

/home/português/python/validando com um esquema XML em Python\_

# Validando com um esquema XML em Python

Eu tenho um arquivo XML e um esquema XML em outro arquivo e gostaria de validar se meu arquivo XML adere ao esquema. Como faço isso em Python?

Prefiro algo usando a biblioteca padrão, mas posso instalar um pacote de terceiros, se necessário.

 **python xml validation xsd**

 92



Eli Courtwright



18 de nov. de 2008



Suponho que você queira dizer usando arquivos XSD. Surpreendentemente, não há muitas bibliotecas XML python que suportam isso. lxml faz no entanto. Marque [Validação com lxml](#) . A página também lista como usar o lxml para validar com outros tipos de esquema.

 55



Keegan Carruthers-Smith



18 de nov. de 2008

ti-enxame.com

The most convenient and re-

# liable file storage service

Receive your personal cloud storage with **2Gb of space for free**

---

Quanto às soluções "python puro": o índice do pacote lista:

- [pyxsd](#) , a descrição diz que usa `xml.etree.cElementTree`, que não é "python puro" (mas incluído no `stdlib`), mas o código fonte indica que ele volta a `xml.etree.ElementTree`, portanto isso contaria como python puro. Não o utilizou, mas de acordo com os documentos, ele faz a validação de esquema.
- [minixsv](#) : 'um validador de esquema XML leve escrito em Python "puro"'. No entanto, a descrição diz "atualmente um subconjunto do padrão do esquema XML é suportado", portanto, isso pode não ser suficiente.
- [XSV](#) , que eu acho que é usado para o validador xsd on-line do W3C (ainda parece usar o pacote `pyxml` antigo, que acho que não é mais mantido)



Steven



6 de abr. de 2011

---

## Um exemplo de um validador simples em Python3 usando a biblioteca popular [lxml](#)

### Instalação lxml

```
pip install lxml
```

Se você receber um erro como *"Não foi possível encontrar a função `xml-CheckVersion` na biblioteca `libxml2`. A `libxml2` está instalada?"* , tente fazer isso primeiro:

```
# Debian/Ubuntu
apt-get install python-dev python3-dev libxml2-dev libxslt-dev

# Fedora 23+
dnf install python-devel python3-devel libxml2-devel libxslt-devel
```

---

## O validador mais simples

Vamos criar mais simples **validator.py**

```
from lxml import etree

def validate(xml_path: str, xsd_path: str) -> bool:

    xmlschema_doc = etree.parse(xsd_path)
    xmlschema = etree.XMLSchema(xmlschema_doc)

    xml_doc = etree.parse(xml_path)
    result = xmlschema.validate(xml_doc)

    return result
```

então escreva e execute **main.py**

```
from validator import validate

if validate("path/to/file.xml", "path/to/scheme.xsd"):
    print('Valid! :)')
else:
    print('Not valid! :(')
```

---

## m pouco de POO

Para validar mais de um arquivo, não há necessidade de criar um objeto *XMLSchema* todas as vezes, portanto:

### validator.py

```
from lxml import etree

class Validator:

    def __init__(self, xsd_path: str):
```

```
def __init__(self, xsd_path: str):
    xmlschema_doc = etree.parse(xsd_path)
    self.xmlschema = etree.XMLSchema(xmlschema_doc)

def validate(self, xml_path: str) -> bool:
    xml_doc = etree.parse(xml_path)

    result = self.xmlschema.validate(xml_doc)

    return result
```

Agora podemos validar todos os arquivos no diretório da seguinte maneira:

### main.py

```
import os
from validator import Validator

validator = Validator("path/to/scheme.xsd")

# The directory with XML files
XML_DIR = "path/to/directory"

for file_name in os.listdir(XML_DIR):
    print('{}: '.format(file_name), end='')

    file_path = '{}/{}/'.format(XML_DIR, file_name)

    if validator.validate(file_path):
        print('Valid! :)')
    else:
        print('Not valid! :(')
```

---

Para mais opções, leia aqui: [Validação com lxml](#)



Sergio



22 de jun. de 2016

---

O pacote PyXB em <http://pyxb.sourceforge.net/> gera ligações de validação para Python \_ a partir de documentos de esquema XML. Ele lida com quase todas as construções de esquema e suporta vários espaços para nome.



pabigot



22 de dez. de 2009

---

Existem duas maneiras (na verdade existem mais) de você fazer isso.

1. usando [lxml](#)

```
pip install lxml
```

```
from lxml import etree, objectify
from lxml.etree import XMLSyntaxError

def xml_validator(some_xml_string, xsd_file='/path/to/my_schema_file.xsd'):
    try:
        schema = etree.XMLSchema(file=xsd_file)
        parser = objectify.makeparser(schema=schema)
        objectify.fromstring(some_xml_string, parser)
        print "YEAH!, my xml file has validated"
    except XMLSyntaxError:
        #handle exception here
        print "Oh NO!, my xml file does not validate"
        pass

xml_file = open('my_xml_file.xml', 'r')
xml_string = xml_file.read()
xml_file.close()

xml_validator(xml_string, '/path/to/my_schema_file.xsd')
```

2. Use [xmllint](#) na linha de comando. O xmllint vem instalado em muitas distribuições linux.

```
>> xmllint --format --pretty 1 --load-trace --debug --schema
/path/to/my_schema_file.xsd /path/to/my_xml_file.xml
```



Komu



26 de ago. de 2015

# Flexible, reliable and affordable cloud hosting



Sign up and get **\$50** bonus within 30-day!

---

o lxml fornece etree.DTD

dos testes em [http://lxml.de/api/lxml.tests.test\\_dtd-pysrc.html](http://lxml.de/api/lxml.tests.test_dtd-pysrc.html)

```
...
root = etree.XML(_bytes("<b/>"))
dtd = etree.DTD(BytesIO("<!ELEMENT b EMPTY>"))
self.assert_(dtd.validate(root))
```



altunyurt



18 de nov. de 2008

---

É possível validar facilmente um arquivo ou árvore XML em um esquema XML (XSD) com o [xmlschema Python package](#) . É Python puro, disponível em [PyPi](#) e não possui muitas dependências.

Exemplo - valide um arquivo:

```
import xmlschema
xmlschema.validate('doc.xml', 'some.xsd')
```

O método gera uma exceção se o arquivo não for validado no XSD. Essa exceção contém alguns detalhes da violação.

Se você deseja validar muitos arquivos, basta carregar o XSD uma vez:

```
xsd = xmlschema.XMLSchema('some.xsd')
for filename in filenames:
    xsd.validate(filename)
```

Se você não precisar da exceção, poderá validar assim:

```
if xsd.is_valid('doc.xml'):
    print('do something useful')
```

Como alternativa, xmlschema funciona diretamente em objetos de arquivo e em árvores XML de memória (criadas com xml.etree.ElementTree ou lxml).  
Exemplo:

```
import xml.etree.ElementTree as ET
t = ET.parse('doc.xml')
result = xsd.is_valid(t)
print('Document is valid? {}'.format(result))
```



maxschleipzig



13 de set. de 2018

# The most convenient and reliable file storage service

Receive your personal cloud storage with **2Gb of space for free**

[Ferramenta de validação do XML Schema \(XSD\)?](#)

[Existe alguma diferença entre 'xml válido' e 'xml bem formado'?](#)

[Como validar xml contra xsd e obter erros \\* ALL \\*?](#)

[Como validar um arquivo XML usando Java com um XSD com inclusão?](#)

[Como corrigir soapenv: problema de envelope no esquema XSD durante a validação com SOAP solicitação/resposta](#)

[Como faço para analisar XML em Python?](#)

[Criando um arquivo XML simples usando python](#)

[Como visualizar um esquema XML?](#)

Quaisquer ferramentas para gerar um esquema XSD de um documento de instância XML?

Para que são utilizados os dados de validação em um modelo Sequencial Keras?

O esquema de SpringLocation falha quando não há conexão com a Internet

Como converter uma string xml em um dicionário?

Analisando um arquivo YAML no Python e acessando os dados?

Validação de XML com XSD no Visual Studio IDE

Gerar esquema Json a partir do esquema XML (XSD)

análise XML do jQuery com namespaces

Como posso incluir um caractere e comercial (&) em um documento XML?

Existe uma extensão de esquema XML para o código VS

Usando o Python Iterparse para grandes arquivos XML

Como obter todos os subelementos de uma árvore de elementos com o Python ElementTree?

---

Content dated before 2011-04-08 (UTC) is licensed under [CC BY-SA 2.5](#). Content dated from 2011-04-08 up to but not including 2018-05-02 (UTC) is licensed under [CC BY-SA 3.0](#). Content dated on or after 2018-05-02 (UTC) is licensed under [CC BY-SA 4.0](#). |

[Privacy](#)