登录 | 注册

# **痴人说梦** Yesterday is history, tomorrow is a mystery, but today is a gift.



```
阅读排行
AJAX POST请求中参数以
mysql安装常见问题(系统
                  (37289)
使用maven profile实现多
                  (22238)
java中创建对象的方法
                  (17939)
iava正则表达式获取指定
                  (15063)
java读取.properties配置
                 (14107)
正则表达式学习(七)重
                 (13681)
过去一年的总结
                  (13610)
eclipse tomcat maven热
                  (12877)
springmvc RestTemplate
                  (10804)
```

评论排行 mysql安装常见问题(系统 (29)java正则表达式获取指定 AJAX POST请求中参数以 (11)使用maven profile实现多 (8) 正则表达式: java中匹配 (7) 过去一年的总结 (6) springmyc RestTemplate (6) 验证IPV4正则表达式 (5)如何防止访问API的frienc (4) 使用java进行文件编码转 (3)

\* Chromium扩展 (Extension) 的页面 (Page) 加载过程分析 \* Android Studio 2.2 来啦 \* 手把手教你做音乐播放器 (二) 技术原理与框架设计

\* 2016 年最受欢迎的编程语言是

推荐文章

\* JVM 性能调优实战之:使用阿里开源工具 TProfiler 在海量业务代码中精确定位性能代码 最新评论

最新评论 过去一年的总结 12期-赵尽朝: 总结展望未来 验证IPV4正则表达式 12期-赵尽朝: biaoshikanbudong mysql安装常见问题(系统找不至 Vipwt!: 感谢,解决了我的问题! java正则表达式获取指定HTML标 MikanMu: @baidu\_20228769: 可以看这篇文章: http://blog.csdn.net/mhmyq...

springmvc RestTemplate文件上 qq\_18723091: 感谢万分! 决定 了我用RestTemplate上传图片的 问题

java正则表达式获取指定HTML标 slow-Walker: 楼主还能回下吗? 获取X110 标签内的内容怎么做 到

java正则表达式获取指定HTML标 MikanMu: @you23hai45:多谢指 正,已修改

java正则表达式获取指定HTML标 MikanMu: @qq\_33200504:多谢 指正,已修改

```
Accept:text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
07.
      Accept-Encoding:gzip,deflate,sdch
08.
      Accept-Language:zh-CN,zh;q=0.8,en;q=0.6
09.
      AlexaToolbar-ALX_NS_PH:AlexaToolbar/alxg-3.2
10.
      Cache-Control:max-age=0
11.
      Connection:keep-alive
      Content-Length: 25
12.
13.
      Content-Type:application/x-www-form-urlencoded
      Cookie: JSESSIONID=74AC93F9F572980B6FC10474CD8EDD8D
14.
15.
      Host:127.0.0.1:8080
      Origin:http://127.0.0.1:8080
16.
17.
      Referer:http://127.0.0.1:8080/test/index.jsp
18.
      User-
      Agent:Mozilla/5.0 (Windows NT 6.1)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.149 Safa
19.
20.
      Form Data
21.
      name:mikan
22.
      address:street
23.
24.
      Response Headers
25.
      Content-Length: 2
26.
      Date:Sun, 11 May 2014 11:05:33 GMT
27.
      Server:Apache-Coyote/1.1
```

这里要注意post请求的Content-Type为application/x-www-form-urlencoded,参数是在请求体中,即上面请求中的Form Data。

在servlet中,可以通过request.getParameter(name)的形式来获取表单参数。

而如果使用原生AJAX POST请求的话:

```
C Y
      [javascript]
01.
      function getXMLHttpRequest() {
02.
                 var xhr;
03.
                 if(window.ActiveXObject) {
04.
                          xhr= new ActiveXObject("Microsoft.XMLHTTP");
05.
                 }else if (window.XMLHttpRequest) {
06.
                          xhr= new XMLHttpRequest();
07.
                 }else {
08.
                          xhr= null;
09.
                 }
10.
                 return xhr;
11.
12.
13.
      function save() {
14.
                 var xhr = getXMLHttpRequest();
                 xhr.open("post","http://127.0.0.1:8080/test/test.do");
15.
16.
                 var data = "name=mikan&address=street...";
17.
                 xhr.send(data);
18.
                 xhr.onreadystatechange= function() {
19.
                          if(xhr.readyState == 4 && xhr.status == 200) {
20.
                                   alert("returned:"+ xhr.responseText);
21.
                          }
22.
                 };
23.
      }
```

通过chrome的开发者工具看到请求头如下:

```
CP
      [plain]
01.
      RequestURL:http://127.0.0.1:8080/test/test.do
02.
      Request Method:POST
03.
      Status Code: 200 OK
94.
05.
      Request Headers
96.
      Accept:*/*
07.
      Accept-Encoding:gzip,deflate,sdch
08.
      Accept-Language:zh-CN,zh;q=0.8,en;q=0.6
09.
      AlexaToolbar-ALX NS PH:AlexaToolbar/alxg-3.2
10.
      Connection:keep-alive
11.
      Content-Length:28
      Content-Type:text/plain;charset=UTF-8
12.
```

java正则表达式获取指定HTML标 段小憨: @qq\_32452623:String reg = "<" + element + "\*?\\s?"...

java正则表达式获取指定HTML标 段小憨:标签内的最后一个属性获 取不到值

```
Cookie: JSESSIONID=C40C7823648E952E7C6F7D2E687A0A89
  13.
  14.
        Host:127.0.0.1:8080
 15.
        Origin:http://127.0.0.1:8080
 16.
        Referer:http://127.0.0.1:8080/test/index.jsp
  17.
        Agent:Mozilla/5.0 (Windows NT 6.1)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.149 Safa
  18.
  19.
        Request Payload
  20.
        name=mikan&address=street
  21.
  22.
        Response Headers
  23.
        Content-Length: 2
        Date:Sun, 11 May 2014 11:49:23 GMT
  24.
  25.
        Server:Apache-Coyote/1.1
→ ■
```

注意请求的Content-Type为text/plain;charset=UTF-8,而请求表单参数在RequestPayload中。

那么servlet中通过request.getParameter(name)却是空。为什么呢?而这样的参数又该怎么样获取呢?

为了搞明白这个问题,查了些资料,也看了Tomcat7.0.53关于请求参数处理的源码,终于搞明白了是怎么回事。

HTTP POST表单请求提交时,使用的Content-Type是application/x-www-form-urlencoded,而使用原生AJAX的 POST请求如果不指定请求头ReguestHeader,默认使用的Content-Type是text/plain;charset=UTF-8。

由于Tomcat对于Content-Type multipart/form-data(文件上传)和application/x-www-form-urlencoded(POST请求)做了"特殊处理"。下面来看看相关的处理代码。

Tomcat的HttpServletRequest类的实现类为org.apache.catalina.connector.Request(实际上是org.apache.coyote.Request),而它对处理请求参数的方法为protected void parseParameters(),这个方法中对Content-Type multipart/form-data(文件上传)和application/x-www-form-urlencoded(POST请求)的处理代码如下:

```
CP
      [java]
01.
      protectedvoid parseParameters() {
                //省略部分代码.....
02.
                parameters.handleQueryParameters();// 这里是处理url中的参数
03.
04.
                if ("multipart/form-data".equals(contentType)) { // 这里是处理文件上传请求
05.
06.
                     parseParts();
07.
                     success = true;
08.
                     return;
09.
                }
10.
                if(!("application/x-www-form-urlencoded".equals(contentType))) {// 这里如果是非POST请求
11.
      直接返回, 不再进行处理
12.
                     success = true:
13.
                     return:
14.
15.
                //下面的代码才是处理POST请求参数
                //省略部分代码.....
16.
17.
                try {
18.
                     if (readPostBody(formData, len)!= len) { // 读取请求体数据
19.
                         return;
20.
21.
                } catch (IOException e) {
22.
                     // Client disconnect
23.
                     if(context.getLogger().isDebugEnabled()) {
                         context.getLogger().debug(
24.
25.
                                 sm.getString("coyoteRequest.parseParameters"),e);
26.
                     }
27.
                     return:
28.
                parameters.processParameters(formData, 0, len); // 处理POST请求参数, 把它放到
29.
      requestparameter map中(即request.getParameterMap获取到的Map, request.getParameter(name)也是从这个
      Map中获取的)
30.
                // 省略部分代码.....
31.
      }
32.
33.
         protected int readPostBody(byte body[], int len)
34.
             throws IOException {
35.
36.
             int offset = 0;
             do {
37.
38.
                int inputLen = getStream().read(body, offset, len - offset);
39.
                if (inputLen <= 0) {</pre>
```

从上面代码可以看出,Content-Type不是application/x-www-form-urlencoded的POST请求是不会读取请求体数据和进行相应的参数处理的,即不会解析表单数据来放到request parameter map中。所以通过request.getParameter(name)是获取不到的。

那么这样提交的参数我们该怎么获取呢?

当然是使用最原始的方式,读取输入流来获取了,如下所示:

```
CP
        [java]
 01.
        privateString getRequestPayload(HttpServletRequest req) {
  02.
                  StringBuildersb = new StringBuilder();
 03.
                  try(BufferedReaderreader = req.getReader();) {
  04.
                           char[]buff = new char[1024];
 05.
                           intlen:
 06.
                           while((len = reader.read(buff)) != -1) {
 07.
                                    sb.append(buff,0, len);
  08.
                  }catch (IOException e) {
 09.
  10.
                           e.printStackTrace();
 11.
                  }
  12.
                  returnsb.toString();
  13.
       }
4
```

当然,设置了application/x-www-form-urlencoded的POST请求也可以通过这种方式来获取。

所以,在使用原生AJAX POST请求时,需要明确设置Request Header,即:

```
[javascript] C &

01. xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
```

另外,如果使用jQuery,我使用1.11.0这个版本来测试,\$.ajax post请求是不需要明确设置这个请求头的,其他版本的本人没有亲自测试过。相信在1.11.0之后的版本也是不需要设置的。不过之前有的就不一定了。这个没有测试过。

### 2015-04-17后记:

最近在看书时才真正搞明白,服务器为什么会对表单提交和文件上传做特殊处理,因为表单提交数据是名值对的方式,且Content-Type为application/x-www-form-urlencoded,而文件上传服务器需要特殊处理,普通的post请求(Content-Type不是application/x-www-form-urlencoded)数据格式不固定,不一定是名值对的方式,所以服务器无法知道具体的处理方式,所以只能通过获取原始数据流的方式来进行解析。

jquery在执行post请求时,会设置Content-Type为application/x-www-form-urlencoded,所以服务器能够正确解析,而使用原生ajax请求时,如果不显示的设置Content-Type,那么默认是text/plain,这时服务器就不知道怎么解析数据了,所以才只能通过获取原始数据流的方式来进行解析请求数据。



上一篇 eclipse tomcat maven热部署

下一篇 jenkins在使用maven部署到nginx+tomcat服务器时出现504错误

#### 我的同类文章

## java (22)

• tomcat如何正确的开启远程... 2015-10-17 阅读 2762 • 细说JDK动态代理的实现原理 2015-09-15 阅读 2914

细说java.util.HashMap

2015-09-01 阅读 795

• 关于@Override在1.5和1.6... 2015-08-29 阅读 580

• java枚举类型的实现原理

2015-08-29 阅读 2290

2015-08-29 阅读 1532 细说java.util.Timer

• java中什么是bridge method... 2015-08-07 阅读 2526

· java如何获取方法参数名 2015-08-05 阅读 6695

• 使用java进行文件编码转换

2014-07-17 阅读 2867 · springmvc RestTemplate文... 2014-05-20 阅读 10780

更多文章

#### 参考知识库



jQuery知识库

4106 关注 | 910 收录



request.form

## 猜你在找

js+ajax+jquery+easyui从入门到精通(项

Ajax与JQuery easy UI实战

windows server 2008 r2服务器操作系统 全网服务器数据备份解决方案案例实践

/a\_WebDevelop页面

jsp网站一个 DST请求request payload转form

基于Servlet+JDBC+Bootstrap+MySQL+AJAX权限管理系纟 angularjs中http模块POST请求request payload转form angularjs中http模块POST请求request payload转form ajaxpost方式请求参数无法传递requestgetParameter无

查看评论

11楼 kelezxw 2016-04-08 13:45发表



必须赞一下,楼主辛苦了

10楼 阿迪-Andy 2016-03-07 14:49发表



getRequestPayload()方法读取了流,该怎么在通过对参数进行处理后放入原位置呢,这个才是重点。

9楼 齐博田 2016-02-26 14:06 45 =



写的很好,楼主辛苦啦

8楼 qq\_31834377 2016 01 06 10:44发表



4楼 jane2016 2015-11-01 07:45发表



写得非常好,深入浅出!



公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持 京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved