

**NAME**

Types::TypeTiny – type constraints used internally by Type::Tiny

**STATUS**

This module is covered by the Type-Tiny stability policy.

**DESCRIPTION**

Dogfooding.

This isn't a real Type::Library-based type library; that would involve too much circularity. But it exports some type constraints which, while designed for use within Type::Tiny, may be more generally useful.

**Types**

StringLike

Accepts strings and objects overloading stringification.

HashLike

Accepts hashrefs and objects overloading hashification.

ArrayLike

Accepts arrayrefs and objects overloading arrayfication.

CodeLike

Accepts coderefs and objects overloading codification.

TypeTiny

Accepts blessed Type::Tiny objects.

`_ForeignTypeConstraint`

Any reference which to\_TypeTiny recognizes as something that can be coerced to a Type::Tiny object.

Yeah, the underscore is included.

**Coercion Functions**

`to_TypeTiny($constraint)`

Promotes (or “demotes” if you prefer) a Moose::Meta::TypeConstraint object to a Type::Tiny object.

Can also handle Validation::Class objects. Type constraints built from Validation::Class objects deliberately *ignore* field filters when they do constraint checking (and go to great lengths to do so); using filters for coercion only. (The behaviour of `coerce` if we don't do that is just too weird!)

Can also handle any object providing `check` and `get_message` methods. (This includes Moose::Meta::TypeConstraint objects.) If the object also provides `has_coercion` and `coerce` methods, these will be used too.

Can also handle coderefs (but not blessed coderefs or objects overloading `&{}`). Coderefs are expected to return true iff `$_` passes the constraint. If `$_` fails the type constraint, they may either return false, or die with a helpful error message.

**Methods**

These are implemented so that `Types::TypeTiny->meta->get_type($foo)` works, for rough compatibility with a real Type::Library type library.

`meta`

`type_names`

`get_type($name)`

`has_type($name)`

`coercion_names`

`get_coercion($name)`

`has_coercion($name)`

**BUGS**

Please report any bugs to <http://rt.cpan.org/Dist/Display.html?Queue=Type-Tiny>.

## **SEE ALSO**

Type::Tiny.

## **AUTHOR**

Toby Inkster <tobyink@cpan.org>.

## **COPYRIGHT AND LICENCE**

This software is copyright (c) 2013–2014, 2017–2019 by Toby Inkster.

This is free software; you can redistribute it and/or modify it under the same terms as the Perl 5 programming language system itself.

## **DISCLAIMER OF WARRANTIES**

THIS PACKAGE IS PROVIDED “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.