

## 58. 验证规则和错误信息

学习要点：

1. 验证规则
2. 错误信息

本节课我们来学习一下数据验证的知识，这节课了解验证规则和错误信息。

### 一. 验证规则

1. 在上一节验证器定义的时候，我们采用的字符串模式，也支持数组模式；

```
protected $rule = [
    'name' => [
        'require',
        'max' => 10,
        'checkName' => '李炎恢'
    ],
    'price' => [
        'number',
        'between' => '1,100'
    ],
    'email' => 'email'
];
```

2. 数组模式在验证规则很多很乱的情况下，更容易管理，可读性更高；
3. 如果你想使用独立验证，就是手动调用验证类，而不是调用 `User.php` 验证类；
4. 这种调用方式，一般来说，就是独立、唯一，并不共享的调用方式；

```
$validate = new Validate();
// $validate->rule('name', 'require|max:10');
$validate->rule([
    'name' => 'require|max:10',
    'price' => 'number|between:1,100',
    'email' => 'email'
]);
```

5. 独立验证默认也是返回一条错误信息，如果要批量返回所有错误使用 `batch()`；

```
$validate->batch()->check($data)
```

6. 独立验证支持对象化的定义方式，但不支持在属性方式的定义；

```
$validate = new Validate();
// $validate->rule('name', ValidateRule::isRequire()->max(10));
$validate->rule([
    'name' => ValidateRule::isRequire()->max(10),
    'price' => ValidateRule::isNumber()->between('1,100'),
```

```

        'email' => ValidateRule::isEmail()
    ]);

```

7. 独立验支持闭包的自定义方式，但不支持属性方式和多规则方式；

```

$validate = new Validate();
$validate->rule([
    'name' => function ($value, $data) {
        return $value != '' ? true : '姓名不得为空';
    },
    'price'=> function ($value) {
        return $value > 0 ? true : '价格不得小于零';
    }
]);

```

## 二. 错误信息

1. 上一节课，我们可以在属性定义错误信息，如果不定义讲使用默认错误信息；
2. 如果使用默认错误信息，将字段设置描述信息，提示也会使用描述的信息；

```

protected $rule = [
    'name|姓名' => 'require|max:10',
];

```

3. 如果在控制器使用独立验证规则，也可以在控制器设置错误信息；

```

$validate->message([
    'name.require' => '姓名不可以是空的呢',
    'name.max' => '姓名太长可不好',
]);

```

4. 你也可以直接把错误信息写入到语言包里，进行调用，语言包：lang/zh-cn.php；

```

'name.require' => 'name_require' //控制器设置
'name_require' => 'name 不得为空' //语言包设置

```