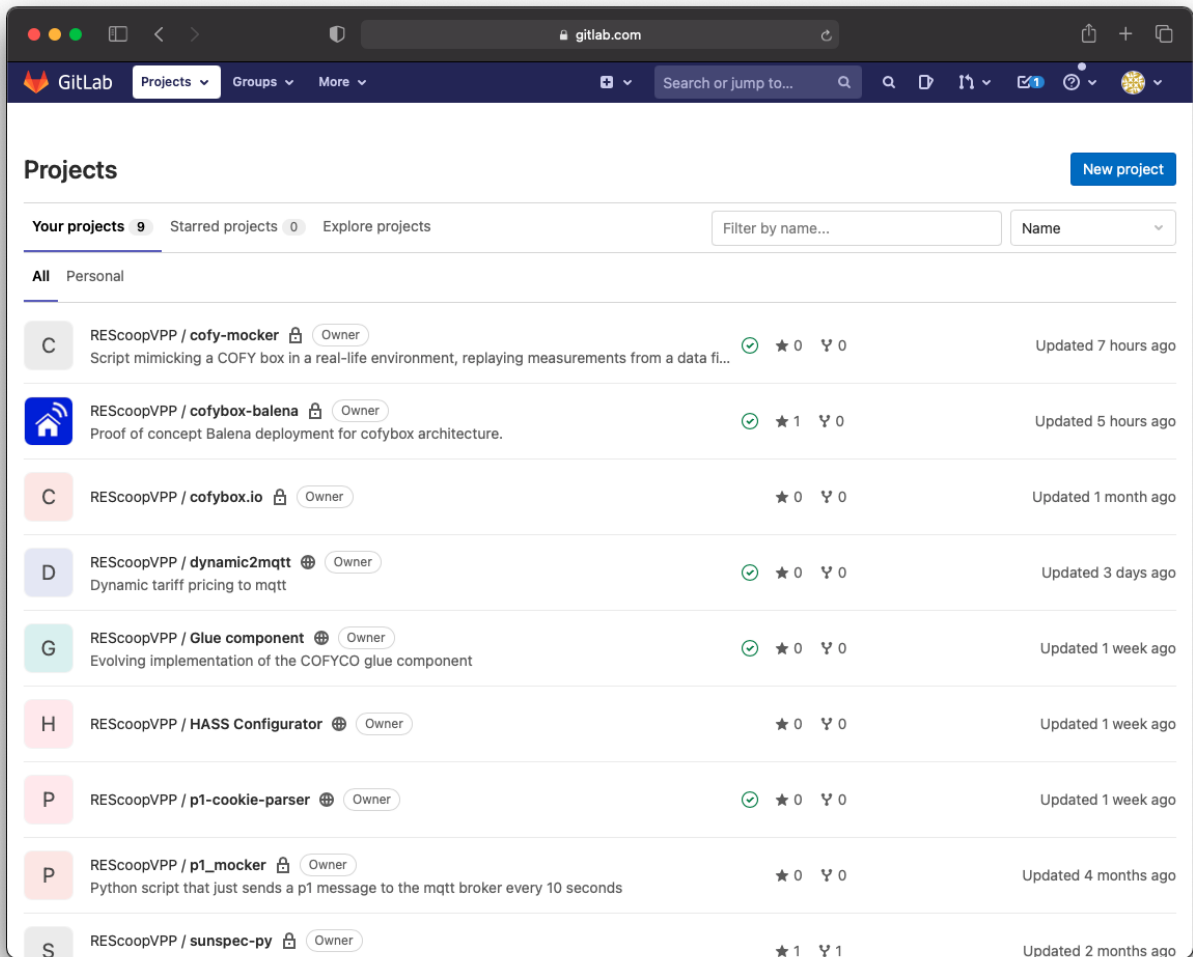
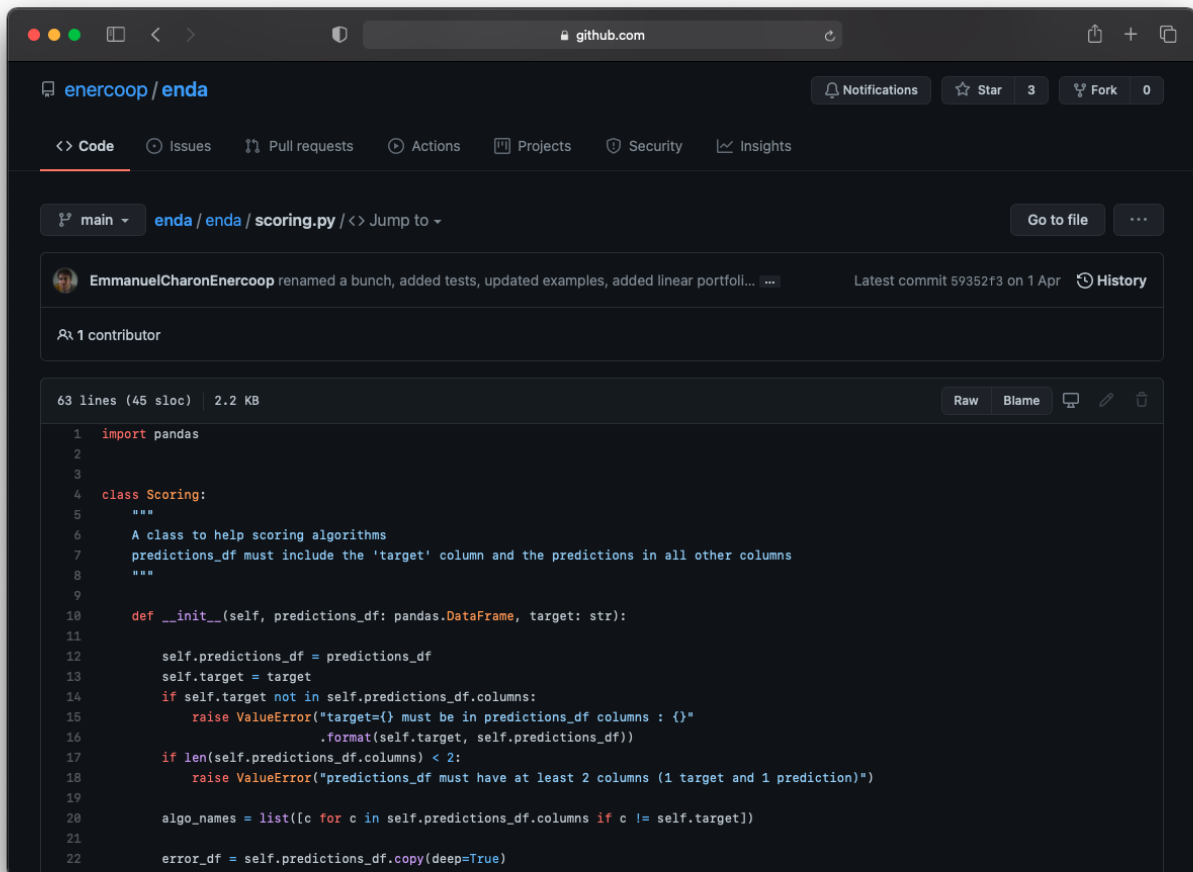


figure 2: Screenshot of the REScoopVPP Gitlab project page

Enda

Enda is the name of the Python library tasked with Portfolio Forecasting, developed in WP4. At <https://github.com/enercoop/enda>, the code can be found alongside some working examples and datasets. Enda is also published on the Python Package Index at <https://pypi.org/project/enda/>.



The screenshot shows the GitHub repository page for 'enercoop/enda'. The file 'scoring.py' is selected, showing 63 lines of Python code. The code defines a 'Scoring' class that takes a pandas DataFrame and a target column name as input. It includes validation logic to ensure the target column exists and that there are at least two columns in total. The class also initializes 'algo_names' based on the DataFrame columns.

```
1 import pandas
2
3
4 class Scoring:
5     """
6     A class to help scoring algorithms
7     predictions_df must include the 'target' column and the predictions in all other columns
8     """
9
10    def __init__(self, predictions_df: pandas.DataFrame, target: str):
11
12        self.predictions_df = predictions_df
13        self.target = target
14        if self.target not in self.predictions_df.columns:
15            raise ValueError("target={} must be in predictions_df columns : {}".format(self.target, self.predictions_df))
16        if len(self.predictions_df.columns) < 2:
17            raise ValueError("predictions_df must have at least 2 columns (1 target and 1 prediction)")
18
19        algo_names = list([c for c in self.predictions_df.columns if c != self.target])
20
21        error_df = self.predictions_df.copy(deep=True)
22        for i in algo_names:
```

figure 3: screenshot of the Enda Github page