Configuring OpenID Connect in PyPI

In this article

Use OpenID Connect within your workflows to authenticate with PyPI.

Overview

OpenID Connect (OIDC) allows your GitHub Actions workflows to authenticate with PyPI to publish Python packages.

This guide gives an overview of how to configure PyPI to trust GitHub's OIDC as a federated identity, and demonstrates how to use this configuration in the `pypa/gh-action-pypi-publish` action to publish packages to PyPI (or other Python package repositories) without any manual API token management.

Prerequisites

- To learn the basic concepts of how GitHub uses OpenID Connect (OIDC), and its architecture and benefits, see "About security hardening with OpenID Connect."
- Before proceeding, you must plan your security strategy to ensure that access tokens are only allocated in a predictable way. To control how your cloud provider issues access tokens, you **must** define at least one condition, so that untrusted repositories can’t request access tokens for your cloud resources. For more information, see "About security hardening with OpenID Connect."

Adding the identity provider to PyPI

To use OIDC with PyPI, add a trust configuration that links each project on PyPI to each repository and workflow combination that's allowed to publish for it.

1. Sign in to PyPI and navigate to the trusted publishing settings for the project you'd like to configure. For a project named `myproject`, this will be at `https://pypi.org/manage/project/myproject/settings/publishing/`.
2. Configure a trust relationship between the PyPI project and a GitHub repository (and workflow within the repository). For example, if your GitHub repository is at `myorg/myproject` and your release workflow is defined in `release.yml` with an environment of `release`, you should use the following settings for your trusted publisher on PyPI.

   Note: Enter these values carefully. Giving the incorrect user, repository, or workflow the ability to publish to your PyPI project is equivalent to sharing an API token.

- Owner: `myorg`
- Repository name: `myproject`
- Workflow name: `release.yml`
- (Optionally) a GitHub Actions environment name: `release`

Updating your GitHub Actions workflow

Once your trusted publisher is registered on PyPI, you can update your release workflow to use trusted publishing.

The `pypa/gh-action-pypi-publish` action has built-in support for trusted publishing, which can be enabled by giving its containing job the `id-token: write` permission and omitting `username` and `password`.

The following example uses the `pypa/gh-action-pypi-publish` action to exchange an OIDC token for a PyPI API token, which is then used to upload a package's release distributions to PyPI.

YAML

```yaml
jobs:
  release-build:
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v4

      - uses: actions/setup-python@v4
        with:
          python-version: "3.x"

      - name: build release distributions
        run: |
          # NOTE: put your own distribution build steps here.
          python -m build

      - name: upload windows dists
        uses: actions/upload-artifact@v3
        with:
          name: release-dists
          path: dist/

  pypi-publish:
    runs-on: ubuntu-latest
    needs:
      - release-build
    permissions:
      id-token: write

    steps:
      - name: Retrieve release distributions
        uses: actions/download-artifact@v3
        with:
          name: release-dists
          path: dist/

      - name: Publish release distributions to PyPI
        uses: pypa/gh-action-pypi-publish@release/v1
```