

SpringWeb基于请求参数值分发HTTP请求

2017/10/09 23:23 - Posted By 起衣 - Tags: spring - Category: Code

1 条评论

最近加入到一个新的项目组，项目里基于 Spring Web 做 HTTP(不敢说是 REST)请求的接入，由于一些历史原因，请求的 URI 都规划成了一样的，比如只有三个请求 URI: /client/api, /server/api, /wap/api。但是规划请求里总会带一个参数 method, 不同业务请求的 method 值是不一样的，服务端不同的 Controller 处理不同 method 的 HTTP 请求。

当前的实现继承了 Spring 的 AbstractUrlHandlerMapping 类，然后使用 XML 配置的方式声明成一个 Bean，并且给 Bean 注入初始化的 map 类型的配置: key是不同的 method, value 是不同的 Controller 实现。Bean 在初始化的时候，还会把 method 名字当作 url 注册到 AbstractUrlHandlerMapping 里去，最主要的是重写了 getHandlerInternal 方法，方法的实现就是取请求的 method 参数值，然后根据初始化 map 找到对应的 Controller Bean 并返回。这里还会额外约束 Controller 的实现必须继承 Spring 的 MultiActionController (其实继承 AbstractController 类应该就OK了)

不过感觉当前的实现方式有些复杂，一是继承了 AbstractUrlHandlerMapping，但其实需要的路由分发逻辑和 Url 是没有关系的，分发逻辑是非常简单的根据 method 请求参数的值，找到映射的 Controller 即可，即使要重写映射逻辑，继承 AbstractHandlerMapping 也就够了。另外，必须使用 xml 进行配置，缺少了现在流行好用的 Java Config 的支持，不利于编译器检查错误和重构。最后还要求实现者必须继承 MultiActionController 类，强制使用原始的 HttpServletRequest 对象，导致业务代码里大量和业务关联不大的取值代码，字符串常量定义，效率低还容易出错。

其实 Spring 的 RequestMapping 注解本身就有提供对参数进行判断映射的能力，使用其 params 属性就可以。样例如下：

第一个 controller 在类上添加 requestMapping 注解，指定 path 是 /e (固定值)，然后 params 指定 method=e1

```
@Controller
@RequestMapping(value = "/e", params = "method=e1")
public class E1 {
    @PostMapping("")
    public void handle(HttpServletRequest request, HttpServletResponse response) throws IOException {
        response.getWriter().write("e1");
        response.setStatus(HttpServletResponse.SC_OK);
    }
}
```

第二个 controller 也是在类上指定 requestMapping 的 path 是 /e(固定值)，然后 params 指定为 method=e2

```
@Controller
@RequestMapping(value = "/e", params = "method=e2")
public class E2 {
    @PostMapping("")
    public void handle(HttpServletRequest request, HttpServletResponse response) throws IOException {
        response.getWriter().write("e2");
        response.setStatus(HttpServletResponse.SC_OK);
    }
}
```

这样一来，不用额外实现 AbstractHandlerMapping 类，我们、完全可以和普通基于 URI Path 进行路由分发的场景一样来实现 controller，controller 路由信息也由 Spring 自行从注解信息上获取，controller 的实现不会要求强制继承某个类，可以充分使用 spring 提供的注解来标记方法参数和 http 请求参数的映射让 spring 自动提取转换。

测试结果：

```
Command Prompt
C:\Users\bphan\Documents\workspace\bin>fetch -X POST -d "method=e1" "http://127.0.0.1:8080/e?k=v"
e1
C:\Users\bphan\Documents\workspace\bin>fetch -X POST -d "method=e2" "http://127.0.0.1:8080/e?k=v"
e2
C:\Users\bphan\Documents\workspace\bin>fetch -X POST "http://127.0.0.1:8080/e?method=e1&k=v"
e1
C:\Users\bphan\Documents\workspace\bin>fetch -X POST "http://127.0.0.1:8080/e?method=e2&k=v"
e2
C:\Users\bphan\Documents\workspace\bin>
```

不管是 query 请求参数，还是 form data 里的参数，spring 都能正确处理。

本文链接: <http://isouth.org/archives/360.html> , 转载请注明出处, 此外还可以[订阅我](#)。

相关日志 Relate Posts


- [SpringBoot应用集成etcd配置源](#)
- [SpringBoot应用配置项加密](#)
- [使用 MariaDB4j 给 Spring Boot 应用做集成测试](#)
- [SpringBoot 集成 Flyway 自动创建数据库表](#)

收藏与分享: [Twitter](#) | [Facebook](#) | [微博](#) | [人人](#) | [Google+](#) | [PDF](#)

“SpringWeb基于请求参数值分发HTTP请求”1条留言


 思空: October 10, 2017 at 12:46 pm [Reply](#)

发表留言 (Ctrl+Enter提交)

昵称 (必填) 

EMail (不公开, 必填)

网站 (选填)



发表

themed by [white](#) 1.18 hosted by [衡天主机](#) copyright © 2019 [isouth.org](#) 27 queries in 1.133 seconds.

[回到顶端](#) ↑