

河北师大软件学院 @Software College

前端开发与HTML5 程序设计基础

王岩

2.13 MVC与ThinkPHP框架（四）

控制器

控制器基础

控制器 (Controller) 作用

- ❖ 接受请求，控制程序的执行
- ❖ 调用模型方法获取数据库数据
- ❖ 进行模板变量赋值、控制模板输出

控制器写法

- ❖ 控制器文件写在 `Application / 模块名 / Controller` 目录中
- ❖ 控制器文件命名为：控制器名称+`Controller.class.php`，控制器名称首字母大写
- ❖ 如：UserController.class.php
- ❖ 控制器类名为：控制器名称+Controller
- ❖ 控制器继承Controller类

URL路由

localhost/app/

Home模块下的Index控制器index方法

localhost/app/index.php/Home/
Index/lists

Home模块下的Index控制器lists方法

localhost/app/index.php/Home/
News/add

Home模块下的News控制器add方法

localhost/app/index.php/Home/
News/edit/id/2

Home模块下的News控制器edit方法，
附带参数id=2

localhost/app/index.php/Admin

Admin模块下的Index控制器index方法

Action参数绑定

参数绑定

- ❖ 参数绑定是通过直接绑定URL地址中的变量作为操作方法的参数，可以简化方法的定义。

```
namespace Home\Controller;
use Think\Controller;
class BlogController extends Controller{
    public function read($id){
        echo 'id=' . $id;
    }

    public function archive($year='2013', $month='01'){
        echo 'year=' . $year . '&month=' . $month;
    }
}
```

参数绑定

- ❖ 按照变量顺序绑定
 - ❖ 配置开启: 'URL_PARAMS_BIND_TYPE' => 1
 - ❖ 使用方式:
 - ❖ Action定义不需要改变
 - ❖ URL格式中, 不需要指定参数名
 - ❖ Action参数列表将按照URL参数的顺序分别获取

输入变量

传统方式

- ❖ `$id = $_GET['id']`
 - ❖ `$name = $_POST['name']`
 - ❖ `$value = $_SESSION['value']`
 - ❖ `$name = $_COOKIE['name']`
 - ❖ `$file = $_SERVER['PHP_SELF']`
-
- ❖ 不建议使用传统方式，因为没有统一的安全处理机制，后期调整将会比较麻烦

获取输入变量

❖ 使用I()方法:

❖ I('变量类型.变量名/修饰符','默认值','过滤方法','额外数据源'])

变量类型

❖ 例：

❖ I('get.id');

❖ I('post.title');

❖ I('session.userid');

❖ I('post.');

变量类型	含义
get	获取GET参数
post	获取POST参数
param	自动判断请求类型获取GET、POST或者PUT参数
request	获取REQUEST 参数
put	获取PUT 参数
session	获取 \$_SESSION 参数
cookie	获取 \$_COOKIE 参数
server	获取 \$_SERVER 参数
globals	获取 \$GLOBALS参数
path	获取 PATHINFO模式的URL参数 (3.2.2新增)
data	获取 其他类型的参数，需要配合额外数据源参数 (3.2.2新增)

变量过滤

- ❖ 没有在调用I函数的时候指定过滤方法时，系统会采用默认的过滤机制（由DEFAULT_FILTER配置）

```
// 系统默认的变量过滤机制  
'DEFAULT_FILTER'          => 'htmlspecialchars'
```

```
// 等同于 htmlspecialchars($_GET['name'])  
I('get.name');
```

自定义过滤

- ❖ 指定过滤方法: `I('get.name','strip_tags');`
- ❖ 配置过滤方法:
 - ❖ `'DEFAULT_FILTER' => 'strip_tags,htmlspecialchars'`
 - ❖ `I('get.name')`相当于
`htmlspecialchars(strip_tags($_GET['name']))`

请求类型判断

请求类型判断

- ❖ 在某些情况下需要根据当前页面的请求类型来做出不同的处理
- ❖ ThinkPHP提供几个常量用于判断请求类型

常量	说明
IS_GET	判断是否是GET方式提交
IS_POST	判断是否是POST方式提交
IS_PUT	判断是否是PUT方式提交
IS_DELETE	判断是否是DELETE方式提交
IS_AJAX	判断是否是AJAX提交
REQUEST_METHOD	当前提交类型

- ❖ 用法：`if (IS_POST) {`
- ❖ `}`

跳转和重定向

页面跳转

- ❖ 根据逻辑处理，在控制器中往往会进行跳转或者重定向页面
- ❖ ThinkPHP的跳转方法：
 - ❖ success（）方法和error（）方法，用于成功/错误提示并跳转
 - ❖ redirect（）方法

success () 、 error ()

❖ 参数：

❖ 参数1：提示信息

❖ 参数2（可选）：信息提示后的自动跳转地址

生成URL

- ❖ ThinkPHP内置提供了U方法，用于URL的动态生成，可以确保项目在移植过程中不受环境的影响。
- ❖ 使用方法：
 - ❖ U('[模块/控制器/动作]?参数' ['参数','伪静态后缀','是否跳转','显示域名'])
- ❖ 示例：

```
U('User/add') // 生成User模块的add操作的URL地址
U('Blog/read?id=1') // 生成Blog模块的read操作 并且id为1的URL地址
U('Blog/cate',array('cate_id'=>1,'status'=>1))
U('Blog/cate','cate_id=1&status=1')
U('Blog/cate?cate_id=1&status=1')
```

配置提示信息模板

- ❖ success和error方法都有对应的模板，并且是可以设置的，默认的设置是系统模板
- ❖ 默认配置信息：
 - ❖ // 默认错误跳转对应的模板文件
 - ❖ 'TMPL_ACTION_ERROR' => THINK_PATH . 'Tpl/dispatch_jump.tpl',
 - ❖ // 默认成功跳转对应的模板文件
 - ❖ 'TMPL_ACTION_SUCCESS' => THINK_PATH . 'Tpl/dispatch_jump.tpl'

redirect () 跳转

- ❖ 不提示任何内容，直接跳转：redirect ()
- ❖ `$this->redirect('News / category', array('cate_id' => 2), 5, '页面跳转中...');`
- ❖ 参数1（必选）：跳转的地址
- ❖ 参数2（可选）：跳转时的参数信息，数组形式
- ❖ 参数3（可选）：等待时间（秒）
- ❖ 参数4（可选）：显示字样

谢谢！