**《Spring Boot基础教程》**

**第23节**

**调用REST服务-如何使用代理**

1. **添加依赖**

<dependency>

<groupId>org.apache.httpcomponents</groupId>

<artifactId>httpclient</artifactId>

</dependency>

1. **代码实现**

/\*\*

\* @author wujing

\*/

@RestController

@RequestMapping(value = "/rest", method = RequestMethod.POST)

public class RestRoncooController {

@Autowired

private RoncooUserLogCache RoncooUserLogCache;

@RequestMapping(value = "/update")

public RoncooUserLog update(@RequestBody JsonNode jsonNode) {

System.out.println("jsonNode=" + jsonNode);

RoncooUserLog bean = RoncooUserLogCache.selectById(jsonNode.get("id").asInt(1));

if(bean == null){

bean = new RoncooUserLog();

}

bean.setUserName("测试");

bean.setCreateTime(new Date());

bean.setUserIp("192.168.1.1");

RoncooUserLogCache.updateById(bean);

return bean;

}

@RequestMapping(value = "/update/{id}", method = RequestMethod.GET)

public RoncooUserLog update2(@PathVariable(value = "id") Integer id) {

RoncooUserLog bean = RoncooUserLogCache.selectById(id);

if(bean == null){

bean = new RoncooUserLog();

}

bean.setUserName("测试");

bean.setCreateTime(new Date());

bean.setUserIp("192.168.1.1");

RoncooUserLogCache.updateById(bean);

return bean;

}

}

1. **测试**

@Autowired

private RestTemplateBuilder restTemplateBuilder;

/\*\*

\* get请求

\*/

@Test

public void getForObject() {

RoncooUserLog bean = restTemplateBuilder.build().getForObject("http://localhost:8080/rest/update/{id}", RoncooUserLog.class, 6);

System.out.println(bean);

Map<String,Object> map = new HashMap<String,Object>();

map.put("id", 7);

bean = restTemplateBuilder.build().postForObject("http://localhost:8080/rest/update", map, RoncooUserLog.class);

System.out.println(bean);

}

**代理实现：**

**static** **class** ProxyCustomizer **implements** RestTemplateCustomizer {

@Override

**public** **void** customize(RestTemplate restTemplate) {

String proxyHost = "59.33.46.187";

**int** proxyPort = 6969;

HttpHost proxy = **new** HttpHost(proxyHost, proxyPort);

HttpClient httpClient = HttpClientBuilder.*create*().setRoutePlanner(**new** DefaultProxyRoutePlanner(proxy) {

@Override

**public** HttpHost determineProxy(HttpHost target, HttpRequest request, HttpContext context) **throws** HttpException {

**return** **super**.determineProxy(target, request, context);

}

}).build();

HttpComponentsClientHttpRequestFactory httpComponentsClientHttpRequestFactory = **new** HttpComponentsClientHttpRequestFactory(httpClient);

httpComponentsClientHttpRequestFactory.setConnectTimeout(10000);

httpComponentsClientHttpRequestFactory.setReadTimeout(60000);

restTemplate.setRequestFactory(httpComponentsClientHttpRequestFactory);

}

}

**代理测试：**

String result = restTemplateBuilder.additionalCustomizers(**new** ProxyCustomizer()).build().getForObject("http://www.roncoo.com", String.**class**);

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

**在线代理：**

**http://ip.zdaye.com/**

**更多课程信息，请关注 龙果学院 官方网站**[**http://www.roncoo.com/**](http://www.roncoo.com/)

**或关注 龙果 微信公众号 RonCoo\_com**

****