

User guide for the repository IO-TEMPLATE-LIB

IO-TEMPLATE-LIB is a template repository for creating Python libraries. This document describes how to use this repository to create a new repository. In the following instructions, we assume that the new repository should be named my-lib and the library to be created with it should be named mylib.

I. Requirements

Regarding operating system, Ubuntu version 20.04 and above and Windows version 10 and above are supported. An existing Python 3 installation is required. Furthermore, the use of an IDE or a text editor that can replace texts across files is useful.

II. Repository creation

1. Create the new repository my-lib

As described [here](#), the new repository my-app must first be created.

2. Create a local copy of the new repository my-lib

```
git clone https://github.com/io-aero/my-lib
```

3. Delete the two files with the User's Guide

```
`user_guide.md`  
`user_guide.pdf`
```

4. Rename the following file directories and files

Old name	New name
iotemplatelib	mylib
run_io_template_lib.bat	run_my_lib.bat
run_io_template_lib.sh	run_my_lib.sh

5. Replacing texts in the new repository my-lib

It is absolutely necessary to respect the capitalization!

Old text	New text
IO-TEMPLATE-LIB	MY-LIB
IO_TEMPLATE_LIB	MY_LIB

Old text	New text
<code>io-template-lib</code>	<code>my-lib</code>
<code>io_template_lib</code>	<code>my_lib</code>
<code>iotemplatelib</code>	<code>mylib</code>

6. Store your AWS access rights in file `~/.aws/credentials`

```
[default]
aws_access_key_id=...
aws_secret_access_key=...
```

7. Create the package index configuration file `~/.pypirc`

```
[distutils]
index-servers =
    codeartifact
    pypi
    testpypi

[codeartifact]
repository = https://io-aero-444046118275.d.codeartifact.us-east-1.amazonaws.com/pypi/io-aero-pypi/
username = aws
password = <password>

[pypi]
repository = https://upload.pypi.org/legacy/
username = io-aero
password = <password>

[testpypi]
repository = https://test.pypi.org/legacy/
username = io-aero-test
password = <password>
```

8. Test the current state of the new library

8.1 If Miniconda is required

- Install Miniconda
- Run `make conda-dev`
- Run `make final`

8.2 If Miniconda is not required

- Run `make pipenv-dev`
- Run `make final`

9. Define GitHub Actions secrets

Under 'settings' -> 'Secrets and variables' -> 'Actions' -> Tab 'Secrets' define the following 'New repository secret's:

```
AWS_ACCESS_KEY_ID
AWS_SECRET_ACCESS_KEY
GLOBAL_USER_EMAIL
```

10. Define GitHub repository variables

Under 'settings' -> 'Secrets and variables' -> 'Actions' -> Tab 'Variables' define the following 'New repository variable's:

Name	Value	Reason
CONDA	true	Include Miniconda
COVERALLS	true	Run coveralls.io

11. Commit and push all changes to the repository as 'Base version'