1. 验证码登录
2. 生成token

JWT工具生成token

* 1. 引入依赖

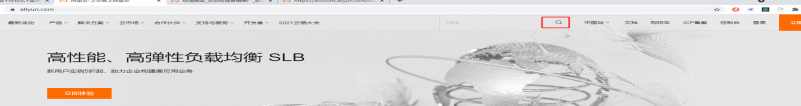
<dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt</artifactId>  
 <version>0.9.0</version>  
 </dependency>

* 1. 生成token的工具类

|  |
| --- |
| **public class** JwtHelper {  //Token过期时间  **private static long** *tokenExpiration* = 24\*60\*60\*1000;  // 签名密钥  **private static** String *tokenSignKey* = **"123456"**;  // 生成token  **public static** String createToken(Long userId, String userName) {  String token = Jwts.*builder*()  .setSubject(**"YYGH-USER"**)  .setExpiration(**new** Date(System.*currentTimeMillis*() + *tokenExpiration*))  .claim(**"userId"**, userId)  .claim(**"userName"**, userName)  // 签名hash  .signWith(SignatureAlgorithm.***HS512***, *tokenSignKey*)  .compressWith(CompressionCodecs.***GZIP***)  .compact();  **return** token;  }  // 获取用户id  **public static** Long getUserId(String token) {  **if**(StringUtils.*isEmpty*(token)) **return null**;  Jws<Claims> claimsJws = Jwts.*parser*().setSigningKey(*tokenSignKey*).parseClaimsJws(token);  Claims claims = claimsJws.getBody();  Integer userId = (Integer)claims.get(**"userId"**);  **return** userId.longValue();  }  // 获取用户名称  **public static** String getUserName(String token) {  **if**(StringUtils.*isEmpty*(token)) **return ""**;  Jws<Claims> claimsJws  = Jwts.*parser*().setSigningKey(*tokenSignKey*).parseClaimsJws(token);  Claims claims = claimsJws.getBody();  **return** (String)claims.get(**"userName"**);  }  // 测试方法  **public static void** main(String[] args) {  // 根据用户名和用户id生成token  String token = JwtHelper.*createToken*(1L, **"55"**);  System.***out***.println(token);  // 获取用户id  System.***out***.println(JwtHelper.*getUserId*(token));  // 获取token  System.***out***.println(JwtHelper.*getUserName*(token));  } } |

1.3、阿里云短信服务

1.3.1、网址 <https://www.aliyun.com/>

1.3.2、点击搜索

1.3.3、输入短信服务跳转到

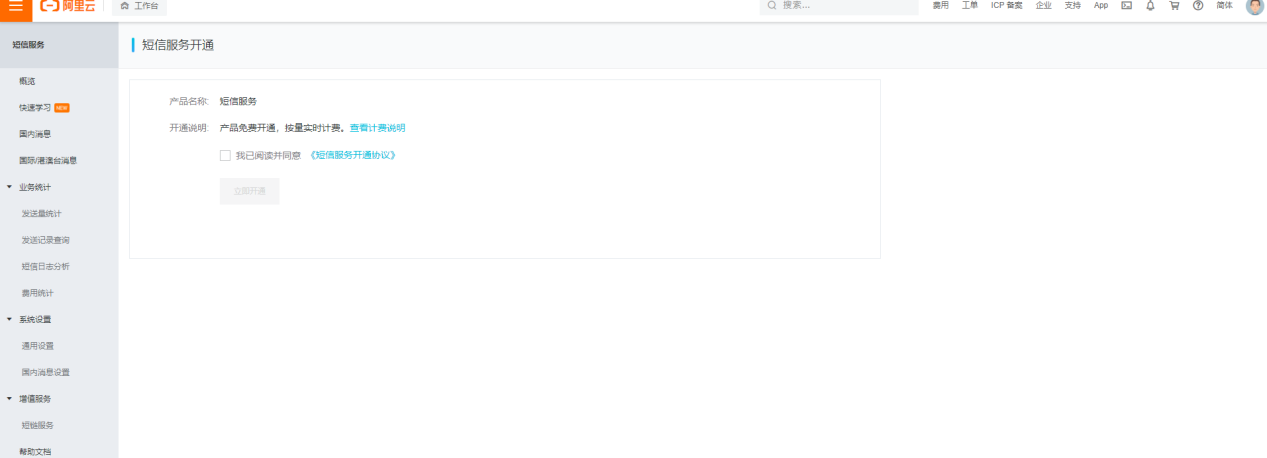


点击免费开通



支付宝扫码登录

按照提示就可以看到短信服务的界面了



点击立即开通之后还得需要实名认证：认证的方式有个人认证和企业认证

实名认证以后就看到这个页面



阿里云的短信服务一共设置3处：

1. 签名管理
2. 模板管理
3. Accesskey

点击accesskey管理， 就可以看到accesskeyId和secret

例如 accesskeyId： LTAI5tLyW8vTXgPKnwZrpe2n

Secret： v9CccaBLvPC8kP6OFLmceCWolWiDrW

1.4、准备工作做完就可以整合sms短信服务了

1.4.1、引入依赖

<dependency>  
 <groupId>com.aliyun</groupId>  
 <artifactId>aliyun-java-sdk-core</artifactId>  
 <version>4.1.1</version>  
 </dependency>

1.4.2、启动redis，短信验证码存在redis里面， 设置过期时间（比如5分钟）

1.4.3、这三个值前面准备工作审核通过之后得到的，调短信发送的时候需要用

aliyun.sms.regionId = default  
aliyun.sms.accessKeyId = LTAI5tLyW8vTXgPKnwZrpe2n  
aliyun.sms.secret = v9CccaBLvPC8kP6OFLmceCWolWiDrW

1.4.4、调用方法进行短信验证码的发送

Controller里面：

@GetMapping("/send/{phone}")  
public Object sendCode(@PathVariable String phone) {  
 // 从redis获取验证码，获取到直接返回 ( 发送一次，5分钟内不能再次发送验证码 )  
 String code = redisTemplate.opsForValue().get(phone).toString();  
 if(!StringUtils.isEmpty(code)) {  
 return "已经发送过验证码， 5分钟内有效";  
 }  
  
 // 获取不到， 生成验证码设置有效时长，放在redis里面  
 double aa = (double)Math.random()\*9000+1000;  
 int createCode = (int)Math.ceil(aa);  
 System.err.println(createCode);  
 boolean isSend = smsService.sendCode(phone, createCode); // 调用阿里云的短信服务  
  
 if(isSend) {  
 // 短信发送成功，存到redis里面设置上过期时间  
 redisTemplate.opsForValue().set(phone, createCode, 5, TimeUnit.MINUTES); // 存到redis中5分钟过期  
 return "验证码发送成功";  
 }else{  
 return "验证码发送失败";  
 }  
}

ServiceImpl里面：

// 调阿里云短信接口  
@Override  
public boolean sendCode(String phone, int createCode) {  
  
 // 判断手机号是否为空  
 if(StringUtils.isEmpty(phone)) {  
 return false;  
 }  
  
 // 以下都是固定写法， 不能变  
 DefaultProfile profile = DefaultProfile.getProfile(REGION\_Id, ACCESS\_KEY\_ID, SECRECT);  
 IAcsClient client = new DefaultAcsClient(profile);  
 CommonRequest request = new CommonRequest();  
 //request.setProtocol(ProtocolType.HTTPS);  
 request.setMethod(MethodType.POST);  
 request.setDomain("dysmsapi.aliyuncs.com");  
 request.setVersion("2017-05-25");  
 request.setAction("SendSms");  
  
  
 // 设置参数  
 //手机号  
 request.putQueryParameter("PhoneNumbers", phone);  
 //签名名称（和申请网站上的是一样的）  
 request.putQueryParameter("SignName", "我的谷粒在线教育网站");  
 //模板code（和申请网站上的是一样的）  
 request.putQueryParameter("TemplateCode", "SMS\_180051135");  
 //验证码 使用json格式 {"code":"123456"}  
 Map<String,Object> param = new HashMap();  
 param.put("code",createCode);  
 request.putQueryParameter("TemplateParam", JSONObject.toJSONString(param));  
  
 //调用方法进行短信发送  
 try {  
 CommonResponse response = client.getCommonResponse(request);  
 System.out.println(response.getData());  
 return response.getHttpResponse().isSuccess();  
 } catch (ServerException e) {  
 e.printStackTrace();

return false;  
 } catch (ClientException e) {  
 e.printStackTrace();

return false;  
 }   
}