

Instructions for Publishing HDMR-Lib on PyPI

This guide provides step-by-step instructions for publishing HDMR-Lib to the Python Package Index (PyPI).

Table of Contents

1. [Prerequisites](#)
 2. [Pre-Release Tasks](#)
 3. [Building the Package](#)
 4. [Testing on TestPyPI](#)
 5. [Publishing to PyPI](#)
 6. [Post-Release Tasks](#)
 7. [Troubleshooting](#)
-

Prerequisites

Required Accounts

1. **PyPI Account:** Register at <https://pypi.org/account/register/>
2. **TestPyPI Account** (recommended): Register at <https://test.pypi.org/account/register/>

Required Tools

Install the necessary Python packages:

```
pip install --upgrade pip
pip install build twine setuptools wheel
```

API Tokens (Recommended)

For security, use API tokens instead of passwords:

1. **PyPI:** Go to <https://pypi.org/manage/account/token/>
 - Create a new API token
 - Scope: "Entire account" for first release, then project-specific
 - Save the token securely (you'll only see it once)
2. **TestPyPI:** Go to <https://test.pypi.org/manage/account/token/>
 - Create a new API token
 - Save the token securely

Configure your tokens in `~/.pypirc`:

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

[pypi]
username = __token__
password = pypi-YOUR-API-TOKEN-HERE

[testpypi]
username = __token__
password = pypi-YOUR-TESTPYPI-TOKEN-HERE
```

Pre-Release Tasks

Step 1: Update Package Metadata

Edit `setup.py`:

```
author="Your Name",                      # ← Update this
author_email="your.email@example.com",   # ← Update this
url="https://github.com/yourusername/HDMR-Lib", # ← Update this
```

Edit `pyproject.toml`:

```
authors = [
    {name = "Your Name", email = "your.email@example.com"} # ← Update this
]

[project.urls]
Homepage = "https://github.com/yourusername/HDMR-Lib" # ← Update this
Repository = "https://github.com/yourusername/HDMR-Lib" # ← Update this
```

Step 2: Verify Package Name Availability

Check if the package name is available:

- Visit <https://pypi.org/project/hdmr-lib/>
- If the page shows "404 Not Found", the name is available ✓
- If taken, choose an alternative name:
 - `hdmr-pro`
 - `pyhdmr`
 - `hdmr-toolkit`
 - `hdmr-decomposition`

If you need to change the name, update it in:

- `setup.py` → `name="your-new-name"`
- `pyproject.toml` → `name = "your-new-name"`

Step 3: Run Tests

Ensure all tests pass:

```
# Run all tests
pytest tests/ -v

# Check test coverage (optional)
pip install pytest-cov
pytest tests/ --cov=. --cov-report=html
```

Expected output: All tests should pass ✓

Step 4: Test Local Installation

Install the package locally in development mode:

```
# Install in development mode
pip install -e .

# Test basic imports
python -c "from hdmr import HDMR; from empr import EMPR; print('✓ Imports working')"

# Check version
python -c "import hdmr_lib; print(f'Version: {hdmr_lib.__version__}')"

# Run an example
python examples/basic_usage.py
```

Step 5: Clean Previous Builds

Remove any old build artifacts:

```
# On Unix/macOS/Linux:
rm -rf build/ dist/ *.egg-info

# On Windows PowerShell:
Remove-Item -Recurse -Force build, dist, *.egg-info -ErrorAction SilentlyContinue
```

Build both source distribution and wheel:

```
python -m build
```

This creates two files in the `dist/` directory:

- `hdmr-lib-0.1.0.tar.gz` (source distribution)
- `hdmr_lib-0.1.0-py3-none-any.whl` (wheel distribution)

Verify the contents:

```
# List files in the distribution
tar -tzf dist/hdmr-lib-0.1.0.tar.gz

# Or use twine to check
twine check dist/*
```

Expected output: `Checking dist/... PASSED`

Testing on TestPyPI

Always test on TestPyPI before uploading to the real PyPI!

Step 1: Upload to TestPyPI

```
twine upload --repository testpypi dist/*
```

You'll see:

```
Uploading distributions to https://test.pypi.org/legacy/
Uploading hdmr_lib-0.1.0-py3-none-any.whl
100% _____
Uploading hdmr-lib-0.1.0.tar.gz
100% _____
```

View at:
<https://test.pypi.org/project/hdmr-lib/0.1.0/>

Step 2: Test Installation from TestPyPI

Create a fresh virtual environment:

```
# Create new environment
python -m venv test_env

# Activate it
# On Unix/macOS:
source test_env/bin/activate
# On Windows:
test_env\Scripts\activate

# Install from TestPyPI
pip install --index-url https://test.pypi.org/simple/ --extra-index-url
https://pypi.org/simple/ hdmr-lib
```

Note: `--extra-index-url https://pypi.org/simple/` allows installing dependencies from real PyPI.

Step 3: Verify Installation

```
# Test imports
python -c "from hdmr import HDMR; from empr import EMPR; print('✓ Package
installed successfully')"

# Check version
python -c "import hdmr_lib; print(f'Version: {hdmr_lib.__version__}')"

# Run a quick test
python -c "
import numpy as np
from hdmr import HDMR
tensor = np.random.rand(3, 3, 3)
model = HDMR(tensor)
result = model.decompose(order=2)
print('✓ HDMR working')
"
```

Step 4: Clean Up Test Environment

```
deactivate
rm -rf test_env # Or: Remove-Item -Recurse test_env (Windows)
```

Publishing to PyPI

Only proceed if TestPyPI testing was successful!

Upload to PyPI

```
twine upload dist/*
```

You'll see:

```
Uploading distributions to https://upload.pypi.org/legacy/
Uploading hdmr_lib-0.1.0-py3-none-any.whl
100% ━━━━━━━━━━━━━━
Uploading hdmr-lib-0.1.0.tar.gz
100% ━━━━━━━━━━━━━━
```

View at:

<https://pypi.org/project/hdmr-lib/0.1.0/>

Verify on PyPI

1. Visit <https://pypi.org/project/hdmr-lib/>
2. Check that the page displays correctly
3. Verify README renders properly
4. Check installation instructions

Test Real Installation

```
# In a fresh environment
pip install hdmr-lib

# Verify
python -c "from hdmr import HDMR; print('✓ Installed from PyPI')"
```

Post-Release Tasks

1. Tag the Release on Git

```
# Create annotated tag
git tag -a v0.1.0 -m "Release version 0.1.0 - Initial public release"

# Push tag to GitHub
git push origin v0.1.0
```

2. Create GitHub Release

1. Go to your repository on GitHub
2. Click "Releases" → "Create a new release"
3. Choose tag: **v0.1.0**

4. Release title: **HDMR-Lib v0.1.0**

5. Description:

```
## HDMR-Lib v0.1.0 - Initial Release
```

First public release of HDMR-Lib!

Features

- ⚡ HDMR (High Dimensional Model Representation) decomposition
- ⚡ EMPR (Enhanced Multivariate Products Representation) decomposition
- 🔐 Multi-backend support (NumPy, PyTorch, TensorFlow)
- 🚀 CUDA/GPU acceleration
- 📈 Sensitivity analysis tools
- 📄 Comprehensive documentation and examples

Installation

```
```bash
pip install hdmr-lib
```

## Quick Start

```
import numpy as np
from hdmr import HDMR

tensor = np.random.rand(5, 5, 5)
model = HDMR(tensor)
result = model.decompose(order=2)
```

See [README](#) for full documentation.

6. Attach distributions (optional): Upload `dist/\*.tar.gz` and `dist/\*.whl`

7. Click "Publish release"

**### 3. Update Documentation**

If you have a documentation site, update it with:

- New version number
- Installation instructions
- API reference
- Examples

**### 4. Announce the Release**

Share on:

- Twitter/X with #Python #MachineLearning #ScientificComputing
- Reddit: r/Python, r/MachineLearning
- LinkedIn

- Your blog or website
- Relevant forums or mailing lists

### ### 5. Monitor

- Check PyPI download statistics: <https://pypistats.org/packages/hdmlr-lib>
- Respond to GitHub issues
- Monitor pull requests
- Update documentation based on user feedback

---

### ## Troubleshooting

#### ### Error: "Package name already taken"

**Solution:** Choose a different name (see Step 2 in Pre-Release Tasks)

#### ### Error: "Invalid distribution file"

**Solution:** Rebuild the package:

```
```bash
rm -rf dist/ build/ *.egg-info
python -m build
```

Error: "Long description has syntax errors"

Solution: Validate README:

```
pip install readme-renderer
python -m readme_renderer README.md -o /tmp/README.html
```

Error: "Uploading package not allowed"

Possible causes:

1. Package name already exists → Choose different name
2. Wrong credentials → Check `~/.pypirc`
3. API token expired → Generate new token

Error: "403 Forbidden"

Solution:

1. Verify you're logged in: `twine upload` will prompt for credentials
2. Check API token is correct
3. Ensure you have permission to upload (for existing packages)

Import Error After Installation

Solution: Check package structure:

```
# Package should be importable as:  
python -c "import hdmr_lib" # Note: underscores, not hyphens  
  
# But installed with:  
pip install hdmr-lib # Note: hyphens in package name
```

Missing Dependencies

Solution: Verify `requirements.txt` and `setup.py` have matching dependencies

Version Updates (Future Releases)

For subsequent releases:

1. Update version in:

- o `setup.py` → `version="0.2.0"`
- o `pyproject.toml` → `version = "0.2.0"`
- o `__init__.py` → `__version__ = "0.2.0"`

2. Update CHangelog (create if doesn't exist):

```
## [0.2.0] - 2025-XX-XX  
### Added  
- New feature X  
### Fixed  
- Bug fix Y
```

3. Follow same release process:

```
rm -rf dist/ build/  
python -m build  
twine upload --repository testpypi dist/* # Test first  
twine upload dist/* # Then real PyPI  
git tag -a v0.2.0 -m "Release v0.2.0"  
git push origin v0.2.0
```

Quick Reference

Essential Commands

```
# Build package
python -m build

# Check package
twine check dist/*

# Upload to TestPyPI
twine upload --repository testpypi dist/*

# Upload to PyPI
twine upload dist/*

# Install from PyPI
pip install hdmr-lib

# Tag release
git tag -a v0.1.0 -m "Release v0.1.0"
git push origin v0.1.0
```

Important Links

- PyPI: <https://pypi.org/>
 - TestPyPI: <https://test.pypi.org/>
 - Python Packaging Guide: <https://packaging.python.org/>
 - Twine Documentation: <https://twine.readthedocs.io/>
 - Setuptools Documentation: <https://setuptools.pypa.io/>
-

Need Help?

- **Python Packaging Issues:** <https://github.com/pypa/packaging-problems/issues>
 - **PyPI Support:** <https://pypi.org/help/>
 - **Community:** Python Packaging Discord or r/Python
-

Good luck with your release! 