

Validator

Feat. Wandu Framework

전창완

발표자 소개



전창완

- Oponiti & Allbus 서버 개발자
- Wandu Framework



목차

1. **핫한** Laravel에서의 Validator
2. 프레임워크 사용하지 않는 사람들의 대안은?
 1. Symfony Validator
 2. Respect Validation
3. Wandu Validator

가야할 길이 멀어요..;;
후딱후딱 보고 지나가쇼.

핫한 Laravel에서의 Validator

Laravel Validation

1. <https://laravel.com/docs/5.3/validation>

핫한 Laravel에서의 Validator

Laravel Validation

1. 장점

1. 역시 라라벨, 직관적이다.

2. 단점

1. 라라벨을 써야한다. (~~단독으로 사용할 수 있으면 제가 고생 할 일이 없었잖아요..~~)

핫한 Laravel에서의 Validator

Laravel Validation

1. 장점

Laravel을 안썼을 때의 대안은?

1. 역시 라라벨, 직관적이다.

2. 단점

1. 라라벨을 써야한다. (~~단독으로 사용할 수 있으면 제가 고생 할 일이 없었잖아요..~~)

Respect

Validation

1. <http://respect.github.io/Validation/docs/>

Respect

Validation

1. 장점

1. 정말 많은 Validation을 지원한다.
2. 간단하게 한두개 체크할 때는 굉장히 편하다.

2. 단점

1. 체크할 값이 많아지고, Validation 룰을 조금만 복잡하게 사용하면 복잡도가 확 증가한다.

Symfony

Validator

1. <https://symfony.com/doc/current/validation.html>

2. 이거 진짜 좋네요. (멍)

Symfony

Validator

1. 장점

1. 어노테이션 지원으로 필요한 부분에서 값을 체크할 수 있다.

2. 단점

1. PHP는 Annotation이 공식지원이 아니라서 String 파싱해서 쓸텐데, 아마도 느리다. (그래야 해요.)

3. 그외..

1. Symfony Validator는 앞쪽 두개의 Validator와는 다르다. 그래서 비교가 무의미하다.



Wandu Validator

Wandu Validator

왜 만들었을까?

1. 회사 소스가 굉장히 레거시인데, 간단하게 붙일 Validation이 필요했어요.
2. `respect/validator`를 사용하다가 조금만 복잡해지면 코드가 엉망.. 조금 더 단순한 표현이 필요했어요..

Wandu Validator

요구사항

1. 복잡한 타입의 데이터도 테스트할 수 있어야 한다. 객체를 확인하는 Validator 생성 룰이 단순해야 한다.
2. 에러 메시지(“제목을 입력해주세요”와 같은)는 나오지 않아도 정확히 어느 지점에서 잘못했는지 알아야 한다.
3. 커스텀 Validator가 쉽게 추가 되어야 한다.
4. 기왕이면 객체 구조가 ~~아름다웠으면...~~ 명확해야한다. 그리고, IDE에서 모든 소스가 추적가능해야 한다.

Wandu Validator

간단한 예제

```
use function Wandu\Validator\validator;

$validator = validator()->from([
    'username' => 'required|string|length_min:4|length_max:20',
    'password' => 'required|string|length_min:5|length_max:20',
    'email' => 'email',
    'company' => [
        'name' => 'required|string',
        'address' => 'string',
    ],
    'addresses' => validator()->collection([
        'name' => 'required|string',
        'lat' => 'float',
        'lng' => 'float',
    ]),
]);
```

Wandu Validator

간단한 예제

```
// assert / validate 두가지 지원
$valvalidator->assert([
    'username' => 'wan2land',
    'password' => 'dhksenWkd!!',
    'company' => [
        'name' => '완두컴퍼니',
        'address' => '서울 어딘가',
    ],
    'addresses' => [
        [
            'name' => '집1',
        ],
        [
            'name' => '집2',
            'lat' => 37.55,
            'lng' => 128.333,
        ],
    ],
]);
```


Wandu Validator

지원하는 두개의 메서드

1. `validate($data)` true / false 둘중 하나의 값을 반환, 잘못된 값 있으면 그 즉시 테스트 종료.
2. `assert($data)` 실패시 Exception 발생, 잘못된 값이 있어도 모든 값을 테스트 함.

Wandu Validator

에러처리

```
try {
    $validator->assert([
        'username' => 'abc',
        'company' => [
            'address' => '서울 어딘가',
        ],
        'addresses' => [
            [
                'name' => '집1',
            ],
            [
                'lat' => '잘못된값',
                'lng' => 128.333,
            ],
        ],
    ]);
} catch (Wandu\Validator\Exception\InvalidValueException $exception) {
    print_r($exception->getTypes());
}
```

Wandu Validator

에러처리

4. wan2land@JunChangwanui-MacBook-Pro-2: ~/Workspace/wandu/Framework (zsh)

➔ Framework git:(develop) ✗ php keynote.php

Array

(

[0] => length_min:4@username

[1] => required@password

[2] => required@company.name

[3] => required@addresses.1.name

[4] => float@addresses.1.lat

)

➔ Framework git:(develop) ✗

Wandu Validator

에러처리

1. 모든 에러는 “에러이름:매개변수1:매개변수2...@에러위치” 의 규칙을 따릅니다.
2. 에러 위치는 오브젝트를 그대로 따릅니다.
3. 에러 메시지를 지정하고 싶으면 배열로 알아서 적당히 사브작사브작...

Wandu Validator

Not Validator

```
$validator = validator()->from([
    'username' => 'string!numeric',
]);
try {
    $validator->assert([
        'username' => '303030',
    ]);
} catch (Wandu\Validator\Exception\InvalidValueException $exception) {
    print_r($exception->getTypes());
}
```

Wandu Validator

Not Validator

4. wan2land@JunChangwanui-MacBook-Pro-2: ~/Workspace/wandu/Framework (zsh)

➔ Framework git:(develop) ✗ php keynote.php

Array

```
(  
    [0] => not.numeric@username  
)
```

➔ Framework git:(develop) ✗

Wandu Validator

Not Validator

1. 그냥 ! 하나만 붙이면 되요.

Wandu Validator

객체로 선언하고자 한다면?

```
use function Wandu\Validator\validator;

$validator = validator()->from([
    'username' => 'required|string|length_min:4|length_max:20',
    'password' => 'required|string|length_min:5|length_max:20',
    'email' => 'email',
    'company' => [
        'name' => 'required|string',
        'address' => 'string',
    ],
    'addresses' => validator()->collection([
        'name' => 'required|string',
        'lat' => 'float',
        'lng' => 'float',
    ]),
]);
```


Wandu Validator

객체로 선언하고자 한다면?

```
$validator = new ArrayValidator([
    'username' => new PipelineValidator([
        new RequiredValidator(),
        new StringValidator(),
        new LengthMinValidator(4),
        new LengthMaxValidator(20),
    ]),
    'password' => new PipelineValidator([ /* 생략 */ ]),
    'company' => new ArrayValidator([
        'name' => new PipelineValidator([ /* 생략 */ ]),
        /* 생략 */
    ]),
    'addresses' => new CollectionValidator([
        'name' => new PipelineValidator([ /* 생략 */ ]),
        /* 생략 */
    ]),
]);
```

Wandu Validator

객체로 선언하고자 한다면?

1. 굳이, 일일이 선언하지 마세요. 내부적으로 알아서 동일한 Validator는 하나의 객체로 관리하고 있어요.
(위의 예시에서 StringValidator)

Wandu Validator

기본으로 제공하는 내역

- `required()`
- `not(\Wandu\Validator\Contracts\ValidatorInterface $validator)`
- `array(array $attributes = [])`
- `collection($rule = null) (=iterable)`
- `arrayable(array $attributes = [])`
- `object(array $properties = [])`
- `integer()`
- `boolean()`
- `float()`
- `string()`
- `integerable()`
- `floatable()`
- `numeric()`
- `stringable()`
- `printable()`
- `pipeline(array $validators = [])`
- `min(int $min)`
- `max(int $max)`
- `lengthMin(int $min)`
- `lengthMax(int $max)`
- `email(\Egulias\EmailValidator\Validation\EmailValidation $validation = null)`
- `regExp(string $pattern)`

Wandu Validator

Float는 뭐고 Floatable은 뭐예요?

1. float = 30.33

1. 타입까지 정확히 일치해야합니다.

2. floatable = “30.33”

1. float로 변환가능한 스트링도 허용합니다.

Wandu Validator

Float는 뭐고 Floatable은 뭐예요?

1. integer / integerable

2. float / floatable

3. stringable

4. array / arrayble

Wandu Validator

Float는 뭐고 Floatable은 뭐예요?

Arrayable을 통해 이런짓이 가능해요..

1. integer / integerable

2. float / floatable

3. stringable

4. array / arrayble

Wandu Validator

이런것도 가능해요...

```
$validator = validator()->collection(  
    validator()->arrayable([  
        'id' => 'integer',  
        'username' => 'required|string',  
    ])  
);
```

Wandu Validator

이런것도 가능해요...

```
$validator->assert(User::all());
```


Wandu Validator

어쩌다 보니 엘로퀀트도 지원합니다.

1. collection은 Traversable 인터페이스를 체크하는 Validator.

1. Laravel Collection도 Traversable을 구현하고 있음.

2. arrable은 ArrayAccess 인터페이스를 체크하는 Validator.

1. Laravel Model도 ArrayAccess를 구현하고 있음.

하나만 더..

Custom Validator!

Wandu Validator

Custom Validator

1. 지원하는 Validator가 몇개 없어서.. Custom Validator를 추가할 수 있게 했습니다.
2. 필요에 맞춰서 정의할 수 있습니다.

Wandu Validator

Custom Validator

```
use Wandu\Validator\Rules\ValidatorAbstract;

class OverTenValidator extends ValidatorAbstract
{
    const ERROR_TYPE = 'over_ten:{{something}}';

    protected $something = "hello";

    public function test($item)
    {
        return $item > 10;
    }
}
```

Wandu Validator

Custom Validator

1. ValidatorInterface를 구현해도 되는데, 귀찮으면 ValidatorAbstract를 구현해도 됩니다.
2. ValidatorInterface를 구현할 때는 `validate()`와 `assert()`를 각각 정의해야 합니다.
3. 하지만, ValidatorAbstract를 구현할 때는 `test()`만 정의하시면 됩니다. (에러내용이랑)

Wandu Validator

Custom Validator

```
$validator = new OverTenValidator();
```

```
$validator = validator()->from([  
    'name' => new OverTenValidator(),  
]);
```

Wandu Validator

더 쉽게 사용하고 싶다면?

```
$factory = new ValidatorFactory();  
$factory->register('Your\\Own\\Validator\\Namespace');  
$factory->setAsGlobal();  
  
$validator = validator()->overTen();  
  
$validator = validator()->from([  
    'name' => 'over_ten',  
]);
```

Wandu Validator

Custom Validator

1. Factory에 Validator의 네임스페이스를 등록합니다.
(네이밍은 반드시 PascalCase + “Validator”의 형태를 따라야 합니다.)
2. validator() 뒤에 camelCase를 통해 해당 Validator를 불러올 수 있습니다.
3. validator()->from() 에서 underbar_case 를 통해 해당 Validator를 불러 올 수 있습니다.



Wandu

슬림을 쓴다면



패키지스트에서 라이브러리를
고르고 구조를 설계하는데
엄청난 시간을 쏟고



마침내 모든 작업이 끝났더니
요새는 버려져 있고 공주님은
이미 다른 성으로 떠났습니다.



라라벨을 쓴다면



구출 작전을 엘레강스한 로직으로
포현하는데 엄청난 시간을 썼지만



결국 포기하고 라라캐스트에서
제X리 웨이에게 공주님을 대신
구해달라고 부탁드립니다





Wandu

완두가 뭐예요.. 먹는건가?

1. 혼자서 2년가까이 끄냥끄냥 거린 소스, 드디어 v3이 나왔습니다.
2. 원래는 라이브러리들이었는데 하나로 묶어서 프레임워크로 합쳤습니다. (그래서 시작버전이 v3..)
3. 메뉴얼이 굉장히 불친절합니다. (핵심) .. 도전해보세요!

Q & A ?

