

HTTP Path 匹配和参数提取

2017/04/09 16:16 - Posted By 起衣 - Tags: [http,regex](#) - Category: [Code](#)

0 条评论

给定一个 REST 的 Path 规则是: `/orgs/<orgName>/apis/<apiName>/assets`, 其中尖括号包含的部分表示是一个可变的参数, 参数一般情况下允许包含字母、数字、下划线。然后要求可以匹配符合规则的 HTTP URI Path 部分, 并可以根据参数名称提取出参数的值来。



参考 [bottle](#) 项目的实现方案, 将整个规则转换成一个正则表达式, 然后利用 `named group` 特性, 给定正则表达式捕获的分组名称来获取参数的值。

1. 把 `/orgs/<orgName>/apis/<apiName>/assets` 转换成 `/orgs/(?<orgName>[^/]+)/apis/(?<apiName>[^/]+)/assets`, 注意在 Java 7 后才支持这个正则表达式的写法, 其中 `(?<orgName>[^/]+)` 表示当前分组捕获结果可以用 `orgName` 来进行命名。为啥使用 `named group` 而不是直接按捕获分组顺序来查找, 是因为情况再复杂一点的时候, 正则表达式里的分组可能就很多很乱了, 数也数不清顺序了。
2. 第一步转换成正则表达式的时候, 同时把参数名称提取出来。
3. 正则表达式匹配, 并根据参数名直接从正则表达式结果里获取参数








Java 代码样例:

```
01. public static void main(String[] args) {
02.     String rule = "/orgs/<orgName>/apis/<apiName>/assets";
03.     String path = "/orgs/dev/apis/dts/assets";
04.     Map params = match(rule, path);
05.     System.out.println(params); // {apiName=dts, orgName=dev}
06. }
07.
08. private static Map match(String rule, String path) {
09.     StringBuilder pathRule = new StringBuilder();
10.     List params = new ArrayList<>();
11.
12.     Pattern ruleSyntax = Pattern.compile("<(\w*)>");
```




 订阅 RSS Feed 

 Follow @qiyl on Twitter 

最新文章 [WebSlice](#)

-  [HTTP Path 匹配和参数提取](#)
-  [Mlb Jerseys Buy Conducted Next](#)
-  [Java 里 long 可以计数多久](#)
-  [The Clay Matthews Retro Alternate](#)
-  [并发编程 Promise, Future 和 Callback](#)
-  [golang 以 dom 方式解析 xml](#)
-  [宏碁读做 Hóngjī](#)

管理功能 [Admin](#)

-  [About](#)
-  [Wiki](#)
-  [Log in](#)

```
13.     Matcher ruleMatcher = ruleSyntax.matcher(rule);
14.     int offset = 0;
15.     while (ruleMatcher.find()) {
16.         int groupOffset = ruleMatcher.start();
17.         pathRule.append(rule.substring(offset, groupOffset));
18.
19.         String groupName = ruleMatcher.group(1);
20.         params.add(groupName);
21.         // 拼接成 (?<name>[^/]+) 的正则表达式
22.         pathRule.append("(?<" .append(groupName) .append(">[^/]+)");
23.         offset = ruleMatcher.end();
24.     }
25.     if (offset < rule.length()) {
26.         pathRule.append(rule.substring(offset, rule.length()));
27.     }
28.
29.     Pattern pathPattern = Pattern.compile("^" + pathRule.toString() + "$");
30.     Matcher pathMatcher = pathPattern.matcher(path);
31.     Map result = new HashMap<>();
32.
33.     if (pathMatcher.matches()) {
34.         for (String param : params) {
35.             result.put(param, pathMatcher.group(param));
36.         }
37.     }
38.     return result;
39. }
```

本文链接: <http://isouth.org/archives/356.html>, 转载请注明出处, 此外还可以[订阅我](#)。

相关日志 Relate Posts

- 无相关日志

发表留言 (Ctrl+Enter提交)

昵称 (必填)

EMail (不公开, 必填)

网站 (选填)



发表