
Typewriter Documentation

Jms Dnns

Oct 21, 2020

USER GUIDE

1	Example	3
2	Documentation	5
2.1	Installation	5
2.2	Quickstart	5
2.3	Type	7
2.4	Primitives	7
2.5	Record	11
2.6	Composite Types	12
2.7	Time Keeping	14
2.8	Network Addresses	15
2.9	Spatial	18
2.10	Cryptography	19
2.11	Views	21
2.12	Schematic	21
2.13	Dev Environment	21
2.14	Testing	22
2.15	Architecture	22
2.16	Contibuting	24
2.17	License	24
	Python Module Index	25
	Index	27

Data Types for Cynical Humans.

Welcome to Typewriter, a project that makes it easy to structure and process data using concepts from type systems.

EXAMPLE

A Type.

```
>>> st = types.StringType(max_length=12)
```

A Record.

```
>>> class Artist(types.Record):  
...     name = types.StringType(required=True)  
...     website = types.URLType()  
...  
>>> artist_type = Artist()
```

Validate data.

```
>>> data = 'Take Five'  
>>> st.validate(data)  
>>>  
>>> data = {  
...     'name': u'American Food',  
...     'website': 'http://soundcloud.com/americanfood'  
... }  
>>> artist_type.validate(data)
```


DOCUMENTATION

2.1 Installation

Tagged releases are available from [PyPI](#):

```
$ pip install typerighter
```

2.1.1 Python Version

TypeRighter is strictly Python 3 and up.

2.1.2 Optional Packages

The optional packages are for running unittests or generation documentation and are explained in the *Dev Environment* doc.

2.2 Quickstart

2.2.1 Types

This is a simple Type.

```
>>> st = types.StringType(max_length=12)
```

A Type doesn't store data, but it knows how to validate it.

```
>>> st.validate('short enough')
```

Errors

Exceptions are specific about errors

```
>>> st.validate('not short enough')
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
File "/home/jmsdnns/Projects/typerighter/typerighter/types/base.py", line 90, in _
    validate
    func(self, native)
File "/home/jmsdnns/Projects/typerighter/typerighter/types/domains.py", line 73, in _
    validate_max_length
    err_msg.format(value, instance.max_length)
typerighter.exceptions.ValidationException: Value length above max: short enoughhhh >
    12
```

2.2.2 Records

A Record is a structure consisting of fields, or *named type instances*.

```
>>> from typerighter import types
>>> class Artist(types.Record):
...     name = types.StringType(required=True)
...     website = types.URLType()
...
>>> artist_type = Artist()
```

A Record is a Type, so it doesn't store data, but knows how to validate it.

```
>>> data = {
...     'name': u'American Food',
...     'website': 'http://soundcloud.com/americanfood'
... }
>>> artist_type.validate(data)
```

2.2.3 Views

A View is a mutable, configurable structure that stores Record data.

```
>>> artist_view = artist_type.make_view(artist)
>>> artist_view.website = 'https://soundcloud.com/americanfood/my-take-on-take-on-me'
```

It also knows how to validate data, but assumes it validates itself.

```
>>> artist_view.validate()
```

2.3 Type

class typerighter.types.base.**Type** (*a, **kw)

This class represents the top, and thus most ambiguous, point of the Typerighter hierarchy.

It's purpose is to define the baseline expectations for every other *Type* in this library.

NATIVE

alias of `builtins.object`

is_coercible (value)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters **value** (*object*) – The value to inspect

is_falsy (value)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters **value** (*object*) – The value to inspect

Returns True or False

is_type_match (value)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native (value)

Converts a value to the native form of this type

Parameters **value** (*object*) – The value to convert

to_primitive (value)

Converts a value to the primitive form of this type

Parameters **value** (*object*) – The value to convert

to_schematic ()

Returns a Type's Schematic

validate (value)

This validation function is the primary function responsible for calling all associated validators and for managing any details related to aggregation of validation results.

Parameters **value** (*object*) – The value to convert

class typerighter.types.base.**UnsetValue**

This class exists to put a label on the type of value that represents when a field does not *_yet_* have a value.

typerighter.types.base.skip_falsy (*method*)

A decorator that intercepts method calls to prevent falsy inputs from being validated unnecessarily.

2.4 Primitives

class typerighter.types.primitives.**BooleanType** (*a, **kw)

NATIVE

alias of `builtins.bool`

is_coercible (*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters *value* (*object*) – The value to inspect

is_falsy (*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters *value* (*object*) – The value to inspect

Returns True or False

is_type_match (*value*)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native (*value*)

Converts a value to the native form of this type

Parameters *value* (*object*) – The value to convert

to_primitive (*value*)

Converts a value to the primitive form of this type

Parameters *value* (*object*) – The value to convert

to_schematic ()

Returns a Type's Schematic

validate (*value*)

This validation function is the primary function responsible for calling all associated validators and for managing any details related to aggregation of validation results.

Parameters *value* (*object*) – The value to convert

class `typerighter.types.primitives.FloatType` (*a, **kw)

NATIVE

alias of `builtins.float`

is_coercible (*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters *value* (*object*) – The value to inspect

is_falsy (*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters *value* (*object*) – The value to inspect

Returns True or False

is_type_match (*value*)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native (*value*)

Converts a value to the native form of this type

Parameters *value* (*object*) – The value to convert

to_primitive (*value*)

Converts a value to the primitive form of this type

Parameters *value* (*object*) – The value to convert

to_schematic()
Returns a Type's Schematic

validate (*value*)
This validation function is the primary function responsible for calling all associated validators and for managing any details related to aggregation of validation results.

Parameters *value* (*object*) – The value to convert

class typerighter.types.primitives.**IntegerType** (**a*, ***kw*)

NATIVE
alias of `builtins.int`

is_coercible (*value*)
Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters *value* (*object*) – The value to inspect

is_falsy (*value*)
Checks a value and responds saying whether the *Type* considers it falsy.

Parameters *value* (*object*) – The value to inspect

Returns True or False

is_type_match (*value*)
Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native (*value*)
Converts a value to the native form of this type

Parameters *value* (*object*) – The value to convert

to_primitive (*value*)
Converts a value to the primitive form of this type

Parameters *value* (*object*) – The value to convert

to_schematic()
Returns a Type's Schematic

validate (*value*)
This validation function is the primary function responsible for calling all associated validators and for managing any details related to aggregation of validation results.

Parameters *value* (*object*) – The value to convert

class typerighter.types.primitives.**Number** (**a*, ***kw*)

NATIVE
alias of `builtins.object`

is_coercible (*value*)
Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters *value* (*object*) – The value to inspect

is_falsy (*value*)
Checks a value and responds saying whether the *Type* considers it falsy.

Parameters **value** (*object*) – The value to inspect

Returns True or False

is_type_match (*value*)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native (*value*)

Converts a value to the native form of this type

Parameters **value** (*object*) – The value to convert

to_primitive (*value*)

Converts a value to the primitive form of this type

Parameters **value** (*object*) – The value to convert

to_schematic ()

Returns a Type's Schematic

validate (*value*)

This validation function is the primary function responsible for calling all associated validators and for managing any details related to aggregation of validation results.

Parameters **value** (*object*) – The value to convert

class typerighter.types.primitives.**Primitive** (*a, **kw)

NATIVE

alias of builtins.object

is_coercible (*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters **value** (*object*) – The value to inspect

is_falsy (*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters **value** (*object*) – The value to inspect

Returns True or False

is_type_match (*value*)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native (*value*)

Converts a value to the native form of this type

Parameters **value** (*object*) – The value to convert

to_primitive (*value*)

Converts a value to the primitive form of this type

Parameters **value** (*object*) – The value to convert

to_schematic ()

Returns a Type's Schematic

validate (*value*)

This validation function is the primary function responsible for calling all associated validators and for managing any details related to aggregation of validation results.

Parameters **value** (*object*) – The value to convert

```
class typerighter.types.primitives.StringType(*a, **kw)
```

NATIVE

alias of `builtins.str`

is_coercible(*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters *value* (*object*) – The value to inspect

is_falsy(*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters *value* (*object*) – The value to inspect

Returns True or False

is_type_match(*value*)

Checks if a value is an instance of this Type's native type. :param *object value*: The value to inspect

to_native(*value*)

Converts a value to the native form of this type

Parameters *value* (*object*) – The value to convert

to_primitive(*value*)

Converts a value to the primitive form of this type

Parameters *value* (*object*) – The value to convert

to_schematic()

Returns a Type's Schematic

validate(*value*)

This validation function is the primary function responsible for calling all associated validators and for managing any details related to aggregation of validation results.

Parameters *value* (*object*) – The value to convert

2.5 Record

```
class typerighter.types.records.Record(*a, **kw)
```

NATIVE

alias of `builtins.dict`

is_coercible(*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters *value* (*object*) – The value to inspect

is_falsy(*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters *value* (*object*) – The value to inspect

Returns True or False

is_type_match (*value*)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_schematic ()

Returns a Type's Schematic

validate (*value*)

This validation function is the primary function responsible for calling all associated validators and for managing any details related to aggregation of validation results.

Parameters *value* (*object*) – The value to convert

2.6 Composite Types

class typerighter.types.composites.**Container** (**a*, ***kw*)

NATIVE

alias of `builtins.object`

is_coercible (*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters *value* (*object*) – The value to inspect

is_falsy (*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters *value* (*object*) – The value to inspect

Returns True or False

is_type_match (*value*)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native (*value*)

Converts a value to the native form of this type

Parameters *value* (*object*) – The value to convert

to_primitive (*value*)

Converts a value to the primitive form of this type

Parameters *value* (*object*) – The value to convert

to_schematic ()

Returns a Type's Schematic

validate (*value*)

This validation function is the primary function responsible for calling all associated validators and for managing any details related to aggregation of validation results.

Parameters *value* (*object*) – The value to convert

class typerighter.types.composites.**ListType** (**a*, ***kw*)

NATIVE

alias of `builtins.list`

is_coercible (*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters *value* (*object*) – The value to inspect

is_falsy (*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters *value* (*object*) – The value to inspect

Returns True or False

is_type_match (*value*)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native (*value*)

Converts a value to the native form of this type

Parameters *value* (*object*) – The value to convert

to_primitive (*value*)

Converts a value to the primitive form of this type

Parameters *value* (*object*) – The value to convert

to_schematic ()

Returns a Type's Schematic

validate (*value*)

This validation function is the primary function responsible for calling all associated validators and for managing any details related to aggregation of validation results.

Parameters *value* (*object*) – The value to convert

class `typerighter.types.composites.SumType (*a, **kw)`

NATIVE

alias of `builtins.object`

is_coercible (*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters *value* (*object*) – The value to inspect

is_falsy (*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters *value* (*object*) – The value to inspect

Returns True or False

is_type_match (*value*)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native (*value*)

Converts a value to the native form of this type

Parameters *value* (*object*) – The value to convert

to_primitive (*value*)

Converts a value to the primitive form of this type

Parameters *value* (*object*) – The value to convert

to_schematic()

Returns a Type's Schematic

validate(*value*)

This validation function is the primary function responsible for calling all associated validators and for managing any details related to aggregation of validation results.

Parameters *value* (*object*) – The value to convert

2.7 Time Keeping

class `typerighter.types.timekeeping.DateTimeType(*a, **kw)`

NATIVE

alias of `datetime.datetime`

is_coercible(*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters *value* (*object*) – The value to inspect

is_falsy(*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters *value* (*object*) – The value to inspect

Returns True or False

is_type_match(*value*)

Checks if a value is an instance of this Type's native type. :param *object value*: The value to inspect

to_schematic()

Returns a Type's Schematic

class `typerighter.types.timekeeping.TimeType(*a, **kw)`

NATIVE

alias of `datetime.time`

is_coercible(*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters *value* (*object*) – The value to inspect

is_falsy(*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters *value* (*object*) – The value to inspect

Returns True or False

is_type_match(*value*)

Checks if a value is an instance of this Type's native type. :param *object value*: The value to inspect

to_schematic()

Returns a Type's Schematic

2.8 Network Addresses

```
class typerighter.types.net.EmailType(*a, **kw)
```

NATIVE

alias of `builtins.str`

is_coercible(*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters *value* (*object*) – The value to inspect

is_falsy(*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters *value* (*object*) – The value to inspect

Returns True or False

is_type_match(*value*)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native(*value*)

Converts a value to the native form of this type

Parameters *value* (*object*) – The value to convert

to_primitive(*value*)

Converts a value to the primitive form of this type

Parameters *value* (*object*) – The value to convert

to_schematic()

Returns a Type's Schematic

```
class typerighter.types.net.IPAddressType(*a, **kw)
```

NATIVE

alias of `builtins.str`

is_coercible(*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters *value* (*object*) – The value to inspect

is_falsy(*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters *value* (*object*) – The value to inspect

Returns True or False

is_type_match(*value*)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native(*value*)

Converts a value to the native form of this type

Parameters *value* (*object*) – The value to convert

to_primitive (*value*)

Converts a value to the primitive form of this type

Parameters **value** (*object*) – The value to convert

to_schematic ()

Returns a Type's Schematic

class `typerighter.types.net.IPv4Type` (**a*, ***kw*)

NATIVE

alias of `builtins.str`

is_coercible (*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters **value** (*object*) – The value to inspect

is_falsy (*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters **value** (*object*) – The value to inspect

Returns True or False

is_type_match (*value*)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native (*value*)

Converts a value to the native form of this type

Parameters **value** (*object*) – The value to convert

to_primitive (*value*)

Converts a value to the primitive form of this type

Parameters **value** (*object*) – The value to convert

to_schematic ()

Returns a Type's Schematic

class `typerighter.types.net.IPv6Type` (**a*, ***kw*)

NATIVE

alias of `builtins.str`

is_coercible (*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters **value** (*object*) – The value to inspect

is_falsy (*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters **value** (*object*) – The value to inspect

Returns True or False

is_type_match (*value*)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native (*value*)

Converts a value to the native form of this type

Parameters **value** (*object*) – The value to convert

to_primitive (*value*)

Converts a value to the primitive form of this type

Parameters **value** (*object*) – The value to convert

to_schematic ()

Returns a Type's Schematic

class typerighter.types.net.**MACAddressType** (**a*, ***kw*)

NATIVE

alias of `builtins.str`

is_coercible (*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters **value** (*object*) – The value to inspect

is_falsy (*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters **value** (*object*) – The value to inspect

Returns True or False

is_type_match (*value*)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native (*value*)

Converts a value to the native form of this type

Parameters **value** (*object*) – The value to convert

to_primitive (*value*)

Converts a value to the primitive form of this type

Parameters **value** (*object*) – The value to convert

to_schematic ()

Returns a Type's Schematic

class typerighter.types.net.**URLType** (**a*, ***kw*)

NATIVE

alias of `builtins.str`

is_coercible (*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters **value** (*object*) – The value to inspect

is_falsy (*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters **value** (*object*) – The value to inspect

Returns True or False

is_type_match (*value*)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native (*value*)

Converts a value to the native form of this type

Parameters **value** (*object*) – The value to convert

to_primitive (*value*)

Converts a value to the primitive form of this type

Parameters **value** (*object*) – The value to convert

to_schematic ()

Returns a Type's Schematic

2.9 Spatial

class typerighter.types.spatial.**GeoPointType** (**a*, ***kw*)

NATIVE

alias of `builtins.object`

is_coercible (*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters **value** (*object*) – The value to inspect

is_falsy (*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters **value** (*object*) – The value to inspect

Returns True or False

is_type_match (*value*)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native (*value*)

Converts a value to the native form of this type

Parameters **value** (*object*) – The value to convert

to_primitive (*value*)

Converts a value to the primitive form of this type

Parameters **value** (*object*) – The value to convert

to_schematic ()

Returns a Type's Schematic

validate (*value*)

This validation function is the primary function responsible for calling all associated validators and for managing any details related to aggregation of validation results.

Parameters **value** (*object*) – The value to convert

2.10 Cryptography

```
class typerighter.types.cryptography.HashType(*a, **kw)
```

NATIVE

alias of `builtins.object`

is_coercible(value)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter `is_type_match` if `self.strict` is True.

Parameters **value** (*object*) – The value to inspect

is_falsy(value)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters **value** (*object*) – The value to inspect

Returns True or False

is_type_match(value)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native(value)

Converts a value to the native form of this type

Parameters **value** (*object*) – The value to convert

to_primitive(value)

Converts a value to the primitive form of this type

Parameters **value** (*object*) – The value to convert

to_schematic()

Returns a Type's Schematic

validate(value)

This validation function is the primary function responsible for calling all associated validators and for managing any details related to aggregation of validation results.

Parameters **value** (*object*) – The value to convert

```
class typerighter.types.cryptography.MD5Type(*a, **kw)
```

NATIVE

alias of `builtins.object`

is_coercible(value)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter `is_type_match` if `self.strict` is True.

Parameters **value** (*object*) – The value to inspect

is_falsy(value)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters **value** (*object*) – The value to inspect

Returns True or False

is_type_match(value)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native (*value*)

Converts a value to the native form of this type

Parameters **value** (*object*) – The value to convert

to_primitive (*value*)

Converts a value to the primitive form of this type

Parameters **value** (*object*) – The value to convert

to_schematic ()

Returns a Type's Schematic

validate (*value*)

This validation function is the primary function responsible for calling all associated validators and for managing any details related to aggregation of validation results.

Parameters **value** (*object*) – The value to convert

class typerighter.types.cryptography.**SHA1Type** (*a, **kw)

NATIVE

alias of `builtins.object`

is_coercible (*value*)

Checks a value for whether or not it can be converted to the correct type. Falls back to the stricter *is_type_match* if *self.strict* is True.

Parameters **value** (*object*) – The value to inspect

is_falsy (*value*)

Checks a value and responds saying whether the *Type* considers it falsy.

Parameters **value** (*object*) – The value to inspect

Returns True or False

is_type_match (*value*)

Checks if a value is an instance of this Type's native type. :param object value: The value to inspect

to_native (*value*)

Converts a value to the native form of this type

Parameters **value** (*object*) – The value to convert

to_primitive (*value*)

Converts a value to the primitive form of this type

Parameters **value** (*object*) – The value to convert

to_schematic ()

Returns a Type's Schematic

validate (*value*)

This validation function is the primary function responsible for calling all associated validators and for managing any details related to aggregation of validation results.

Parameters **value** (*object*) – The value to convert

2.11 Views

class typerighter.views.**Field**(*name*)

A descriptor used to join a mutable View instance, that stores data, with the immutable Type instance, that only defines methods for operating on data.

class typerighter.views.**View**(*record*, *data=None*)

A View combines a *Record* with a dictionary to provide an object modeled after the record that can store data in a familiar object oriented manner.

typerighter.views.**make_view**(*record*, *data=None*)

Takes both a record and some data and produces View instance.

Parameters

- **record** (*Type*) – The type that defines the view’s shape
- **data** (*dict*) – Any initial data for the view’s fields

2.12 Schematic

class typerighter.schematics.**Schematic**(*klass*)

A Schematic is a object that maintains a Type’s argspec. It exists as a class to provide a namespace for relevant values.

typerighter.schematics.**extract_argspec**(*klass*)

Inspects a class and creates a dict of keyword arguments and their default values.

Parameters **klass** (*class*) – The class definition to inspect

Returns a dictionary of default values found in argspec for class’s init

typerighter.schematics.**init_arg_capture**(*method*)

A decorator that wraps a Type’s `__init__` method for the purpose of capturing the arguments used when a Type is instantiated so it can then update the instance’s argspec with what was actually used.

2.13 Dev Environment

First, install Typerighter from source and the packages we use for development.

```
$ git clone https://github.com/jmsdnns/typerighter
$ cd typerighter
$ pip install -e .[dev,docs]
```

Verify all tests are passing

```
$ pytest tests
===== test session starts =====
...
===== 69 passed in 0.16s =====
```

Nice.

2.14 Testing

Typewriter uses *pytest*.

```
$ pytest tests
===== test session starts =====
platform darwin -- Python 3.7.4, pytest-5.2.1, py-1.8.0, pluggy-0.13.0
rootdir: ...
plugins: cov-2.8.1
collected 69 items

tests/test_booleantypes.py .... [ 5%]
tests/test_cache.py . [ 7%]
tests/test_datetimetypes.py .. [ 10%]
tests/test_emailtype.py .. [ 13%]
tests/test_ipaddresstypes.py .... [ 18%]
tests/test_listtypes.py ..... [ 27%]
tests/test_macaddresses.py .. [ 30%]
tests/test_primitives.py ..... [ 43%]
tests/test_records.py ..... [ 60%]
tests/test_schematics.py . [ 62%]
tests/test_stringtypes.py ..... [ 72%]
tests/test_sumtypes.py .... [ 78%]
tests/test_types.py ..... [ 92%]
tests/test_urltypes.py .. [ 95%]
tests/test_views.py ... [100%]

===== 69 passed in 0.16s =====
```

Nice.

2.15 Architecture

TypeRighter is a toolkit for structuring, validating, and reshaping data.

Using the toolkit means using one or more of the following things:

- **Type**: a classification of some data which describes how to verify arbitrary data for coherence.
- **Record**: a structure of data that has type instances, called `_fields_`, for attributes.
- **Schematic**: the map of arguments used to instantiate either a **Type** or a **Record**.
- **View**: a class that let's you interact with a **Record** and some data as though it were an object instance.

2.15.1 Metaprogramming

The design of both `Type` and `Record` relies on metaprogramming to collect information about the way you choose to use them.

Generally speaking, metaprogramming is a way for programs to treat code like data. We can read, generate, analyze, or transform code, or modify it while running.

More specifically, TypeRighter can inspect the attributes and functions on any type at the moment a user creates one. This allows it to:

- make lists of all member variables

- make a list of all functions that start with *someprefix_*

And with that metadata users can:

- map out the steps for complex data validation
- generate a SQL statement automatically
- easily define datatype conversion pipelines

2.15.2 Attributes

All Type and Record definitions have values for:

- `_validate_functions`
- `_schematic`

Records use two extra fields:

- `_fields`
- `_field_functions`

Types

A type's `validate()` function will call each function listed in `_validate_functions` on its input.

The metaclass can be told about new validation functions by adding functions with `validate_` as a prefix, ie. `validate_uppercase`.

```
class StrictStringType(StringType):
    def validate_strict(self, value):
        ...
```

Records

Records introduce the concept of a field by associating a name with a type. It adds two fields of metadata to the class definition.

Let's define a record with a string stored as field `s`.

```
class Foo(Record):
    s = StringType(required=True)
```

Fields defined like this are stored as `_fields`.

It is also possible to use a function to generate field values.

```
class Foo(Record):
    def field_s(self):
        return 'an actual string'
```

Functions that behave like fields have a prefix `field_`, similar to the behavior for validation functions. This field is stored as `_field_functions`.

2.16 Contibuting

aww yeah

2.17 License

aww yeah

PYTHON MODULE INDEX

t

- `typerighter.schematics`, [21](#)
- `typerighter.types.base`, [7](#)
- `typerighter.types.composites`, [12](#)
- `typerighter.types.cryptography`, [19](#)
- `typerighter.types.net`, [15](#)
- `typerighter.types.primitives`, [7](#)
- `typerighter.types.records`, [11](#)
- `typerighter.types.spatial`, [18](#)
- `typerighter.types.timekeeping`, [14](#)
- `typerighter.views`, [21](#)

INDEX

B

`BooleanType` (class in `typerighter.types.primitives`), 7

C

`Container` (class in `typerighter.types.composites`), 12

D

`DateTimeType` (class in `typerighter.types.timekeeping`), 14

E

`EmailType` (class in `typerighter.types.net`), 15

`extract_argspec()` (in module `typerighter.schematics`), 21

F

`Field` (class in `typerighter.views`), 21

`FloatType` (class in `typerighter.types.primitives`), 8

G

`GeoPointType` (class in `typerighter.types.spatial`), 18

H

`HashType` (class in `typerighter.types.cryptography`), 19

I

`init_arg_capture()` (in module `typerighter.schematics`), 21

`IntegerType` (class in `typerighter.types.primitives`), 9

`IPAddressType` (class in `typerighter.types.net`), 15

`IPv4Type` (class in `typerighter.types.net`), 16

`IPv6Type` (class in `typerighter.types.net`), 16

`is_coercible()` (`typerighter.types.base.Type` method), 7

`is_coercible()` (`typerighter.types.composites.Container` method), 12

`is_coercible()` (`typerighter.types.composites.ListType` method), 12

`is_coercible()` (`typerighter.types.composites.SumType` method), 13

`is_coercible()` (`typerighter.types.cryptography.HashType` method), 19

`is_coercible()` (`typerighter.types.cryptography.MD5Type` method), 19

`is_coercible()` (`typerighter.types.cryptography.SHA1Type` method), 20

`is_coercible()` (`typerighter.types.net.EmailType` method), 15

`is_coercible()` (`typerighter.types.net.IPAddressType` method), 15

`is_coercible()` (`typerighter.types.net.IPv4Type` method), 16

`is_coercible()` (`typerighter.types.net.IPv6Type` method), 16

`is_coercible()` (`typerighter.types.net.MACAddressType` method), 17

`is_coercible()` (`typerighter.types.net.URLType` method), 17

`is_coercible()` (`typerighter.types.primitives.BooleanType` method), 7

`is_coercible()` (`typerighter.types.primitives.FloatType` method), 8

`is_coercible()` (`typerighter.types.primitives.IntegerType` method), 9

`is_coercible()` (`typerighter.types.primitives.Number` method), 9

`is_coercible()` (`typerighter.types.primitives.Primitive` method), 10

`is_coercible()` (typer-

<code>ighter.types.primitives.StringType</code>	<code>method</code>),	<code>is_falsy()</code> (<code>typerighter.types.timekeeping.DateTimeType</code>	<code>method</code>), 14
11		<code>method</code>), 14	
<code>is_coercible()</code> (<code>typerighter.types.records.Record</code>	<code>method</code>), 11	<code>is_falsy()</code> (<code>typerighter.types.timekeeping.TimeType</code>	<code>method</code>), 14
<code>is_coercible()</code>	(<code>typer-</code>	<code>is_type_match()</code> (<code>typerighter.types.base.Type</code>	<code>method</code>), 7
<code>ighter.types.spatial.GeoPointType</code>	<code>method</code>),	<code>is_type_match()</code>	(<code>typer-</code>
18		<code>ighter.types.composites.Container</code>	<code>method</code>),
<code>is_coercible()</code>	(<code>typer-</code>	12	
<code>ighter.types.timekeeping.DateTimeType</code>	<code>method</code>), 14	<code>is_type_match()</code>	(<code>typer-</code>
<code>is_coercible()</code>	(<code>typer-</code>	<code>ighter.types.composites.ListType</code>	<code>method</code>),
<code>ighter.types.timekeeping.TimeType</code>	<code>method</code>),	13	
14		<code>is_type_match()</code>	(<code>typer-</code>
<code>is_falsy()</code> (<code>typerighter.types.base.Type</code> <code>method</code>), 7		<code>ighter.types.composites.SumType</code>	<code>method</code>),
<code>is_falsy()</code> (<code>typerighter.types.composites.Container</code>	<code>method</code>), 12	13	
<code>is_falsy()</code> (<code>typerighter.types.composites.ListType</code>	<code>method</code>), 13	<code>is_type_match()</code>	(<code>typer-</code>
<code>is_falsy()</code> (<code>typerighter.types.composites.SumType</code>	<code>method</code>), 13	<code>ighter.types.cryptography.HashType</code>	<code>method</code>),
<code>is_falsy()</code> (<code>typerighter.types.cryptography.HashType</code>	<code>method</code>), 19	19	
<code>is_falsy()</code> (<code>typerighter.types.cryptography.MD5Type</code>	<code>method</code>), 19	<code>is_type_match()</code>	(<code>typer-</code>
<code>is_falsy()</code> (<code>typerighter.types.cryptography.SHA1Type</code>	<code>method</code>), 20	<code>ighter.types.cryptography.MD5Type</code>	<code>method</code>),
<code>is_falsy()</code> (<code>typerighter.types.cryptography.SHA1Type</code>	<code>method</code>), 20	19	
<code>is_falsy()</code> (<code>typerighter.types.net.EmailType</code> <code>method</code>),	15	<code>is_type_match()</code>	(<code>typer-</code>
<code>is_falsy()</code> (<code>typerighter.types.net.IPAddressType</code>	<code>method</code>), 15	<code>ighter.types.cryptography.SHA1Type</code>	<code>method</code>),
<code>is_falsy()</code> (<code>typerighter.types.net.IPv4Type</code> <code>method</code>),	16	20	
<code>is_falsy()</code> (<code>typerighter.types.net.IPv6Type</code> <code>method</code>),	16	<code>is_type_match()</code> (<code>typerighter.types.net.EmailType</code>	<code>method</code>), 15
<code>is_falsy()</code> (<code>typerighter.types.net.MACAddressType</code>	<code>method</code>), 17	<code>is_type_match()</code>	(<code>typer-</code>
<code>is_falsy()</code> (<code>typerighter.types.net.URLType</code> <code>method</code>),	17	<code>ighter.types.net.IPAddressType</code>	<code>method</code>),
<code>is_falsy()</code> (<code>typerighter.types.primitives.BooleanType</code>	<code>method</code>), 8	15	
<code>is_falsy()</code> (<code>typerighter.types.primitives.FloatType</code>	<code>method</code>), 8	<code>is_type_match()</code> (<code>typerighter.types.net.IPv4Type</code>	<code>method</code>), 16
<code>is_falsy()</code> (<code>typerighter.types.primitives.IntegerType</code>	<code>method</code>), 9	<code>is_type_match()</code> (<code>typerighter.types.net.IPv6Type</code>	<code>method</code>), 16
<code>is_falsy()</code> (<code>typerighter.types.primitives.Number</code>	<code>method</code>), 9	<code>is_type_match()</code>	(<code>typer-</code>
<code>is_falsy()</code> (<code>typerighter.types.primitives.Primitive</code>	<code>method</code>), 10	<code>ighter.types.net.MACAddressType</code>	<code>method</code>),
<code>is_falsy()</code> (<code>typerighter.types.primitives.StringType</code>	<code>method</code>), 11	17	
<code>is_falsy()</code> (<code>typerighter.types.records.Record</code>	<code>method</code>), 11	<code>is_type_match()</code> (<code>typerighter.types.net.URLType</code>	<code>method</code>), 17
<code>is_falsy()</code> (<code>typerighter.types.spatial.GeoPointType</code>	<code>method</code>), 18	<code>is_type_match()</code>	(<code>typer-</code>
		<code>ighter.types.primitives.BooleanType</code>	<code>method</code>),
		8	
		<code>is_type_match()</code>	(<code>typer-</code>
		<code>ighter.types.primitives.FloatType</code>	<code>method</code>),
		8	
		<code>is_type_match()</code>	(<code>typer-</code>
		<code>ighter.types.primitives.IntegerType</code>	<code>method</code>),
		9	
		<code>is_type_match()</code>	(<code>typer-</code>
		<code>ighter.types.primitives.Number</code>	<code>method</code>),
		10	
		<code>is_type_match()</code>	(<code>typer-</code>
		<code>ighter.types.primitives.Primitive</code>	<code>method</code>),
		10	
		<code>is_type_match()</code>	(<code>typer-</code>

`ighter.types.primitives.StringType` *method*),
11
`is_type_match()` (*typerighter.types.records.Record*
method), 11
`is_type_match()` (*typer-*
ighter.types.spatial.GeoPointType *method*),
18
`is_type_match()` (*typer-*
ighter.types.timekeeping.DateTimeType
method), 14
`is_type_match()` (*typer-*
ighter.types.timekeeping.TimeType *method*),
14

L

`ListType` (class in *typerighter.types.composites*), 12

M

`MACAddressType` (class in *typerighter.types.net*), 17
`make_view()` (in module *typerighter.views*), 21
`MD5Type` (class in *typerighter.types.cryptography*), 19
module
 typerighter.schematics, 21
 typerighter.types.base, 7
 typerighter.types.composites, 12
 typerighter.types.cryptography, 19
 typerighter.types.net, 15
 typerighter.types.primitives, 7
 typerighter.types.records, 11
 typerighter.types.spatial, 18
 typerighter.types.timekeeping, 14
 typerighter.views, 21

N

NATIVE (*typerighter.types.base.Type* attribute), 7
NATIVE (*typerighter.types.composites.Container* at-
tribute), 12
NATIVE (*typerighter.types.composites.ListType* at-
tribute), 12
NATIVE (*typerighter.types.composites.SumType* at-
tribute), 13
NATIVE (*typerighter.types.cryptography.HashType* at-
tribute), 19
NATIVE (*typerighter.types.cryptography.MD5Type* at-
tribute), 19
NATIVE (*typerighter.types.cryptography.SHA1Type* at-
tribute), 20
NATIVE (*typerighter.types.net.EmailType* attribute), 15
NATIVE (*typerighter.types.net.IPAddressType* attribute),
15
NATIVE (*typerighter.types.net.IPv4Type* attribute), 16
NATIVE (*typerighter.types.net.IPv6Type* attribute), 16
NATIVE (*typerighter.types.net.MACAddressType* at-
tribute), 17

NATIVE (*typerighter.types.net.URLType* attribute), 17
NATIVE (*typerighter.types.primitives.BooleanType* at-
tribute), 7
NATIVE (*typerighter.types.primitives.FloatType* at-
tribute), 8
NATIVE (*typerighter.types.primitives.IntegerType* at-
tribute), 9
NATIVE (*typerighter.types.primitives.Number* attribute),
9
NATIVE (*typerighter.types.primitives.Primitive* at-
tribute), 10
NATIVE (*typerighter.types.primitives.StringType* at-
tribute), 11
NATIVE (*typerighter.types.records.Record* attribute), 11
NATIVE (*typerighter.types.spatial.GeoPointType* at-
tribute), 18
NATIVE (*typerighter.types.timekeeping.DateTimeType*
attribute), 14
NATIVE (*typerighter.types.timekeeping.TimeType* at-
tribute), 14
`Number` (class in *typerighter.types.primitives*), 9

P

`Primitive` (class in *typerighter.types.primitives*), 10

R

`Record` (class in *typerighter.types.records*), 11

S

`Schematic` (class in *typerighter.schematics*), 21
`SHA1Type` (class in *typerighter.types.cryptography*), 20
`skip_falsy()` (in module *typerighter.types.base*), 7
`StringType` (class in *typerighter.types.primitives*), 10
`SumType` (class in *typerighter.types.composites*), 13

T

`TimeType` (class in *typerighter.types.timekeeping*), 14
`to_native()` (*typerighter.types.base.Type* method), 7
`to_native()` (*typerighter.types.composites.Container*
method), 12
`to_native()` (*typerighter.types.composites.ListType*
method), 13
`to_native()` (*typerighter.types.composites.SumType*
method), 13
`to_native()` (*typer-*
ighter.types.cryptography.HashType *method*),
19
`to_native()` (*typer-*
ighter.types.cryptography.MD5Type *method*),
19
`to_native()` (*typer-*
ighter.types.cryptography.SHA1Type *method*),
20

<code>to_native()</code> (<i>typerighter.types.net.EmailType method</i>), 15	<code>to_primitive()</code> (<i>typerighter.types.net.IPv6Type method</i>), 17
<code>to_native()</code> (<i>typerighter.types.net.IPAddressType method</i>), 15	<code>to_primitive()</code> (<i>typerighter.types.net.MACAddressType method</i>), 17
<code>to_native()</code> (<i>typerighter.types.net.Ipv4Type method</i>), 16	<code>to_primitive()</code> (<i>typerighter.types.net.URLType method</i>), 18
<code>to_native()</code> (<i>typerighter.types.net.Ipv6Type method</i>), 16	<code>to_primitive()</code> (<i>typerighter.types.primitives.BooleanType method</i>), 8
<code>to_native()</code> (<i>typerighter.types.net.MACAddressType method</i>), 17	<code>to_primitive()</code> (<i>typerighter.types.primitives.FloatType method</i>), 8
<code>to_native()</code> (<i>typerighter.types.net.URLType method</i>), 18	<code>to_primitive()</code> (<i>typerighter.types.primitives.IntegerType method</i>), 9
<code>to_native()</code> (<i>typerighter.types.primitives.BooleanType method</i>), 8	<code>to_primitive()</code> (<i>typerighter.types.primitives.Number method</i>), 10
<code>to_native()</code> (<i>typerighter.types.primitives.FloatType method</i>), 8	<code>to_primitive()</code> (<i>typerighter.types.primitives.Primitive method</i>), 10
<code>to_native()</code> (<i>typerighter.types.primitives.IntegerType method</i>), 9	<code>to_primitive()</code> (<i>typerighter.types.primitives.StringType method</i>), 11
<code>to_native()</code> (<i>typerighter.types.primitives.Number method</i>), 10	<code>to_primitive()</code> (<i>typerighter.types.spatial.GeoPointType method</i>), 18
<code>to_native()</code> (<i>typerighter.types.primitives.Primitive method</i>), 10	<code>to_primitive()</code> (<i>typerighter.types.base.Type method</i>), 7
<code>to_native()</code> (<i>typerighter.types.primitives.StringType method</i>), 11	<code>to_primitive()</code> (<i>typerighter.types.composites.Container method</i>), 12
<code>to_native()</code> (<i>typerighter.types.spatial.GeoPointType method</i>), 18	<code>to_primitive()</code> (<i>typerighter.types.composites.ListType method</i>), 13
<code>to_primitive()</code> (<i>typerighter.types.base.Type method</i>), 7	<code>to_primitive()</code> (<i>typerighter.types.composites.SumType method</i>), 13
<code>to_primitive()</code> (<i>typerighter.types.composites.Container method</i>), 12	<code>to_primitive()</code> (<i>typerighter.types.cryptography.HashType method</i>), 19
<code>to_primitive()</code> (<i>typerighter.types.composites.ListType method</i>), 13	<code>to_primitive()</code> (<i>typerighter.types.cryptography.MD5Type method</i>), 20
<code>to_primitive()</code> (<i>typerighter.types.composites.SumType method</i>), 13	<code>to_primitive()</code> (<i>typerighter.types.cryptography.SHA1Type method</i>), 20
<code>to_primitive()</code> (<i>typerighter.types.cryptography.HashType method</i>), 19	<code>to_primitive()</code> (<i>typerighter.types.net.EmailType method</i>), 15
<code>to_primitive()</code> (<i>typerighter.types.cryptography.MD5Type method</i>), 20	<code>to_primitive()</code> (<i>typerighter.types.net.IPAddressType method</i>), 15
<code>to_primitive()</code> (<i>typerighter.types.cryptography.SHA1Type method</i>), 20	<code>to_primitive()</code> (<i>typerighter.types.net.Ipv4Type method</i>), 16
<code>to_primitive()</code> (<i>typerighter.types.net.EmailType method</i>), 15	<code>to_primitive()</code> (<i>typerighter.types.net.IPv6Type method</i>), 17
<code>to_primitive()</code> (<i>typerighter.types.net.IPAddressType method</i>), 15	<code>to_primitive()</code> (<i>typerighter.types.net.MACAddressType method</i>), 17
<code>to_primitive()</code> (<i>typerighter.types.net.Ipv4Type method</i>), 16	<code>to_primitive()</code> (<i>typerighter.types.net.URLType method</i>), 18
<code>to_primitive()</code> (<i>typerighter.types.net.IPv6Type method</i>), 17	<code>to_primitive()</code> (<i>typerighter.types.primitives.BooleanType method</i>), 8
<code>to_primitive()</code> (<i>typerighter.types.net.MACAddressType method</i>), 17	<code>to_primitive()</code> (<i>typerighter.types.primitives.FloatType method</i>), 8
<code>to_primitive()</code> (<i>typerighter.types.net.URLType method</i>), 18	<code>to_primitive()</code> (<i>typerighter.types.primitives.IntegerType method</i>), 9
<code>to_primitive()</code> (<i>typerighter.types.primitives.BooleanType method</i>), 8	<code>to_primitive()</code> (<i>typerighter.types.primitives.Number method</i>), 10
<code>to_primitive()</code> (<i>typerighter.types.primitives.FloatType method</i>), 8	<code>to_primitive()</code> (<i>typerighter.types.primitives.Primitive method</i>), 10
<code>to_primitive()</code> (<i>typerighter.types.primitives.IntegerType method</i>), 9	<code>to_primitive()</code> (<i>typerighter.types.primitives.StringType method</i>), 11
<code>to_primitive()</code> (<i>typerighter.types.primitives.Number method</i>), 10	<code>to_primitive()</code> (<i>typerighter.types.spatial.GeoPointType method</i>), 18
<code>to_primitive()</code> (<i>typerighter.types.primitives.Primitive method</i>), 10	<code>to_schematic()</code> (<i>typerighter.types.base.Type method</i>), 7
<code>to_primitive()</code> (<i>typerighter.types.primitives.StringType method</i>), 11	<code>to_schematic()</code> (<i>typerighter.types.composites.Container method</i>), 12
<code>to_primitive()</code> (<i>typerighter.types.spatial.GeoPointType method</i>), 18	<code>to_schematic()</code> (<i>typerighter.types.composites.ListType method</i>), 13
<code>to_schematic()</code> (<i>typerighter.types.base.Type method</i>), 7	<code>to_schematic()</code> (<i>typerighter.types.composites.SumType method</i>), 13
<code>to_schematic()</code> (<i>typerighter.types.composites.Container method</i>), 12	<code>to_schematic()</code> (<i>typerighter.types.cryptography.HashType method</i>), 19
<code>to_schematic()</code> (<i>typerighter.types.composites.ListType method</i>), 13	<code>to_schematic()</code> (<i>typerighter.types.cryptography.MD5Type method</i>), 20
<code>to_schematic()</code> (<i>typerighter.types.composites.SumType method</i>), 13	<code>to_schematic()</code> (<i>typerighter.types.cryptography.SHA1Type method</i>), 20
<code>to_schematic()</code> (<i>typerighter.types.cryptography.HashType method</i>), 19	<code>to_schematic()</code> (<i>typerighter.types.net.EmailType method</i>), 15
<code>to_schematic()</code> (<i>typerighter.types.cryptography.MD5Type method</i>), 20	<code>to_schematic()</code> (<i>typerighter.types.net.IPAddressType method</i>), 16
<code>to_schematic()</code> (<i>typerighter.types.cryptography.SHA1Type method</i>), 20	<code>to_schematic()</code> (<i>typerighter.types.net.Ipv4Type method</i>), 16
<code>to_schematic()</code> (<i>typerighter.types.net.EmailType method</i>), 15	
<code>to_schematic()</code> (<i>typerighter.types.net.IPAddressType method</i>), 16	
<code>to_schematic()</code> (<i>typerighter.types.net.Ipv4Type method</i>), 16	

`method)`, 16
`to_schematic()` (*typerighter.types.net.IPv6Type*
`method)`, 17
`to_schematic()` (*typer-*
`ighter.types.net.MACAddressType` *method),
 17
`to_schematic()` (*typerighter.types.net.URLType*
`method)`, 18
`to_schematic()` (*typer-*
`ighter.types.primitives.BooleanType` *method),
 8
`to_schematic()` (*typer-*
`ighter.types.primitives.FloatType` *method),
 8
`to_schematic()` (*typer-*
`ighter.types.primitives.IntegerType` *method),
 9
`to_schematic()` (*typer-*
`ighter.types.primitives.Number` *method),
 10
`to_schematic()` (*typer-*
`ighter.types.primitives.Primitive` *method),
 10
`to_schematic()` (*typer-*
`ighter.types.primitives.StringType` *method),
 11
`to_schematic()` (*typerighter.types.records.Record*
`method)`, 12
`to_schematic()` (*typer-*
`ighter.types.spatial.GeoPointType` *method),
 18
`to_schematic()` (*typer-*
`ighter.types.timekeeping.DateTimeType`
`method)`, 14
`to_schematic()` (*typer-*
`ighter.types.timekeeping.TimeType` *method),
 14
Type (class in *typerighter.types.base*), 7
typerighter.schematics
 module, 21
typerighter.types.base
 module, 7
typerighter.types.composites
 module, 12
typerighter.types.cryptography
 module, 19
typerighter.types.net
 module, 15
typerighter.types.primitives
 module, 7
typerighter.types.records
 module, 11
typerighter.types.spatial
 module, 18*********

typerighter.types.timekeeping
 module, 14
typerighter.views
 module, 21

U

UnsetValue (class in *typerighter.types.base*), 7
URLType (class in *typerighter.types.net*), 17

V

`validate()` (*typerighter.types.base.Type* *method*), 7
`validate()` (*typerighter.types.composites.Container*
method), 12
`validate()` (*typerighter.types.composites.ListType*
method), 13
`validate()` (*typerighter.types.composites.SumType*
method), 14
`validate()` (*typerighter.types.cryptography.HashType*
method), 19
`validate()` (*typerighter.types.cryptography.MD5Type*
method), 20
`validate()` (*typerighter.types.cryptography.SHA1Type*
method), 20
`validate()` (*typerighter.types.primitives.BooleanType*
method), 8
`validate()` (*typerighter.types.primitives.FloatType*
method), 9
`validate()` (*typerighter.types.primitives.IntegerType*
method), 9
`validate()` (*typerighter.types.primitives.Number*
method), 10
`validate()` (*typerighter.types.primitives.Primitive*
method), 10
`validate()` (*typerighter.types.primitives.StringType*
method), 11
`validate()` (*typerighter.types.records.Record*
method), 12
`validate()` (*typerighter.types.spatial.GeoPointType*
method), 18
View (class in *typerighter.views*), 21