

PIOTR SOLNICA

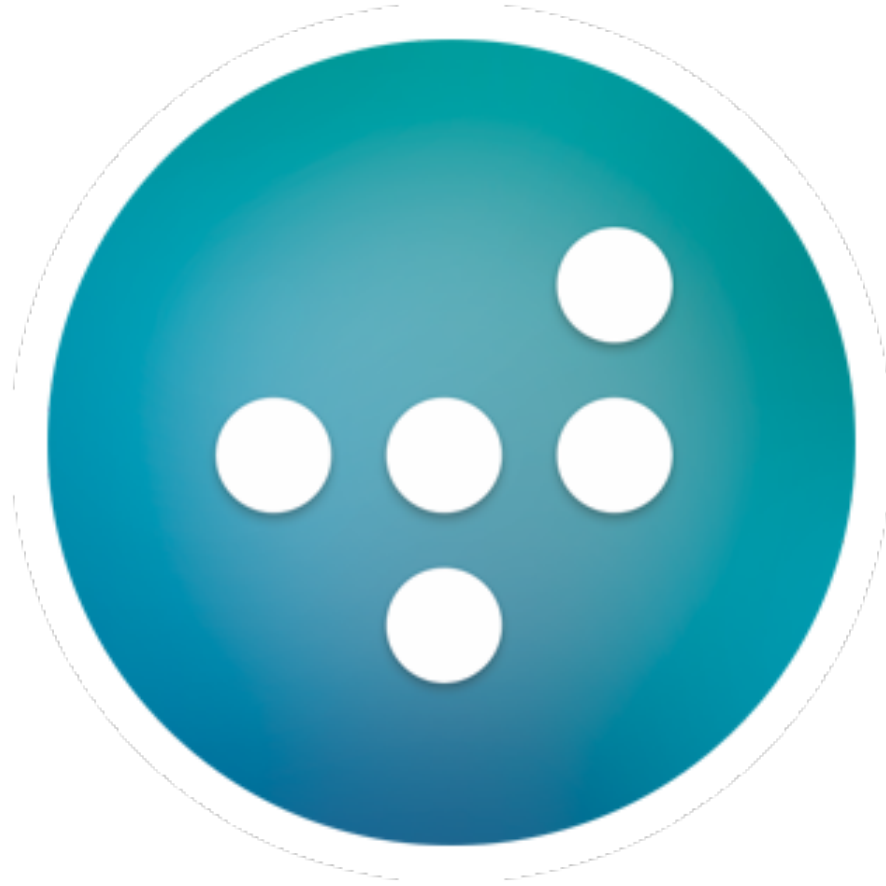
DRY-RB – NEXT GENERATION RUBY LIBRARIES



rom-rb.org



dry-rb.org



DRY-RB

22 GEMS

**ADOPTED BY OTHER
PROJECTS....**



Hanami is a modern web framework for Ruby.

TRAILBLAZER

A High-level Architecture For The Web



dry-rb is a collection of next-generation Ruby libraries,
each intended to encapsulate a common task



Non-intrusive



Flexible



Fast!

NON-INTRUSIVE



```
class MyLovelyClass
  include Dry::AmazingFunctionality::VeryPowerful::Wow
end
```

```
MyLovelyClass.methods.count
# 1202941204812049128412043
```

6 new methods / 4 methods as public API

```
class MyApp  
end
```

```
MyApp.methods.count  
# 101
```

```
MyApp.extend(Dry::Configurable)
```

```
MyApp.methods.count  
# 107
```

NO MONKEY-PATCHING

FLEXIBLE

DRY-RB POWERED VALIDATION

```
# apps/web/controllers/signup/create.rb
module Web::Controllers::Signup
  class Create
    include Web::Action

    params do
      required(:email).filled
      required(:password).filled

      required(:address).schema do
        required(:country).filled
      end
    end

    def call(params)
      puts params[:email]           # => "alice@example.org"
      puts params[:password]        # => "secret"
      puts params[:address][:country] # => "Italy"

      puts params[:admin]           # => nil
    end
  end
end
```

hanamirb.org

DRY-RB POWERED
TYPE COERCION

```
class AlbumTwin < Disposable::Twin  
  feature Coercion  
  
  property :id, type: Types::Form::Int
```

trailblazer.to

```
require 'rom-sql'
require 'rom/sql/types/pg'

class Users < ROM::Relation[:sql]
  schema do
    attribute :meta, Types::PG::JSON
    attribute :tags, Types::PG::Array
    attribute :info, Types::PG::Hash
  end
end
```

DRY-RB POWERED
SCHEMA TYPES

rom-rb.org

FAST

Calculating -----

virtus	32.346k	(± 4.7%)	i/s	-	162.894k	in	5.047797s
dry-struct	202.988k	(± 4.5%)	i/s	-	1.025M	in	5.061732s

Comparison:

dry-struct:	202988.4	i/s		
virtus:	32345.7	i/s	- 6.28x	slower

Calculating -----

ActiveModel::Validations

1.567k (± 3.1%) i/s - 7.904k in 5.047307s

dry-validation / schema

8.046k (±11.1%) i/s - 39.312k in 5.002043s

dry-validation / form

7.134k (±11.3%) i/s - 35.040k in 5.024701s

Comparison:

dry-validation / schema: 8045.9 i/s

dry-validation / form: 7134.1 i/s - same-ish: difference falls within error

ActiveModel::Validations: 1567.4 i/s - 5.13x slower

DRY-VALIDATION

**FAST, POWERFUL DATA
VALIDATION WITH
TYPE-SAFETY & SAFE COERCIONS**

VALIDATION SCHEMAS

```
require 'dry-validation'
```

```
UserSchema = Dry::Validation.Schema do  
  required(:login).value(:str?, size?: 2..64)  
  required(:age).maybe(:int?, gt?: 18)  
end
```

IS IT A STRING WITH A SIZE
BETWEEN 2 AND 64 CHARS?

CAN BE NIL BUT
IF IT'S NOT THEN IS IT AN
INTEGER AND IS IT GREATER
THAN 18?



SHORTCUT FOR `#CALL` :)

```
UserSchema.(login: "j", age: 19).errors  
# {login=>["length must be within 2 - 64"]}
```

```
UserSchema.(login: "jane", age: nil).errors  
# {}
```

FEATURES

- ▶ Powerful validation DSL with support for coercions and rules
- ▶ Support for custom predicates
- ▶ High-level rules depending on specific values
- ▶ i18n support with, or without I18n gem (yep)
- ▶ Schema, Form and JSON for different types of input
- ▶ Errors convertible to literally anything you want via custom message compilers
- ▶ Custom coercions via your own types

DRY-TYPES

A TYPE SYSTEM FOR RUBY WITH COERCIONS AND CONSTRAINTS

```
require 'dry-types'
```

```
module Types
```

```
  include Dry::Types.module
```

```
end
```

KERNEL COERCIONS

```
Types::Coercible::String[1]  
# => "1"
```

JSON COERCIONS

```
Types::Json::Date["2012-12-12"]  
# #<Date: 2012-12-12 ((2456274j,0s,0n),+0s,2299161j)>
```

```
Types::Form::Float["1.23"]  
# 1.23
```

HTTP FORM COERCIONS



DEFAULT VALUE

```
DefaultAge = Types::Strict::Int.default(33)
```

```
DefaultAge[nil] # 33
```

```
DefaultAge[35] # 35
```



NIL OR STRING

```
NilOrString = Types::Strict::Nil | Types::Strict::String
```

```
NilOrString[nil] # nil
```

```
NilOrString["foo"] # "foo"
```



VALUE CONSTRAINT

```
UserAge = Types::Strict::Int.constrained(gt: 18)
```

```
UserAge[19]
```

```
# 19
```

```
UserAge[18]
```

```
# Dry::Types::ConstraintError: 17 violates constraints (gt?(18)  
failed)
```

```
require 'dry-struct'
```

```
class User < Dry::Struct
  attribute :id, Types::Strict::Int
  attribute :name, Types::Strict::String
end
```

```
User.new(id: '1', name: "Jane")
# [User.new] "1" (String) has invalid type for :id
# (Dry::Struct::Error)
```

FEATURES

- ▶ Plenty of built-in, reusable and extendible types
- ▶ Constrained types with a built-in support for all predicates from dry-logic
- ▶ Custom type constructors
- ▶ Default values
- ▶ Enums
- ▶ Hash schemas
- ▶ Array with member type
- ▶ Maybe values using dry-monads
- ▶ Meta-data for annotations with arbitrary information

github.com/dry-rb/dry-struct

ATTRIBUTE DSL BUILT ON TOP OF DRY-TYPES

github.com/dry-rb/dry-transaction

COMPOSABLE TRANSACTIONS WITH GREAT ERROR HANDLING

github.com/dry-rb/dry-system

APPLICATION STATE MANAGEMENT WITH BOOTABLE COMPONENTS

github.com/dry-rb/dry-web-roda

WEB APPLICATION CONTAINER BASED ON DRY-SYSTEM AND RODA ROUTING TREE

github.com/dry-rb/dry-logic

**COMPOSABLE RULES WITH PREDICATE LOGIC
(DRY-TYPES & DRY-VALIDATION BACKEND)**

github.com/dry-rb/dry-initializer

FLEXIBLE DSL FOR DEFINING CONSTRUCTORS

github.com/dry-rb/dry-monads

COMMON MONADS IMPLEMENTED AS IDIOMATIC RUBY

AND MORE...



dry-rb.org

@dry_rb

Piotr Solnica

solnic.eu

@_solnic_

THANK YOU :)