

个人资料



??toy

关注 发私信

访问：511次

积分：184

等级：

排名：千里之外

原创：3篇 转载：77篇

译文：0篇 评论：0条

文章搜索

文章分类

- 工具 (8)
- JavaSE (2)
- JavaEE (4)
- js (5)
- echart (1)
- ssm (8)
- json-httpclient (3)
- ftp (4)
- 前端 (2)
- 技术杂谈 (1)
- tomcat (5)

文章存档

- 2016年11月 (16)
- 2016年10月 (64)

[移动信息安全的漏洞和逆向原理](#) [程序员11月书讯，评论得书啦](#) [Get IT技能知识库，50个领域一键直达](#)

HttpClient和Spring的整合

2016-10-27 23:01 9人阅读 评论(0)

版权声明：本文为博主原创文章，未经博主允许不得转载。

[html]

```
01. public class RequestConfigDemo {
02.
03.     public static void main(String[] args) throws Exception {
04.
05.         // 创建HttpClient对象
06.         CloseableHttpClient httpClient = HttpClients.createDefault();
07.
08.         // 创建http GET请求
09.         HttpGet httpGet = new HttpGet("http://www.baidu.com/");
10.
11.         // 构建请求配置信息
12.         RequestConfig config = RequestConfig.custom().setConnectTimeout(1000)
长时间
13.             .setConnectionRequestTimeout(500) // 从连接池中获取到连接的最长时
14.             .setSocketTimeout(10 * 1000) // 数据传输的最长时间
15.             .setStaleConnectionCheckEnabled(true) // 提交请求前测试连接是否
16.             .build();
17.         // 设置请求配置信息
18.         httpGet.setConfig(config);
19.
20.         CloseableHttpResponse response = null;
21.         try {
22.             // 执行请求
23.             response = httpClient.execute(httpGet);
24.             // 判断返回状态是否为200
25.             if (response.getStatusLine().getStatusCode() == 200) {
26.                 String content = EntityUtils.toString(response.getEntity());
27.                 System.out.println(content);
28.             }
29.         } finally {
30.             if (response != null) {
31.                 response.close();
32.             }
33.             httpClient.close();
34.         }
35.
36.     }
```

阅读排行

java对文件压缩与解...	(20)
FTP文件上传与下载	(20)
is获取后台Model中的值	(19)
常用的MvEclipse快捷	(17)
2 Slogar获取CPU信息	(16)
Win7下的内置FTP组	(15)
丁县	(14)
如何在Windows Serve	(13)
Mvhatic (一) -----setti	(12)
neinx反向代理结合FT...	(12)

评论排行

小丁县 - 批量删除Ma...	(0)
Eclipse常用的设置	(0)
使用erchart画图表	(0)
正则删除字符串左	(0)
正则删除字符串左	(0)
maven聚合工程的创	(0)
JSON详解	(0)
is获取后台Model中的值	(0)
java 文件及文件夹操...	(0)
常用的MvEclipse快捷	(0)

推荐文章

- * 程序员10月书讯，评论得书
- * Android中Xposed框架篇---修改系统位置信息实现自身隐藏功能
- * Chromium插件（Plugin）模块（Module）加载过程分析
- * Android TV开发总结--构建一个TV app的直播节目实例
- * 架构设计：系统存储--MySQL简单主从方案及暴露的问题



```
37. |
38. | }
```

导入依赖

```
[html]
01. <!-- httpClient -->
02. <dependency>
03.     <groupId>org.apache.httpcomponents</groupId>
04.     <artifactId>httpClient</artifactId>
05.     <version>${httpClient.version}</version>
06. </dependency>
```

application-httpclient.xml

```
[html]
01. <beans xmlns="http://www.springframework.org/schema/beans"
02.     xmlns:context="http://www.springframework.org/schema/context" xmlns:p="http://www.springframework.org/schema/p"
03.     xmlns:aop="http://www.springframework.org/schema/aop" xmlns:tx="http://www.springframework.org/schema/tx"
04.     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
05.     xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans-4.0.xsd
06.         http://www.springframework.org/schema/context http://www.springframework.org/schema/context-4.0.xsd
07.         http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop-4.0.xsd
08.         http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx-4.0.xsd
09.         http://www.springframework.org/schema/util http://www.springframework.org/schema/util-4.0.xsd">
10.
11.     <!-- 定义连接管理器 -->
12.     <bean id="connectionManager"
13.         class="org.apache.http.impl.conn.PoolingHttpClientConnectionManager"
14.         <!-- 最大连接数 -->
15.         <property name="maxTotal" value="${http.maxTotal}" />
16.         <property name="defaultMaxPerRoute" value="${http.defaultMaxPerRoute}" />
17.     </bean>
18.
19.     <!-- 定义HttpClient构造器 -->
20.     <bean id="httpClientBuilder" class="org.apache.http.impl.client.HttpClientBuilder"
21.         <property name="connectionManager" ref="connectionManager" />
22.     </bean>
23.
24.     <!-- 定义HttpClient对象 -->
25.     <bean class="org.apache.http.impl.client.CloseableHttpClient"
26.         factory-bean="httpClientBuilder" factory-method="build" scope="prototype" />
27.
28.     <!-- 请求参数的构造器 -->
29.     <bean id="requestConfigBuilder" class="org.apache.http.client.config.RequestConfig"
30.         <!-- 创建连接的最长时间 -->
31.         <property name="connectTimeout" value="${http.connectTimeout}" />
32.         <!-- 从连接池中获取到连接的最长时间 -->
33.         <property name="connectionRequestTimeout" value="${http.connectionRequestTimeout}" />
34.         <!-- 数据传输的最长时间 -->
35.         <property name="socketTimeout" value="${http.socketTimeout}" />
36.         <!-- 提交请求前测试连接是否可用 -->
37.         <property name="staleConnectionCheckEnabled" value="${http.staleConnectionCheckEnabled}" />
38.     </bean>
```

```

39.      <!-- 定义请求参数 -->
40.      <bean class="org.apache.http.client.config.RequestConfig"
41.          factory-bean="requestConfigBuilder" factory-method="build" />
42.
43.      <!-- 定期关闭无效连接 -->
44.      <bean class="com.taotao.web.httpclient.IdleConnectionEvictor">
45.          <constructor-arg index="0" ref="connectionManager" />
46.      </bean>
47.
48.  </beans>

```

httpClient.properties

```

[html]
01. http.maxTotal=200
02. http.defaultMaxPerRoute=50
03. http.connectTimeout=1000
04. http.connectionRequestTimeout=500
05. http.socketTimeout=10000
06. http.staleConnectionCheckEnabled=true

```

封装ApiService

```

[html]
01. /**
02.  * httpClient 常用操作的 工具类
03.  *
04.  * @author Administrator
05.  *
06.  */
07. @Service
08. public class ApiService implements BeanFactoryAware{
09.
10.     // @Autowired
11.     // private CloseableHttpClient httpClient;
12.
13.     @Autowired
14.     private RequestConfig requestConfig;
15.
16.     /**
17.      * post请求, get请求
18.      */
19.
20.     /**
21.      * 发送get请求, 并且没有参数
22.      *
23.      * @param url
24.      * @return 请求的响应内容
25.      * @throws IOException
26.      * @throws ClientProtocolException
27.      */
28.     public String doGet(String url) throws ClientProtocolException, IOException {
29.         // 创建http GET请求
30.         HttpGet httpGet = new HttpGet(url);
31.         httpGet.setConfig(requestConfig);
32.
33.         CloseableHttpResponse response = null;
34.         try {
35.             // 执行请求
36.             response = getHttpClient().execute(httpGet);
37.             // 判断返回状态是否为200
38.             if (response.getStatusLine().getStatusCode() == 200) {
39.                 return EntityUtils.toString(response.getEntity(), "UTF-8");

```

```

40.         }
41.     } finally {
42.         if (response != null) {
43.             response.close();
44.         }
45.     }
46.     return null;
47. }
48.
49. /**
50.  * 发送带有参数的get请求
51.  *
52.  * @param url
53.  * @param param
54.  * @return
55.  * @throws URISyntaxException
56.  * @throws ClientProtocolException
57.  * @throws IOException
58.  */
59. public String doGet(String url, Map<String, String> param)
60.     throws URISyntaxException, ClientProtocolException, IOException
61. {
62.     URIBuilder builder = new URIBuilder(url);
63.     if (param != null) {
64.         for (String key : param.keySet()) {
65.             builder.setParameter(key, param.get(key));
66.         }
67.     }
68.     return this.doGet(builder.build().toString());
69. }
70.
71. /**
72.  * 带有参数的post请求
73.  * @param url
74.  * @param param
75.  * @return
76.  * @throws IOException
77.  */
78. public HttpResult doPost(String url, Map<String, String> param)
79.     throws IOException {
80.
81.     // 创建http POST请求
82.     HttpPost httpPost = new HttpPost(url);
83.
84.     if (param != null) {
85.         // 设置2个post参数，一个是scope、一个是q
86.         List<NameValuePair> parameters = new ArrayList<NameValuePair>(0);
87.         for (String key : param.keySet()) {
88.             parameters.add(new BasicNameValuePair(key, param.get(key)));
89.         }
90.         // 构造一个form表单式的实体
91.         UrlEncodedFormEntity formEntity = new UrlEncodedFormEntity(
92.             parameters);
93.         // 将请求实体设置到httpPost对象中
94.         httpPost.setEntity(formEntity);
95.     }
96.     httpPost.setConfig(requestConfig);
97.
98.     CloseableHttpResponse response = null;
99.     try {
100.        // 执行请求
101.        response = getHttpClient().execute(httpPost);
102.        return new HttpResult(response.getStatusLine().getStatusCode(),

```

```

103.         EntityUtils.toString(response.getEntity(), "UTF-8"));
104.     } finally {
105.         if (response != null) {
106.             response.close();
107.         }
108.     }
109. }
110.
111. /**
112.  * post请求, 无参
113.  * @param url
114.  * @return
115.  * @throws IOException
116.  */
117. public HttpResult doPost(String url) throws IOException{
118.     return this.doPost(url, null);
119. }
120.
121. private BeanFactory beanFactory;
122.
123. /**
124.  * 向当前的bean 里面注入BeanFactory
125.  */
126. @Override
127. public void setBeanFactory(BeanFactory beanFactory) throws BeansException {
128.     this.beanFactory = beanFactory;
129. }
130.
131. private CloseableHttpClient getHttpClient(){
132.     return this.beanFactory.getBean(CloseableHttpClient.class);
133. }
134.
135. }

```

顶 踩
0 0

- [上一篇 echarts-去掉X轴、Y轴和网格线](#)
- [下一篇 httpClient使用总结](#)

参考知识库



Java EE知识库

1210 关注 | 581 收录

猜你在找

Python编程基础视频教...
HTML 5移动开发从入门...
零基础学HTML 5实战开...
Swift视频教程(第四季)
Swift视频教程(第六季)

JA-SIGCAS学习笔记3
Android Github开源项目...
GitHub 优秀的 Android ...
linux后台系统
面试总结



查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker Of
Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity Splashtoj
components Windows Mobile Rails QEMU KDE Cassandra CloudStack FTC coremail
CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo Compuware 大数据
Tornado Ruby Hibernate ThinkPHP HBase Pure Solr Angular Cloud Redis
Django Bootstrap

[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 