
lxml-wrapper Documentation

Release 0.4a

Mateusz 'matee' Pawlik

February 04, 2016

1	What is it doing?	3
2	Installation	5
3	Basic usage	7
4	Indices and tables	9

Contents:

What is it doing?

Have you ever needed to write many complex XML generators? We have.

```
root = Element('root', atr=str(100))
root.text = 'text1'
child = SubElement(root, 'child', atr="atr")
superchild = SubElement(root, 'superchild', atr="" if value is None else value)
superchild.text = 'sctext1'
superchild.tail = 'tail1'
superchild.tail += 'tail2'
child.tail = 'tail'
child = SubElement(root, 'child', atr="")
child.text = 'text'
```

Code shown above is generating relatively simple XML:

```
<root atr="100">
  text1
  <child atr="atr">
    <superchild atr="">sctext1</superchild>
  tail1
  tail2
  </child>
  tail
  <child atr="">text</child>
</root>
```

If we wanted to generate a bigger, more complex XML with many deeply nested lists and conditional elements, we needed to write more boilerplate code. Then I wrote a little helper which helped us a lot, and reduced highly time spent on writing and also reading a code that generates XMLs.

Code below shows our new way (for us it's not so new, it's used in our production environments since 2010):

```
root = E('root', atr=100).add(
    'text1',
    E('child', atr="atr").add(
        E('superchild', atr=None).add('sctext1'),
        'tail1',
        'tail2'
    ),
    'tail',
    E('child', atr="").add(
        'text'
    )
)
```

Now, as you can see, all boilerplate goes away. This library gives you one more little bonus. Normally you need to pass strings as `attrs`, `text`, `tail`, etc but `lxml-wrapper` also simplifies that for you. In the last example we've just passed an integer as a first attribute.

Installation

```
pip install lxml-wrapper
```

Basic usage

```
from lxmlwrapper import E, etree
e = E('root', param='foo').add(
    E('SubElement', param='bar').add('just text')
)
print etree.tostring(e)
<root param="foo"><SubElement param="bar">just text</SubElement></root>
```

Indices and tables

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