

需要技术储备 SpringSecurity+jwt权限认证相关基本知识

一、概述

微服务架构，前后端分离目前已成为互联网项目开发的业界标准，其核心思想就是前端（APP、小程序、H5页面等）通过调用后端的API接口，提交及返回JSON数据进行交互。在前后端分离项目中，首先要解决的就是登录及授权的问题。微服务架构下，传统的session认证限制了应用的扩展能力，无状态的JWT认证方法应运而生，该认证机制特别适用于分布式站点的单点登录（SSO）场景。

二、组件应用

1. 引入组件

可引用jar包，也可以源码作为子模块



jwt-module-1.3.jar
21.25KB



jwt-module.zip
18.76KB

2. 过滤器注入

启动类注入过滤器bean

@Bean

```
public FilterRegistrationBean  
registration(JwtAuthenticationTokenFilter filter) {  
    FilterRegistrationBean registration = new  
FilterRegistrationBean(filter);  
    registration.setEnabled(false);  
    return registration;  
}
```

3. yml配置jwt信息

#jwt

jwt:

header: authorization

令牌前缀

token-start-with: bearer

使用Base64对该令牌进行编码 (制定您的密钥) 256

base64-secret:

yuantiaokj62672000yuantiaokj62672000yuantiaokj62672000cnzz

令牌过期时间 此处单位/毫秒

token-validity-in-seconds: 14400000

#放行url (静态资源已放行)

permitMatchers:

- /JwtTokenController/getToken

4. 认证, 权限校验, 用户信息获取demo

```
1  /**
2   * *****
3   * Copyright © 2020 远眺科技 Inc.All rights reserved. * **
4   * *****
5   *
6   * @program: module-lib
7   * @description: 测试
8   * @author: cnzz
9   * @create: 2020-06-04 16:39
10  **/
11  @RestController
12  @Slf4j
13  @RequestMapping("/JwtTokenController")
14  public class JwtTokenController {
15      @Resource
16      JwtTokenUtils jwtTokenUtils;
17      @Resource
18      JwtSecurityProperties jwtSecurityProperties;
```

```
19
20  /**
21   * 登录认证生成token
22   *
23   * @return
24   */
25  @GetMapping("/getToken")
26  public String getToken() {
27      List<String> permissions = new ArrayList<>();
28      permissions.add("admin");
29      permissions.add("afds");
30      String token = jwtTokenUtils.createToken("1223", "abcc", permissions);
31      log.info("获取的|token={}", token);
32      return token;
33  }
34
35
36  /**
37   * 从token中获取用户信息
38   *
39   * @param httpRequest
40   * @return
41   */
42  @GetMapping("/noToken")
43  public String noToken(HttpServletRequest httpRequest) {
44      //获取token
45      String bearerToken =
46      httpRequest.getHeader(jwtSecurityProperties.getHeader());
47      log.info("获取token|bearerToken={}", bearerToken);
48      //获取token用户信息
49      Claims claims = jwtTokenUtils.getClaimsFromToken(bearerToken);
50      String id = claims.getId();
51      log.info("获取token用户信息|id={}", id);
52      return id;
53  }
54
55  /**
56   * 权限认证注解
57   *
58   * @return
```

```
59  */
60  @RequestMapping("/validPermission")
61  @PreAuthorize("hasAnyAuthority('admin')")
62  public String validPermission() {
63      log.info("进入|validPermission");
64      return "OK";
65  }
66 }
```