

## 09. 表单提交验证器

学习要点：

### 1. 验证器设置

本节课我们来开始进入学习一下添加用户表单的验证功能。

#### 一. 验证器设置

1. 首先，我们安装一个验证器类：User.php，用命令生成给你自动格式；

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

2. 提交时，调用这个验证类，并实现验证并输出错误；

```
protected $rule = [
    'username'      => 'require|min:2|max:10|chsDash|unique:user',
    '__token__'     => 'token',
    'password'      => 'require|min:6',
    'passwordnot'   => 'require|confirm:password',
    'email'         => 'require|email|unique:user',
    'agree'         => 'require|accepted',
];

protected $message = [
    'username.require'      => '用户名不得为空~',
    'username.min'         => '用户名不得小于 2 位~',
    'username.max'         => '用户名不得大于 10 位~',
    'username.chsDash'     => '用户名只能是汉字、字母、数字或下划线以及破折号~',
    'username.unique'      => '用户名已存在~',
    'password.require'     => '密码不得为空~',
    'password.min'         => '密码不得小于 6 位~',
    'passwordnot.require'  => '密码确认不得为空~',
    'passwordnot.confirm'  => '密码确认和密码不一致~',
    'email.require'        => '电子邮件不得为空~',
    'email.email'          => '电子邮件格式不正确~',
    'email.unique'         => '电子邮件已存在~',
    'agree.require'        => '必须确认协议~',
    'agree.accepted'       => '必须认同协议~'
];
```

3. 资源控制器中，save()方法进行验证调用；

```
public function save(Request $request)
{
    //将提交过来的数据交给验证类验证
    //支持批量验证和提醒
```

```
try {  
    validate(UserValidate::class)  
        ->batch(true)  
        ->check($request->param());  
} catch (ValidateException $exception) {  
    dd($exception->getError());  
}  
}
```