

54. 验证器定义

学习要点：

1. 验证器定义

本节课我们来学习一下数据验证的知识，首先就是验证器定义。

一. 验证器定义

1. 验证器的使用，我们必须先定义它，系统提供了一条命令直接生成想要的类；

```
php think make:validate User
```

2. 这条命令会自动在应用目录下生成一个 `validate` 文件夹，并生成 `User.php` 类；

```
class User extends Validate
```

3. 自动生成了两个属性：`$rule` 表示定义规则，`$message` 表示错误提示信息；

```
protected $rule = [  
    'name'      =>    'require|max:20',          //不得为空，不得大于 20 位  
    'price'     =>    'number|between:1,100',     //必须是数值，1-100 之间  
    'email'     =>    'email'                     //邮箱格式要正确  
];
```

```
protected $message = [  
    'name.require'      =>    '姓名不得为空',  
    'name.max'          =>    '姓名不得大于 20 位',  
    'price.number'      =>    '价格必须是数字',  
    'price.between'     =>    '价格必须 1-100 之间',  
    'email'             =>    '邮箱的格式错误'  
];
```

4. 如果不设置`$message`定义的话，将提示默认的错误信息；
5. 验证器定义好了之后，我们需要进行调用测试，创建一个 `Verify.php` 控制器；

```
try {  
    validate(User::class)->check([  
        'name' => '蜡笔小新',  
        'price' => 90,  
        'email' => 'xiaoxin@163.com'  
    ]);  
} catch (ValidateException $e) {  
    dump($e->getError());  
}
```

6. 默认情况下，出现一个错误就会停止后面字段的验证，我们也可以设置批量验证；
`validate(User::class)->batch(true)...`

7. 系统提供了常用的规则让开发者直接使用，也可以自行定义独有的特殊规则；

```
protected $rule = [  
    'name' => 'require|max:20|checkName:李炎恢',  
];
```

//自定义规则，名称中不得是“李炎恢”

```
protected function checkName($value, $rule)  
{  
    return $rule != $value ? true : '名称存在非法称谓';  
}
```

8. 对于自定义规则中，一共可以有五个参数，我们分别了解一下；

```
protected function checkName($value, $rule, $data, $field, $title)  
{  
    dump($data);           //所有数据信息  
    dump($field);          //当前字段名  
    dump($title);          //字段描述，没有就是字段名  
}
```

9. 如何设置字段描述，只要在字段名用|后设置即可：

```
'name|用户名' => 'require|max:20|checkName:李炎恢',
```