
untangle Documentation

Release 1.1.0

Christian Stefanescu

Jul 01, 2022

Contents

1	Usage	3
2	Example	5
3	Installation	7
4	Motivation	9
5	Limitations	11
6	Changelog	13
7	Indices and tables	15
	Python Module Index	17
	Index	19

untangle is a tiny Python library which converts an XML document to a Python object. It is available under the [MIT license](#).

Contents

- *untangle: Convert XML to Python objects*
 - *Usage*
 - *Example*
 - *Installation*
 - *Motivation*
 - *Limitations*
 - *Changelog*
- *Indices and tables*

CHAPTER 1

Usage

untangle has a very simple API. You just need to call the parse function to get back a Python object. The parameter can be:

- a string
- a filename
- a URL

`untangle.parse(filename)`

Interprets the given string as a filename, URL or XML data string, parses it and returns a Python object which represents the given document.

Raises `ValueError` if the argument is `None` / empty string.

Raises `xml.sax.SAXParseException` if something goes wrong during parsing.

If you are looking for information on a specific function, class or method, this part of the documentation is for you.

The object you get back represents the complete XML document. Child elements can be accessed with `parent.child`, attributes with `element['attribute']`. Siblings with similar names are grouped into a list.

CHAPTER 2

Example

Considering this XML document:

```
<?xml version="1.0"?>
<root>
  <child name="child1"/>
</root>
```

and assuming it's available in a variable called *xml*, we could use untangle like this:

```
doc = untangle.parse(xml)
child_name = doc.root.child['name'] # 'child1'
```

For more examples, have a look at (and launch) [examples.py](#).

CHAPTER 3

Installation

It is recommended to use pip, which will always download the latest stable release:

```
pip install untangle
```

untangle works with Python versions 2.4, 2.5, 2.6, 2.7, 3.0, 3.1, 3.2 and pypy

CHAPTER 4

Motivation

untangle is available for that use case, where you have a 20-line XML file you got back from an API and you just need to extract some values out of it. You might not want to use regular expressions, but just as well you might not want to install a complex libxml2-based solution (and look up its terse API).

Performance and memory usage might be bad, but these tradeoffs were made in order to allow a simple API and no external dependencies. See also: *Limitations*.

Limitations

untangle trades features for a simple API, which is why untangle substitutes `-`, `.` and `:` with `_`:

- `<foobar><foo-bar/></foobar>` can be accessed with `foobar.foo_bar`
- `<foo.bar.baz/>` can be accessed with `foo_bar_baz`
- `<foo:bar><foo:baz/></foo:bar>` can be accessed with `foo_bar.foo_baz`

CHAPTER 6

Changelog

1.1.0 - `__dir__` support for untangled objects - code cleanups

1.0.0 - first official release

CHAPTER 7

Indices and tables

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

u

untangle, 3

P

`parse()` (*in module untangle*), 3

U

`untangle(module)`, 3