schematics-xml Documentation

Release 0.2.1

Alex Hayes

November 11, 2016

Contents

Python schematics models for converting to and from XML.

CHAPTER 1

Contents

1.1 Installation

You can install schematics-xml either via the Python Package Index (PyPI) or from github.

To install using pip;

```
$ pip install schematics-xml
```

From github;

```
$ pip install git+https://github.com/alexhayes/schematics-xml.git
```

Currently *schematics-xml* only supports Python 3 - if you'd like feel free to submit a PR adding support for Python 2, however note that I love type annotations and any Python 2 PR will need to include a stub file.

1.2 Usage

Simply inherit `XMLModel`.

```
from schematics_xml import XMLModel

class Person(XMLModel):
    name = StringType()

john = Person(dict(name='John'))

xml = john.to_xml()
```

XML now contains;

```
<?xml version='1.0' encoding='UTF-8'?>
<person>
  <name>John</name>
</person>
```

And back the other way;

```
john = Person.from_xml(xml)
```

1.2.1 Root Node

To set the root node simply define class property `xml_root`, as follows;

```
from schematics_xml import XMLModel

class Animal(XMLModel):
    kind = StringType()

    @property
    def xml_root(self):
        return self.kind

garfield = Animal(dict(kind='cat'))

xml = garfield.to_xml()
```

XML now contains;

```
<?xml version='1.0' encoding='UTF-8'?>
<cat>
  <kind>cat</kind>
</cat>
```

1.2.2 Encoding

By default the encoding returned *XMLModel.to_xml()* is *UTF-8* however this can be changed either by setting the *xml_encoding* attribute on the model or by setting the *encoding* kwarg when calling *XMLModel.to_xml()*.

```
from schematics_xml import XMLModel

class Animal(XMLModel):
    xml_encoding = 'UTF-8'
    kind = StringType()

garfield = Animal(dict(kind='cat'))

xml = garfield.to_xml()
```

XML now contains;

```
<?xml version='1.0' encoding='UTF-8'?>
<cat>
  <animal>cat</animal>
</cat>
```

1.3 Developer Documentation

1.3.1 Contributions

Contributions are more than welcome!

To get setup do the following;

```
mkvirtualenv --python=/usr/bin/python3.5 schematics-xml
git clone https://github.com/alexhayes/schematics-xml.git
cd schematics-xml
pip install -r requirements/dev.txt
pip install Django>=1.9,<1.10
```

1.3.2 Running Tests

Once you've checked out you should be able to run the tests;

```
tox
```

Or run all environments at once using detox;

```
detox
```

1.3.3 Creating Documentation

```
cd docs
make clean html
```

1.4 Internal Module Reference

Release 0.2.1

Date November 11, 2016

1.4.1 schematics_xml package

Subpackages

schematics_xml.tests package

Submodules

schematics_xml.tests.test_models module

Module contents

Submodules

schematics_xml.models module

schematics_xml.models

Base models that provide to/from XML methods.

class schematics_xml.models.**XMLModel**(*raw_data=None*, *trusted_data=None*, *deserialize_mapping=None*, *init=True*, *partial=True*, *strict=True*, *validate=False*, *app_data=None*, ***kwargs*)

Bases: schematics.models.Model

A model that can convert it's fields to and from XML.

classmethod from_xml(*xml: str*) → schematics.models.Model

Convert XML into a model.

Parameters **xml** – A string of XML that represents this Model.

primitive_to_xml(*primitive: dict*, *parent: 'lxml.etree._Element'=None*)

primitive_value_to_xml(*key*, *parent*, *value*)

to_xml(*role: str=None*, *app_data: dict=None*, *encoding: str=None*, ***kwargs*) → str

Return a string of XML that represents this model.

Currently all arguments are passed through to schematics.Model.to_primitive.

Parameters

- **role** – schematics Model to_primitive role parameter.
- **app_data** – schematics Model to_primitive app_data parameter.
- **encoding** – xml encoding attribute string.
- **kwargs** – schematics Model to_primitive kwargs parameter.

xml_encoding = 'UTF-8'

Override this attribute to set the encoding specified in the XML returned by *XMLModel.to_xml()*.

xml_root

Override this attribute to set the XML root returned by *XMLModel.to_xml()*.

schematics_xml.models.**ensure_lists_in_model**(*raw_data: dict*, *model_cls: schematics_xml.models.XMLModel*)

Ensure that single item lists are represented as lists and not dicts.

In XML single item lists are converted to dicts by xmltodict - there is essentially no way for xmltodict to know that it *should* be a list not a dict.

Parameters

- **raw_data** –
- **model_cls** –

schematics_xml.models.**ensure_lists_in_value**(*value: 'typing.Any'*, *field: schematics.types.base.BaseType*)

schematics_xml.models.**field_has_type**(*needle: schematics.types.base.BaseType*, *field: schematics.types.base.BaseType*) → bool

Return True if field haystack contains a field of type needle.

Parameters

- **needle** – A schematics field class to search for.
- **haystack** – An instance of a schematics field within a model.

schematics_xml.models.**model_has_field_type**(*needle: schematics.types.base.BaseType*, *haystack: schematics.models.Model*) → bool

Return True if haystack contains a field of type needle.

Iterates over all fields (and into field if appropriate) and searches for field type *needle* in model *haystack*.

Parameters

- **needle** – A schematics field class to search for.
- **haystack** – A schematics model to search within.

Module contents

Python schematics models for converting to and from XML.

class schematics_xml.**VersionInfo**(*major*, *minor*, *micro*, *releaselevel*, *serial*)
Bases: tuple

major
Alias for field number 0

micro
Alias for field number 2

minor
Alias for field number 1

releaselevel
Alias for field number 3

serial
Alias for field number 4

1.4.2 schematics_xml

CHAPTER 2

License

This software is licensed under the *MIT License*. See the LICENSE.

CHAPTER 3

Author

Alex Hayes <alex@alution.com>

S

E

F

M

P

R

S

T

V

X