
tri.named*_structDocumentation*

Release 0.12.0

Anders Hovmöller

October 26, 2016

1	tri.named_struct	3
1.1	Example	3
1.2	Running tests	4
1.3	License	4
1.4	Documentation	4
2	Contents:	5
2.1	Installation	5
2.2	API documentation	5
2.3	History	6
2.4	Credits	6
2.5	Contributing	6
3	Indices and tables	7

target <http://codecov.io/github/TriOptima/tri.named-struct?branch=master>

tri.named_struct

tri.named_struct supplies classes that can be used like dictionaries, but with a predefined set of possible key values.

1.1 Example

```
from tri.named_struct import NamedStruct

class MyNamedStruct(NamedStruct):
    foo = NamedStructField()
    bar = NamedStructField()

m = MyNamedStruct(17, 42)
assert m['foo'] == 17
assert m.foo == 17
assert m == dict(foo=17, bar=42)

m.not_foo  # Will raise an AttributeError
```

Default values can be provided:

```
from tri.named_struct import NamedStruct

class MyNamedStruct(NamedStruct):
    foo = NamedStructField()
    bar = NamedStructField()
    baz = NamedStructField(default='default')

assert MyNamedStruct(17) == dict(foo=17, bar=None, baz='default')
```

Default values can alternatively be provided by a factory method:

```
from tri.named_struct import NamedStruct

class MyNamedStruct(NamedStruct):
    foo = NamedStructField(default_factory=list)

assert MyNamedStruct().foo == []
```

There is also a functional way to defined a NamedStruct subclass:

```
from tri.named_struct import named_struct
```

```
MyNamedStruct = named_struct('foo', bar')
m = MyNamedStruct(17, 42)
assert m.foo == 17
assert m.bar == 42
```

1.2 Running tests

You need tox installed then just `make test`.

1.3 License

BSD

1.4 Documentation

<http://trinamedstruct.readthedocs.org>.

Contents:

2.1 Installation

At the command line:

```
$ pip install tri.named_struct
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv tri.named_struct
$ pip install tri.named_struct
```

2.2 API documentation

class `tri.named_struct.NamedFrozenStruct (*args, **kwargs)`

Class extending `tri.struct.FrozenStruct` to only allow a defined subset of string keys.

__weakref__

list of weak references to the object (if defined)

classmethod `get_declared (parameter='members')`

Get the `OrderedDict` value of the parameter collected by the `@declarative` class decorator. This is the same value that would be submitted to the `__init__` invocation in the `members` argument (or another name if overridden by the `parameter` specification) @type cls: class @type parameter: str @return `OrderedDict`

classmethod `set_declared (value, parameter='members')`

@type cls: class @type value: `OrderedDict` @type parameter: str

class `tri.named_struct.NamedStruct (*args, **kwargs)`

Class extending `tri.struct.Struct` to only allow a defined subset of string keys.

classmethod `get_declared (parameter='members')`

Get the `OrderedDict` value of the parameter collected by the `@declarative` class decorator. This is the same value that would be submitted to the `__init__` invocation in the `members` argument (or another name if overridden by the `parameter` specification) @type cls: class @type parameter: str @return `OrderedDict`

classmethod `set_declared (value, parameter='members')`

@type cls: class @type value: `OrderedDict` @type parameter: str

```
class tri.named_struct.NamedStructField(*args, **kwargs)
    Field declaration for NamedStruct classes

    __ge__(other)
        x.__ge__(y) <==> x>=y

    __gt__(other)
        x.__gt__(y) <==> x>y

    __le__(other)
        x.__le__(y) <==> x<=y

    __weakref__
        list of weak references to the object (if defined)

tri.named_struct.named_frozen_struct(field_names, typename='FrozenNamedStruct')
    Procedural way to define a FrozenNamedStruct subclass, similar to the named_tuple builtin.

tri.named_struct.named_struct(field_names, typename='NamedStruct')
    Procedural way to define a NamedStruct subclass, similar to the named_tuple builtin.
```

2.3 History

2.3.1 0.12.0 (2016-08-30)

- Fix class level override of named struct fields.

2.3.2 0.11.0 (2016-05-06)

- Fix default values shadowing values set explicitly in subclass `__init__` method.

2.3.3 0.10.0 (2016-01-12)

- Added *default_factory* argument to specify default values via a callback.

2.4 Credits

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

2.5 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.

Indices and tables

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

Symbols

`__ge__()` (`tri.named_struct.NamedStructField` method), 6
`__gt__()` (`tri.named_struct.NamedStructField` method), 6
`__le__()` (`tri.named_struct.NamedStructField` method), 6
`__weakref__` (`tri.named_struct.NamedFrozenStruct` attribute), 5
`__weakref__` (`tri.named_struct.NamedStructField` attribute), 6

G

`get_declared()` (`tri.named_struct.NamedFrozenStruct` class method), 5
`get_declared()` (`tri.named_struct.NamedStruct` class method), 5

N

`named_frozen_struct()` (in module `tri.named_struct`), 6
`named_struct()` (in module `tri.named_struct`), 6
`NamedFrozenStruct` (class in `tri.named_struct`), 5
`NamedStruct` (class in `tri.named_struct`), 5
`NamedStructField` (class in `tri.named_struct`), 5

S

`set_declared()` (`tri.named_struct.NamedFrozenStruct` class method), 5
`set_declared()` (`tri.named_struct.NamedStruct` class method), 5

T

`tri.named_struct` (module), 5