NAME

MooseX::Meta::TypeConstraint::ForceCoercion − Force coercion when validating type constraints

VERSION

version 0.01

SYNOPSIS

```
    use MooseX::Types:::Moose qw/Str Any/;
    use Moose::Util::TypeConstraints;
    use MooseX::Meta::TypeConstraint::ForceCoercion;

    # get any type constraint
    my $tc = Str;

    # declare one or more coercions for it
    coerce $tc,
        from Any,
        via { ... };

    # wrap the $tc to force coercion
    my $coercing_tc = MooseX::Meta::TypeConstraint::ForceCoercion−>new(
        type_constraint => $tc,
    );

    # check a value against new type constraint. this will run the type
    # coercions for the wrapped type, even if the value already passes
    # validation before coercion. it will fail if the value couldn't be
    # coerced
    $coercing_tc−>check('Affe');
```

DESCRIPTION

This class allows to wrap any `Moose::Meta::TypeConstraint` in a way that will force coercion of the value when checking or validating a value against it.

ATTRIBUTES

type_constraint

The type constraint to wrap. All methods except for `validate` and `check` are delegated to the value of this attribute.

METHODS

check ($value)

Same as `Moose::Meta::TypeConstraint::check`, except it will always try to coerce `$value` before checking it against the actual type constraint. If coercing fails the check will fail, too.

validate ($value, `$coerced_ref`?)

Same as `Moose::Meta::TypeConstraint::validate`, except it will always try to coerce `$value` before validating it against the actual type constraint. If coercing fails the validation will fail, too.

If coercion was successful and a `$coerced_ref` references was passed, the coerced value will be stored in that.

AUTHOR

```
  Florian Ragwitz <rafl AT debian DOT org>
```

COPYRIGHT AND LICENSE