

The five roles shared		
Role	pip side	Poetry side
Frontend	pip, python -m build	poetry build
Backend	hatchling, setuptools, flit-core, pdm-backend, uv_build	poetry-core (any since 2.1)
Installer	pip install	poetry install/sync
Index	PyPI; upload with twine	poetry publish
Environment	python -m venv .venv	one per project, managed
sdist name-1.0.tar.gz: source + PKG-INFO, needs the backend to install. wheel .whl: ready to unpack, no build. Names normalised (Bird_Tally = bird-tally; files use _).		

pyproject.toml shared	
[build-system]	
requires = ["hatchling>=1.27"]	
build-backend = "hatchling.build"	
[project]	
name = "birdtally"	
version = "0.1.0"	
requires-python = ">=3.10"	
license = "MIT"	# SPDX
license-files = ["LICENSE"]	
dependencies = ["rich>=13"]	
[project.optional-dependencies]	
pretty = ["rich>=13"]	# birdtally[pretty]
[project.scripts]	
birdtally = "birdtally.cli:main"	
[dependency-groups]	# PEP 735
test = ["pytest>=8"]	
dev = [{"include-group = "test"}, {"ruff"}]	
[tool.poetry]	
packages = [{"include = "birdtally", from = "src"}]	

Specifiers & markers PEP 440 / 508	
==1.4.* !=1.5 >=1.4,<2 ~=1.4.2 (==>=1.4.2,==1.4.*)	
=== (arbitrary). Pre 2.0a1 b1 rc1, post .post1, dev .dev3, local +ubuntu1 (not on PyPI), epoch 1!.	
rich>=13 ; python_version >= "3.11" and sys_platform != "win32"	
birdtally[pretty] @ git+https://host/b.git@v0.2.0	
Poetry → PEP 440: ^1.2.3 = >=1.2.3,<2.0.0; ^0.2.3 = >=0.2.3,<0.3.0; ^0.0.3 = >=0.0.3,<0.0.4; ~1.2.3 = >=1.2.3,<1.3.0; ~1.2 = >=1.2,<1.3; 1.2.* = ==1.2.*.	

pip ↔ Poetry	
Task	pip
Install a project	pip install -e .
Add a dependency	edit pyproject.toml, reinstall
Remove one	edit, pip uninstall rich
Upgrade one	pip install -U rich
Lock	pip lock -o pylock.toml / pip-compile
Install from lock	pip install -r pylock.toml / --require-hashes -r
Sync exactly	pip-sync (pip-tools)
Dependency tree	pip show, pipdeptree
Build / publish	python -m build / twine upload
Private index	--index-url, PIP_INDEX_URL
Credentials	keyring, .netrc
Groups	pip install --group dev
Run in env	.venv/bin/python ...

Lock files shared			
	Scope	Hashes	Read by
pip freeze output	this env	no	pip
pip-compile / --require-hashes reqs	one platform	yes	pip
pylock.toml (PEP 751)	pip lock: one platform; Poetry export: all	yes	pip 26.1+ (-r), uv, PDM
poetry.lock	all platforms, groups, markers	yes	Poetry

pip install the options used most	
python -m pip install "birdtally[pretty]>=0.1"	
python -m pip install -e .	
python -m pip install -r requirements.txt -c constraints.txt	
python -m pip install --group dev	
python -m pip install --upgrade pip	
-e: PEP 660 editable; --group: PEP 735 (25.1). -U --upgrade-strategy eager --pre --no-deps --only-deps (26.2) --only-binary :all: --no-binary --index-url --extra-index-url (dependency confusion!) --find-links --no-index --dry-run --report --target --user --require-virtualenv --uploaded-prior-to (cooldown) --break-system-packages (PEP 668 override).	
Manage: pip list -o pip show pip freeze pip check pip inspect pip uninstall pip download pip wheel pip index versions pip cache dir[list purge pip debug --verbose (tags).	

requirements, constraints pip	
# requirements.txt	
--index-url https://pypi.org/simple	
rich==15.0.0 \	
-hash=sha256:33bd4ef7...	
-r base.txt	
-c constraints.txt	
-e ./libs/birdtally	
Constraints pin versions without requiring installs. Build constraints (--build-constraint, 25.3; since 26.2 -c no longer applies to builds) pin build backends. Hash mode: --require-hashes (auto when any hash present; every file needs one; pip hash). Lock: pip lock -r requirements.txt -o pylock.toml (25.1, experimental), install with pip install -r pylock.toml (26.1).	

Configuration pip	
Files ([NI, [global], [install]...), loaded in order: global /etc/pip.conf → user ~/.config/pip/pip.conf (%APPDATA%\pip\pip.ini) → site \$VIRTUAL_ENV/pip.conf → PIP_CONFIG_FILE (os.devnull = none). Precedence: command line > PIP_* env > files. PIP_INDEX_URL, PIP_NO_CACHE_DIR=1 (empty string ≠ false). pip config list[get set unset edit debug, --isolated. Credentials: keyring, .netrc, URL. Certs: system store (24.2+), --cert, REQUESTS_CA_BUNDLE.	

Error references pip	
externally-managed-environment (use a venv) · subprocess-exited-with-error · metadata-generation-failed · failed-wheel-build-for-install · failed-build-dependency-install · ssl-verification-failed · proxy-connection-failed · resolution-too-deep · ResolutionImpossible (read the conflict, loosen a pin). More detail: -v/-vvv, --log.	

Commands by task Poetry	
poetry new --name birdtally birdtally	
poetry init -n	
poetry add "rich>=13" --optional=pretty	
poetry add --group test pytest	
poetry add rich@latest	
poetry remove --group test pytest	
poetry show --tree	
poetry show --outdated	
poetry lock	
poetry lock --regenerate	
poetry install --with docs -E pretty	
poetry sync --only main,test	
poetry update rich	
poetry run pytest	
eval \$(poetry env activate)	
poetry env use 3.14	
poetry check --lock --strict	
poetry version minor	
poetry build	
poetry publish -r local	
new uses the src layout (2.1); lock keeps locked versions (2.0); sync removes extras; @latest goes past the constraint. Global: -C dir (switches dir), -P dir, -n, --no-cache, --no-plugins, -vvv.	

Groups, install / update / sync Poetry	
--with adds optional groups; non-optional groups (and all PEP 735 groups) install by default, so use --only main,test for exact sets. --without, --all-groups, --only-root, --no-root --no-directory (Docker layer).	
install	lock → env; adds missing, keeps extras
sync	lock → env; also removes what the lock lacks (2.0)
update	re-resolves within constraints, rewrites lock, installs
lock	lock only; check --lock in CI

Sources, settings, plugins Poetry	
poetry source add --priority=explicit birdwatch https://idx/simple/	
poetry add --source birdwatch birdtally	
Priority: constraint source= → only that source; else primary (any primary disables implicit PyPI) → supplemental; explicit only when named. Settings (poetry config [--local] key value --unset --list; env POETRY_+UPPER, dots/dashes → _; env > poetry.toml > config.toml > default): virtualenvs.in-project virtualenvs.create installer.re-resolve (false, 2.3) solver.min-release-age (2.4) installer.only-binary keyring.enabled http-basic.NAME → POETRY_HTTP_BASIC_NAME_PASSWORD, pypi-token.NAME → POETRY_PYPI_TOKEN_NAME. Plugins: poetry self add poetry-plugin-export (export), -shell, -bundle; per project [tool.poetry.requires-plugins]. poetry export -f requirements.txt constraints.txt pylock.toml.	

Build, check, upload publishing	
rm -rf dist ; python -m build	
twine check --strict dist/*	
unzip -l dist/*.whl ; tar tzf dist/*.tar.gz	
twine upload -r testpypi dist/*	
twine upload dist/*	
poetry build ; poetry publish --build -r testpypi	
build makes the sdist, then the wheel from it; PyPI user __token__, password pypi-.... ~/.pypirc (chmod 600; prefer keyring or TWINE_PASSWORD):	
[distutils]	
index-servers = pypi testpypi	
[pypi]	
username = __token__	
[testpypi]	
username = __token__	

Trusted Publishing job publishing	
publish:	
needs: {build}	
if: startsWith(github.ref, 'refs/tags/')	
runs-on: ubuntu-latest	
environment: pypi	
permissions:	
id-token: write	
steps:	
- uses: actions/download-artifact@v8	
with: {name: dist, path: dist/}	
- uses: pypa/gh-action-pypi-publish@release/v1	
OIDC token → /_oidc/mint-token → 15-min project token; attestations (PEP 740) by default. Poetry: build with Poetry, upload with the action.	

Release checklist & after publishing	
• metadata, SPDX license, license-files match; new PEP 440 version, no local part; changelog; tag	
• empty dist/; build; twine check --strict; install wheel + sdist in fresh venvs; tests	
• TestPyPI rehearsal with a dev version; publish from CI; check the page	
• Yank (PEP 592): skipped unless pinned ==; delete is permanent and burns filenames; archive (no uploads); new files only within 14 days of a release	
Wheel name: birdtally-1.2.0-py3-none-any.whl = name-version(-build)-python-abi-platform; e.g. cp314-cp314-manylinux_2_28_x86_64, cp310-ab13-...	