一、发布 jwt 鉴权服务

1、选择 jwtToken

基本配置		
* API名称	testapi	
* API描述	test	
安全认证	jwtTokenAuth ▼	
*认证配置文件	{"apiTokenName":"apiToken","userTokenName":"apiName","s ecretKeys":[{"admin":"sinopec"},{"test":"123456"}]}	
访问限制/单位	不限制 ▼	

2、认证配置文件说明 需使用 JSON 对象描述

二、JAVA 程序生成 token 示例

```
Maven 工具类包
<dependency>
<groupId>com.auth0</groupId>
<artifactId>java-jwt</artifactId>
<version>3.1.0</version>
</dependency>
参考文章:
https://blog.csdn.net/u011277123/article/details/78918390
https://yq.aliyun.com/php/1290
```

```
JwtToken.java ×
       JwtToken createToken()
       package com.ph.jwt;
       import com.auth0.jwt.JWT;
       import com.auth0.jwt.JWTVerifier;
 4
       import com.auth0.jwt.algorithms.Algorithm;
       import com.auth0.jwt.interfaces.Claim;
       import com.auth0.jwt.interfaces.DecodedJWT;
       import java.util.Calendar;
       import java.util.Date;
       import java.util.HashMap;
       import java.util.Map;
14
        * @author Free码农
15
        * @version I.0
          @since 2017-12-27
18
19
       public class JwtToken {
20
            * 公用密钥-保存在服务端,客户端是不会知道密钥的,以防被攻击
24
            public static String SECRET = "FreeMaNong";
            * 生成Token
             * @return
30
             * @throws Exception
            public static String createToken() throws Exception{
34
                // 签发时间
                Date iatDate = new Date();
36
                // 过期时间- 1分钟过期
38
                Calendar nowTime = Calendar.getInstance();
39
                nowTime.add(Calendar.MINUTE, 1);
                Date expiresDate = nowTime.getTime();
41
42
                Map<String, Object> map = new HashMap<String, Object>();
               map.put("alg", "HS256");
map.put("typ", "JWT");
String token = JWT.create()
44
46
                         .withHeader(map)//header
                        withClaim("name", "Free的农")//payload
.withClaim("age", "28")
.withClaim("org", "今日头条")
47
48
49
                        .withExpiresAt(expiresDate) // 设置过期时间-过期时间要大于签发时间
.withIssuedAt(iatDate) // 设置签发时间
50
                         .sign(Algorithm.HMAC256(SECRET));//加密
                return token;
             * 解密Token
58
59
               @param token
68
               Othrows Exception
            public static Map<String,Claim> verifyToken(String token) throws Exception{
63
                JWTVerifier verifier = JWT.require(Algorithm.HMAC256(SECRET))
64
                        .build();
65
                DecodedJWT jwt = null;
66
                try {
67
                    jwt = verifier.verify(token);
                }catch (Exception e){
                    throw new RuntimeException("登录凭证已过去,请重新登录");
78
                return jwt.getClaims();
74
       }
```

建议: 出于安全考虑,请添加签发时间及过期时间

其它语言 SDK 工具包支持:(详情查看 http://www.jwt.io)

.NET

1C

С

C++

Clojure

Crystal

D

Delphi

Elixir

Erlang

Go

Groovy

Haskell

Haxe

Java

JavaScript

kdb+/Q

Lua

Node.js

Objective-C

Perl

PHP

PostgreSQL

Python

Ruby

Rust

Scala

Swift

三、工具测试方法

- 1、登录网址: https://jwt.io/
- 2、区域 2,3 不用更改
- 3、区域 4 写入密钥
- 4、区域1内容为token值

Encoded PASTE A TOKEN HERE

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30 .EQuqGOwFvi7NRf1SUcUjQnO1drSzwg7Xr1HerrNJ2Y 区域1

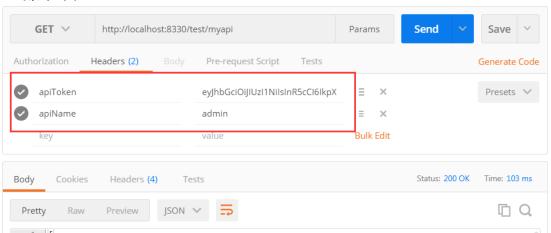
Decoded EDIT THE PAYLOAD AND SECRET



⊗ Signature Verified

SHARE JWT

5、测试工具



用户不正确,token 无效,密钥不对等错误,会返回:

```
"status": 401,
  "msg": "Unauthorized",
  "data": null
}
```