
lazyxml Documentation

Release 1.2.1

Zonglong Fan

Jun 29, 2017

Contents

1	API Documentation	1
1.1	lazyxml – A simple xml parse and build lib	1
2	Changes	9
3	About This Documentation	11
4	Indices and tables	13
	Python Module Index	15

lazyxml – A simple xml parse and build lib

loads() – Load xml content to python object.

A simple xml parse and build lib.

lazyxml.**loads**(*content*, ***kw*)

Load xml content to python object.

```
>>> import lazyxml
```

```
>>> xml = '<demo><foo>foo</foo><bar>bar</bar></demo>'
>>> lazyxml.loads(xml)
{'bar': 'bar', 'foo': 'foo'}
```

```
>>> xml = '<demo><foo>foo</foo><bar>bar</bar></demo>'
>>> lazyxml.loads(xml, strip_root=False)
{'demo': {'bar': 'bar', 'foo': 'foo'}}
```

```
>>> xml = '<demo><foo>foo</foo><bar>1</bar><bar>2</bar></demo>'
>>> lazyxml.loads(xml)
{'bar': ['1', '2'], 'foo': 'foo'}
```

```
>>> xml = '<root xmlns:h="http://www.w3.org/TR/html4/">&lt;demo&gt;&lt;foo&gt;foo&
↳lt;/foo&gt;&lt;bar&gt;bar&lt;/bar&gt;&lt;/demo&gt;</root>'
>>> lazyxml.loads(xml, unescape=True, strip_root=False)
{'root': {'demo': {'bar': 'bar', 'foo': 'foo'}}}
```

Parameters *content* (*str*) – xml content

kw arguments below here.

Parameters

- **encoding** – XML utf-8.
- **unescape** (*bool*) – HTML False.
- **strip_root** (*bool*) – True.
- **strip_attr** (*bool*) – True.
- **strip** (*bool*) – True.
- **errors** – :str.decode() strict.

Return type `dict`

Changed in version 1.2.1: The `strip_attr` option supported to decide whether return the element attributes for parse result.

load() – Load xml content from file and convert to python object.

A simple xml parse and build lib.

`lazyxml.load(fp, **kw)`

Load xml content from file and convert to python object.

```
>>> import lazyxml
>>> with open('demo.xml', 'rb') as fp:
>>>     lazyxml.load(fp)
```

```
>>> from cStringIO import StringIO
>>> buffer = StringIO('<?xml version="1.0" encoding="utf-8"?><demo><foo><![CDATA[
↪<foo>]]></foo><bar><![CDATA[1]]></bar><bar><![CDATA[2]]></bar></demo>')
>>> lazyxml.load(buffer)
{'bar': ['1', '2'], 'foo': '<foo>'}
>>> buffer.close()
```

Note: kw argument have the same meaning as in `loads()`

Parameters **fp** – a file or file-like object that support `.read()` to read the xml content

Return type `dict`

dumps() – Dump python object to xml.

A simple xml parse and build lib.

`lazyxml.dumps(obj, **kw)`

Dump python object to xml.

```
>>> import lazyxml
```

```
>>> data = {'demo':{'foo': '<foo>', 'bar': ['1', '2']}}
```

```
>>> lazyxml.dumps(data)
'<?xml version="1.0" encoding="utf-8"?><demo><foo><![CDATA[<foo>]]></foo><bar><![
↪CDATA[1]]></bar><bar><![CDATA[2]]></bar></demo>'
```

```
>>> lazyxml.dumps(data, header_declare=False)
'<demo><foo><![CDATA[<foo>]]></foo><bar><![CDATA[1]]></bar><bar><![CDATA[2]]></
↳bar></demo>'
```

```
>>> lazyxml.dumps(data, cdata=False)
'<?xml version="1.0" encoding="utf-8"?><demo><foo>&lt;foo&gt;</foo><bar>1</bar>
↳<bar>2</bar></demo>'
```

```
>>> print lazyxml.dumps(data, indent=' ' * 4)
<?xml version="1.0" encoding="utf-8"?>
<demo>
  <foo><![CDATA[<foo>]]></foo>
  <bar><![CDATA[1]]></bar>
  <bar><![CDATA[2]]></bar>
</demo>
```

```
>>> lazyxml.dumps(data, ksort=True)
'<?xml version="1.0" encoding="utf-8"?><demo><bar><![CDATA[1]]></bar><bar><![
↳[CDATA[2]]></bar><foo><![CDATA[<foo>]]></foo></demo>'
```

```
>>> lazyxml.dumps(data, ksort=True, reverse=True)
'<?xml version="1.0" encoding="utf-8"?><demo><foo><![CDATA[<foo>]]></foo><bar><![
↳[CDATA[1]]></bar><bar><![CDATA[2]]></bar></demo>'
```

Note: Data that has attributes convert to xml see demo/dump.py.

Parameters `obj` – data for dump to xml.

kw arguments below here.

Parameters

- **encoding** – XML utf-8.
- **header_declare** (*bool*) – XML True.
- **root** – XML None.
- **cdata** (*bool*) – XML CDATA True.
- **indent** – XML None.
- **ksort** (*bool*) – XML False.
- **reverse** (*bool*) – XML False.
- **errors** – see: `str.decode()` strict.
- **hasattr** (*bool*) – False.
- **attrkey** – key {attrs}.
- **valuekey** – key {values}.

Return type `str`

dump () – Dump python object to file.

A simple xml parse and build lib.

`lazyxml.dump (obj, fp, **kw)`
Dump python object to file.

```
>>> import lazyxml
>>> data = {'demo': {'foo': 1, 'bar': 2}}
>>> lazyxml.dump(data, 'dump.xml')
>>> with open('dump-fp.xml', 'w') as fp:
>>>     lazyxml.dump(data, fp)
```

```
>>> from cStringIO import StringIO
>>> data = {'demo': {'foo': 1, 'bar': 2}}
>>> buffer = StringIO()
>>> lazyxml.dump(data, buffer)
>>> buffer.getvalue()
<?xml version="1.0" encoding="utf-8"?><demo><foo><![CDATA[1]]></foo><bar><![
↪ [CDATA[2]]></bar></demo>
>>> buffer.close()
```

Note: kw argument have the same meaning as in `dumps ()`

Parameters

- **obj** – data for dump to xml.
- **fp** – a filename or a file or file-like object that support `.write ()` to write the xml content

Changed in version 1.2: The *fp* is a filename of string before this. It can now be a file or file-like object that support `.write ()` to write the xml content.

builder – XML Builder Module

class `lazyxml.builder.Builder (**kw)`
XML Builder

set_options (kw)**
Set Builder options.

See also:

kw argument have the same meaning as in `lazyxml.dumps ()`

get_options ()
Get Builder options.

dict2xml (data)
Convert dict to xml.

Warning: DEPRECATED: `dict2xml ()` is deprecated. Please use `object2xml ()` instead.

Deprecated since version 1.2.

object2xml (*data*)

Convert python object to xml string.

Parameters **data** – data for build xml. If don't provide the `root` option, type of `data` must be dict and `len(data) == 1`.

Return type `str` or `unicode`

New in version 1.2.

build_xml_header ()

Build xml header include version and encoding.

Return type `str`

build_tree (*data, tagname, attrs=None, depth=0*)

Build xml tree.

Parameters

- **data** – data for build xml.
- **tagname** – element tag name.
- **attrs** (*dict or None*) – element attributes. DefaultNone.
- **depth** (*int*) – element depth of the hierarchy. Default0.

check_structure (*keys*)

Check structure availability by `attrkey` and `valuekey` option.

pickdata (*data*)

Pick data from `attrkey` and `valuekey` option.

Returns a pair of (`attrs`, `values`)

Return type `tuple`

safedata (*data, cdata=True*)

Convert xml special chars to entities.

Parameters

- **data** – the data will be converted safe.
- **cdata** (*bool*) – whether to use cdata. DefaultTrue. If not, use `cgi.escape()` to convert data.

Return type `str`

build_tag (*tag, text='', attrs=None*)

Build tag full info include the attributes.

Parameters

- **tag** – tag name.
- **text** – tag text.
- **attrs** (*dict or None*) – tag attributes. DefaultNone.

Return type `str`

build_attr (*attrs*)

Build tag attributes.

Parameters **attrs** (*dict*) – tag attributes

Return type `str`

tag_start (*tag*, *attrs=None*)
Build started tag info.

Parameters

- **tag** – tag name
- **attrs** (*dict* or *None*) – tag attributes. DefaultNone.

Return type `str`

tag_end (*tag*)
Build closed tag info.

Parameters **tag** – tag name

Return type `str`

parser – XML Parser Module

class `lazyxml.parser.Parser` (**kw)
XML Parser

set_options (**kw)
Set Parser options.

See also:

`kw` argument have the same meaning as in `lazyxml.loads()`

get_options ()
Get Parser options.

xml2dict (*content*)
Convert xml content to dict.

Warning: DEPRECATED: `xml2dict()` is deprecated. Please use `xml2object()` instead.

Deprecated since version 1.2.

xml2object (*content*)
Convert xml content to python object.

Parameters **content** – xml content

Return type `dict`

New in version 1.2.

xml_filter (*content*)
Filter and preprocess xml content

Parameters **content** – xml content

Return type `str`

guess_xml_encoding (*content*)
Guess encoding from xml header declaration.

Parameters **content** – xml content

Return type `str` or `None`

strip_xml_header (*content*)

Strip xml header

Parameters **content** – xml content

Return type `str`

parse (*element*)

Parse xml element.

Parameters **element** – an `Element` instance

Return type `dict`

parse_full (*element*)

Parse xml element include the node attributes.

Parameters **element** – an `Element` instance

Return type `dict`

New in version 1.2.1.

get_node (*element*)

Get node info.

Parse element and get the element tag info. Include tag name, value, attribute, namespace.

Parameters **element** – an `Element` instance

Return type `dict`

split_namespace (*tag*)

Split tag namespace.

Parameters **tag** – tag name

Returns a pair of (namespace, tag)

Return type `tuple`

CHAPTER 2

Changes

See the changelog for a full list of changes to *lazyxml*.

About This Documentation

This documentation is generated using the [Sphinx](#) documentation generator. The source files for the documentation are located in the *docs/* directory of the **lazyxml** distribution. To generate the docs locally run the following command from the *docs/* directory of the **lazyxml** source:

```
$ cd docs
$ make html
```

or use `make help` to generate other format.

CHAPTER 4

Indices and tables

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

|

lazyxml, 1
lazyxml.builder, 4
lazyxml.parser, 6

B

build_attr() (lazyxml.builder.Builder method), 5
build_tag() (lazyxml.builder.Builder method), 5
build_tree() (lazyxml.builder.Builder method), 5
build_xml_header() (lazyxml.builder.Builder method), 5
Builder (class in lazyxml.builder), 4

C

check_structure() (lazyxml.builder.Builder method), 5

D

dict2xml() (lazyxml.builder.Builder method), 4
dump() (in module lazyxml), 4
dumps() (in module lazyxml), 2

G

get_node() (lazyxml.parser.Parser method), 7
get_options() (lazyxml.builder.Builder method), 4
get_options() (lazyxml.parser.Parser method), 6
guess_xml_encoding() (lazyxml.parser.Parser method), 6

L

lazyxml (module), 1, 2, 4
lazyxml.builder (module), 4
lazyxml.parser (module), 6
load() (in module lazyxml), 2
loads() (in module lazyxml), 1

O

object2xml() (lazyxml.builder.Builder method), 4

P

parse() (lazyxml.parser.Parser method), 7
parse_full() (lazyxml.parser.Parser method), 7
Parser (class in lazyxml.parser), 6
pickdata() (lazyxml.builder.Builder method), 5

S

safedata() (lazyxml.builder.Builder method), 5

set_options() (lazyxml.builder.Builder method), 4
set_options() (lazyxml.parser.Parser method), 6
split_namespace() (lazyxml.parser.Parser method), 7
strip_xml_header() (lazyxml.parser.Parser method), 7

T

tag_end() (lazyxml.builder.Builder method), 6
tag_start() (lazyxml.builder.Builder method), 6

X

xml2dict() (lazyxml.parser.Parser method), 6
xml2object() (lazyxml.parser.Parser method), 6
xml_filter() (lazyxml.parser.Parser method), 6