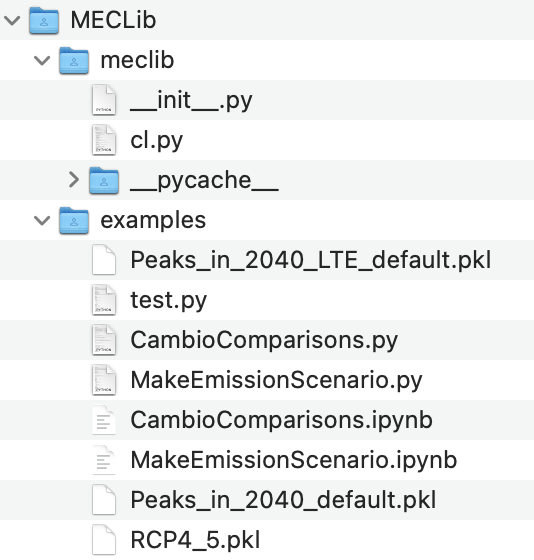
**First-time prep**

**In Firefox**

1. Set up a username at PyPI.org

**In the Finder**

1. Set up a directory structure. Here, the top MECLib folder is called the “project folder,” and the nested MECLib folder is called the “package folder”.  
     
   

**In VSCode**

1. Edit \_\_init\_\_.py so that it has the line “from .cl import \*”
2. In a terminal window of VSCode, navigate to the project folder, and create a virtual environment with the command

> python3 -m venv .venv

This should create a folder by that name.

1. Activate the virtual environment: > source .venv/bin/activate
2. Start tracking with git:  
   > git init  
   > git branch -m main
3. Create a .gitignore that has the following:  
   # Environment

.venv

# Mac system

.DS\_Store

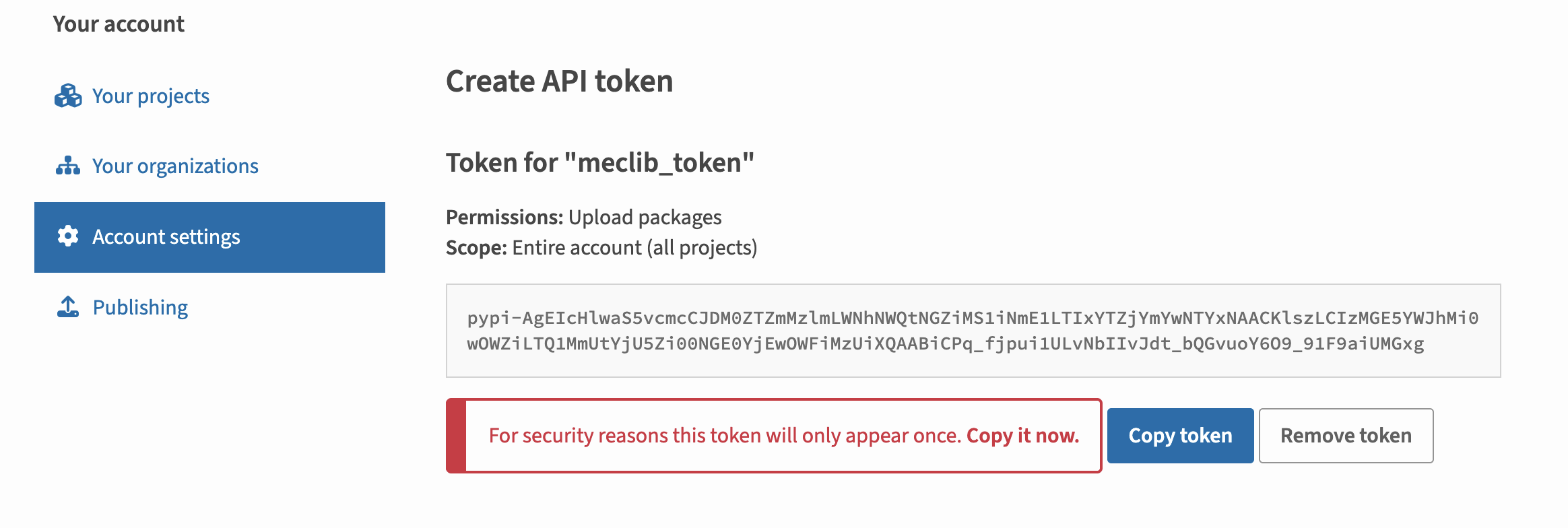
# Python files

\_\_pycache\_\_/

\*.pyc

\*.pkl

\*.ipynb\_checkpoints

1. Check the status of git: > git status … and make sure that everything it’s going to push, you want to be pushed. Might have to add at least one file from each subfolder and then git status again to be sure it’s there: > git add examples/MakeEmissionScenario.py
2. Optionally, add them all: > git add .
3. Commit them all: > git commit -m “first commit”
4. To verify, use the “source code in-place” method. This acts as if the package is installed. The pyproject.toml file is used to specify what folder to use (in this case, meclib). To do this, while still in the project folder (and the dot is really there): > pip install -e .
5. Test that the code works: > python examples/MakeEmissionScenario.py
6. We’ll need “build” and “twine” (which uploads to PyPI): > pip install build twine
7. Build the package: > python -m build … this should create a folder called dist
8. Assuming the build is successful, uninstall meclib with pip: > pip uninstall meclib … this can be verified with: > pip list
9. Now install the whl version we just made: > pip install dist/meclib-0.1.0-py3-none-any.whl
10. Test that the code still works: > python examples/MakeEmissionScenario.py
11. Assuming it works, go to PyPI and get a token  
    
12. Use twine to upload: > twine upload -u \_\_token\_\_ -p pypi-xyz dist/\* (where pypi-xyz is the token copied from PyPI)

**Somewhere else (like on the VM), to testing that the code works from PyPI**

1. Uninstall the old version of MECLib: > python -m pip uninstall MECLib
2. Install the new version: > python -m pip install MECLib.
3. Run test codes. Don’t forget that in .ipynb files, the syntax will be: import meclib.cl as CL

**What if we want to modify the package?**

1. Back in VSCode, make the modifications, then (presumably) use git to add and commit them (repeating steps 8 and 10)
2. Verify (repeating steps 11 and 12)
3. Remove the distribution and build folders: > rm -rf dist/ build/ \*.egg-info
4. Edit pyproject.toml to specify a new version number (like version = “0.1.1”)
5. Use build and twine again (repeating steps 14-19)
6. Somewhere else, like in the VM, repeat steps 20-22