schema

presented by Alex Gherega
from icslab.eu

Prismatic

SMALL COMPANY

Making real-time, personally ranked newsfeeds.

6 engineers

99% Clojure and ClojureScript

PROJECTS

Schema - forged by necessity

Plumbing/Graph

Dommy

HipHip

https://github.com/plumatic/schema

Schema rationale

It has to do with expectations

Functions

- $f: A^n \rightarrow B$
- $f(a_1, a_2, a_3, \ldots.) = b$
- Pure and solely defined by arguments

What if you don’t know A and B?

Types

Using types gives all the documentation you need

Restrictive *type-coupling*

“

we built the ***Schema*** *library for declaring and validating* *data shapes. Schema isn't a full-blown type system, but a lightweight* ***flexible DSL*** *for describing data requirements, more approriate to how Clojure functions should delimit use*

Schema usage

It has to do with expectations

41
Versions

1,393,847
Downloads on Clojars

1662
Current release

208
GItHub forks

Schema's concepts

in some detail

Schema data shapes

A Schema
A Clojure(Script) data structure describing a data shape

spec
explain

Shapes
Leafs
Variants
Collections

Schema function arguments

Schema/defn

You can use the schema defn to add argument validation

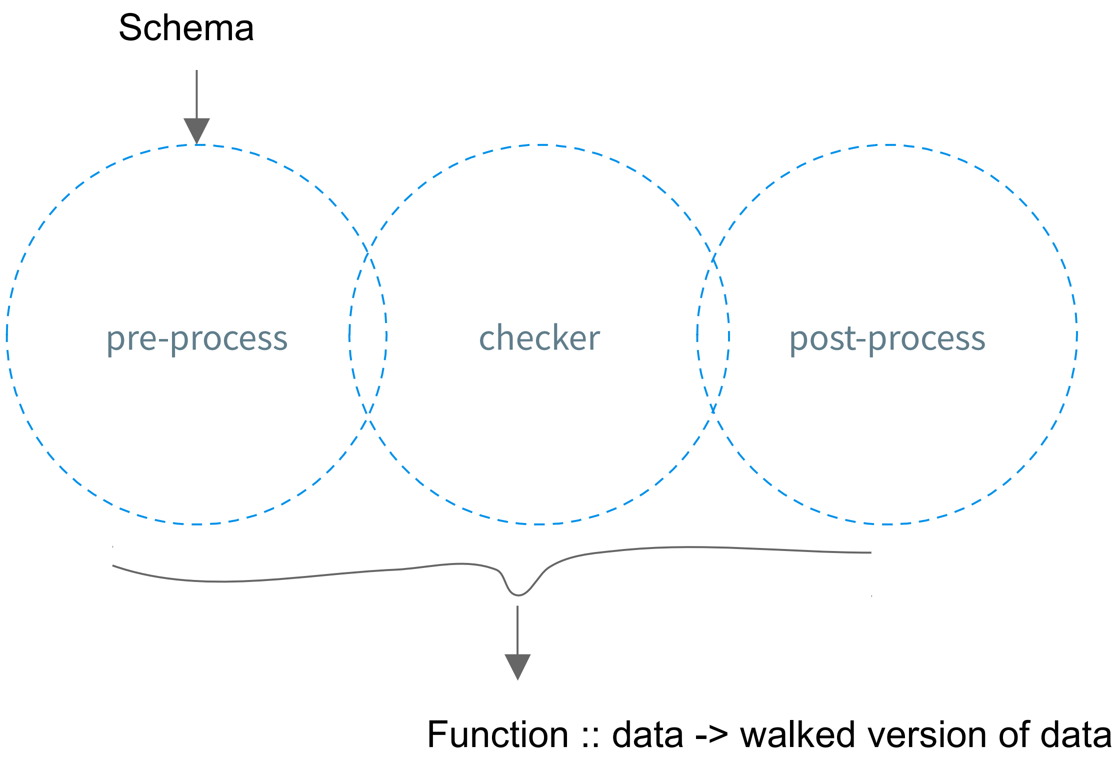
Schema Transform & Coerce

Transformation
Based on a custom walker method out of the Schema protocol.

One can write it’s own.

Two concrete impl
Validation
&
Coercion

Walker: the gist!



Let's review some concepts

Data DSL
It is a data DSL for describing data shapes in terms of Clojure's data

Gives Clarity
It solves the issue of getting the insights given by a formal type-system while avoiding the drawbacks

Non-obtuse
It works with Clojure and it does not obscure code.

Abstraction
It has an elegant way of defining the concepts of schema in terms of transformations.

Used
Impressive Clojars numbers!

Maintained?
Might fall back in the light of spec.

Thanks!

Any questions?

You can find us:

alex.gherega@gmail.com

icslabcrew@gmail.com

http://www.icslab.eu/slack