NAME

Type::Tiny::Manual::Policies – Type::Tiny policies

MANUAL

Type::Tiny Stability Policy

Type::Tiny 1.000000 is considered stable. Any changes to the API that are big enough to *require* changes to the test suite will be preceded by a *six month* notice period, with the following exceptions:

- Any changes which are necessary to maintain compatibility with new releases of Moose, Moo, and other software that Type::Tiny needs to integrate with.

- Changes to maintain compatibility with future versions of Perl itself.

- Where a change fixes a contradiction between the implementation and documentation of Type::Tiny.

- Where a feature is explicitly documented as being ''experimental'' or ''unstable''.

- Improvements to the text of error messages.

Experimental and Unstable Type::Tiny Features

The following list is currently non-exhaustive.

- The following type checks result may vary based on your version of Perl and whether Type::Tiny::XS is installed. Their outcome is currently considered undefined, and may change in future versions.

  - Using **Object** and similar type checks on unblessed regular expression references, the outcome is undefined.

  - On all current versions of Perl, false (!!0) stringifies to the empty string (but using Devel::Peek you can tell the difference between this value and a normal empty string), so **Int** and subtypes of **Int** do not consider it to be an integer. If Perl's behaviour ever changes, you might not be able to rely on this outcome. True (!!1) stringifies as ''1'', so is considered an integer.

  - A glob (not a globref but an actual glob) currently passes the **StringLike** type constraint but not the **Str** type constraint.

- Type::Tiny's `my_methods` attribute and the functionality it provides is experimental.

- The parameterizable coercion API is subject to change.

- The interaction of deep coercions and mutable coercions currently results in ill-defined behaviour. This could change at any time.

- Type::Registry's ability to import MooseX::Types and MouseX::Types type libraries is experimental.

- These modules are considered part of Type::Tiny's internals, and not covered by the stability policy: Devel::TypeTiny::Perl56Compat, Devel::TypeTiny::Perl58Compat, Type::Coercion::FromMoose, Type::Tiny::_HalfOp, Types::Standard::ArrayRef, Types::Standard::CycleTuple, Types::Standard::Dict, Types::Standard::HashRef, Types::Standard::Map, Types::Standard::ScalarRef, Types::Standard::StrMatch, Types::Standard::Tied, and Types::Standard::Tuple.

- Reply::Plugin::TypeTiny is not covered by the stability policy.

Type::Tiny Versioning Policy

As of 1.000000, this distribution follows a versioning scheme similar to Moo, which is based on a semver <http://semver.org/>–like three component version number, but with the last two components each represented by three decimal digits in the fractional part of the version number. That is, version 1.003002 of the software represents ''1.3.2''.

Additionally, releases where the second component is an odd number will be considered unstable/trial releases. (These will also include an underscore in the version number as per the usual CPAN convention.)

NEXT STEPS

Here's your next step:

- Type::Tiny::Manual::Contributing

  Contributing to Type::Tiny development.

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 MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.