

---

# py-deps Documentation

*Release 1.0.0*

Kouhei Maeda <mkouhei@palmtb.net>

May 16, 2020



<b>1</b>	<b>About py-dep</b>	<b>3</b>
1.1	Status . . . . .	3
1.2	Requirements . . . . .	3
1.3	Features . . . . .	3
<b>2</b>	<b>Basic usage</b>	<b>5</b>
2.1	Search packages . . . . .	5
2.2	Show latest version . . . . .	6
2.3	Initialize . . . . .	6
2.4	Generate rendering data . . . . .	7
2.5	Check cache . . . . .	8
<b>3</b>	<b>References</b>	<b>11</b>
<b>4</b>	<b>History</b>	<b>17</b>
4.1	1.0.0 (2020-05-16) . . . . .	17
4.2	0.5.5 (2015-08-19) . . . . .	17
4.3	0.5.4 (2015-07-22) . . . . .	17
4.4	0.5.3 (2015-07-19) . . . . .	17
4.5	0.5.2 (2015-07-15) . . . . .	18
4.6	0.5.1 (2015-07-12) . . . . .	18
4.7	0.5.0 (2015-06-22) . . . . .	18
4.8	0.4.6 (2015-06-11) . . . . .	18
4.9	0.4.5 (2015-06-07) . . . . .	18
4.10	0.4.4 (2015-06-03) . . . . .	18
4.11	0.4.3 (2015-06-02) . . . . .	18
4.12	0.4.2 (2015-05-31) . . . . .	19
4.13	0.4.1 (2015-05-28) . . . . .	19
4.14	0.4.0 (2015-05-20) . . . . .	19
4.15	0.3.0 (2015-05-12) . . . . .	19
4.16	0.2.0 (2015-05-10) . . . . .	19
4.17	0.1.1 (2015-05-08) . . . . .	19
4.18	0.1.0 (2015-05-07) . . . . .	19
4.19	0.0.1 (2015-04-29) . . . . .	20
<b>5</b>	<b>Indices and tables</b>	<b>21</b>

<b>Python Module Index</b>	<b>23</b>
<b>Index</b>	<b>25</b>

Contents:



# CHAPTER 1

---

## 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](#), [Linkdraw](#), etc.

### 1.1 Status



### 1.2 Requirements

- Python 3.8
- setuptools 46.1.0 over
- pip 20.0. over
- NetworkX 2.4 over
- pylibmc 1.6.1 over (optional)

### 1.3 Features

- Generating Linkdraw data (JSON and decoded JSON).

- Generating Networkx DiGraph object data.
- Cache the parsed dependencies.
- Searching packages from PyPI.



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

## 2.1 Search packages

Search packages from PyPI.:

```
$ python
>>> from py_deps.deps import search
>>> 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',
```

(continues on next page)

(continued from previous page)

```
'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 Show latest version

Retrieve latest package from PyPI.:

```
$ python
>>> from py_deps.deps import latest_version
>>> latest_version('deps')
'0.1.0'
```

## 2.3 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')
```

`py-deps` retrieve latest version from PyPI without `version` argument as above. Specify version use `version` argument.:

```
>>> pkg = Package('py-deps', version='0.5.5')
>>> pkg.version
0.5.5
```

### 2.3.1 Change cache file

Use `cache_name` argument.:

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

### 2.3.2 Override cache forcibly

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

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

### 2.3.3 Changes the cache backend to Memcached

Installing `libmemcached-dev` package and `pylibmc`.:

```
$ sudo apt-get install libmemcached-dev
$ . /path/to/venv/bin/activate
(venv)$ cd /path/to/py-deps
(venv)$ python setup.py memcache
```

Use `servers` argument. The argument syntax follows `pylibmc.Client`.:

```
>>> pkg = Package('py-deps', servers=['127.0.0.1:11211'])
```

## 2.4 Generate rendering data

Supports follows currently.

- pretty print
- Linkdraw

### 2.4.1 Pretty print

```
>>> print(pkg.draw())
py-deps -> [networkx, pip, setuptools, wheel]
networkx -> [decorator]
>>>
```

### 2.4.2 Linkdraw

```
>>> import pprint
>>> pp = pprint.PrettyPrinter(width=120)
>>> pp.pprint(pkg.draw('linkdraw'))
{'descr': 'py-deps dependencies',
 'lines': [{ 'color': '#d1e0fa', 'descr': '->', 'link': '', 'source': 'py-deps',
             ↪ 'target': 'networkx', 'width': '1'},
            { 'color': '#a4c1f4', 'descr': '->', 'link': '', 'source': 'networkx',
             ↪ 'target': 'decorator', 'width': '1'},
```

(continues on next page)

(continued from previous page)

```
{'color': '#d1e0fa', 'descr': '->', 'link': '', 'source': 'py-deps',  
→ 'target': 'pip', 'width': '1'},  
{'color': '#d1e0fa', 'descr': '->', 'link': '', 'source': 'py-deps',  
→ 'target': 'setuptools', 'width': '1'},  
{'color': '#d1e0fa', 'descr': '->', 'link': '', 'source': 'py-deps',  
→ 'target': 'wheel', 'width': '1'}],  
'nodes': [{ 'color': '#d1e0fa',  
            'depth': 0,  
            'link': 'https://github.com/mkouhei/py-deps',  
            'name': 'py-deps',  
            'r': '6',  
            'version': '0.5.5'},  
{'color': '#a4c1f4',  
            'depth': 1,  
            'link': 'http://networkx.github.io/',  
            'name': 'networkx',  
            'r': '6',  
            'version': '2.4'},  
{'color': '#76a1ef',  
            'depth': 2,  
            'link': 'https://github.com/micheles/decorator',  
            'name': 'decorator',  
            'r': '6',  
            'version': '4.4.2'},  
{'color': '#a4c1f4', 'depth': 1, 'link': 'https://pip.pypa.io/', 'name':  
→ 'pip', 'r': '6', 'version': '20.1'},  
{'color': '#a4c1f4',  
            'depth': 1,  
            'link': 'https://github.com/pypa/setuptools',  
            'name': 'setuptools',  
            'r': '6',  
            'version': '46.3.1'},  
{'color': '#a4c1f4',  
            'depth': 1,  
            'link': 'https://github.com/pypa/wheel',  
            'name': 'wheel',  
            'r': '6',  
            'version': '0.34.2'}],  
'time': '2020-05-16T14:06:52.790139'}
```

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

## 2.4.3 NetworkX

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

## 2.5 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.cache import backend
>>> backend().list_data()
{'py-deps', '0.5.5': [py-deps]}
(snip)
```

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

```
>>> Container().read_data(('py-deps', '0.5.5'))
[py-deps]
```



`py_deps.deps` module.

`py_deps.deps.SUFFIX = '-py_deps'`  
suffix of temporary directory name

`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.deps.search(pkg_name, exactly=False)`  
Search package.

**Return type** list

**Returns** search packages

**Parameters**

- **pkg\_name** (*str*) – package name.
- **exactly** (*bool*) – exactly match only.

`py_deps.deps.latest_version(pkg_name)`  
Retrieve latest version.

**Return type** str

**Returns** latest version

**Parameters** **pkg\_name** (*str*) – package name.

`py_deps.deps.create_nodes(package_names, depth=0)`  
Show information about installed package.

```
class py_deps.deps.Package (name, version=None, update_force=False, **kwargs)
    Bases: object

    Package class.

    index_url = 'https://pypi.python.org/simple'
        index_url

    name = None
        package name

    cleanup (alldir=False)
        Cleanup temporary build directory.

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

        Return type None

    install ()
        Install packages to build_dir.

    draw (draw_type=None, link_prefix=None)
        Generate drawing data.

        Parameters draw_type (str) – [dot|blockdiag|linkdraw]

class py_deps.deps.Node (name, version=None, url=None, requires=None, depth=0)
    Bases: object

    Node object class.

    name = None
        name

    version = None
        version

    url = None
        project url

    requires = None
        requires

    targets = None
        targets

    test_targets = None
        test targets

    depth = None
        base dependency depth level

py_deps.graph module.

py_deps.graph.router (package, draw_type=None, link_prefix=None)
    Routing drawing tool.

py_deps.graph.edge_key (source_node, target_node)
    Edge source_node -> target_node key.

py_deps.graph.generate_data (chain_data, func)
    Generate dependencies graph.

py_deps.graph.pretty_print (chain_data)
    Pretty print on terminal.
```



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

**class** `py_deps.graph.Graph` (*package*, *link\_prefix=None*)  
Bases: `object`

Graph data generate abstract class.

**class** `py_deps.graph.Linkdraw` (*package*, *link\_prefix=None*)  
Bases: `py_deps.graph.Graph`

Linkdraw object class.

**generate\_data** ()  
Generate Linkdraw data.

**class** `py_deps.graph.Networkx` (*package*, *link\_prefix=None*)  
Bases: `py_deps.graph.Graph`

Networkx object class.

**generate\_edges** ()  
Generate edges data.

**generate\_data** ()  
Generate networkx graph data.

`py_deps.graph.color` (*depth*)  
Color by depth level.

**Return type** `str`

**Returns** hex color code based blue.

**Parameters** `depth` (*int*) – dependency level

`py_deps.cache` module.

`py_deps.cache.backend` (*\*\*kwargs*)  
Specify cache backend.

**Return type** `py_deps.cache.Container`

**Returns** Pickle object or Memcached object.

**Parameters** `kwargs` – parameters

**servers** Memcached servers (required in Memcached)

Required when Memcached. Using Pickle in default without `servers`.

**username** Memcached SASL username (optional)

**password** Memcached SASL password (optional)

**cache\_name** Pickle filename (default, optional)

**class** `py_deps.cache.Container` (*cache\_name=None*)  
Bases: `object`

Package container class.

**store\_data** (*key*, *data*)  
Store `traced_chain` data.

**read\_data** (*key*)  
Read `traced_chain` data.

**Return type** list

**Returns** dependency chain list

**Parameters** **key** (*tuple*) – package name, version

**list\_data** ()

Return dictionary stored package metadata.

**Return type** dict

**Returns** packages metadata

**class** `py_deps.cache.Pickle` (*cache\_name=None*)

Bases: `py_deps.cache.Container`

Cache backend is Pickle.

**default\_cache\_name** = `'py-deps.pickle'`

default cache file name

**load\_cache** ()

Load cache file.

**save\_cache** ()

Save cache file.

**store\_data** (*key, data*)

Store traced\_chain data.

**Parameters**

- **key** (*tuple*) – package name, version
- **data** (*list*) – traced dependency chain data

**list\_data** ()

Return dictionary stored package metadata.

**Return type** dict

**Returns** packages metadata

**read\_data** (*key*)

Read traced\_chain data.

**Return type** list

**Returns** dependency chain list

**Parameters** **key** (*tuple*) – package name, version

**class** `py_deps.cache.Memcached` (*servers=None, username=None, password=None, behaviors=None*)

Bases: `py_deps.cache.Container`

Cache backend is Memcached.

**store\_data** (*key, data*)

Store traced\_chain data.

**Parameters**

- **key** (*tuple*) – package name, version
- **data** (*list*) – traced dependency chain data

**list\_data()**

Return dictionary stored package metadata.

**Return type** dict

**Returns** packages metadata

**read\_data(key)**

Read traced\_chain data.

**Return type** list

**Returns** dependency chain list

**Parameters** **key** (*tuple*) – package name, version

py\_deps.exceptions module.

**exception** py\_deps.exceptions.**Error**

Bases: Exception

Base error class.

**with\_traceback()**

Exception.with\_traceback(tb) – set self.\_\_traceback\_\_ to tb and return self.

**exception** py\_deps.exceptions.**NotFound**

Bases: [py\\_deps.exceptions.Error](#)

Not Found.

**with\_traceback()**

Exception.with\_traceback(tb) – set self.\_\_traceback\_\_ to tb and return self.

**exception** py\_deps.exceptions.**BrokenPackage**

Bases: [py\\_deps.exceptions.Error](#)

BrokenPackage.

**with\_traceback()**

Exception.with\_traceback(tb) – set self.\_\_traceback\_\_ to tb and return self.

**exception** py\_deps.exceptions.**InvalidMetadata**

Bases: [py\\_deps.exceptions.Error](#)

Invalid package metadata.

**with\_traceback()**

Exception.with\_traceback(tb) – set self.\_\_traceback\_\_ to tb and return self.

**exception** py\_deps.exceptions.**BackendFailure**

Bases: [py\\_deps.exceptions.Error](#)

PyPI service down.

**with\_traceback()**

Exception.with\_traceback(tb) – set self.\_\_traceback\_\_ to tb and return self.



### 4.1 1.0.0 (2020-05-16)

- Updates dependencies.
- Refactors.
- Supports Python 3.8 only.

### 4.2 0.5.5 (2015-08-19)

- Adds TimeoutError, ConnectionRefusedError / socket.error exceptions.
- Adds error handling the PyPI service down.
- Changes Sphinx theme to sphinx\_rtd\_theme.
- Adds counter of the each depth.

### 4.3 0.5.4 (2015-07-22)

- Adds latest\_version function.
- Adds link\_prefix for overriding node link.

### 4.4 0.5.3 (2015-07-19)

- Adds exception InvalidMetadata type.

## 4.5 0.5.2 (2015-07-15)

- Fixes duplicated line of linkdraw.
- Changes linkdraw colors by dependencies depth.
- Adds depth property to graph nodes.
- Adds parsing the package dependency depth.

## 4.6 0.5.1 (2015-07-12)

- Changes Package.search method to a function.
- Fixes infinity loop trace\_chain.
- Fixes *None* redundant second argument of *dict.get()*.
- Fixes *len()* used to check if collection has items.
- Fixes old-style string formatting.

## 4.7 0.5.0 (2015-06-22)

- Supports memcached as the backend of cache.

## 4.8 0.4.6 (2015-06-11)

- Fixes not control the version of package correctly.

## 4.9 0.4.5 (2015-06-07)

- Adds disable time, descr for Linkdraw.

## 4.10 0.4.4 (2015-06-03)

- Removes debug print.

## 4.11 0.4.3 (2015-06-02)

- Adds JSON decoder for Linkdraw.

## 4.12 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.

## 4.13 0.4.1 (2015-05-28)

- Adds Container.list\_data method.
- Unsupports wheel format for distribution.

## 4.14 0.4.0 (2015-05-20)

- Searching packages from PyPI.

## 4.15 0.3.0 (2015-05-12)

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

## 4.16 0.2.0 (2015-05-10)

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

## 4.17 0.1.1 (2015-05-08)

- Fixes test data of pretty\_print, linkdraw.

## 4.18 0.1.0 (2015-05-07)

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

## 4.19 0.0.1 (2015-04-29)

- First release



## CHAPTER 5

---

### Indices and tables

---

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



### p

- `py_deps`, [5](#)
- `py_deps.cache`, [13](#)
- `py_deps.deps`, [11](#)
- `py_deps.exceptions`, [15](#)
- `py_deps.graph`, [12](#)



## B

backend() (in module *py\_deps.cache*), 13  
BackendFailure, 15  
BrokenPackage, 15

## C

cleanup() (*py\_deps.deps.Package* method), 12  
color() (in module *py\_deps.graph*), 13  
Container (class in *py\_deps.cache*), 13  
create\_nodes() (in module *py\_deps.deps*), 11

## D

default\_cache\_name (*py\_deps.cache.Pickle* attribute), 14  
depth (*py\_deps.deps.Node* attribute), 12  
draw() (*py\_deps.deps.Package* method), 12

## E

edge\_key() (in module *py\_deps.graph*), 12  
Error, 15

## G

generate\_data() (in module *py\_deps.graph*), 12  
generate\_data() (*py\_deps.graph.Linkdraw* method), 13  
generate\_data() (*py\_deps.graph.Networkx* method), 13  
generate\_edges() (*py\_deps.graph.Networkx* method), 13  
Graph (class in *py\_deps.graph*), 13

## I

index\_url (*py\_deps.deps.Package* attribute), 12  
install() (*py\_deps.deps.Package* method), 12  
InvalidMetadata, 15

## L

latest\_version() (in module *py\_deps.deps*), 11  
Linkdraw (class in *py\_deps.graph*), 13

list\_data() (*py\_deps.cache.Container* method), 14  
list\_data() (*py\_deps.cache.Memcached* method), 14  
list\_data() (*py\_deps.cache.Pickle* method), 14  
load\_cache() (*py\_deps.cache.Pickle* method), 14

## M

Memcached (class in *py\_deps.cache*), 14

## N

name (*py\_deps.deps.Node* attribute), 12  
name (*py\_deps.deps.Package* attribute), 12  
Networkx (class in *py\_deps.graph*), 13  
Node (class in *py\_deps.deps*), 12  
NotFound, 15

## P

Package (class in *py\_deps.deps*), 11  
Pickle (class in *py\_deps.cache*), 14  
pretty\_print() (in module *py\_deps.graph*), 12  
py\_deps (module), 5  
py\_deps.cache (module), 13  
py\_deps.deps (module), 11  
py\_deps.exceptions (module), 15  
py\_deps.graph (module), 12

## R

read\_data() (*py\_deps.cache.Container* method), 13  
read\_data() (*py\_deps.cache.Memcached* method), 15  
read\_data() (*py\_deps.cache.Pickle* method), 14  
requires (*py\_deps.deps.Node* attribute), 12  
router() (in module *py\_deps.graph*), 12

## S

save\_cache() (*py\_deps.cache.Pickle* method), 14  
search() (in module *py\_deps.deps*), 11  
store\_data() (*py\_deps.cache.Container* method), 13  
store\_data() (*py\_deps.cache.Memcached* method), 14  
store\_data() (*py\_deps.cache.Pickle* method), 14

SUFFIX (*in module py\_deps.deps*), 11

## T

targets (*py\_deps.deps.Node attribute*), 12

test\_targets (*py\_deps.deps.Node attribute*), 12

## U

u2h () (*in module py\_deps.deps*), 11

url (*py\_deps.deps.Node attribute*), 12

## V

version (*py\_deps.deps.Node attribute*), 12

## W

with\_traceback () (*py\_deps.exceptions.BackendFailure  
method*), 15

with\_traceback () (*py\_deps.exceptions.BrokenPackage  
method*), 15

with\_traceback () (*py\_deps.exceptions.Error  
method*), 15

with\_traceback () (*py\_deps.exceptions.InvalidMetadata  
method*), 15

with\_traceback () (*py\_deps.exceptions.NotFound  
method*), 15