60. 验证静态调用和令牌

学习要点：

1.静态调用

2.表单令牌

本节课我们来学习一下数据验证的知识，这节了解静态调用和表单令牌。

一. 静态调用

1. 静态调用，即使用 facade 模式进行调用验证，非常适合单个数据的验证；
2. 引入 facade 中的 Validate 时，和 think\Validate 冲突，只需引入一个即可；

```
//验证邮箱是否合法
dump(Validate::isEmail('bnbbs@163.com'));
//验证是否为空
dump(Validate::isRequre(''));
//验证是否为数值
dump(Validate::isNumber(10));
```

3. 静态调用返回的结果是 false 和 true，错误信息需要自行错误；
4. 静态调用，也是支持多规则验证的，使用 checkRule()方法实现；

```
//验证数值合法性
dump(Validate::checkRule(10, 'number|between:1,10'));
```

5. checkRule()不支持错误信息，需要自己实现，但支持对象化规则定义；

```
dump(Validate::checkRule(10, ValidateRule::isNumber()->between('1,10')));
```

二. 表单令牌

1. 表单令牌就是在表单中增加一个隐藏字段，随机生成一串字符，确定不是伪造；
2. 这种随机产生的字符和服务器的 session 进行对比，通过则是合法表单；
3. 首先，创建一个带有令牌的表单，在 See 控制器下的 vali 方法模版实现；

```
<form action="http://localhost/tp5.1test3/public/verify/form" method="post">
    <input type="hidden" name="__token__" value="{$Request.token}">
    <input type="submit" value="提交表单">
</form>
```

4. 为了验证系统内部的机制，可以通过打印测试出内部的构造；

```
//打印出生成的 token 随机数
echo Request::token();
//打印出保存到 session 的 token
echo Session::get('__token__');
```

5. 验证器部分，只要使用内置规则 token 即可验证，具体流程如下：

```
$data = [
    'user'          =>      input('post.user'),
    '__token__'     =>      input('post.__token__'),
];

$validate = new \think\Validate();
$validate->rule([
    'user' => 'require|token',
]);

if (!$validate->batch()->check($data)) {
    dump($validate->getError());
}
```