Atitit webclient httpclient技术总结 RestTemplate

Atitit CateIT重要技术httpclient iduah2 impt

体系树路径：CS IT> net . http ftp

密级和保密期限：：

Keywords和摘要：none

目录

[1. 内部概念 1](#_Toc22054)

[2. 常用工具技术 1](#_Toc31309)

[2.1. HttpClient OkHttp 2](#_Toc2687)

[2.2. Rest工具 RestTemplate Feign 2](#_Toc2983)

[2.3. Netty、Mina 这样的网络 IO 框架过于底层 2](#_Toc15336)

[2.4. Curl postman 2](#_Toc29998)

[3. 类似概念 2](#_Toc31662)

[3.1. WebClient（它会自动根据url来识别是FTP还是Http） 3](#_Toc24547)

[4. RestTemplate 的代码范例 3](#_Toc24598)

[4.1. url 拦截转发器 3](#_Toc23861)

[4.2. 提取标头 参数等 5](#_Toc8230)

[4.3. 并附带带cookie头请求 6](#_Toc19994)

[5. Ref 8](#_Toc11991)

# 内部概念

1. 概念 与组成 2

1.1. Httpentity 2

1.1.1. Fileentity 2

1.1.2. Stringentiry 2

1.1.3. Urlencodeformentity 2

1.2. Method 2

1.2.1. Get post put del 2

# 常用工具技术

## HttpClient OkHttp

在 Java 社区中，HTTP Client 主要有 JDK 的 HttpURLConnection、Apache Commons HttpClient（或被称为 Apache HttpClient 3.x）、Apache HttpComponents Client（或被称为 Apache HttpClient 4.x）、Square 公司开源的 OkHttp。

## Rest工具 RestTemplate Feign

除了这几个纯粹的 HTTP Client 类库以外，还有 Spring 的 RestTemplate、Square 公司的 Retrofit、Netflix 公司的 Feign，以及像 Apache CXF 中的 client 组件。这些框架和类库主要是针对 Web Service 场景，尤其是 RESTful Web Service。它们往往是基于前面提到的 HTTP Client 实现，并在其基础上提供了消息转换、参数映射等对于 Web Service 来说十分必要的功能。

## Netty、Mina 这样的网络 IO 框架过于底层

（当然，像 Netty、Mina 这样的网络 IO 框架，实现 HTTP 自然也不再话下，但这些框架通常过于底层，不会被直接使用）

## Curl postman

# 类似概念

## WebClient（它会自动根据url来识别是FTP还是Http）

就是通过创建WebRequest（它会自动根据url来识别是FTP还是Http）来操作的。可以说，与其你费劲去创建HttpWebRequest，再创建WebResponse，最后才读取数据，当然不如直接使用WebClient方便啦！但是这只是方便程度不同，对于你的这个问题没有差别。

# RestTemplate 的代码范例

## url 拦截转发器

\* \*/

**public** **class** TransInterceptor **extends** HandlerInterceptorAdapter {

/\*\*

\* 。该方法的返回至是 Boolean 类型，当它返回 false 时，表示请求结束，后续的 Interceptor 和 Controller 都不会再执行；当它返回为 true 时会继续调用下一个 Interceptor 的 preHandle 方法，如果已经是最后一个 Interceptor 的时候就会调用当前请求的 Controller 方法。

\*/

@Override

**public** **boolean** preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)

**throws** Exception {

String destUrl = "http://localhost:80"+request.getRequestURI();

**long** startTime = System.*currentTimeMillis*();

System.***out***.println("\n--------LogInterception.preHandle---");

**if**( request.getRequestURI().toLowerCase().startsWith("/actuator"))

**return** **true**;

System.***out***.println("RequestURL:" + request.getRequestURL());

System.***out***.println("StartTime:" + System.*currentTimeMillis*());

// response.getWriter().write( "拦截内容lejye neiron from svr proder.. ");

//

// if("1".equals("1"))

// return false;

// System.out.println(json);

// 构造URI。必须拼接出String url然后创建URI，否则会出现queryString %符号转%25的问题

HttpMethod method = *getHttpmethod*(request);

URI destUri = **new** URI(destUrl+"?"+ *getQueryString*(request));

System.***out***.println(destUri);

HttpEntity<Object> requestEntity = *getHttpHearderAndParams*(request);

// 向服务请求

RestTemplate restTemplate = **new** RestTemplate();

ResponseEntity<String> responseEntity = restTemplate.exchange(destUri, method, requestEntity, String.**class**);

response.getWriter().write( responseEntity.getBody().toString());

// return responseEntity;

// request.setAttribute("startTime", startTime);

**return** **false**; //cancel next process

}

## 提取标头 参数等

**import** javax.servlet.http.HttpServletRequest;

**import** org.springframework.http.HttpEntity;

**import** org.springframework.http.HttpHeaders;

**import** org.springframework.http.HttpMethod;

**import** org.springframework.util.LinkedMultiValueMap;

**import** org.springframework.util.MultiValueMap;

**public** **class** RestTemplateUtil {

**public** **static** HttpMethod getHttpmethod(HttpServletRequest request) {

HttpMethod method = **null**;

**if** (request.getMethod() == HttpMethod.***POST***.name())

method = HttpMethod.***POST***;

**if** (request.getMethod() .equals(HttpMethod.***GET***.name()) )

method = HttpMethod.***GET***;

**return** method;

}

**public** **static** String getQueryString(HttpServletRequest request) {

String destUrl = "";

**if** (request.getQueryString()!=**null** && !request.getQueryString().isEmpty())

destUrl += "?" + request.getQueryString();

**return** destUrl;

}

//请求实体

**public** **static** HttpEntity<Object> getHttpHearderAndParams(HttpServletRequest request) {

// 根据request，构造HttpHeaders

String body=**null**;

HttpHeaders headers = **new** HttpHeaders();

Enumeration<String> headerNames = request.getHeaderNames();

**while**(headerNames.hasMoreElements()){

String name = (String)headerNames.nextElement();

String value = request.getHeader(name);

headers.add(name, value);

}

// 复制 request 的参数

Map<String, String[]> parameterMap = request.getParameterMap();

MultiValueMap<String, Object> params = **new** LinkedMultiValueMap<String, Object>();

// 附加参数值

Set<String> keySet = parameterMap.keySet();

**for** (String key : keySet) {

String[] value = parameterMap.get(key);

params.add(key, value[0]);

}

// 根据body内容填充requestEntity。对于form-data，body为空但parameterMap有值；对于raw，body不为空。

HttpEntity<Object> requestEntity = (body!=**null** && !body.isEmpty())? **new** HttpEntity<>(body, headers): **new** HttpEntity<>(params, headers);

**return** requestEntity;

}

}

## 并附带带cookie头请求

**import** org.springframework.http.HttpEntity;

**import** org.springframework.http.HttpHeaders;

**import** org.springframework.http.HttpMethod;

**import** org.springframework.http.ResponseEntity;

**import** org.springframework.web.client.RestTemplate;

**public** **class** CookieTest {

/\*\*

\*

\* array(1) {

["aaa"]=>

string(2) "11"

}

\* **@param** args

\* **@throws** URISyntaxException

\*/

**public** **static** **void** main(String[] args) **throws** URISyntaxException {

HttpHeaders headers = **new** HttpHeaders();

headers.add("Cookie", "aaa=11" );

RestTemplate restTemplate = **new** RestTemplate();

// URI url= new URI("http://localhost:9088/cktest");

// System.out.println( restTemplate.getForObject(url, String.class));

String url = "http://localhost/showck.php";

url="http://localhost:9088/cktest";

ResponseEntity response = restTemplate.exchange(url,

HttpMethod.***GET***,

**new** HttpEntity (headers) ,String.**class**);

System.***out***.println(response.getBody());

//http://localhost/showck.php

}

}

# Ref

\0 it impttech topic\http https\Atiti HttpClient阻塞的问题.docx

\0 it impttech topic\http https\Atitit apach httpclient 新特性 new feature(1).docx

\0 it impttech topic\http https\Atitit apach httpclient 新特性 new feature.docx

\0 it impttech topic\http https\Atitit apache httpclient encode html编码(1).docx

\0 it impttech topic\http https\Atitit apache httpclient encode html编码.docx

\0 it impttech topic\http https\Atitit avax.ws.rs.NotFoundException HTTP 404 Not Found(1).docx

\0 it impttech topic\http https\Atitit avax.ws.rs.NotFoundException HTTP 404 Not Found.docx

\0 it impttech topic\http https\Atitit http 1 2.0 3.0 的区别与不同.docx

\0 it impttech topic\http https\Atitit http 1 2.0 3.0 的区别与不同.docx.txt

\0 it impttech topic\http https\Atitit HTTP AUTH 安全与认证机制attilax总结.docx

\0 it impttech topic\http https\Atitit http get java 的实现(1).docx

\0 it impttech topic\http https\Atitit http get java 的实现. HttpURLConnection httpclient v2(1).docx

\0 it impttech topic\http https\Atitit http get java 的实现. HttpURLConnection httpclient v2.docx

\0 it impttech topic\http https\Atitit http get java 的实现. HttpURLConnection httpclient v3 t55.docx

\0 it impttech topic\http https\Atitit http get java 的实现.docx

\0 it impttech topic\http https\Atitit http get 功能实现(1).docx

\0 it impttech topic\http https\Atitit http get 功能实现.docx

\0 it impttech topic\http https\atitit http httpclient实践java c# .net php attilax总结.docx

\0 it impttech topic\http https\Atitit http stat 400错误解决.docx

\0 it impttech topic\http https\Atitit http statecode 204(1).docx

\0 it impttech topic\http https\Atitit http statecode 204.docx

\0 it impttech topic\http https\Atitit http 缓存机制.docx

\0 it impttech topic\http https\Atitit http2 新特性.docx

\0 it impttech topic\http https\Atitit HttpClient 4.5.2 (GA) apache ssl 打解决.docx

\0 it impttech topic\http https\Atitit HttpClient post请求功能总结.docx

\0 it impttech topic\http https\Atitit httpclient 概念图gainyetu 与脑图 目录mindchart inde.docx

\0 it impttech topic\http https\Atitit httpclient 概述 课程 v2 t66.docx

\0 it impttech topic\http https\Atitit httpclient 概述 课程 v3 t66.docx

\0 it impttech topic\http https\Atitit httpclient 概述.docx

\0 it impttech topic\http https\Atitit httpclient4.0 get big file down(1).docx

\0 it impttech topic\http https\Atitit httpclient4.0 get big file down.docx

\0 it impttech topic\http https\Atitit HttpClientUtilV2Saa 新特性.docx

\0 it impttech topic\http https\Atitit HttpClient模块 需改记录 attilax总结.docx

\0 it impttech topic\http https\atitit https原理.docx

\0 it impttech topic\http https\atitit http代理 代码 atitit.docx

\0 it impttech topic\http https\Atitit http代理转发映射以及httpclient的选项总结.docx

\0 it impttech topic\http https\atitit http原理与概论attilax总结.docx

\0 it impttech topic\http https\Atitit http参数 模式：request-line方式与request-body方式。(1).docx

\0 it impttech topic\http https\Atitit http参数 模式：request-line方式与request-body方式。.docx

\0 it impttech topic\http https\Atitit mq与http的通讯.docx

\0 it impttech topic\http https\Atitit Rsa 系统解决方案，解决非https系统的通讯安全性.docx

\0 it impttech topic\http https\Atitit server服务器编写总结httphandler .docx

\0 it impttech topic\http https\Atitit web httphandler的实现 java python node.js c# net php.docx

\0 it impttech topic\http https\Atitit WebClient HttpWebRequest HttpClient webrequest diff区别.docx

\0 it impttech topic\http https\Atitit 发出http请求的lib设计.docx

\0 it impttech topic\http https\Atitit 基于http的接口attilax大总结.docx

\0 it impttech topic\http https\Atitit 基于http的接口attilax大总结.docx.txt

\0 it impttech topic\http https\Atitit 流媒体协议sip rtsp http 比较与区别.docx

\0 it impttech topic\http https\Atitit 获取ip的http标头.docx

\0 it impttech topic\http https\Atitit 返回http500返回码，以及自定义返回提示.docx

\0 it impttech topic\http https\Atitit 通过http get api 解决方案.docx

\0 it impttech topic\http https\Atitit 项目的http rest接口.docx

\0 it impttech topic\http https\Atitit 项目的http rest接口v2 - 副本.docx

\0 it impttech topic\http https\Atitit 项目的http rest接口v2.docx

\0 it impttech topic\http https\Atitit 项目的http rest接口v2.docx.txt

\0 it impttech topic\http https\Atitit. http get post 的本质区别 为什么要区分get post ----安全原因.docx

\0 it impttech topic\http https\atitit. http json 接口的测试工具.doc

\0 it impttech topic\http https\atitit. http json 接口的测试工具.doc.txt

\0 it impttech topic\http https\atitit.http get post的原理以及框架实现java php.doc

\0 it impttech topic\http https\Atitit.http httpclient实践java c# .net php attilax总结.docx

\0 it impttech topic\http https\Atitit.HTTP 代理原理及实现 正向代理与反向代理attilax总结.docx

\0 it impttech topic\http https\Atitit.http代理 代码 Atitit.docx

\0 it impttech topic\http https\atitit.http原理与概论attilax总结.docx

\0 it impttech topic\http https\Atitit.?servlet 与 IHttpHandler?的 原理理论 架构设计 实现机制 解决方案 理念模式 实践java php c#.net js javascript c++ python.docx

\0 it impttech topic\http https\Atitit.协议的转换smb2http.docx

\0 it impttech topic\http https\Atitit.简化信息防篡改 md5签名 ---https.doc

\0 it impttech topic\http https\atitit。Error%3A [$injector%3Amodulerr] http%3A%2F%2Ferrors.angularjs.org%2F1.2.9%2F$injector%2Fmodulerr%3Fp0=atiMod&p1=.doc

\0 it impttech topic\http https\Atitit。http?下载细节问题qa.docx

\0 it impttech topic\http https\http.txt

\0 it impttech topic\http https\httpclient-4.5.2.jar 类库冲突.docx

\0 it impttech topic\http https\I’ve just posted a new blog New post in 2blogger car 311 352 httpst.couyqng7aLzJ httpst.co1PIkNGNJuE (3).gdoc

\0 it impttech topic\http https\Jpg http 显示err.docx

\0 it impttech topic\http https\laravel 5.1 unexpected T\_STRING Illuminate Contracts—Http Kernel lass.docx

\0 it impttech topic\http https\NoSuchFieldError INSTANCE at org.apache.http.impl.io.DefaultHttpRequestWriterFactory.docx

\0 it impttech topic\http https\PAIP.http post 400错误.doc

\0 it impttech topic\http https\paip.提升安全性-----使用HTTPS SSL.doc