

35. 路由介绍和定义

学习要点：

1. 路由简介
2. 路由定义

本节课我们来了解一下路由方面的知识，然后简单的使用一下路由的功能。

一. 路由简介

1. 路由的作用就是让 URL 地址更加的规范和优雅，或者说更加简洁；
2. 设置路由对 URL 的检测、验证等一系列操作提供了极大的便利性；
3. 在 ThinkPHP5.1 中，路由是默认开启的，没有配置开关，不需要手动配置；
4. 创建一个 Address 控制器类，创建两个方法，具体如下：

```
class Address extends Controller
{
    public function index()
    {
        return 'index';
    }

    public function details($id)
    {
        return 'details 目前调用的 id : '.$id;
    }
}
```

5. 为了让我们路由的课程观看更加直观，我们采用内置服务器的方式来演示；
6. 通过命令行模式键入到当前项目目录后输入命令：php think run 启动；
7. 此时，public 目录会自动被绑定到顶级域名：127.0.0.1:8000 上；
8. 我们只要在地址栏键入：http://localhost:8000 或(127.0.0.1:8000)即可；

二. 路由定义

1. 在没有定义路由规则的情况下，我们访问 address/details 包含 id 的 URL 为：
http://localhost:8000/address/details/id/5 //或者.../id/5.html
2. 将这个 URL 定义路由规则，在根目录 route 下的 Route.php 里配置；
Route::get('details/:id', 'Address/details');
3. 当配置好路由规则后，会出现非法请求的错误，我们需要用路由规则的 URL 访问；
http://localhost:8000/details/5 //或者.../details/5.html

4. 一般来说 GET 方法是用的最多的，所以我们使用 `Route::get()` 最多，其它如下：
`Route::rule('details/:id', 'Address/xxx', 'GET');` //GET
`Route::rule('details/:id', 'Address/xxx', 'POST');` //POST
`Route::rule('details/:id', 'Address/xxx', 'GET|POST');` //GET 或 POST
5. 所有请求方式(快捷方式): GET(get)、POST(post)、DELETE(delete)、PUT(put)
PATCH(patch)、*(any, 任意请求方式)
6. 快捷方式，就是直接用 `Route::get`、`Route::post` 等方式即可，无须第三参数；
7. 当我们设置了强制路由的时候，访问首页就会报错，必须强制设置首页路由；
8. 开始强制路由，需要在 `app.php` 里面进行配置，然后配置首页路由；
`'url_route_must' => true,`
`Route::get('/', 'index');` //当写一个 `index`，表面控制器是 `Index`
9. 在路由的规则表达式中，有多种地址的配置规则，具体如下：
//静态路由
`Route::get('ad', 'address/index');`
//静态动态结合的地址
`Route::get('details/:id', 'address/details');`
//多参数静态动态结合的地址
`Route::get('search/:id/:uid', 'address/search');`
//全动态地址，不限制是否 `search` 固定
`Route::get(':search/:id/:uid', 'address/search');`
//包含可选参数的地址
`Route::get('find/:id/[:content]', 'address/find');`
//规则完全匹配的地址
`Route::get('search/:id/:uid$', 'address/search');`
//也可以开启全局完全匹配，在 `app.php` 中配置
`'route_complete_match' => true,`
10. 路由定义好之后，我们在控制器要创建这个路由地址，可以通过 `url()` 方法实现：
//不定义标识的做法
`return url('address/details', ['id'=>10]);`

//定义标识的做法
`Route::get('details/:id', 'address/details')->name('det');`
`return url('det', ['id'=>10]);`