WFP Documentation

Release 0.0.2

Stefano Apostolico

April 16, 2013

CONTENTS

PyPPI (Python Private Package Index) is a Django application that provides a private Python Package Index. Originally forked from the DjangoPyPi project, this project gets ideas and code from chishop and DjangoPyPi2.

The reason of this project is that in big environment we need features not provided by the orginal packages:

- Ability to allow/deny package registration
- Ability to allow/deny package distribution upload
- Different levels of visibility and access to the packages
- Create Teams with proper access level
- webhooks
- improved rest interface

CHAPTER

ONE

PROJECT INFO

- Project Page: https://github.com/saxix/pyppi
- Bug Tracking: https://github.com/saxix/pyppi/issues
- Docs: https://pyppi.readthedocs.org/en/latest/

1.1 Installation & Configuration

1.1.1 Installation

PyPPi is a self-contained Django project as a pluggable application.

1.1.2 Run integrated server

The most simple way to install PyPPi is by:

```
# Make sure we run with Bash, create a virtualenv and install packages
$ bash
$ virtualenv pyppi-site
$ source pyppi-site/bin/activate
$ pip install pyppi

# Initialize our installation
$ pyppi init
# Run the server
$ pyppi start
```

That's it, we're now ready to surf to http://localhost:8000/ .

1.1.3 Configuration

By default PyPPi installs and runs from `~/.pippy` where the default configuration file `pyppi.conf.py` is created

You can use a different config file by setting the `PYPPI_CONF` environment variable or passing `--config` to `pyppi`. This must point to a valid Django settings file.

For advanced configuration please check *Settings*

1.1.4 Supervisor

For a permanent setup, simply create a supervisor configuration (you can omit the `environment` setting if you didn't specify a different project root):

```
[program:pyppi]
user = www-data
directory = /path/to/virtualenv
command = /path/to/virtualenv/bin/pyppi start -D
environment = PYPPI_CONF='/path/to/pyppi.conf.py'
```

1.2 Developers: How to upload packages

1.2.1 Uploading to your PyPPI

Assuming you are running your PyPPI site locally for now, add the following to your `~/.pypirc` file:

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

[pypi]
username:user
password:secret

[local]
username:user
password:secret
repository:http://localhost:8000/pypi/
```

Uploading a package: Python >=2.6

To push the package to the local pypi:

```
$ python setup.py register -r local sdist upload -r local
```

Uploading a package: Python <2.6

If you don't have Python 2.6 please run the command below to install the backport of the extension for multiple repositories:

```
$ easy_install -U collective.dist
```

Instead of using register and dist command, you can use `mregister` and `mupload` which are a backport of python 2.6 register and upload commands that supports multiple servers.

To push the package to the local pypi:

```
$ python setup.py mregister -r local sdist mupload -r local
```

1.3 Users: How to use this server

1.3.1 Installing a package with pip

To install your package with pip:

```
$ pip install -i http://my.pypiserver.com/simple/ <PACKAGE>
```

If you want to fall back to PyPi or another repository in the event the package is not on your new server, or in particular if you are installing a number of packages, some on your private server and some on another, you can use pip in the following manner:

```
$ pip install -i http://pyppiserver/simple/ \
  --extra-index-url=http://pypi.python.org/simple/ \
  -r requirements.txt
```

The downside is that each install of a package hosted on the repository in `--extra-index-url` will start with a call to the first repository which will fail before pip falls back to the alternative.

1.4 Settings

Here's a full list of all available settings, in alphabetical order, and their default values.

Warning: To configure the integrated server refer to *Server*

Note: Each entry must be prefixed with `PYPPI_`

1.4.1 ALLOW_VERSION_OVERWRITE

Defaut `None`

Allows you to selectively allow user with *overwrite_file (Can overwrite uploaded file)* permission to overwrite package distributions based on the version number. This is a regular expression, with the default empty string meaning 'deny all'. A common use-case example of this is to allow development versions to be overwritten, but not released versions:

```
"ALLOW_VERSION_OVERWRITE": "\\.dev.*$"
```

This will match `1.0.0.dev`, `1.0.0.dev3`, but not `1.0.0`. Note the escaping of the backslash character - this is required to conform to the json format.

1.4.2 RELEASE_UPLOAD_TO

Defaut `dists`

1.4.3 RELEASE_STORAGE_PATH

Defaut `None`

Full path to the distribution upload directory. `None` means `MEDIA_ROOT`, if you are running the integrated PyPPi server chek `MEDIA_ROOT2` The final full pathname of uploaded file will be:

<RELEASE_STORAGE_PATH> / <RELEASE_UPLOAD_TO> / <filename>

1.4.4 XMLRPC_COMMANDS

1.5 Permissions

1.5.1 `register_package` (Can register package)

1.5.2 `upload_package` (Can upload package)

1.5.3 `download_package` (Can download package)

1.5.4 `overwrite_file` (Can overwrite uploaded file)

1.6 Server

1.6.1 Start/Stop

```
$ pyppi start
```

```
$ pyppi stop
```

1.6.2 Configuration

Here's a full list of the settings of the PyPPi integrated server. To check the PyPPi applicatin settings refet to *Settings*

CONF_ROOT

Default `~/.pyppi/`

EMAIL_BACKEND

Default `django.core.mail.backends.smtp.EmailBackend`

EMAIL_HOST

Default `localhost`

EMAIL_HOST_PASSWORD

Default ‘‘

EMAIL_HOST_USER

Default ‘‘

EMAIL_HOST_PASSWORD

Default ‘‘

EMAIL_PORT

Default `25`

EMAIL_USE_TLS

Default `False`

LISTEN

Default `127.0.0.1:8000`

HAYSTACK_SEARCH_ENGINE

Default `whoosh`

See Also:

haystack, whoosh

HAYSTACK_SITECONF

Default `pyppi.server.search_sites`

See Also:

haystack

HAYSTACK_WHOOSH_PATH

Default `CONF_ROOT`/`whoosh`

See Also:

haystack, whoosh

MEDIA_ROOT

Default CONF_ROOT/media

See Also:

Django MEDIA_ROOT

PYPPI_LOG_DIR

Default CONF_ROOT/logs

SERVER_EMAIL

Default pyppi@localhost

See Also:

Django SERVER_EMAIL

STATIC_ROOT

Default CONF_ROOT/static

1.7 Changelog

This sections lists the biggest changes done on each release.

1.7.1 0.0.2

- fixed some permissions isssue

1.7.2 0.0.1

initial release