Quickstart: uploading, testing, pushing rel

This quickstart document walks you through setting up a self-contained pypi release upload, staging system for your Python packages.

Installing devpi client and server

We want to run the full devpi system on our laptop:

```
pip install -U devpi-web devpi-client
```

Note that the `devpi-web` package will pull in the core `devpi-server` package. If you don't war face you can just install the latter only.

initializing a basic server and index

We need to perform a couple of steps to get an index where we can upload and test packag

- start a background devpi-server at `http://localhost:3141`
- configure the client-side tool `devpi` to connect to the newly started server
- create and login a user, using as defaults your current login name and an empty passwc
- create an index and directly use it.

So let's first initialize devpi-server:

```
$ devpi-init
INFO  NOCTX Loading node info from /tmp/home/.devpi/server/.nodeinfo
INFO  NOCTX generated uuid: 446e22e0db5e41a5989fd671e98ec30b
INFO  NOCTX wrote nodeinfo to: /tmp/home/.devpi/server/.nodeinfo
INFO  NOCTX DB: Creating schema
INFO  [Wtx-1] setting password for user 'root'
INFO  [Wtx-1] created user 'root'
INFO  [Wtx-1] created root user
INFO  [Wtx-1] created root/pypi index
INFO  [Wtx-1] fswriter0: committed at 0
```

To start `devpi-server` in the background we use supervisor as an example. First we create t for it:

```
$ devpi-gen-config
It is highly recommended to use a configuration file for devpi-server, see --c
wrote gen-config/nginx-devpi-lockdown.conf
wrote gen-config/crontab
wrote gen-config/net.devpi.plist
wrote gen-config/launchd-macos.txt
```

```
wrote gen-config/nginx-devpi.conf
wrote gen-config/nginx-devpi-caching.conf
wrote gen-config/supervisor-devpi.conf
wrote gen-config/supervisord.conf
wrote gen-config/devpi.service
wrote gen-config/windows-service.txt
```

Then we start supervisord using a config which includes the generated file, see Quickstart: p stall on server/laptop for more details:

```
$ supervisord -c gen-config/supervisord.conf
```

Then we point the devpi client to it:

```
$ devpi use http://localhost:3141
using server: http://localhost:3141/ (not logged in)
no current index: type 'devpi use -l' to discover indices
venv for install/set commands: /tmp/docenv
only setting venv pip cfg, no global configuration changed
/tmp/docenv/pip.conf: no config file exists
always-set-cfg: no
```

Then we add our own “testuser”:

```
$ devpi user -c testuser password=123
user created: testuser
```

Then we login:

```
$ devpi login testuser --password=123
logged in 'testuser', credentials valid for 10.00 hours
```

And create a “dev” index, telling it to use the `root/pypi` cache as a base so that all of pypi.o will appear on that index:

```
$ devpi index -c dev bases=root/pypi
http://localhost:3141/testuser/dev?no_projects=:
  type=stage
  bases=root/pypi
  volatile=True
  acl_upload=testuser
  acl_toxresult_upload=:ANONYMOUS:
  mirror_whitelist=
  mirror_whitelist_inheritance=intersection
```

Finally we use the new index:

```
$ devpi use testuser/dev
current devpi index: http://localhost:3141/testuser/dev (logged in as testuser)
supported features: server-keyvalue-parsing
venv for install/set commands: /tmp/docenv
only setting venv pip cfg, no global configuration changed
/tmp/docenv/pip.conf: no config file exists
always-set-cfg: no
```

We are now ready to go for uploading and testing packages.

devpi install: installing a package

We can now use the `devpi` command line client to trigger a `pip install` of a pypi package u from our already running server:

```
$ devpi install pytest
--> .$ /tmp/docenv/bin/pip --version
--> .$ /tmp/docenv/bin/pip install -U pytest  [PIP_INDEX_URL=URL('http://****:*
Looking in indexes: http://testuser:****@localhost:3141/testuser/dev/+simple/
Collecting pytest
  Downloading http://localhost:3141/root/pypi/%2Bf/1d8/81c6124e08ff0/pytest-7.4
Collecting iniconfig (from pytest)
  Downloading http://localhost:3141/root/pypi/%2Bf/b6a/85871a79d2e3b/iniconfig-
Requirement already satisfied: packaging in /tmp/docenv/lib/python3.8/site-pack
Collecting pluggy<2.0,>=0.12 (from pytest)
  Downloading http://localhost:3141/root/pypi/%2Bf/d89/c696a773f8bd3/pluggy-1.3
Collecting exceptiongroup>=1.0.0rc8 (from pytest)
  Downloading http://localhost:3141/root/pypi/%2Bf/343/280667a4585d1/exceptiong
Requirement already satisfied: tomli>=1.0.0 in /tmp/docenv/lib/python3.8/site-p
Installing collected packages: pluggy, iniconfig, exceptiongroup, pytest
Successfully installed exceptiongroup-1.1.3 iniconfig-2.0.0 pluggy-1.3.0 pytest
```

The `devpi install` command configured a pip call, using the pypi-compatible `+simple/` page `testuser/dev` index for finding and downloading packages. The `pip` executable was searche and found in `docenv/bin/pip`.

Let's check that `pytest` was installed correctly:

```
$ py.test --version
pytest 7.4.2
```

You may invoke the `devpi install` command a second time which will even work when you network.

devpi upload: uploading one or more packages

We are going to use `devpi` command line tool facilities for performing uploads (you can also setup.py).

Let's verify we are logged in to the correct index:

```
$ devpi use
current devpi index: http://localhost:3141/testuser/dev (logged in as testuser)
supported features: server-keyvalue-parsing
venv for install/set commands: /tmp/docenv
only setting venv pip cfg, no global configuration changed
/tmp/docenv/pip.conf: no config file exists
always-set-cfg: no
```

Now go to the directory of a `setup.py` file of one of your projects (we assume it is named `exa` and upload your package to our `testuser/dev` index:

```
example $ devpi upload
using workdir /tmp/devpi0
pre-build: cleaning dist
--> .$ /tmp/docenv/bin/python -m build
built: dist/example-1.0.tar.gz 1kb
built: dist/example-1.0-py3-none-any.whl 1kb
file_upload of example-1.0.tar.gz to http://localhost:3141/testuser/dev/
file_upload of example-1.0-py3-none-any.whl to http://localhost:3141/testuser/d
```

There are three triggered actions:

- detection of a VCS (git/hg/svn/bazaar) repository, leading to copying all versioned files to work dir. If you are not using mercurial, the copy-step is skipped and the upload operate your source tree.
- registering the `example` release as defined in `setup.py` to our current index
- building and uploading a `gztar` formatted release file from the workdir to the current inde `setup.py` invocation under the hood).

We can now install the freshly uploaded package:

```
$ devpi install example
--> .$ /tmp/docenv/bin/pip --version
--> .$ /tmp/docenv/bin/pip install -U example  [PIP_INDEX_URL=URL('http://****
Looking in indexes: http://testuser:****@localhost:3141/testuser/dev/+simple/
Collecting example
  Downloading http://localhost:3141/testuser/dev/%2Bf/0b1/6414c21b576b1/example
Installing collected packages: example
Successfully installed example-1.0
```

This installed your just uploaded package from the `testuser/dev` index where we previously package.

Note:

`devpi upload` allows to simultaneously upload multiple different formats of your release files `sdist.zip` or `bdist_egg`. The default is `sdist.tgz`.

devpi test: testing an uploaded package

If you have a package which uses tox for testing you may now invoke:

```
$ devpi test --tox-args="-q" example  # package needs to contain tox.ini
using workdir /tmp/devpi-test0
only universal wheels supported, found example-1.0-py3-none-any.whl
received http://testuser:****@localhost:3141/testuser/dev/+f/853/34ff3d48c83ba/
unpacking /tmp/devpi-test0/downloads/example-1.0.tar.gz to /tmp/devpi-test0/tar
--> .$ /home/devpi/devpi/bin/tox --installpkg /tmp/devpi-test0/downloads/exampl
============================= test session starts ==============================
platform darwin -- Python 3.8.12, pytest-7.4.2, pluggy-1.3.0
cachedir: .tox/py/.pytest_cache
rootdir: /private/tmp/devpi-test0/targz/example-1.0
collected 1 item

test_example.py .                                                        [100%]

============================== 1 passed in 0.01s ===============================
  py: OK (10.57 seconds)
  congratulations :) (10.85 seconds)
posting tox result data to http://localhost:3141/testuser/dev/+f/853/34ff3d48c8
successfully posted tox result data
```

Here is what happened:

- devpi got the latest available version of `example` from the current index
- it unpacked it to a temp dir, found the `tox.ini` and then invoked tox, pointing it to our `example-1.0.tar.gz`, forcing all installations to go through our current `testuser/dev/+si` and instructing it to create a `json` report.
- after all tests ran, we send the `toxreport.json` to the devpi server where it will be attach to our release file.

We can verify that the test status was recorded via:

```
$ devpi list example
http://localhost:3141/testuser/dev/+f/853/34ff3d48c83ba/example-1.0.tar.gz
http://localhost:3141/testuser/dev/+f/0b1/6414c21b576b1/example-1.0-py3-none-ar
```

New in version 2.6.

With `--index` you can get the release from another index. Full URLs to another devpi-server supported.

> **Note:**
>
> Since version 2.2.0 testing of universal wheels is supported if there also is an sdist which co necessary tox.ini and tests files. Wheels typically don't contain them as they are a pure insta package.

devpi push: staging a release to another index

Once you are happy with a release file you can push it either to another devpi-managed inde outside pypi index server.

Let's create another `staging` index:

```
$ devpi index -c staging volatile=False
http://localhost:3141/testuser/staging?no_projects=:
  type=stage
  bases=
  volatile=False
  acl_upload=testuser
  acl_toxresult_upload=:ANONYMOUS:
  mirror_whitelist=
  mirror_whitelist_inheritance=intersection
```

We created a non-volatile index which means that one can not overwrite or delete release fil Volatile Indexes for more info on this setting.

We can now push the `example-1.0.tar.gz` from above to our `staging` index:

```
$ devpi push example==1.0 testuser/staging
   200 register example 1.0 -> testuser/staging
   200 store_releasefile testuser/staging/+f/853/34ff3d48c83ba/example-1.0.tar
   200 store_toxresult testuser/staging/+f/853/34ff3d48c83ba/example-1.0.tar.g
   200 store_releasefile testuser/staging/+f/0b1/6414c21b576b1/example-1.0-py3
```

This will determine all files on our `testuser/dev` index belonging to the specified `example==1` copy them to the `testuser/staging` index.

devpi push: releasing to an external index

Let's check again our current index:

```
$ devpi use
current devpi index: http://localhost:3141/testuser/dev (logged in as testuser)
supported features: server-keyvalue-parsing
venv for install/set commands: /tmp/docenv
only setting venv pip cfg, no global configuration changed
/tmp/docenv/pip.conf: no config file exists
always-set-cfg: no
```

Let's now use our `testuser/staging` index:

```
$ devpi use testuser/staging
current devpi index: http://localhost:3141/testuser/staging (logged in as testu
supported features: server-keyvalue-parsing
venv for install/set commands: /tmp/docenv
only setting venv pip cfg, no global configuration changed
/tmp/docenv/pip.conf: no config file exists
always-set-cfg: no
```

and check the test result status again:

```
$ devpi list example
http://localhost:3141/testuser/staging/+f/853/34ff3d48c83ba/example-1.0.tar.gz
http://localhost:3141/testuser/staging/+f/0b1/6414c21b576b1/example-1.0-py3-nor
```

Good, the test result status is still available after the push from the last step.

We may now decide to push this release to an external pypi-style index which we have confi
`.pypirc` file:

```
$ devpi push example-1.0 pypi:testrun
no pypirc file found at: /tmp/home/.pypirc
```

this will push all release files of the `example-1.0` release to the external `testrun` index server
dentials and the URL found in the `pypi` section in your `.pypirc`.

index inheritance re-configuration

At this point we have the `example-1.0` release and release file on both the `testuser/dev` and
`testuser/staging` indices. If we rather want to always use staging packages in our developr
can reconfigure the inheritance `bases` for `testuser/dev`:

```
$ devpi index testuser/dev bases=testuser/staging
/testuser/dev bases=testuser/staging
http://localhost:3141/testuser/dev?no_projects=:
  type=stage
  bases=testuser/staging
  volatile=True
```

```
  acl_upload=testuser
  acl_toxresult_upload=:ANONYMOUS:
  mirror_whitelist=
  mirror_whitelist_inheritance=intersection
```

If we now switch back to using `testuser/dev`:

```
$ devpi use testuser/dev
current devpi index: http://localhost:3141/testuser/dev (logged in as testuser)
supported features: server-keyvalue-parsing
venv for install/set commands: /tmp/docenv
only setting venv pip cfg, no global configuration changed
/tmp/docenv/pip.conf: no config file exists
always-set-cfg: no
```

and look at our example release files:

```
$ devpi list example
http://localhost:3141/testuser/dev/+f/853/34ff3d48c83ba/example-1.0.tar.gz
http://localhost:3141/testuser/dev/+f/0b1/6414c21b576b1/example-1.0-py3-none-ar
http://localhost:3141/testuser/staging/+f/853/34ff3d48c83ba/example-1.0.tar.gz
http://localhost:3141/testuser/staging/+f/0b1/6414c21b576b1/example-1.0-py3-nor
```

we'll see that `example-1.0.tar.gz` is contained in both indices. Let's remove the `testuser/de` release:

```
$ devpi remove -y example
About to remove the following releases and distributions
version: 1.0
  - http://localhost:3141/testuser/dev/+f/853/34ff3d48c83ba/example-1.0.tar.gz
  - http://localhost:3141/testuser/dev/+f/0b1/6414c21b576b1/example-1.0-py3-nor
  - http://localhost:3141/testuser/dev/+f/853/34ff3d48c83ba/example-1.0.tar.gz
Are you sure (yes/no)? yes (autoset from -y option)
```

If you don't specify the `-y` option you will be asked to confirm the delete operation interactive

The `example-1.0` release remains accessible through `testuser/dev` because it inherits all rel `testuser/staging` base:

```
$ devpi list example
http://localhost:3141/testuser/staging/+f/853/34ff3d48c83ba/example-1.0.tar.gz
http://localhost:3141/testuser/staging/+f/0b1/6414c21b576b1/example-1.0-py3-nor
```

Now shutdown supervisord which was started at the beginning of this tutorial:

```
$ supervisorctl -c gen-config/supervisord.conf shutdown
Shut down
```

running devpi-server permanently

If you want to configure a permanent devpi-server install, you can go to Quickstart: permane