tri.struct Documentation

Release 4.0.0

Anders Hovmöller

Aug 20, 2020

Contents

tri.struct supplies classes that can be used like dictionaries and as objects with attribute access at the same time. There are two versions:

- Struct: mutable struct
- FrozenStruct: immutable struct

Some niceties include:

- Predictable repr() so it's easy to write tests
- *merged* function call to merge different types of dicts into a new: *merged(Struct(a=1), FrozenStruct(b=1), c=1) == Struct(a=1, b=1, c=1)*)
- Accelerated implementation in c for improved speed. (With python-only fallback reference implementation)

CHAPTER 1

Example

```
>>> from tri_struct import Struct
>>> foo = Struct()
>>> foo.a = 1
>>> foo['a']
1
>>> foo['a'] = 2
>>> foo.a
2
```

CHAPTER 2

Running tests

You need tox installed then just *make test*.

CHAPTER 3

License

BSD

CHAPTER 4

Documentation

http://tristruct.readthedocs.org.

CHAPTER 5

Contents:

5.1 Installation

At the command line:

```
$ pip install tri.struct
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv tri.struct
$ pip install tri.struct
```

5.2 API documentation

class tri_struct.**Struct**

Struct is a dict that can be accessed like an object. It also has a predictable repr so it can be used in tests for example.

```
>>> bs = Struct(a=1, b=2, c=3)
>>> bs
Struct(a=1, b=2, c=3)
>>> bs.a
1
```

- Struct(**kwargs) -> new Struct initialized with the name=value pairs in the keyword aasdrgument list. For example: Struct(one=1, two=2)
- Struct() -> new empty Struct
- Struct(mapping) -> new Struct initialized from a mapping object's (key, value) pairs
- **Struct(iterable) -> new Struct initialized as if via:**

```
s = Struct()
for k, v in iterable:
    s[k] = v
```

__delattr__(*item*)
x.__delattr__('name') <==> del x.name

__getattribute__(*item*)
x.__getattribute__('name') <==> x.name

__repr__() <==> *repr(x)*

__setattr__(*key*, *value*)
x.__setattr__('name', value) <==> x.name = value

__str__()
x.__repr__() <==> repr(x)

copy() → a shallow copy of D

class tri_struct.**FrozenStruct**

tri_struct.**merged**(**dicts*, ***kwargs*)
Merge dictionaries. Later keys overwrite.

```
merged(dict(a=1), dict(b=2), c=3, d=1)
```

class tri_struct.**DefaultStruct**(*default_factory=None*, **args*, ***kwargs*)

__init__(*default_factory=None*, **args*, ***kwargs*)
x.__init__(...) initializes x; see help(type(x)) for signature

5.3 History

- Split c implementation to separate *FastStruct*. The native python implementation is now always the *Struct*

5.3.1 3.1.0 (2019-08-14)

- Override __copy__ for Frozen. This is an optimization that was found in big uses of tri.token.

5.3.2 3.0.1 (2019-06-12)

- Problems with pypi, this is the same as 3.0.1

5.3.3 3.0.0 (2019-06-04)

- Renamed module from *tri.struct* to *tri_struct*. This is a breaking change.
- Dropped python2 support

5.3.4 2.5.7 (2018-11-16)

- Fixed performance issue with *Frozen*/*FrozenStruct*: the hash was recalculated on each use instead of cached.

5.3.5 2.5.6 (2018-06-26)

- Fixed release functionality

5.3.6 2.5.5 (2018-02-20)

- Fixed segfault in repr when running under Python 3

5.3.7 2.5.4 (2017-06-13)

- Added *DefaultStruct* in the spirit of the standard library *defaultdict*. Also added a *to_default_struct* for recursively converting dicts recursively.

5.3.8 2.5.3 (2017-02-10)

- Fix use-after-free when repring a *Struct* that contains itself more than once.

5.3.9 2.5.2 (2016-04-07)

- Fix make and tox targets for build and release.
- Fix lint issues.

5.3.10 2.5.1 (2015-12-15)

- Bugfix: Fix compilation of the _cstruct module.

5.3.11 2.5.0 (unreleased)

- Build changes.

5.3.12 2.4.0 (2015-12-08)

- Improvement: If a Struct subclass implements the *__missing__* method, it will not be called before GetAttr on attribute access, but will be called before GetAttr on dict access.

5.3.13 2.3.1 (2015-12-07)

- Bugfix: Clear KeyError in CStruct getattr before trying GetAttr, otherwise the KeyError may "leak out". Also, only clear the error and attempt GetAttr if the original error was a KeyError.

5.3.14 2.3.0 (2015-12-02)

- Add mixin class *Frozen* to make read-only versions of a dict-derived class (typically a Struct or a subclass there of.)
- Use the *Frozen* mixin to implement *FrozenStruct*

5.3.15 2.2.0 (2015-11-12)

- Add keyword arguments to *merged* function.

5.3.16 2.1.2 (2015-11-11)

- Redo the C implementation to be a "heaptype", and remove the hack of setting *__dict__ = self*. Instead, *object* will control the type storage, letting us "insert" attributes into the object without polluting the *dict*.

5.3.17 2.0 (unreleased)

- slim down interface to again match dict
- add tri.struct.merged function to join structs
- add optional C implementation to speed up instantiation

5.3.18 1.0 (2015-09-29)

- Struct with attribute & dict interface
- __add__ and __or__ to combine structs

5.4 Credits

- Johan Lübcke <johan.lubcke@trioptima.com>
- Anders Hovmöller <anders.hovmoller@trioptima.com>

5.5 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

Issues, feature requests, etc are handled on github.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Index

Symbols

C

D

F

M

S

T