
app*versionDocumentation*

Release

Alisue

Jul 14, 2017

Contents

1	app_version	1
1.1	Installation	1
1.2	Usage	1
2	API documents	3
2.1	app_version Package	3
2.1.1	app_version Package	3
2.1.2	core Module	5
3	Indices and tables	7
	Python Module Index	9

CHAPTER 1

app_version

Do you write the version information on `setup.py` and `__init__.py`?

This tiny application allow you to access version information of `setup.py` from `__init__.py`.

Based on [this post](#), I wrote this tiny application for convinience.

Check [online documentation](#) for more details.

Installation

Use `pip` like:

```
$ pip install app_version
```

Usage

The following code is an example `__init__.py`.

```
from app_version import get_versions
__version__, VERSION = get_versions('your app name')
```

Then you can access the version string with `__version__` and version tuple with `VERSION`. The version tuple is useful for comparing versions like

```
>>> VERSION = (0, 1, 2)
>>> VERSION > (0, 1, 0)
True
>>> VERSION > (0, 1, 1)
True
>>> VERSION > (0, 1, 2)
False
```


app_version Package

app_version Package

`app_version.get_versions` (*name*, *default_string*='Please install this application with setup.py', *default_tuple*=(0, 0, 0), *allow_ambiguous*=True)

Get string and tuple versions from installed package information

It will return *default_string* and *default_tuple* values when the named package is not installed.

Parameters

- **name** (*string*) – An application name used to install via `setuptools`.
- **default** (*string*) – A default returning value used when the named application is not installed yet
- **default_tuple** (*tuple*) – A default returning value used when the named application is not installed yet
- **allow_ambiguous** (*boolean*) – True for allowing ambiguous version information.

Returns A version string and version tuple

Return type tuple

Examples

```
>>> import re
>>> v1, v2 = get_versions('app_version', allow_ambiguous=True)
>>> isinstance(v1, str)
True
>>> isinstance(v2, tuple)
True
```

```
>>> get_versions('distribution_which_is_not_installed')
('Please install this application with setup.py', (0, 0, 0))
```

`app_version.get_tuple_version` (*name*, *default*=(0, 0, 0), *allow_ambiguous*=True)

Get tuple version from installed package information for easy handling.

It will return default value when the named package is not installed.

Parameters

- **name** (*string*) – An application name used to install via `setuptools`.
- **default** (*tuple*) – A default returning value used when the named application is not installed yet
- **allow_ambiguous** (*boolean*) – True for allowing ambiguous version information.

Returns A version tuple

Return type string

Examples

```
>>> v = get_tuple_version('app_version', allow_ambiguous=True)
>>> len(v) >= 3
True
>>> isinstance(v[0], int)
True
>>> isinstance(v[1], int)
True
>>> isinstance(v[2], int)
True
>>> get_tuple_version('distribution_which_is_not_installed')
(0, 0, 0)
```

`app_version.get_string_version` (*name*, *default*='Please install this application with setup.py', *allow_ambiguous*=True)

Get string version from installed package information.

It will return default value when the named package is not installed.

Parameters

- **name** (*string*) – An application name used to install via `setuptools`.
- **default** (*string*) – A default returning value used when the named application is not installed yet
- **allow_ambiguous** (*boolean*) – True for allowing ambiguous version information. Turn this argument to False if `get_string_version` report wrong version.

Returns A version string or not found message (default)

Return type string

Examples

```

>>> import re
>>> v = get_string_version('app_version', allow_ambiguous=True)
>>> re.match('^\d+\.\d+\.\d+', v) is not None
True
>>> get_string_version('distribution_which_is_not_installed')
'Please install this application with setup.py'

```

core Module

app_{version}

Get version information from setup.py via pkg_resources.

The concept is taken from this answer

<http://stackoverflow.com/a/17638236>

written by Martijn Pietersp

app_{version}.core.get_string_version(name, default='Please install this application with setup.py', allow_ambiguous=True)

Get string version from installed package information.

It will return default value when the named package is not installed.

Parameters

- **name** (*string*) – An application name used to install via setuptools.
- **default** (*string*) – A default returning value used when the named application is not installed yet
- **allow_ambiguous** (*boolean*) – True for allowing ambiguous version information. Turn this argument to False if get_string_version report wrong version.

Returns A version string or not found message (default)

Return type string

Examples

```

>>> import re
>>> v = get_string_version('app_version', allow_ambiguous=True)
>>> re.match('^\d+\.\d+\.\d+', v) is not None
True
>>> get_string_version('distribution_which_is_not_installed')
'Please install this application with setup.py'

```

app_{version}.core.get_tuple_version(name, default=(0, 0, 0), allow_ambiguous=True)

Get tuple version from installed package information for easy handling.

It will return default value when the named package is not installed.

Parameters

- **name** (*string*) – An application name used to install via setuptools.
- **default** (*tuple*) – A default returning value used when the named application is not installed yet

- **allow_ambiguous** (*boolean*) – True for allowing ambiguous version information.

Returns A version tuple

Return type string

Examples

```
>>> v = get_tuple_version('app_version', allow_ambiguous=True)
>>> len(v) >= 3
True
>>> isinstance(v[0], int)
True
>>> isinstance(v[1], int)
True
>>> isinstance(v[2], int)
True
>>> get_tuple_version('distribution_which_is_not_installed')
(0, 0, 0)
```

`app_version.core.get_versions` (*name*, *default_string*='Please install this application with `setup.py`', *default_tuple*=(0, 0, 0), *allow_ambiguous*=True)
Get string and tuple versions from installed package information

It will return `default_string` and `default_tuple` values when the named package is not installed.

Parameters

- **name** (*string*) – An application name used to install via `setuptools`.
- **default** (*string*) – A default returning value used when the named application is not installed yet
- **default_tuple** (*tuple*) – A default returning value used when the named application is not installed yet
- **allow_ambiguous** (*boolean*) – True for allowing ambiguous version information.

Returns A version string and version tuple

Return type tuple

Examples

```
>>> import re
>>> v1, v2 = get_versions('app_version', allow_ambiguous=True)
>>> isinstance(v1, str)
True
>>> isinstance(v2, tuple)
True
>>> get_versions('distribution_which_is_not_installed')
('Please install this application with setup.py', (0, 0, 0))
```

CHAPTER 3

Indices and tables

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

a

app_version, 3

app_version.core, 5

A

app_version (module), 3
app_version.core (module), 5

G

get_string_version() (in module app_version), 4
get_string_version() (in module app_version.core), 5
get_tuple_version() (in module app_version), 4
get_tuple_version() (in module app_version.core), 5
get_versions() (in module app_version), 3
get_versions() (in module app_version.core), 6