

Valua - Validation for Lua

A module for making chained validations. Create your objects, append your tests, use and reuse it!

Originally bundled with Sailor MVC Web Framework, now released as a separated module. <https://github.com/Etienne/sailor>

This module provides tools for validating values, very useful in forms, but also usable elsewhere. It works in appended chains. Create a new validation object and start chaining your test functions. If your value fails a test, it breaks the chain and does not evaluate the rest of it. It returns a boolean and an error string (nil when tests succeeded).

Valua is also available through luarocks

luarocks install valua

Usage

Example 1 - Just create, chain and use:

```
valua:new().type("string").len(3,5)("test string!") -- false, "should have 3-5 characters"
```

Example 2 - Create, chain and later use it multiple times:

```
local reusable_validation = valua:new().type("string").len(3,5)
reusable_validation("test string!") -- false, "should have 3-5 characters"
reusable_validation("test!") -- true
```

Current validation functions

- `alnum()` - Checks if string is alphanumeric.
- `boolean()` - Checks if value is a boolean.
- `compare(another_value)` - Checks if value is equal to another value.
- `contains(substr)` - Checks if a string contains a substring.
- `date()` or `date(format)` -
Checks if a string is a valid date. Default format is UK (dd/mm/yyyy). Also checks for US and ISO formats.
- `email()` - Checks if a string is a valid email address.
- `empty()` - Checks if a value is empty.

- integer() - Checks if a number is an integer;
- in_list(list) - Checks if a value is inside an array.
- len(min,max) - Checks if a string's length is between min and max.
- match(pattern) - Checks if a string matches a given pattern.
- max(n) - Checks if a number is equal or less than n.
- min(n) - Checks if a number is equal or greater than n.
- not_empty() - Checks if a value is not empty.
- no_white() - Checks if a string contains no white spaces.
- number() - Checks if a value is a number.
- string() - Checks if a value is a string.
- type(t) - Checks if a value is of type t.

Copyright (c) 2014 Etienne Dalcol

<http://etiene.net>

http://twitter.com/etiene_d

License: MIT

Inspired by Respect Validation for PHP

<https://github.com/Respect/Validation>