

---

# **fairworkflows**

***Release 1.2.7***

**Robin Richardson, Sven van der Burg**

**Jun 25, 2021**



## API REFERENCE

1	<code>fairworkflows.fairworkflow</code>	1
2	<code>fairworkflows.prov</code>	3
3	<i>fairworkflows</i> python library	5



**FAIRWORKFLOWS.FAIRWORKFLOW**



**FAIRWORKFLOWS.PROV**





## FAIRWORKFLOWS PYTHON LIBRARY

`fairworkflows` is a high-level, user-friendly python library that supports the construction, manipulation and publishing of FAIR scientific workflows using semantic technologies.

### 3.1 Background

`fairworkflows` is developed as a component of the FAIR Workbench, as part of the FAIR is FAIR project.

The focus is on description of workflows consisting of manual and computational steps using semantic technology, such as the ontology described in the publication:

*Celebi, R., Moreira, J. R., Hassan, A. A., Ayyar, S., Ridder, L., Kuhn, T., & Dumontier, M. (2019). Towards FAIR protocols and workflows: The OpenPREDICT case study. \*arXiv:1911.09531.\**

The goals of the project are: 1. To facilitate the construction of RDF descriptions of a variety of scientific ‘workflows’, in the most general sense. This includes experimental procedures, ipython notebooks, computational analysis of results, etc. 2. To allow validation and publication of the resultant RDF (for example, by means of nanopublications). 3. Re-use of previously published steps, in new workflows. 4. FAIR data flow from end-to-end.

We seek to provide an easy-to-use python interface for achieving the above.

### 3.2 Installation

The most recent release can be installed from the python package index using `pip`:

```
pip install fairworkflows
```

To publish workflows to the nanopub server you need to setup your nanopub profile. This allows the nanopub server to identify you. Run the following in the terminal after installation:

```
setup_nanopub_profile
```

This will add and store RSA keys to sign your nanopublications, publish a nanopublication with your name and ORCID iD to declare that you are using these RSA keys, and store your ORCID iD to automatically add as author to the provenance of any nanopublication you will publish using this library.

## 3.3 Quick demo

Try out the library in this online executable notebook:

## 3.4 Quick Start

### 3.4.1 Import from fairworkflows library

```
from fairworkflows import is_fairworkflow, is_fairstep, FairWorkflow
```

### 3.4.2 Define a step for your workflow

Mark a function as a FAIR step using the `is_fairstep` decorator. Use keyword arguments to semantically annotate the step. In this example to provide a label and describe that this is a script task.

```
@is_fairstep(label='Addition', is_script_task=True)
def add(x: float, y: float) -> float:
    """Adding up numbers."""
    return x + y
```

### 3.4.3 Define your workflow

Define your workflow by calling previously defined step functions. Mark the function as a workflow using the `is_fairworkflow` decorator.

```
@is_fairworkflow(label='My Workflow')
def my_workflow(in1, in2):
    """
    A simple workflow
    """
    t1 = add(in1, in2)
    return t1
```

### 3.4.4 Construct and publish a workflow

Construct a `FairWorkflow` object from the function defining the workflow and publish as nanopublication.

```
workflow = FairWorkflow.from_function(my_workflow)
workflow.publish_as_nanopub(use_test_server=True, publish_steps=True)
```

### 3.4.5 Execute the workflow

Execute the workflow and inspect the prospective provenance

```
result, prov = workflow.execute(1, 4)
print(prov)
```

### 3.4.6 Example notebook

- See [examples/fairworkflows-quick-start.ipynb](#) for a current example of using the fairworkflows library to build a workflow using plex rdf

## 3.5 How is the fairworkflows library expected to be used?

While this library could be used as a standalone tool to build/publish RDF workflows, it is intended more as a component to be used in a variety of other tools that seek to add FAIR elements to workflows. At present the library is used in the following tools:

- [FAIRWorkflowsExtension](#): A Jupyter Lab extension that adds a widget for searching for previously published FairSteps or FairWorkflows. These can then be loaded into the notebook for modification or combination into new workflows.

It is expected that the library will soon interact with FAIR Data Points as well e.g. [fairdatapoint](#).

## 3.6 Relation to existing workflow formats/engines (e.g. CWL, WDL, Snakemake etc)

This library is not intended to replace or compete with the hundreds of existing computational workflow formats, but rather to aid in RDF description and comparison of workflows in the most general sense of the term (including manual experimental steps, notebooks, and so on). Steps in a FAIRWorkflow may very well be ‘run this CWL workflow’ or ‘run this script’, so such workflows are expected to sit more on a meta-level, describing the before-and-after of running one of these fully automated computational workflows as well.