

# 安全-服务间交互

示例项目：  
\* spring-boot-ms1  
\* spring-boot-ms2

## 场景描述

两个服务：**spring-boot-ms1**和**spring-boot-ms2**，分别对外提供Rest API服务。当ms2服务访问ms2服务中的某一API时，应该如何做？下面给出简单示例工程。

## 准备

1. 认证服务器UAA：<http://192.168.66.158:18080>
2. 创建Spring boot 项目ms1和ms2，二者满足：
  - ms1 端口：18080; ms2端口：28080；
  - 通过 @EnableResourceServer ,表述两者都是资源服务器，对外提供安全的api服务；
  - ms1 使用的OAuth Client：wanghq；ms2使用OAuth Client:zhuqinghua；
3. ms1提供服务API: /api/user

```
@RequestMapping("/user")
public String user() {
    return "ms1 user...";
}
```

## 目标

ms2 提供服务API：/api/ms1/user，在该接口中调用了ms1的API:/api/user。

## 实现

ms2作为OAuth Client，通过OAuth2RestTemplate达到访问ms1 API的目的。

RestTemplate代码：

```
@Configuration
public class OAuthConfig {
    private final String GRANT_CLIENT_CREDENTIALS = "client_credentials";

    @Value("${ms1.clientId}")
    private String oAuth2ClientId;

    @Value("${ms1.clientSecret}")
    private String oAuth2ClientSecret;

    @Value("${ms1.accessTokenUri}")
    private String accessTokenUri;

    @Autowired
    private OAuth2ClientContext oAuth2Context;

    @Bean
    OAuth2ProtectedResourceDetails resource() {
        ClientCredentialsResourceDetails resource = new ClientCredentialsResourceDetails();
```

ms2服务API定义：

期望输出：**ms1 user...**

## 测试

- ```
curl -X POST \
    'http://192.168.66.158:18080/oauth/token?
client_id=wanghq&client_secret=wanghq&redirect_uri=http%3A%2F%2Flocalhost%3A489%2F&response_type=token' \
-H 'cache-control: no-cache' \
-H 'content-type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW' \
-H 'postman-token: 68440efd-4c00-f5fd-5c83-7455f1650182' \
-F grant_type=client_credentials

{
  "access_token":
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ6aWQiOiJlYWEiLCJyZWZSI6WyJvcGVuaHQiLCJJjb3JlIiwidWFhLmVbmUiLCJhyMucG9saWNPZXMu
cmVhZCIsImFjcj5wb2xpY2llcy53cmI0ZSIsImFjcy5hdHRyaWJldGVzLnJlYWQilCJwcmVkagTWNzLnvpvbmVzMRLbw8udXNlcisImFjcy5hdHRYawJldG
VzLndyaXRlIiwiaWF0eSBmbWVuIiwicmVhZCI6ImlhOjEIMTA2MDg4MDIsImldhcDGI6MCwicianRpIjoimjUxMTZKYMUtYjBLYSOMMDZjLWFhNGItOGZjMGUXNmM5OWFHiiwiY2xpZW50OXZ
b2FiIdGgdvdG9rZW4iLCJleHAiojEIMTA2MDg4MDIsImldhcDGI6MCwicianRpIjoimjUxMTZKYMUtYjBLYSOMMDZjLWFhNGItOGZjMGUXNmM5OWFHiiwiY2xpZW50OXZ
lkIjojd2FuZ2hxIn0.BPyNVjr0LqUMHH4IUH26V3tNZnNUQNHTmfSyjWlopZC0-
TZpawhObmyblACAcymptdwyEDvb3gnzQIGr74Q7jym5AeiQjqOfMLqmOU32xfN6rbz-
dt5yyI6it2xt7Vwbx99PcbwyrfeqJYJXT08Y460sZS_nG9fmU8dDjkN49PryhXffBza-ZLNHLR_uK_jou_nP-
6N_4YNURITSIfiazZFLSBFjDOeCsuuZhSc8nyDG_iRCp7GEvaAo25SLffbBH2G0-
e4IZq7svLEka9F5Z4qv1qwILSemwxIBNCydgTyEwnIgCKi4nqqmkWeAVjFGPCSVmziUBefZ7fhCEg",
  "token_type": "bearer",
  "expires_in": 43199,
  "scope": "openid core uaa.none acs.policies.read acs.policies.write acs.attributes.read prefix-ac.s zones.demo.user
acs.attributes.write acs.zones.admin prefix-ac.s zones.demo.admin",
  "jti": "2513fdbbe-b0ea-406c-aa4b-8fc0e17399aa"
```

1. 测试ms2的API接口： /api/ms1/user

输出: ms1 user...

输出和预期一致。