
py-deps Documentation

Release 0.2.0

Kouhei Maeda <mkouhei@palmtb.net>

June 02, 2015

1	About py-dep	3
1.1	Status	3
1.2	Requirements	3
1.3	Features	3
2	Basic usage	5
2.1	Search packages	5
2.2	Initialize	6
2.3	Generate rendering data	6
2.4	Check cache	8
3	References	11
4	Design notes	15
4.1	Memo	15
5	History	17
5.1	0.4.3 (2015-06-02)	17
5.2	0.4.2 (2015-05-31)	17
5.3	0.4.1 (2015-05-28)	17
5.4	0.4.0 (2015-05-20)	17
5.5	0.3.0 (2015-05-12)	17
5.6	0.2.0 (2015-05-10)	18
5.7	0.1.1 (2015-05-08)	18
5.8	0.1.0 (2015-05-07)	18
5.9	0.0.1 (2015-04-29)	18
6	Indices and tables	19
	Python Module Index	21

Contents:

About py-dep

The `py-dep` provides parsing the dependencies of Python packages and generating the metadata for graph. The graph data is for [NetworkX](#), [Graphviz](#), [blockdiag](#), [Linkdraw](#), etc.

1.1 Status

1.2 Requirements

- Python 2.7 over or Python 3.3 over or PyPy 2.4.0 over
- pip 1.5.6 or 6.1.1 over
- wheel 0.24.0 over
- NetworkX 1.9 over

1.3 Features

- Generating Linkdraw data (JSON and decoded JSON).
- Generating Networkx DiGraph object data.
- Cache the parsed dependencies.
- Searching packages from PyPI.

Basic usage

py-deps provides parsing the Python deps and generating graph data.

2.1 Search packages

Search packages from PyPI:

```
$ python
>>> from py_deps import Package
>>> Package.search('deps')
[{'_pypi_ordering': False,
  'name': 'anybox.recipe.sysdeps',
  'summary': 'A buildout recipe to check system dependencies',
  'version': '0.5'},
 {'_pypi_ordering': False,
  'name': 'appdynamics-bindeps-linux-x64',
  'summary': 'Dependencies for AppDynamics Python agent',
  'version': '4.0.5.0'},
 {'_pypi_ordering': False,
  'name': 'appdynamics-bindeps-linux-x86',
  'summary': 'Dependencies for AppDynamics Python agent',
  'version': '4.0.5.0'},
 {'_pypi_ordering': False,
  'name': 'appdynamics-bindeps-osx-x64',
  'summary': 'Dependencies for AppDynamics Python agent',
  'version': '4.0.5.0'},
 {'_pypi_ordering': False,
  'name': 'deps',
  'summary': 'deps discovers your Python dependencies',
  'version': '0.1.0'},
 {'_pypi_ordering': False,
  'name': 'gtkeggdeps',
  'summary': 'Interactive egg dependency browser',
  'version': '0.0.7'},
 {'_pypi_ordering': False,
  'name': 'htmldeps',
  'summary': 'Expand CSS and javascript dependency links in HTML',
  'version': '1.2.1'},
 {'_pypi_ordering': False,
  'name': 'py-deps',
  'summary': 'parsing the Python deps and generating graph data',
  'version': '0.3.0'},
```

```
{'_pypi_ordering': False,
 'name': 'pydeps',
 'summary': 'Display module dependencies',
 'version': '1.2.1'},
{'_pypi_ordering': False,
 'name': 'runestone-test-deps',
 'summary': 'This is dependencies for RSI',
 'version': '0.1'},
{'_pypi_ordering': False,
 'name': 'tl.eggdeps',
 'summary': 'Compute a dependency graph between active Python eggs.',
 'version': '0.4'},
{'_pypi_ordering': False,
 'name': 'tt.eggdeps',
 'summary': 'Compute a dependency graph between active Python eggs.',
 'version': '0.5'}]
```

2.2 Initialize

Cache the parsed dependencies into the `py-deps.pickle` on current working directory. This file format is [pickle](#).

```
$ python
>>> from py_deps import Package
>>> pkg = Package('py-deps')
```

2.2.1 Change cache file

Use `cache_name` argument:

```
>>> pkg = Package('py-deps', cache_name='some-cache.name')
```

2.2.2 Override cache forcibly

Use `update_force` argument. (default: `False`):

```
>>> pkg = Package('py-deps', update_force=True)
```

2.3 Generate rendering data

Supports follows currently.

- pretty print
- Linkdraw

2.3.1 Pretty print

```
>>> print(pkg.draw())
py-deps -> [Sphinx, setuptools, pip, wheel, tox]
setuptools -> [certifi, wincertstore, setuptools[ssl], pytest]
pip -> [pytest, virtualenv, scripttest, mock, pytest, virtualenv,
       scripttest, mock]
wheel -> [ed25519ll, keyring, argparse, pyxdg, jsonschema, pytest,
         coverage, pytest-cov]
>>>
```

2.3.2 Linkdraw

```
>>> import json
>>> json.loads(pkg.draw('linkdraw'))
{'descr': 'py-deps dependencies',
 'lines': [{u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'py-deps', u'target': u'Sphinx', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'py-deps', u'target': u'setuptools', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'py-deps', u'target': u'pip', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'py-deps', u'target': u'wheel', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'py-deps', u'target': u'tox', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'setuptools', u'target': u'certifi', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'setuptools', u'target': u'wincertstore', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'setuptools', u'target': u'setuptools____ssl',
            u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'setuptools', u'target': u'pytest', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'pip', u'target': u'pytest', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'pip', u'target': u'virtualenv', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'pip', u'target': u'scripttest', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'pip', u'target': u'mock', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'pip', u'target': u'pytest', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'pip', u'target': u'virtualenv', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'pip', u'target': u'scripttest', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'pip', u'target': u'mock', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'wheel', u'target': u'ed25519ll', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'wheel', u'target': u'keyring', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'wheel', u'target': u'argparse', u'width': u'1'},
           {u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
            u'source': u'wheel', u'target': u'pyxdg', u'width': u'1'},
```

```
{u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
 u'source': u'wheel', u'target': u'jsonschema', u'width': u'1'},
{u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
 u'source': u'wheel', u'target': u'pytest', u'width': u'1'},
{u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
 u'source': u'wheel', u'target': u'coverage', u'width': u'1'},
{u'color': u'#5F9EA0', u'descr': u'->', u'link': u'',
 u'source': u'wheel', u'target': u'pytest-cov', u'width': u'1'}],
u'nodes': [{u'color': u'', u'link': u'https://github.com/mkouhei/py-deps',
 u'name': u'py-deps', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'', u'name': u'Sphinx', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'https://bitbucket.org/pypa/setuptools',
 u'name': u'setuptools', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'https://pip.pypa.io/',
 u'name': u'pip', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'http://bitbucket.org/pypa/wheel/',
 u'name': u'wheel', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'', u'name': u'tox', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'', u'name': u'certifi', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'', u'name': u'wincertstore',
 u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'', u'name': u'setuptools____ssl',
 u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'', u'name': u'pytest', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'', u'name': u'virtualenv', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'', u'name': u'scripttest', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'', u'name': u'mock', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'', u'name': u'ed25519ll', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'', u'name': u'keyring', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'', u'name': u'argparse', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'', u'name': u'pyxdg', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'', u'name': u'jsonschema', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'', u'name': u'coverage', u'r': u'6'},
{u'color': u'#5F9EA0', u'link': u'', u'name': u'pytest-cov',
 u'r': u'6'}],
u'time': u'2015-05-08T03:52:59.542732'}
```

See also [How to use linkdraw](#).

2.3.3 NetworkX

```
>>> pkg.draw('networkx')
>>> <networkx.classes.digraph.DiGraph at 0x7fbe2311dbd0>
```

2.4 Check cache

Stores parsed dependency metadata to pickles data file. The file name is `py-deps.pickle` in default.

Listing cached data with the `list_data` method of `Container`..

```
>>> from py_deps import Container
>>> Container().list_data()
{'py-deps': [py-deps, setuptools, pip, wheel, networkx, decorator],
 (snip)}
```

Read the cached package with `read_data` method of `Container`. This method returns `Package.traced_chain`.

```
>>> Container().read_data('py-deps')  
[py-deps, setuptools, pip, wheel, networkx, decorator]
```

References

py_deps.deps module.

class `py_deps.deps.Container` (*cache_name=None*)
Bases: `object`

Package container class.

list_data ()
Return dictionary stored package metadata.

Return type `dict`

Returns packages metadata

load_cache ()
Load cache file.

read_data (*key*)
Read traced_chain data.

Return type `list`

Returns dependency chain list

Parameters **key** (*str*) – package name

save_cache ()
Save cache file.

store_data (*key, data*)
Store traced_chain data.

Parameters

- **key** (*str*) – package name
- **data** (*list*) – traced dependency chain data

class `py_deps.deps.Node` (*name, version=None, url=None*)
Bases: `object`

Node object class.

add_targets (*nodes*)
Add targets.

add_test_targets (*nodes*)
Add test targets.

remove_targets (*nodes)

Remove targets.

class `py_deps.deps.Package` (name, cache_name=None, update_force=False)

Bases: `object`

Package class.

cleanup (alldir=False)

Cleanup temporary build directory.

Parameters **alldir** (bool) – Remove all temporary directories. (default: False)

Return type None

draw (draw_type=None, decode_type='')

Generate drawing data.

Parameters

- **draw_type** (str) – [dot|blockdiag|linkdraw]
- **decode_type** (str) – ['**l**json(linkdraw)]

classmethod **search** (pkg_name, exactly=False)

Search package.

Return type list

Returns search packages

Parameters

- **pkg_name** (str) – package name.
- **exactly** (bool) – exactly match only.

trace_chain (pkg_name=None)

Trace dependency chain.

Parameters **pkg_name** (str) – package name

class `py_deps.deps.Target` (nodename, specs, extras=False)

Bases: `py_deps.deps.Node`

Target objects.

add_targets (nodes)

Add targets.

add_test_targets (nodes)

Add test targets.

remove_targets (*nodes)

Remove targets.

`py_deps.deps.u2h` (name)

Change underscore to hyphen of package name.

Return type str

Returns string replaced underscore with hyphen

Parameters **name** (str) – package name

`py_deps.graph` module.

class `py_deps.graph.Graph(chain_data)`

Bases: `object`

Graph data generate abstract class.

generate_nodes()

Generate nodes data.

class `py_deps.graph.Linkdraw(chain_data)`

Bases: `py_deps.graph.Graph`

Linkdraw object class.

generate_data()

Generate Linkdraw data.

generate_edges()

Generate edges data.

generate_nodes()

Generate nodes data.

class `py_deps.graph.Metadata`

Bases: `object`

Metadata object class.

class `py_deps.graph.Networkx(chain_data)`

Bases: `py_deps.graph.Graph`

Networkx object class.

generate_data()

Generate networkx graph data.

generate_edges()

Generate edges data.

generate_nodes()

Generate nodes data.

`py_deps.graph.pretty_print(chain_data)`

Pretty print on terminal.

Parameters `chain_data` (*list*) – List of *deps.Node*

`py_deps.graph.router(chain_data, draw_type=None, decode_type='')`

Routing drawing tool.

Design notes

4.1 Memo

purpose	setup.py	Egg	wheel (metadata.json)
requires	install_requires	requires.txt	run_requires.requires
no PyPI	dependency_links	dependency_links.txt	
optional	extras_require		extras
extra resource			run_requires.extra
test			test_requires.requires

History

5.1 0.4.3 (2015-06-02)

- Adds JSON decoder for Linkdraw.

5.2 0.4.2 (2015-05-31)

- Fixes #7 not handling the failure of python setup egg_info.
- Adds py_deps.exceptions module.
- Adds py_deps.logger module.
- Fixes issues of DistributionNotFound, InstallationError.

5.3 0.4.1 (2015-05-28)

- Adds Container.list_data method.
- Unsupports wheel format for distribution.

5.4 0.4.0 (2015-05-20)

- Searching packages from PyPI.

5.5 0.3.0 (2015-05-12)

- Supports NetworkX DiGraph objects.
- Changes to use mock instead of pip.req.RequirementSet.prepare_files.
- Coverage 98% over.

5.6 0.2.0 (2015-05-10)

- Cache the parsed dependencies.
- Fixes setting the url of node and targets.

5.7 0.1.1 (2015-05-08)

- Fixes test data of pretty_print, linkdraw.

5.8 0.1.0 (2015-05-07)

- Supports generating linkdraw data.
- Supports pip 6.1.1 over.
- Supports wheel format for distribution.
- Adds unit tests.

5.9 0.0.1 (2015-04-29)

- First release

Indices and tables

- `genindex`
- `modindex`
- `search`

p

`py_deps`, [5](#)
`py_deps.deps`, [11](#)
`py_deps.graph`, [12](#)

A

`add_targets()` (`py_deps.deps.Node` method), 11
`add_targets()` (`py_deps.deps.Target` method), 12
`add_test_targets()` (`py_deps.deps.Node` method), 11
`add_test_targets()` (`py_deps.deps.Target` method), 12

C

`cleanup()` (`py_deps.deps.Package` method), 12
`Container` (class in `py_deps.deps`), 11

D

`draw()` (`py_deps.deps.Package` method), 12

G

`generate_data()` (`py_deps.graph.Linkdraw` method), 13
`generate_data()` (`py_deps.graph.Networkx` method), 13
`generate_edges()` (`py_deps.graph.Linkdraw` method), 13
`generate_edges()` (`py_deps.graph.Networkx` method), 13
`generate_nodes()` (`py_deps.graph.Graph` method), 13
`generate_nodes()` (`py_deps.graph.Linkdraw` method), 13
`generate_nodes()` (`py_deps.graph.Networkx` method), 13
`Graph` (class in `py_deps.graph`), 12

L

`Linkdraw` (class in `py_deps.graph`), 13
`list_data()` (`py_deps.deps.Container` method), 11
`load_cache()` (`py_deps.deps.Container` method), 11

M

`Metadata` (class in `py_deps.graph`), 13

N

`Networkx` (class in `py_deps.graph`), 13
`Node` (class in `py_deps.deps`), 11

P

`Package` (class in `py_deps.deps`), 12
`pretty_print()` (in module `py_deps.graph`), 13
`py_deps` (module), 5

`py_deps.deps` (module), 11
`py_deps.graph` (module), 12

R

`read_data()` (`py_deps.deps.Container` method), 11
`remove_targets()` (`py_deps.deps.Node` method), 11
`remove_targets()` (`py_deps.deps.Target` method), 12
`router()` (in module `py_deps.graph`), 13

S

`save_cache()` (`py_deps.deps.Container` method), 11
`search()` (`py_deps.deps.Package` class method), 12
`store_data()` (`py_deps.deps.Container` method), 11

T

`Target` (class in `py_deps.deps`), 12
`trace_chain()` (`py_deps.deps.Package` method), 12

U

`u2h()` (in module `py_deps.deps`), 12