

# Python Package Project Structure

Reusable library with src layout, type hints, docs, and PyPI-ready packaging.

#python #package #library #pypi #module

PNG

PDF

Copy

</> Prompt

## Project Directory

### mypackage/

pyproject.toml All project con...

README.md

LICENSE

.gitignore

.python-version

CHANGELOG.md Version history

> src/ Source layout

> mypackage/

\_\_init\_\_.py Public API expo...

py.typed PEP 561 marker

core.py Main functional...

models.py Data classes an...

exceptions.py Custom exceptio...

> \_internal/ Private impleme...

\_\_init\_\_.py

utils.py

parsers.py

> tests/

\_\_init\_\_.py

conftest.py

test\_core.py

test\_models.py

> docs/ Documentation s...

index.md

api.md API reference

guide.md Usage guide

## Why This Structure?

The `src/` layout is the PyPA-recommended structure for libraries. It ensures your tests import the installed package, not the local source. The `py.typed` marker enables type checking for consumers. Private code lives in `_internal/` by convention.

## Key Directories

**src/mypackage/** - Package source, what users import

**\_\_init\_\_.py** - Public API—export only what users need

**\_internal/** - Implementation details, not public API

**py.typed** - Tells type checkers this package has types

## Public API in \_\_init\_\_.py

```
# src/mypackage/__init__.py
"""MyPackage - A brief description."""

from mypackage.core import process, validate
from mypackage.models import Config, Result
from mypackage.exceptions import MyPackageError

__version__ = "0.1.0"
__all__ = ["process", "validate", "Config", "Result", "MyP...
```

## Getting Started

- `uv init mypackage --lib`
- Add dependencies to `pyproject.toml`
- Export public API in `__init__.py`
- `uv pip install -e .`
- Run tests with `pytest`

## When To Use This

- Building a reusable library
- Code shared across multiple projects
- Publishing to PyPI or private index
- Library with typed API
- Internal company packages

## Trade-offs

**More structure** - Overhead for small utilities

**Editable installs** - Need `uv pip install -e .` for dev

**Docs maintenance** - Keep docs in sync with code

## Best Practices

- Export only public API from `__init__.py`
- Use `__all__` to define public symbols
- Prefix private modules with `_`
- Include `py.typed` for type checker support
- Version in one place (pyproject.toml or `__init__.py`)

## Naming Conventions

**Package** - lowercase, underscores ok (my\_package)

**Private modules** - Prefix with `_` (`_internal/`, `_utils.py`)

**Exceptions** - Suffix with Error (ValidationError)