

2-9 创建JWT工具类

140.143.132.225:8000/project-1/doc-21

JWT的 **Token** 要经过加密才能返回给客户端，包括客户端上传的 **Token**，后端项目需要验证核实。于是我们需要一个JWT工具类，用来 **加密Token** 和 **验证Token** 的有效性。

一、导入依赖库

```
1. <dependency>
2.     <groupId>org.apache.shiro</groupId>
3.     <artifactId>shiro-web</artifactId>
4.     <version>1.5.3</version>
5. </dependency>
6. <dependency>
7.     <groupId>org.apache.shiro</groupId>
8.     <artifactId>shiro-spring</artifactId>
9.     <version>1.5.3</version>
10. </dependency>
11. <dependency>
12.     <groupId>com.auth0</groupId>
13.     <artifactId>java-jwt</artifactId>
14.     <version>3.10.3</version>
15. </dependency>
16. <dependency>
17.     <groupId>org.springframework.boot</groupId>
18.     <artifactId>spring-boot-configuration-processor</artifactId>
19.     <optional>true</optional>
20. </dependency>
21. <dependency>
22.     <groupId>org.apache.commons</groupId>
23.     <artifactId>commons-lang3</artifactId>
24.     <version>3.11</version>
25. </dependency>
26. <dependency>
27.     <groupId>org.apache.httpcomponents</groupId>
28.     <artifactId>httpcore</artifactId>
29.     <version>4.4.13</version>
30. </dependency>
31. <dependency>
32.     <groupId>org.springframework.boot</groupId>
33.     <artifactId>spring-boot-starter-aop</artifactId>
34. </dependency>
```

二、定义密钥和过期时间

我建议大家把密钥和过期时间定义到SpringBoot配置文件中，然后再值注入到JavaBean中，这样维护起来比较方便。

1. emos:
2. jwt:
3. #密钥
4. secret: abc123456
5. #令牌过期时间（天）
6. expire: 5
7. #令牌缓存时间（天数）
8. cache-expire: 10

二、创建JWT工具类

在 `com.example.emos.wx.config.shiro` 中创建 `JwtUtil` 类。



```
1. package com.example.emos.wx.config.shiro;
2. import cn.hutool.core.date.DateField;
3. import cn.hutool.core.date.DateUtil;
4. import com.auth0.jwt.JWT;
5. import com.auth0.jwt.JWTCreator;
6. import com.auth0.jwt.JWTVerifier;
7. import com.auth0.jwt.algorithms.Algorithm;
8. import com.auth0.jwt.interfaces.DecodedJWT;
9. import com.example.emos.wx.exception.EmosException;
10. import lombok.extern.slf4j.Slf4j;
11. import org.springframework.beans.factory.annotation.Value;
12. import org.springframework.stereotype.Component;
13. import java.util.Date;
14. @Component
15. @Slf4j
16. public class JwtUtil {
17.     //密钥
18.     @Value("${emos.jwt.secret}")
19.     private String secret;
20.     //过期时间（天）
21.     @Value("${emos.jwt.expire}")
22.     private int expire;
23.     public String createToken(int userId) {
24.         Date date = DateUtil.offset(new Date(), DateField.DAY_OF_YEAR,
25.             expire).toJdkDate();
26.         Algorithm algorithm = Algorithm.HMAC256(secret); //创建加密算法对象
27.         JWTCreator.Builder builder = JWT.create();
28.         String token = builder.withClaim("userId",
29.             userId).withExpiresAt(date).sign(algorithm);
30.         return token;
31.     }
32.     public int getUserId(String token) {
33.         try {
34.             DecodedJWT jwt = JWT.decode(token);
35.             return jwt.getClaim("userId").asInt();
36.         } catch (Exception e) {
37.             throw new EmosException("令牌无效");
38.         }
39.     }
40.     public void verifierToken(String token) {
41.         Algorithm algorithm = Algorithm.HMAC256(secret); //创建加密算法对象
42.         JWTVerifier verifier = JWT.require(algorithm).build();
43.         verifier.verify(token);
44.     }
45. }
```