Pkgdb2 Documentation

Release

Pierre-Yves Chibon <pingou@pingoured.fr>

May 10, 2017

Contents

The Pkgdb project is the application handling who is allowed to work on which package present in the Fedora repositories.

Resources:

- Home page
- Documentation
- Git repository
- Github mirror
- Discussion mailing-list

Contents:

CHAPTER 1

Deployment

From sources

Clone the source:

```
git clone https://git.fedorahosted.org/git/pkgdb2.git
```

Install the dependencies listed in the `requirements.txt` file.

Note: The `requirements.txt` file require flask>=0.10 but this is only required for the unit-tests and in fact flask<0.10 is **required** for python-fedora to work at the moment. The next release of python-fedora should fix this problem.

Copy the configuration files:

```
cp pkgdb2.cfg.sample pkgdb2.cfg
```

Adjust the configuration files (secret key, database URL, admin group...). See *Configuration* for detailed information about the configuration.

Create the database scheme:

```
PKGDB2_CONFIG=/path/to/pkgdb2.cfg python createdb.py
```

Set up the WSGI as described below.

From system-wide packages

Start by install pkgdb2:

```
yum install pkgdb2
```

Adjust the configuration files: `/etc/pkgdb2/pkgdb2.cfg`. See *Configuration* for detailed information about the configuration.

Find the file used to create the database:

```
rpm -ql pkgdb2 |grep createdb.py
```

Create the database scheme:

```
PKGDB2_CONFIG=/etc/pkgdb2/pkgdb2.cfg python path/to/createdb.py
```

Set up the WSGI as described below.

Set-up WSGI

Start by installing `mod_wsgi`:

```
yum install mod_wsgi
```

Then configure apache:

```
sudo vim /etc/httd/conf.d/pkgdb2.conf
```

uncomment the content of the file and adjust as desired.

Then edit the file `/usr/share/pkgdb2/pkgdb2.wsgi` and adjust as needed.

Then restart apache and you should be able to access the website on http://localhost/pkgdb

Note: Flask provides also some documentation on how to deploy Flask application with WSGI and apache.

For testing

See *Development* if you want to run pkgdb2 just to test it.

CHAPTER 2

Configuration

There are the main configuration options to set to have pkgdb2 running. These options are all present and described in the pkgdb2.cfg file.

Here are listed some configuration options specific to pkgdb2, but as a Flask application, you may also use the Flask configuration options.

The secret key

Set in the configuration file under the key `SECRET_KEY`, this is a unique, random string which is used by Flask to generate the CSRF key unique for each user.

You can easily generate one using pwgen for example to generate a 50 characters long random key

```
pwgen 50
```

The database URL

PackageDB uses SQLAlchemy has Object Relationship Mapper and thus to connect to the database. You need to provide under the key `DB_URL` in the configuration file the required information to connect to the database.

Examples URLs are:

```
DB_URL=mysql://user:pass@host/db_name
DB_URL=postgres://user:pass@host/db_name
DB_URL=sqlite:////full/path/to/database.sqlite
```

Note: The key `sqlalchemy.url` of the `alembic.ini` file should have the same value as the `DB_URL` described here.

The admin group

PackageDB relies on a group of administrator to create calendar which are then managed by people from this group. The `ADMIN_GROUP` field in the configuration file refers to the FAS group that manages this pkgdb2 instance.

Default: `ADMIN_GROUP = ['sysadmin-main', 'sysadmin-cvs']`.

Items per page

The `ITEMS_PER_PAGE` allows setting how many items should be presented per page. Items in this case may be packages, packagers or collections.

Default: `ITEMS_PER_PAGE = 50`.

Auto-approve ACLs

The `AUTO_APPROVE` lists the ACLs we handle, there are a couple which that can be automatically approved when a user requests them.

Default: `AUTO_APPROVE = ['watchcommits', 'watchbugzilla']`.

Caching configuration

Pkgdb2 uses dogplie.cache for caching. This caching is used in the extra API endpoints.

There are two configuration keys for this caching system.

`PKGDB2_CACHE_BACKEND` which specifies which backend to use for the caching

`PKGDB2_CACHE_KWARGS` which allows passing arguments to this backend

Default:

```
PKGDB2_CACHE_BACKEND = 'dogpile.cache.memcached'
PKGDB2_CACHE_KWARGS = {
    'arguments': {
        'url': "127.0.0.1:11211",
    }
}
```

More information about the possible backends and configurations can be found in the dogpile.cache documentation.

Bugzilla integration

`PKGDB2_BUGZILLA_IN_TESTS` is used to test the integration of pkgdb2 with bugzilla in the unit-tests. This setting has no effect with the actual application, as such there is no point changing it in production.

Default: `PKGDB2_BUGZILLA_IN_TESTS = False`.

`PKGDB2_BUGZILLA_NOTIFICATION` is used to change the owner of a component in bugzilla upon changes of the point of contact of a package. If False, the owner of the component in bugzilla will not reflect the change in the point of contact in packagedb. This should set to `True` in production.

Default: `PKGDB2_BUGZILLA_NOTIFICATION = False`.

`PKGDB2_BUGZILLA_URL` is the url to the bugzilla instance the packagedb application should synchronize with.

Default: `PKGDB2_BUGZILLA_URL = 'https://bugzilla.redhat.com'`.

`PKGDB2_BUGZILLA_USER` is the bugzilla user the packagedb application can log in with onto the bugzilla server set.

Default: `PKGDB2_BUGZILLA_USER = None`.

`PKGDB2_BUGZILLA_PASSWORD` is the password of the bugzilla user the packagedb application can log in with onto the bugzilla server set.

Default: `PKGDB2_BUGZILLA_PASSWORD = None`.

FAS integration

PackageDB queries a FAS instance to ensure users asking for ACL on a package are in fact already approved packagers.

`PKGDB2_FAS_URL` is the URL to the FAS instance pkgdb2 should query.

Default: `PKGDB2_FAS_URL = None`.

`PKGDB2_FAS_USER` is the FAS user pkgdb2 can log in with on the FAS server.

Default: `PKGDB2_FAS_USER = None`.

`PKGDB2_FAS_PASSWORD` is the FAS user password, pkgdb2 can log in with on the FAS server.

Default: `PKGDB2_FAS_PASSWORD = None`.

PackageDB authenticates its users with a FAS instance through FedOAuth. To do so it relies on the `flask-fas-openid` plugin which can be configured to usea different endpoint, thus allowing other project to use pkgdb2.

Here below are listed the configuration keys to use to authenticate your users against another FedOAuth instance than the default one.

`FAS_OPENID_ENDPOINT` is the URL for the FedOAuth instance.

Default: `FAS_OPENID_ENDPOINT = https://id.fedoraproject.org`

`FAS_OPENID_CHECK_CERT` is a boolean to specify if FedOAuth will verify SSL certificates.

Default: `FAS_OPENID_CHECK_CERT = True`

Notification settings

`PKGDB2_FEDMSG_NOTIFICATION` boolean specifying if the pkgdb2 application should broadcast notifications via fedmsg.

Default: `PKGDB2_FEDMSG_NOTIFICATION = True`.

`PKGDB2_EMAIL_NOTIFICATION` is a boolean specifying if the pkgdb2 application should send its notificationds by email.

Default: `PKGDB2_EMAIL_NOTIFICATION = False`.

`PKGDB2_EMAIL_TO` is a template to specify to which email the email notifications should be set. This implies there are number of aliases set redirecting from these emails to the users.

Default: `PKGDB2_EMAIL_TO = '{pkg_name}-owner@fedoraproject.org'`.

`PKGDB2_EMAIL_FROM` specifies the from field used if the notifications are sent by emails.

Default: `PKGDB2_EMAIL_FROM = 'nobody@fedoraproject.org'`.

`PKGDB2_EMAIL_SMTP_SERVER` specifies the SMTP server to use to send the notifications if they are set to be sent by emails.

Default: `PKGDB2_EMAIL_SMTP_SERVER = 'localhost'`.

Email stacktraces

PkgDB2 sends email when it faces an exception (trying to add an existing package or something alike. These emails are sent to the address set in the configuration key `MAIL_ADMIN`

Default: `MAIL_ADMIN = '<my personnal email>'`.

Packages not accessible to provenpackagers

On Fedora, some packages are restricted to their maintainers only, even members of the provenpackager group cannot access them (while they can access every other packages), this mostly when there is trademark regulations involved.

These packages are listed in the configuration under the configuration key `PKGS_NOT_PROVENPACKAGER`

Default `PKGS_NOT_PROVENPACKAGER = ['firefox', 'thunderbird', 'xulrunner']`.

Security

It is a good practice to have the cookies require a https connection for security reason. However, while developing this can prevent the authentication from working. So by default this is turned off to provide an out-of-the-box working configuration, however you will want to change it in production.

The setting to change is `SESSION_COOKIE_SECURE`.

Default `SESSION_COOKIE_SECURE = False`

To change to `SESSION_COOKIE_SECURE = True`.

Cookie conflicts

If you run multiple applications at different level of your server, by default the `path` of the cookie will be `/`, eventually leading to cookie conflict but providing a working configuration out of the box

To prevent this, adjust the `APPLICATION_ROOT` or `SESSION_COOKIE_NAME` as needed (in Fedora we used `APPLICATION_ROOT`).

Default `APPLICATION_ROOT = '/'`

Note: The application root should start with a `/` otherwise the `path` of the cookie is not set correctly

Note: More configuration information are described in the flask documentation.

CHAPTER 3

Group maintainership

PkgDB2 integrates the possibility for FAS group to get `watchcommits`, `watchbugzilla` and `commit` ACLs.

There are some requirements for the FAS group:

- name must end with `-sig`
- must be of type `pkgdb`
- must require people to be in the `packager` group
- must have a mailing list address
- must require sponsoring

One requirement for the mailing list address:

- The mailing list address given to the FAS group must have a corresponding bugzilla account

Note: If you wish to share you ACLs with a FAS group, open a new ticket on the infrastructure pagure.io tracker.

Once the group has been created in FAS, you may give it `commit`, `watchcommits` and `watchbugzilla` ACLs using the `Manage` button on the package's page.

On the manage page, you will have to click on `Add someone` and specify which ACL you want to give and on which branch.

Note: For groups, the packager name will then have the format `group::<fas_group_name>`. If you do not respect this format, pkgdb2 will refuse to add the group as co-maintainer.

CHAPTER 4

Development

Get the sources

Anonymous:

```
git clone http://git.fedorahosted.org/git/pkgdb2.git
```

Contributors:

```
git clone ssh://<FAS user>@git.fedorahosted.org/git/pkgdb2.git
```

Dependencies

The dependencies of pkgdb2 are listed in the file `requirements.txt` at the top level of the sources.

Note: if you work in a virtualenv the installation of python-fedora might fail the first time you try, just try to run the command twice, the second time it should work.

Run pkgdb for development

Copy the configuration file:

```
cp pkgdb2.cfg.sample pkgdb2.cfg
```

Adjust the configuration file (secret key, database URL, admin group...) See *Configuration* for more detailed information about the configuration.

Create the database scheme:

```
./createdb
```

Run the server:

```
./runserver
```

You should be able to access the server at http://localhost:5000

Every time you save a file, the project will be automatically restarted so you can see your change immediatly.

Get a working database

We provide a daily dump of the pkgdb2 database used in production in Fedora. You can load this database dump and use it for your tests and hacking.

You will need to set up your postgresql server first.

Once you postgresql database is running, download the latest database dump:

```
wget http://infrastructure.fedoraproject.org/infra/db-dumps/pkgdb2.dump.xz
```

Create the database itself:

```
sudo -u postgres createdb pkgdb2
```

Load the dump:

```
xzcat pkgdb2.dump.xz | sudo -u postgres psql pkgdb2
```

Please refer to `createdb --help` and `pg_restore --help` for further help on using these commands.

Coding standards

We are trying to make the code PEP8-compliant. There is a pep8 tool that can automatically check your source.

We are also inspecting the code using pylint and aim of course for a 10/10 code (but it is an assymptotic goal).

Note: both pep8 and pylint are available in Fedora via yum:

```
yum install python-pep8 pylint
```

Send patch

The easiest way to work on pkgdb2 is to make your own branch in git, make your changes to this branch, commit whenever you want, rebase on master, whenever you need and when you are done, send the patch either by email, via the trac or a pull-request (using git or github).

The workflow would therefore be something like:

```
git branch <my_shiny_feature>
git checkout <my_shiny_feature>
<work>
git commit file1 file2
<more work>
git commit file3 file4
git checkout master
git pull
git checkout <my_shiny_feature>
git rebase master
git format-patch -2
```

This will create two patch files that you can send by email to submit in the trac.

Note: You can send your patch by email to the packagedb mailing-list

Unit-tests

Pkgdb2 has a number of unit-tests providing at the moment a full coverage of the backend library (pkgdb.lib).

We aim at having a full (100%) coverage of the whole code (including the Flask application) and of course a smart coverage as in we want to check that the functions work the way we want but also that they fail when we expect it and the way we expect it.

Tests checking that function are failing when/how we want are as important as tests checking they work the way they are intended to.

`runtests.sh`, located at the top of the sources, helps to run the unit-tests of the project with coverage information using python-nose.

Note: You can specify additional arguments to the nose command used in this script by just passing arguments to the script.

For example you can specify the `-x` / `--stop` argument: *Stop running tests after the first error or failure* by just doing

```
./runtests.sh --stop
```

Each unit-tests files (located under `tests/`) can be called by alone, allowing easier debugging of the tests. For example:

```
python tests/test_collection.py
```

Similarly as for nose you can also ask that the unit-test stop at the first error or failure. For example, the command could be:

```
PKGDB2_CONFIG=tests/pkgd2b_test.cfg python -m unittest -f -v pkgdb2.tests.test_
↪collection
```

Note: In order to have coverage information you might have to install `python-coverage`

```
yum install python-coverage
```

One of our test requires network access, which means when you are working offline (when traveling for example) the test will fail. In order to have the whole test suite succeed when working offline, you can skip this test by setting an `OFFLINE` environment variable. For example:

```
OFFLINE=1 ./runtests.sh -x
```

Troubleshooting

- Login fails in development mode

  The Flask FAS extension requires a secure cookie which ensures that it is always encrypted during client/server exchanges. This makes the authentication cookie less likely to be exposed to cookie theft by eavesdropping.

  You can disable the secure cookie for testing purposes by setting the configuration key `FAS_HTTPS_REQUIRED` to False.

  Warning: Do not use this option in production as it causes major security issues

CHAPTER 5

Contributing

If you're submitting patches to pkgdb2, please observe the following:

- Check that your python code is PEP8-compliant. There is a pep8 tool that can automatically check your source.
- Check that your code doesn't break the test suite. The test suite can be run using the `runtests.sh` shell script at the top of the sources. See *Development* for more information about the test suite.
- If you are adding new code, please write tests for them in `tests/`, the `runtests.sh` script will help you to see the coverage of your code in unit-tests.
- If your change warrants a modification to the docs in `doc/` or any docstrings in `pkgdb2/` please make that modification.

Note: You have a doubt, you don't know how to do something, you have an idea but don't know how to implement it, you just have something bugging you?

Come to see us on IRC: `#fedora-apps` on irc.freenode.net or via the trac of the project.

CHAPTER 6

Contributors to packagedb

Pkgdb2 would be nothing without its contributors.

On April 25, 2017 (release 2.6.2), the list looks as follow:

Number of commits	Contributor
2814	Pierre-Yves Chibon <pingou@pingoured.fr>
60	Ralph Bean <rbean@redhat.com>
18	Till Maas <opensource@till.name>
9	Devyani Kota <divs.passion.18@gmail.com>
8	Ryan Lerch <rlerch@redhat.com>
4	Patrick Uiterwijk <puiterwijk@redhat.com>
4	trishnaguha <trishnaguha17@gmail.com>
3	Chaoyi Zha <Cydrobolt@users.noreply.github.com>
3	Johan Cwiklinski <johan@x-tnd.be>
3	Michael Cronenworth <mike@cchtml.com>
3	Mikolaj Izdebski <mizdebsk@redhat.com>
3	Ricky Elrod <ricky@elrod.me>
3	Vivek Anand <vivekanand1101@gmail.com>
2	Ryan Lerch <ryanlerch@users.noreply.github.com>
2	farhaanbukhsh <farhaan.bukhsh@gmail.com>
1	Alan Pevec <alan.pevec@redhat.com>
1	Christopher Meng <cickumqt@gmail.com>
1	Micah Denn <micah.denn@gmail.com>
1	Michael Haynes <mhaynes.linux@gmail.com>
1	Patrick Uiterwijk <patrick@puiterwijk.org>
1	Ralph Bean <ralph.bean@gmail.com>
1	Ratnadeep Debnath <ratnadeep.debnath@ibibogroup.com>
1	Subho-bcrec <subho.prp@gmail.com>
1	Vít Ondruch <v.ondruch@tiscali.cz>
1	Vít Ondruch <vondruch@redhat.com>
1	mhaynes121 <mhaynes.linux@gmail.com>

This list is generated using

```
git shortlog -s -n -e
```

The link icon (`static/link.png`) was designed by Máirín Duffy.

The package review icon (`static/fedora-infra-icon_review.png`) was designed by Micah Denn under CC-BY license.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

CHAPTER 8

FAQ

Here are some answers to frequently-asked questions from IRC and elsewhere. Got a question that isn't answered here? Try IRC, the mailing list.

CHAPTER 9

How do I...

...specify comaintainers on a new package request?

Once the package is created you can add other packagers, pseudo-users or groups:

1. Go to one of your package: https://admin.fedoraproject.org/pkgdb/package/<package_name>/
2. Click on: Manage the committers/watchers/package administrators/main contacts (any of these)
3. There is then an `Add` someone button you can use.