

## 45. 创建手动验证

学习要点：

### 1. 手动验证

本节课我们来开始学习手动验证的创建方法。

#### 一. 手动验证

1. 手动验证，和第一节快速验证是一样的，都是在控制器验证；
2. 只不过，快速验证是全自动化的，无法局部控制更多的细节；

//快速验证方式回顾

```
$validateData = $request->validate([
    'username'      => 'required|min:2|max:10',
    'password'      => 'required|min:6',
]);
```

//验证通过后得到提交的值

```
dd($validateData);
```

3. 创建手动验证，具体如下：

//手动创建验证器

```
$validator = Validator::make($request->post(), [
    'username'      => 'required|min:2|max:10',
    'password'      => 'required|min:6',
]);

//如果没有通过的话
if ($validator->fails()) {
    //跳转后，并将错误保存，还可以返回上一次填的信息
    return redirect('/task/form')->withErrors($validator)->withInput();
}
```

用户名：<input type="text" name="username" value="{{old('username')}}">

PS：快速验证，也支持 old 记忆上一次表单数据的功能；

4. 如果有多个表单，那可以设置\$errors 的多个错误包，有两种方法：

```
$validator = Validator::make($request->post(), [
    'username'      => 'required|min:2|max:10',
    'password'      => 'required|min:6',
])->validateWithBag('post');    //设置快速验证和手动验证
```

```
if ($validator->fails()) {  
    //适合手动验证  
    return redirect('/task/form')->withErrors($validator,  
    'post')->withInput();  
}  
  
{--多个错误包调用--}  
{{ $errors->post }}
```

5. 我们也可以在验证完毕之前，通过附加回调添加更多的自定义错误信息；

```
//验证钩子，  
$validator->after(function ($validator) {  
    $validator->errors()->add('info', '隐藏字段 info 的值不存在');  
});
```

6. 关于\$errors 对象，除了 any()判断， all()获取全部，还有一些别的方法；

```
{--获取指定字段第一条信息--}  
{{ $errors->first('username') }}
```

```
{--判断指定字段是否有错误信息--}  
{{ $errors->has('username') }}
```