
lazr.uri Documentation

Release 1.0.6

LAZR Developers

Oct 31, 2021

CONTENTS

1	The URI class	3
2	Finding URIs in Text	5
2.1	NEWS for lazr.uri	5

The lazr.uri package includes code for parsing and dealing with URIs.

```
>>> import lazr.uri
>>> print('VERSION:', lazr.uri.__version__)
VERSION: ...
```


THE URI CLASS

```
>>> from lazr.uri import URI
>>> uri1 = URI('http://localhost/foo/bar?123')
>>> uri2 = URI('http://localhost/foo/bar/baz')
>>> uri1.contains(uri2)
True
```

These next two are equivalent, so the answer should be True, even though the “outside” one is shorter than the “inside” one.

```
>>> uri1 = URI('http://localhost/foo/bar/')
>>> uri2 = URI('http://localhost/foo/bar')
>>> uri1.contains(uri2)
True
```

The next two are exactly the same. We consider a url to be inside itself.

```
>>> uri1 = URI('http://localhost/foo/bar/')
>>> uri2 = URI('http://localhost/foo/bar/')
>>> uri1.contains(uri2)
True
```

In the next case, the string of url2 starts with the string of url1. But, because url2 continues within the same path step, url2 is not inside url1.

```
>>> uri1 = URI('http://localhost/foo/ba')
>>> uri2 = URI('http://localhost/foo/bar')
>>> uri1.contains(uri2)
False
```

Here, url2 is url1 plus an extra path step. So, url2 is inside url1.

```
>>> uri1 = URI('http://localhost/foo/bar/')
>>> uri2 = URI('http://localhost/foo/bar/baz')
>>> uri1.contains(uri2)
True
```

Once the URI is parsed, its parts are accessible.

```
>>> uri = URI('https://fish.tree:8666/blee/blah')
>>> uri.scheme
'https'
```

(continues on next page)

(continued from previous page)

```
>>> uri.host
'fish.tree'
>>> uri.port
'8666'
>>> uri.authority
'fish.tree:8666'
>>> uri.path
'/blee/blah'
```

```
>>> uri = URI('https://localhost/blee/blah')
>>> uri.scheme
'https'
>>> uri.host
'localhost'
>>> uri.port is None
True
>>> uri.authority
'localhost'
>>> uri.path
'/blee/blah'
```

The grammar from RFC 3986 does not allow for square brackets in the query component, but Section 3.4 does say how such delimiter characters should be handled if found in the component.

```
>>> uri = URI('http://www.apple.com/store?delivery=[slow]#horse+cart')
>>> uri.scheme
'http'
>>> uri.host
'www.apple.com'
>>> uri.port is None
True
>>> uri.path
'/store'
>>> uri.query
'delivery=[slow]'
>>> uri.fragment
'horse+cart'
```


FINDING URIS IN TEXT

lazr.uri also knows how to retrieve a list of URIs from a block of text. This is intended for uses like finding bug tracker URIs or similar.

The `find_uris_in_text()` function returns an iterator that yields URI objects for each URI found in the text. Note that the returned URIs have been canonicalised by the URI class:

```
>>> from lazr.uri import find_uris_in_text
>>> text = '''
... A list of URIs:
... * http://localhost/a/b
... * http://launchpad.net
... * MAILTO:joe@example.com
... * xmpp:fred@example.org
... * http://bazaar.launchpad.net/~name12/firefox/foo
... * http://somewhere.in/time?track=[02]#wasted-years
... '''
```

```
>>> for uri in find_uris_in_text(text):
...     print(uri)
http://localhost/a/b
http://launchpad.net/
mailto:joe@example.com
xmpp:fred@example.org
http://bazaar.launchpad.net/~name12/firefox/foo
http://somewhere.in/time?track=[02]#wasted-years
```

2.1 NEWS for lazr.uri

2.1.1 1.0.6 (2021-09-13)

- Adjust versioning strategy to avoid importing `pkg_resources`, which is slow in large environments.

2.1.2 1.0.5 (2020-06-29)

- Add an explicit `__hash__` method to `lazr.uri.URI`.

2.1.3 1.0.4 (2020-06-12)

- Install `version.txt` with `package_data` (Stefano Rivera, <https://bugs.launchpad.net/lazr.uri/+bug/918660>).
- Switch from `buildout` to `tox`.

2.1.4 1.0.3 (2012-01-18)

- Add compatibility with Python 3 (Thomas Kluyver).

2.1.5 1.0.1 (2009-06-01)

- Eliminate dependency on `setuptools_bzr` so `sdist`s do not bring `bzr` `ini`, among others.

2.1.6 1.0 (2009-03-23)

- Initial release on PyPI