

59. 验证场景和路由验证

学习要点：

1. 验证场景
2. 路由验证

本节课我们来学习一下数据验证的知识，这节课了解验证场景和路由验证。

一. 验证场景

1. 有时，我们并不希望所有的字段都得到验证，比如修改更新时不验证邮箱；
2. 可以在验证类 `User.php` 中，设置一个 `$scene` 属性，用来限定场景验证；
3. 上述中，`insert` 新增需要验证三个字段，而 `edit` 更新则只要验证两个字段；
4. 在控制器端，验证时，根据不同的验证手段，绑定验证场景有下面两种方式：

```
$validate->scene('edit')->check($data)
'\app\validate\User.edit'
```

5. 在验证类端，可以设置一个公共方法对场景的细节进行定义；

```
public function sceneEdit()
{
    $edit = $this->only(['name', 'price'])    //仅对两个字段验证
        ->remove('name', 'max')             //移出掉最大字段的限制
        ->append('price', 'require');       //增加一个不能为空的限制
    return $edit;
}
```

6. 注意：请不要对一个字段进行两个或以上的移出和添加，会被覆盖；
`remove('name', 'xxx|yyy|zzz')`或 `remove('name', ['xxx', 'yyy', 'zzz'])`；
而不是
`remove('name', 'xxx')->remove('name', 'yyy')->remove('name', 'zzz')`；

二. 路由验证

1. 路由验证，即在路由的参数来调用验证类进行验证，和字段验证一模一样；

```
protected $rule = [
    'id'          => 'number|between:1,10'
];
```

```
protected $message = [  
    'id.between'      =>    'id 只能是 1-10 之间',  
    'id.number'       =>    'id 必须是数字'  
];
```

```
Route::rule('read/:id','Verify/read')  
    ->validate(\app\validate\User::class, 'eidt');
```

2. 如果不实用验证器类，也可以使用独立的验证方式，也可以使用对象化；

```
Route::rule('read/:id','Verify/read')  
    ->validate([  
        'id'          =>    'number|between:1,10',  
        'email'       =>    \think\validate\ValidateRule::isEmail()  
    ], 'edit', [  
        'id.between'  =>    'id 限定在 1-10 之间',  
        'email'       =>    '邮箱格式错误'  
    ], true);
```