Gemfury Dev Center

Installing private Python packages

Table of Contents

Your secret Repository URL

Building and uploading packages

Install packages via Pipfile

Securely install private packages via Pipfile

Install packages via command-line

Install packages via requirements.txt

Once you have signed up for a Gemfury account and uploaded a few packages, you can install them via command-line, or with requirements.txt.

Your secret Repository URL

The secret repository URL is the PyPI endpoint for your Gemfury account and packages. Do not share this URL to keep your account private.

Your PyPI URL has the following format:

```
https://TOKEN:@pypi.fury.io/USERNAME/
```

You can find your Repo URL via the dashboard following these steps.

Building and uploading packages

After using one of many guides on the web about creating a Python package, you can now upload it to Gemfury.

Use the **"+Upload"** button in the dashboard or the following command:

```
$ curl -F package=@pkg-0.0.1.tgz
https://TOKEN@push.fury.io/USERNAME/
```

Uploading packages can also be done using `distutils` commands. You'll first need to add Gemfury to your `~/.pypirc` :

```
# ~/.pypirc
[distutils]
index-servers =
    pypi
    fury

# https://www.python.org/pypi
[pypi]
username:
password:

[fury]
repository: https://pypi.fury.io/USERNAME/
username: SeCrEt-ToKeN
password:
```

You can then use the following command to build and upload your package:

```
python setup.py sdist upload -r fury
```

You can additionally use `--show-response` flag to see more upload details.

Install packages via Pipfile

Installing your packages with `pipenv` , using a `Pipfile` is now supported.

You may learn more about pipenv and how to use it, from the official website.

First, add your repository URL as a source in your Pipfile:

```
[[source]]
url = "https://pypi.fury.io/USERNAME/"
verify_ssl = true
name = "fury"
```

Then, add packages from your Gemfury account in the Pipfile as:

```
[packages]
<package_name> = "*"
```

Or, you may explicitly specify the index for each package as:

```
[packages]
<package_name> = { version = "*", index = "fury" }
```

And, afterwards, update your local environment:

```
pipenv install
```

Securely install private packages via Pipfile

If you want to install private packages using Pipfile, you may want to keep your secret token outside your `Pipfile` .

You may do so, using the following:

```
[[source]]
url = "https://${FURY_AUTH}:@pypi.fury.io/USERNAME/"
verify_ssl = true
name = "fury"
```

And then specify your secret token in the environment when invoking `pipenv` .

```
$ env FURY_AUTH=TOKEN pipenv install
```

You may also persist the token in your shell session using:

```
$ export FURY_AUTH=TOKEN
$ pipenv install
```

Install packages via command-line

Once you have your Repository URL, you can install packages via the following command:

```
pip install py-sample --extra-index-url <Repo-URL>
```

We use `--extra-index-url` to allow `pip` to keep the original Index URL. This allows `pip` to implicitly install public packages that your private package may depend on.

Install packages via requirements.txt

Add the `--extra-index-url` option at the top of your `requirements.txt` :

```
--extra-index-url <Repo-URL>
my-pkg==0.0.1
...
```

Test it out by running:

```
pip install -r ./requirements.txt
```