

# Language Independent Validation Rules (LIVR)

Viktor Turskyi  
CTO at WebbyLab

Kiev 2013

# Why we need it?

- We develop under different platforms
- We validate different types of data
- There is no universal validator in the world

**Multipurpose (user input validation,  
configs validation, contracts  
programming etc)**

**Rules must be declarative and  
language independent**

**Any number of rules for each field**

**Should return together errors for all fields**

**Exclude all fields that do not have  
validation rules described**

**Possibility to validate complex hierarchical structures**



**Easy to describe and understand  
validation**

**Should return understandable error codes(not error messages)**

**Easy to implement own rules**

**Rules should be able to change  
results output**

# Example: validate registration data

```
{  
  name: 'required',  
  email: ['required', 'email'],  
  gender: { one_of: [['male', 'female']] },  
  phone: {max_length: 10},  
  password: ['required', {min_length: 10} ]  
  password2: { equal_to_field: 'password' }  
}
```

# Syntax details

# Short form

```
password: [ 'required', {min_length: 10} ]
```

# Full form

```
password: [ {required: [ ] }, { min_length: [10] } ]
```

# Example: validate nested object

```
{  
  name: 'required',  
  phone: {max_length: 10},  
  address: { 'nested_object': {  
    city: 'required',  
    zip: ['required', 'positive_integer']  
  }}  
}
```

# Example: validate list

```
{  
  order_id: ['required', 'positive_integer'],  
  product_ids: { 'list_of': [[  
    'required',  
    'positive_integer'  
  ]] }  
}
```



# Example: validate list of objects

```
{  
  order_id: ['required', 'positive_integer'],  
  products: [ 'not_empty_list', { 'list_of_objects': {  
    product_id: ['required', 'positive_integer'],  
    quantity: ['required', 'positive_integer']  
  }}]  
}
```

# Example: output modification

```
{  
  email: [ 'trim', 'required', 'email', 'to_lc' ]  
}
```

# Example: validate different objects

```
{
  "order_id": ["required", "positive_integer"],
  "products": ["required", { "list_of_different_objects": [
    "product_type", {
      "material": {
        "product_type": "required",
        "material_id": ["required", "positive_integer"],
        "quantity": ["required", {"min_number": 1} ],
        "warehouse_id": "positive_integer"
      },
      "service": {
        "product_type": "required",
        "name": ["required", {"max_length": 10} ]
      }
    }
  ]
}
```

# Links

## LIVR:

- Specification <https://github.com/koorchik/LIVR>
- Universal test suite [https://github.com/koorchik/LIVR/tree/master/test\\_suite](https://github.com/koorchik/LIVR/tree/master/test_suite)
- Article about LIVR - <http://goo.gl/g5P6mJ>

## Perl:

- Validator::LIVR (on cpan) - validator
- LIVR::Contract (on cpan) - contract programming with LIVR

## JavaScript:

- Validator for browser and node js (BETA) - <https://github.com/koorchik/js-validator-livr>

## PHP:

- validator (in progress) - <https://github.com/koorchik/php-validator-livr>

# Viktor Turskyi

viktor@webbylab.com

<http://koorchik.blogspot.com>

<http://search.cpan.org/~koorchik/>

<https://github.com/koorchik>

**WebbyLab**

<http://webbylab.com>