

73. 多语言

学习要点：

1. 多语言

本节课我们来学习一下如何配置的多语言功能，并实现调用方法。

一. 多语言

1. 如果要开启多语言切换功能，中间件定义文件 `middleware.php` 开启：

```
// 多语言加载
\think\middleware\LoadLangPack::class,
```

2. 配置文件在 `config/lang.php` 文件，默认设置的是 `zh-cn` 中文语言：

```
'zh-hans-cn' => 'zh-cn',
'detect_var'   => 'lang',           //自动监测的变量为 lang
```

3. 默认应用目录会调用 `app/lang` 目录下的语言包，我们创建三个：

```
//错误信息, zh-cn.php
```

```
return [
    'require_name' => '用户名不得为空！',
    'email_error'  => '邮箱地址不正确！',
];
```

```
//error message, en-us.php
```

```
return [
    'require_name' => 'The user name cannot be empty!',
    'email_error'  => 'Incorrect email address!',
];
```

```
//エラーメッセージ, ja-jp.php
```

```
return [
    'require_name' => 'ユーザ名は空ではいけません！',
    'email_error'  => 'メールアドレスが間違っています！',
];
```

4. 系统默认会指定：`zh-cn` 这个语言包，我们通过 `::get()` 来输出错误信息：

```
Lang::get('require_name');
lang('require_name');           //助手函数
```

5. 通过 URL 方式来切换语言，`?lang=en-us` 即可，`cookie` 自动生成；

6. 而下一次就算不带 `?lang=en-us`，默认就会执行 `cookie` 的语言包；

7. 也可以在配置文件中，设置允许的语言包，默认为空，具体如下：

```
// 允许的语言列表
'allow_lang_list' => ['zh-cn', 'en-us', 'ja-jp'],
```

8. 在模版中调用语言信息，可以用`{Think.lang.xxx}`;

```
{Think.lang.require_name}
{:lang('email_error')}
```

9. 可以在配置文件中开启多语言分组，运行使用二维数组来实现语言包定义；

```
// 是否支持语言分组
'allow_group' => true,

'user' => [
  'welcome' => '欢迎光临！',
  'login' => '登入成功！',
  'logout' => '退出成功！',
]
```

```
Lang::get('user.login');
```