
metaextract Documentation

Release 1.0.7

Thomas Bechtold

Nov 04, 2020

Contents

1	Installation	3
2	Usage	5
3	API documentation	7
3.1	The <code>metaextract.utils</code> Module	7
4	Indices and tables	9

metaextract is a tool to collect metadata about a python module. For example you may have a sdist tarball from the [Python Package Index](#) and you want to know it's dependencies. metaextract can collect theses dependencies. The tool was first developed in [py2pack](#) but is now it's own module to be useful for others, too.

CHAPTER 1

Installation

To install metaextract from the [Python Package Index](#), simply:

```
$ pip install metaextract
```


CHAPTER 2

Usage

To extract the metadata for a python module using `setup.py`, do:

```
$ metaextract my-archive-file.tar.gz
```

This will print a json blob to stdout which contains i.e. `install_requires`, `extras_require` and `friends` extracted from the given archive file.

If you already have some source code available (i.e. a git checkout) for some project you can also run the `setup.py` file with the `metaextract distutils` command:

```
$ python setup.py --command-packages=metaextract metaextract
```

This will print the metadata as json. If you want to write the data to a file, do:

```
$ python setup.py --command-packages=metaextract metaextract -o output-file
```


3.1 The `metaextract.utils` Module

`metaextract.utils.from_archive` (*archive_filename*, *py_interpreter*=`'/home/docs/checkouts/readthedocs.org/user_builds/m'`)
extract metadata from a given sdist archive file

Parameters

- **`archive_filename`** – a sdist archive file
- **`py_interpreter`** – The full path to the used python interpreter

Returns a json blob with metadata

CHAPTER 4

Indices and tables

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