|  |  |
| --- | --- |
| **summary:** | HMS Core WalletKit接入 |
| **url:** | HMSWalletKit |
| **categories:** | HMS |
| **status:** | Published |
| **feedback link:** | <https://developer.huawei.com/consumer/cn/support/feedback> |
| **authors:** | HUAWEI |

# HMS Core WalletKit接入

## **介绍**

Duration: 1:00

### 简介

WalletKit支持用户领取商户卡券到华为钱包，包括会员卡、礼品卡、优惠券、登机牌、活动门票、交通券。您可以采用集成按钮或者提供链接的方式，以便用户将上述卡券保存到华为钱包，并在华为钱包中查看及管理。

开发者需要：

* 在华为开发者联盟注册开发者帐号，并在AppGallery Connect上创建应用以及配置应用信息和开通WalletKit API
* 在应用内选择并注册WalletKit服务，设置商户服务号和回调地址。并使用工具生成公私钥后，将公钥同步到华为服务器。
* 使用Android Studio IDE创建Project，并配置接入WalletKit SDK
* 调用WalletKit API 接口，推送商户模板信息到华为服务器。
* 根据WalletKit开发指南中集成钱包服务SDK部分，实现WalletKit SDK的调用以及调试

### 您将建立什么

在这个Codelab中，你将使用已经创建好的Demo Project实现对WalletKit SDK的调用，通过Demo Project你可以体验到：

* 调用WalletKit SDK 构建对象信息，添加卡券到华为钱包

### 您将会学到什么

* 如何在华为开发者联盟上创建应用
* 如何开通WalletKit API
* 如何导入WalletKit SDK
* 如何调用WalletKit SDK添加卡券到华为钱包

## 您需要什么

Duration: 1:00

### 硬件要求

* 开发计算机（台式机或笔记本电脑），运行Windows 10操作系统。
* 预装了华为移动服务3.0.3.300以上版本的HUAWEI手机一部，用于真机调试运行demo。

|  |
| --- |
| **提示：**以上硬件环境和设备需要开发者提前准备好。 |

### 软件要求

* [Android Studio 3.X](https://developer.android.com/studio)
* JDK 1.8 及以上
* SDK Platform 19及以上
* Gradle 4.6及以上

|  |
| --- |
| **提示：**以上软件开发环境需要开发者提前准备好。 |

## 能力接入准备

Duration: 20:00

集成HUAWEI HMS Core能力，需要完成以下准备工作

* 创建AGC应用
* 创建Android Studio工程
* 生成签名证书
* 生成签名证书指纹
* 配置签名证书指纹
* 添加应用包名并保存配置文件
* 配置Maven仓地址及AGC gradle插件
* 在Android Studio配置签名文件

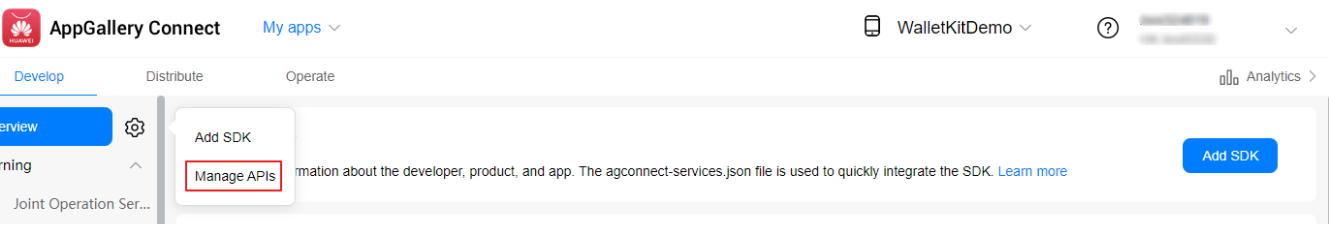
具体操作，请按照《[HUAWEI HMS Core集成准备](https://developer.huawei.com/consumer/cn/codelab/HMSPreparation/index.html)》中详细说明来完成。

|  |
| --- |
| **提示：**需要通过注册成开发者才能完成集成准备中的操作。 |

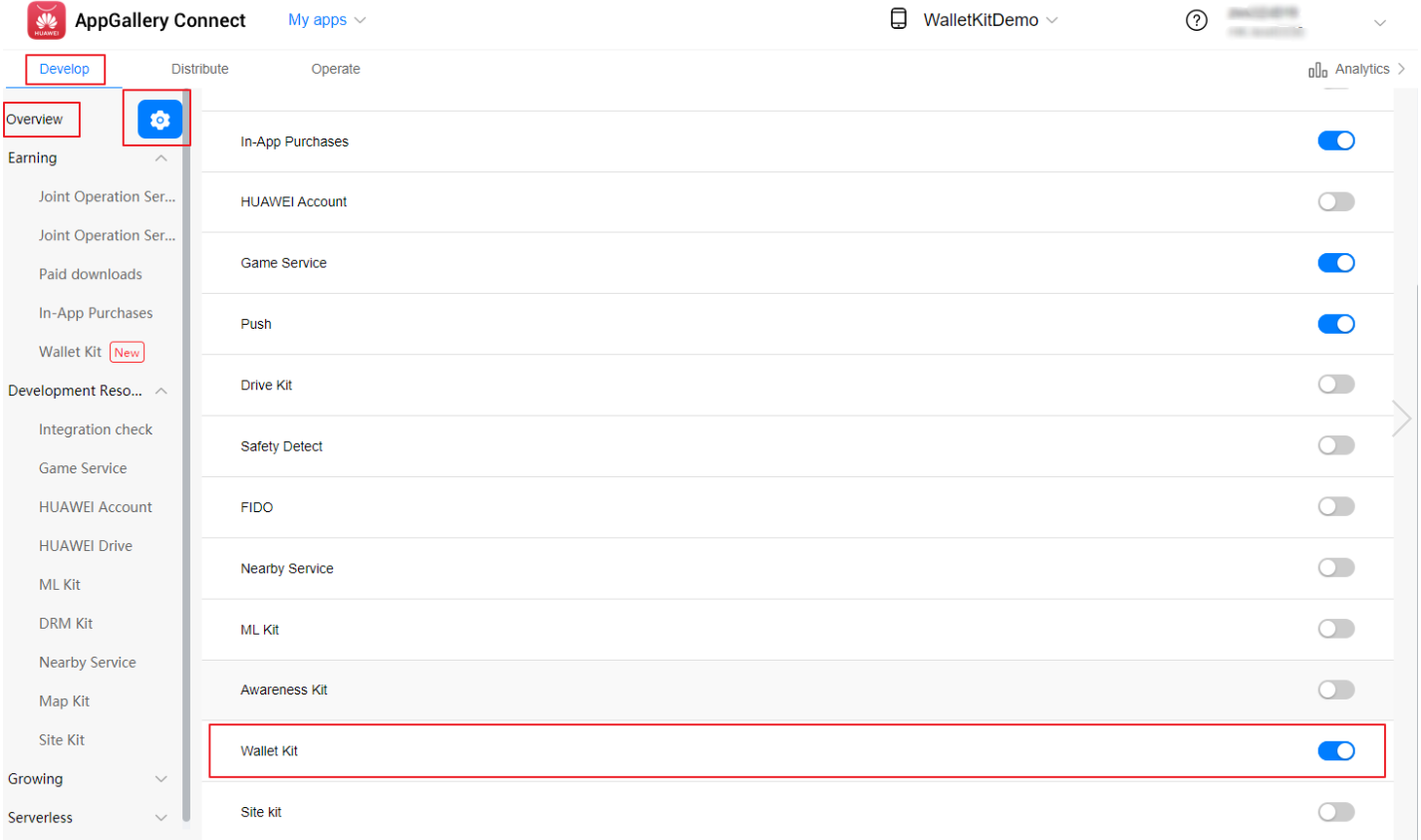
## 开通WalletKit API服务

Duration: 1:00

1. 在[华为开发者联盟AppGallery Connect](https://developer.huawei.com/consumer/cn/service/josp/agc/index.html)的应用页面中点击“开发”页签，点击“概览”后的图标，选择“API管理”。



1. 打开“Wallet Kit”API开关。

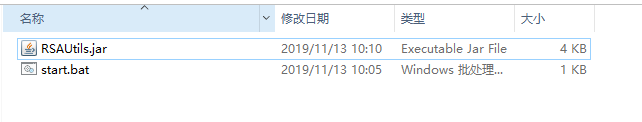


至此，已经为创建的应用开通了“WalletKit”API服务。

## 生成公私钥对

Duration: 2:00

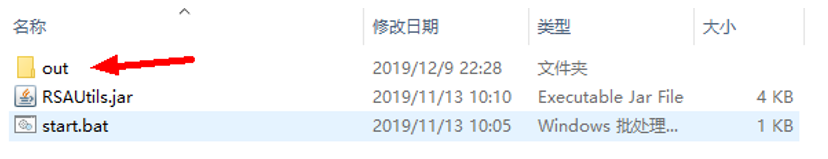
1. 下载RSAUtils.rar压缩包，解压并打开RSAUtils文件夹：



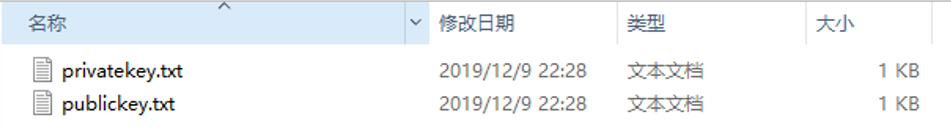
1. 双击运行**start.bat**文件



1. 运行成功后，界面会提示公私钥生成的文件路径，默认为当前路径的out文件夹下：



1. out文件夹内容如下：

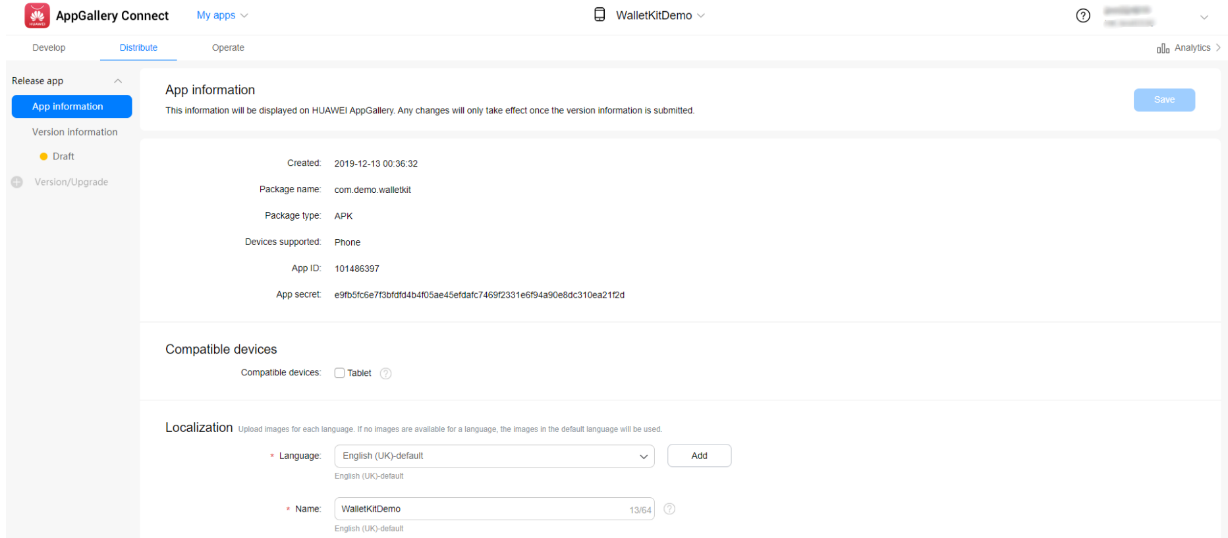


|  |
| --- |
| **说明：**  publickey.txt：公钥文件，注册WalletKit服务时使用。用于用户添加卡券到华为钱包时，华为侧对请求进行验签。  privateKey.txt：私钥文件，请妥善保存。后续生成对象JSON字符串后签名时使用。 |

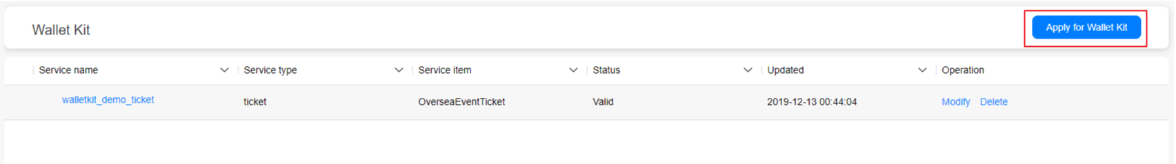
## 注册WalletKit服务

Duration: 3:00

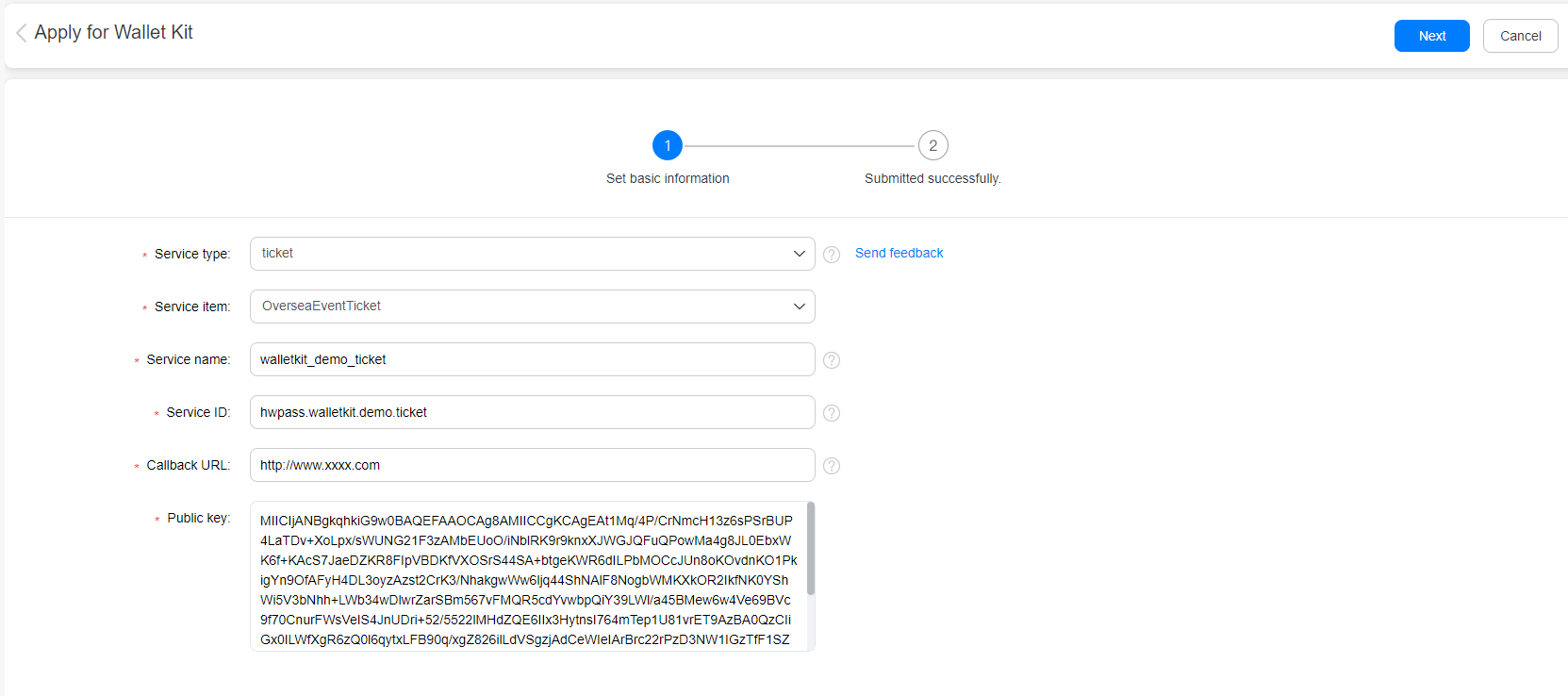
1. 登录[AppGallery Connect网站](https://developer.huawei.com/consumer/cn/service/josp/agc/index.html)，点击“我的应用”，进入应用信息页面。



1. 在“开发”页签，左侧导航选择“盈利 > wallet kit”， 在页面右上角申请wallet kit服务。



1. 设置服务基本信息，类型选择App应用，注册服务号，及回调地址，并将前述步骤的公钥内容填入后点击下一步完成创建：

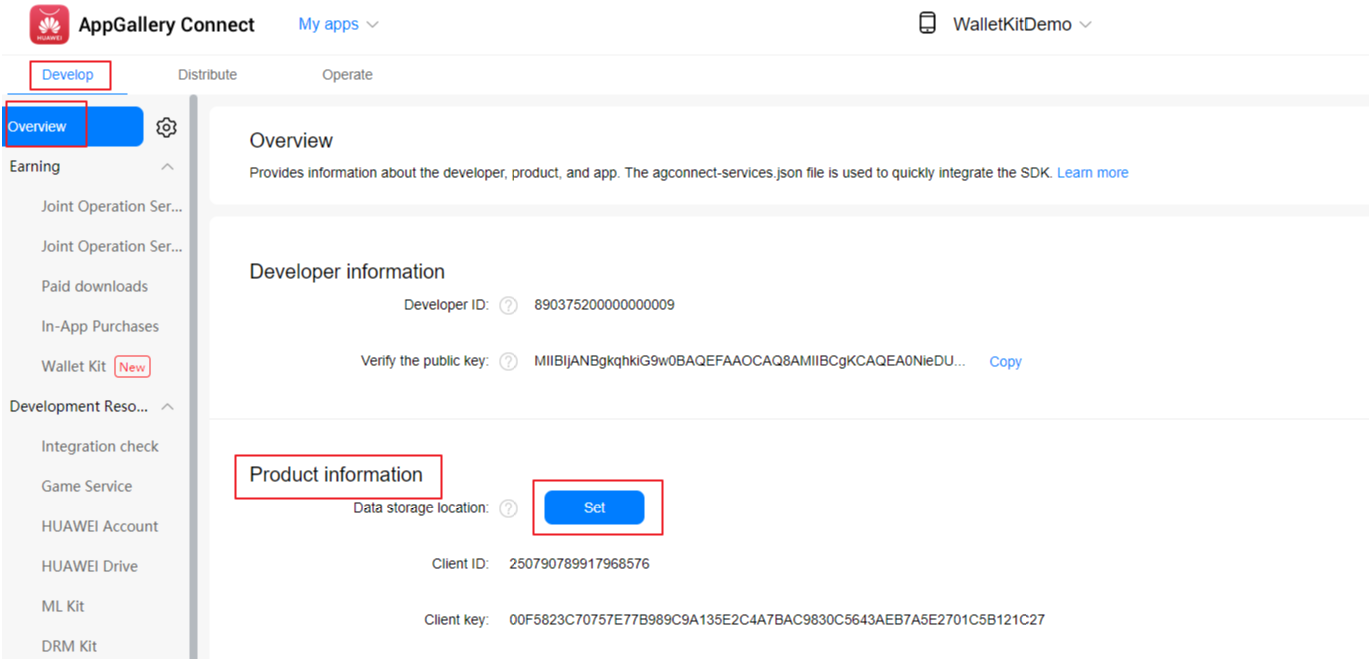


## 集成WalletKit SDK

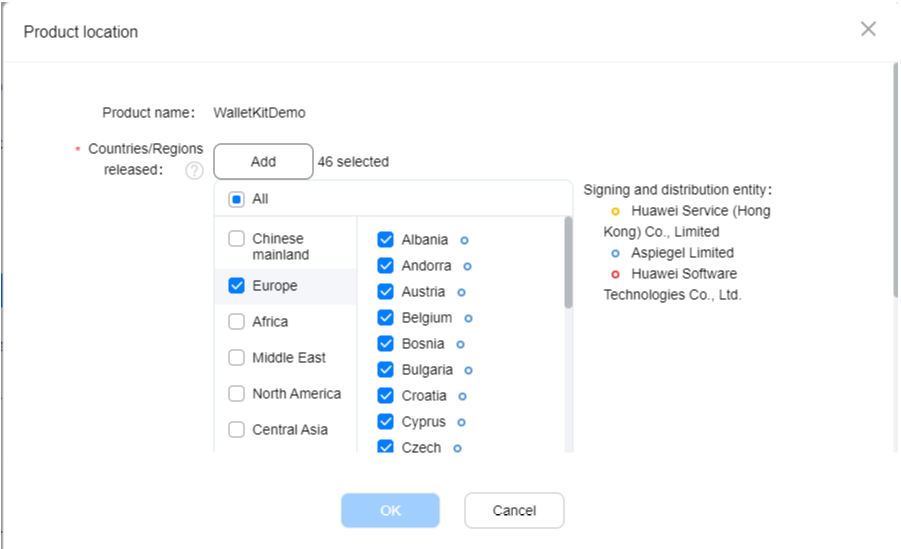
Duration: 5:00

### 获取配置文件

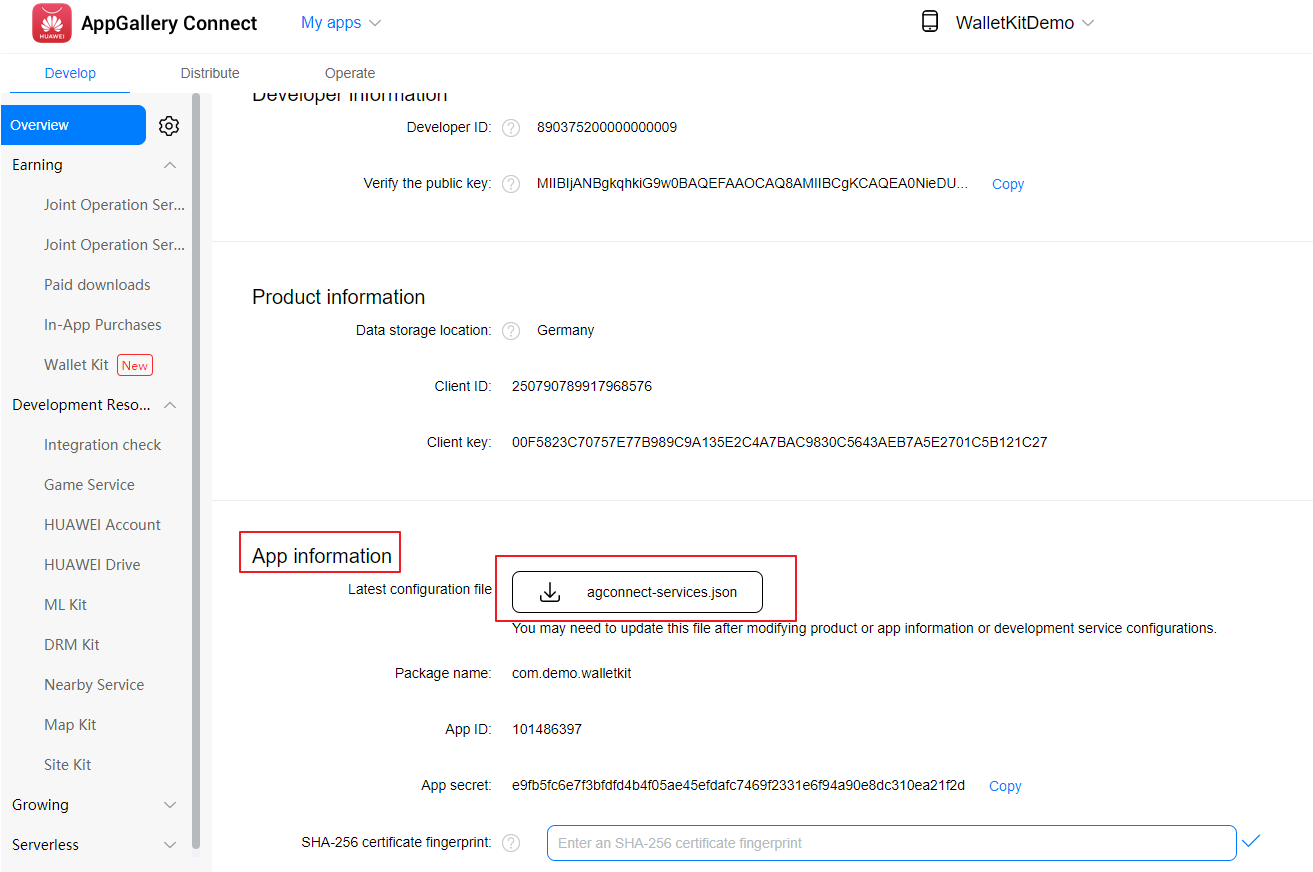
1. 打开[华为开发者联盟AppGallery Connect](https://developer.huawei.com/consumer/cn/service/josp/agc/index.html)应用管理中之前创建的应用，并选择“**开发->概览**”，点击数据存储位置设置按钮



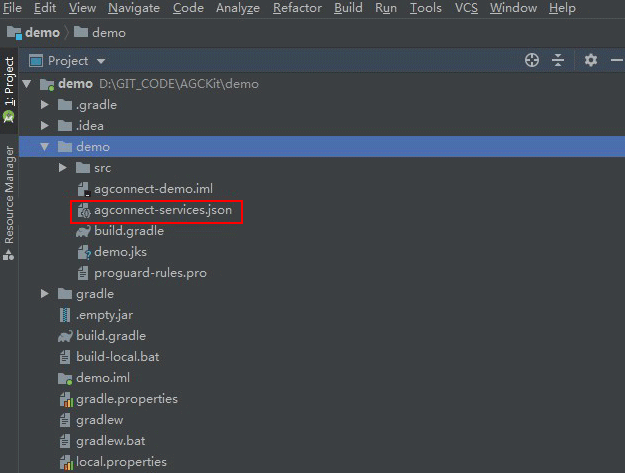
1. 根据应用实际情况，设置对应数据存储位置，然后点击确认按钮



1. 下载应用中的“**agconnect-services.json**”

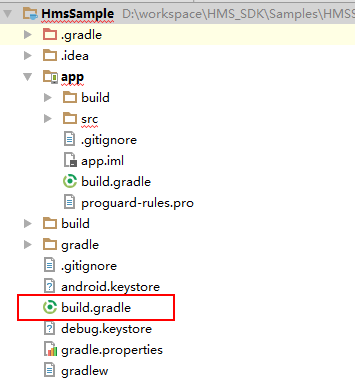


1. 将 “agconnect-services.json”文件移至Android Studio开发工程应用模块的根目录



### 配置WalletKit SDK依赖

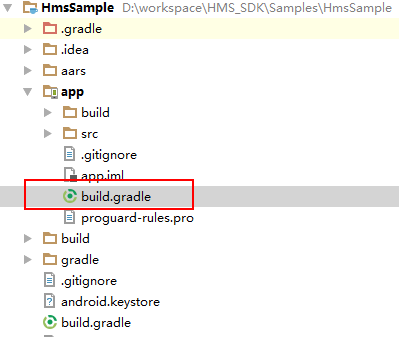
1. 打开AndroidStudio项目级build.gradle文件。



1. 在buildscript -> repositories和allprojects ->repositories里面都配置HMS SDK的maven仓地址以及agc编译依赖。

|  |
| --- |
| buildscript {  repositories {  maven { url 'http://developer.huawei.com/repo/' }  }  dependencies {  classpath 'com.android.tools.build:gradle:3.3.2'  classpath 'com.huawei.agconnect:agcp:1.0.0.300'  } }  allprojects {  repositories {  maven { url 'http://developer.huawei.com/repo/' }  } } |

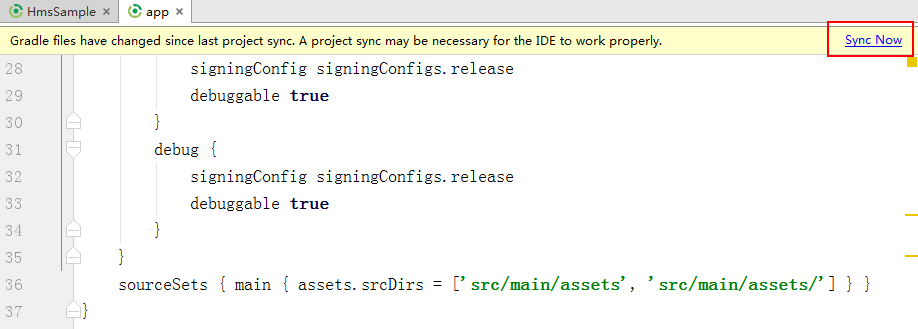
1. 打开应用级的**build.gradle**文件。



1. 在“**dependencies**”中添加如下WalletKit SDK。

|  |
| --- |
| dependencies {  implementation 'com.huawei.hms:wallet:2.0.0.300' } |

1. 点击sync now 同步工程



### 配置混淆脚本

|  |
| --- |
| **提示：如果开发者有使用到代码混淆功能，请配置混淆配置文件避免SDK被混淆导致功能异常**。 |

1. 打开Android Studio工程的混淆配置文件**proguard-rules.pro**。
2. 加入混淆配置。

|  |
| --- |
| -ignorewarnings  -keepattributes \*Annotation\*  -keepattributes Exceptions  -keepattributes InnerClasses  -keepattributes Signature  -keepattributes SourceFile,LineNumberTable  -keep class com.hianalytics.android.\*\*{\*;}  -keep class com.huawei.updatesdk.\*\*{\*;}  -keep class com.huawei.hms.\*\*{\*;} |

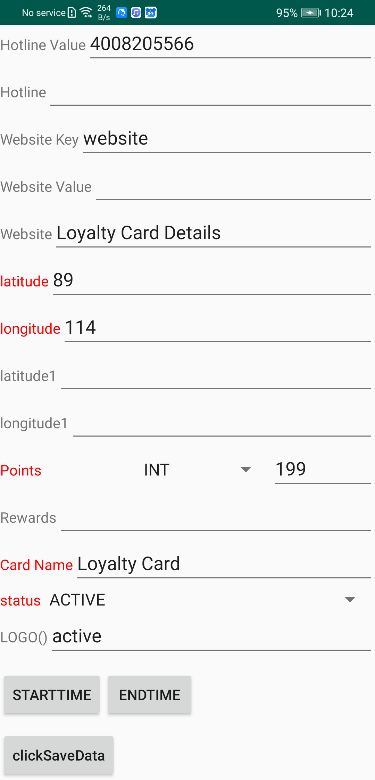
1. 如果开发者使用了AndResGuard，需要在混淆配置文件中加入AndResGuard白名单

|  |
| --- |
| "R.string.hms\*",  "R.string.connect\_server\_fail\_prompt\_toast",  "R.string.getting\_message\_fail\_prompt\_toast",  "R.string.no\_available\_network\_prompt\_toast",  "R.string.third\_app\_\*",  "R.string.upsdk\_\*",  "R.layout.hms\*",  "R.layout.upsdk\_\*",  "R.drawable.upsdk\*",  "R.color.upsdk\*",  "R.dimen.upsdk\*",  "R.style.upsdk\*" |

## 界面设计

Duration: 20:00

WalletKit SDK的交互接口主要用于添加卡券到华为钱包，需要设置会员卡号、积分、会员名称等信息。本次Codelab中您可以在您的Android Studio工程中创建一个布局页面，参照下图进行UI设计。



## 添加卡券到华为钱包功能

Duration: 30:00

|  |
| --- |
| **提示：WalletKit使用JWT作为参数发送到华为服务器，完成卡券生成并绑定到用户的动作**。 |

添加会员卡到华为钱包代码样例

1. 展示华为保存到华为钱包按钮。

应用在页面展示保存到华为钱包的按钮，保存到华为钱包按钮规范可参考[华为图标规范](https://developer.huawei.com/consumer/cn/service/hms/catalog/HwJointOperationAppV3.html?page=hmssdk_jointOper_devguide_icon_rules)。

|  |
| --- |
| **提示：提供的Codelab样例是用Button实现的，开发者实际接入请按规范使用华为保存到华为钱包图标**。 |

1. 构建请求并发送到华为服务器

* 构建会员卡对象。

|  |
| --- |
| LoyaltyWalletObject.PassBuilder builder = LoyaltyWalletObject.newPassBuilder();  builder.setPassTypeIdentifier(typeId)  .setCardNumber(cardNumber)  .setPassBarcodeAlternateText(barcodeAlternateText)  .setPassBarcodeType(barcodeType)  .setPassBarcodeValue(barcodeValue)  .setPassStyleIdentifier(styleIdentifier)  .setOrganizationPassId(organizationPassId)  .setPassIssuerName(issuerName)  .setPoints(points)  .setPassProgramName(programName)  .setPassState(state)  .setPassTimeDuration(timeDuration);  //添加经纬度List<LatLng> locations = new ArrayList<>();  locations.add(new LatLng(latitudSecond,longitudSecond));  builder.addPassLocations(locations);  //生成会员卡对象  LoyaltyWalletPass walletPassObj = builder.build(); |

* 将对象转化为JWT。

|  |
| --- |
| // Fixed public key provided by HUAWEI used to encrypt the sessionKey  String sessionKeyPublicKey = "";  // sessionkey is used to encrypt the payload  data.String sessionKey = RandomUtils.generateSecureRandomFactor(16);  // Encrypt JSON as payload of JWT  byte[] sessionKeyBytes = EncodeUtil.hex2Byte(sessionKey);  String objectJson = hwWalletObject.toJson();  String payLoadEncrypt = AESUtils.encryptAESCBC(objectJson.getBytes(EncodeUtil.UTF\_8), sessionKeyBytes);  // Encrypt sessionKey and set into header of JWT  Map<String, Object> headerMap = new HashMap<>();  try {  String sessionKeyPlaintext = RSA.encrypt(sessionKey.getBytes(), sessionKeyPublicKey, "RSA/ECB/OAEPwithSHA-256andMGF1Padding", "UTF-8");  headerMap.put("sessionKey", sessionKeyPlaintext);  } catch (Exception e) {  System.out.println(e.getMessage());  }  JwtBuilder builder = Jwts.builder().setHeader(headerMap).setPayload(payLoadEncrypt);String userToken = builder.compact();  String content = userToken.substring(0, userToken.length() - 1);  String sign = "";  String charset = "utf-8";  try {  // 使用注册时生成的privateKey签名  PKCS8EncodedKeySpec priPKCS8 = new PKCS8EncodedKeySpec(Base64.getDecoder().decode(privateKey));  KeyFactory keyf = KeyFactory.getInstance("RSA");  PrivateKey priKey = keyf.generatePrivate(priPKCS8);  java.security.Signature signatureObj = java.security.Signature.getInstance(SIGN\_ALGORITHMS256);  signatureObj.initSign(priKey);  signatureObj.update(content.getBytes(charset));  byte[] signed = signatureObj.sign();  sign = Base64.getEncoder().encodeToString(signed);  } catch (Exception ex) {  …  }  String jwt\_content = userToken + sign; |

* 创建CreateWalletPassRequest

|  |
| --- |
| CreateWalletPassRequest request = CreateWalletPassRequest.newBuilder()  .setJwt(jwt\_content)  .build(); |

* 创建Client实例

|  |
| --- |
| // walletOptions目前暂不支持配置，可以设置为null  WalletPassClient walletObjectsClient = Wallet.getWalletPassClient(PassTestActivity.this, walletOptions); |

* 创建Task

|  |
| --- |
| WalletPassClient walletObjectsClient = Wallet.getWalletPassClient(PassTestActivity.this, walletOptions);  其中walletOptions目前暂不支持配置，可以直接设置null |

* 执行Task

|  |
| --- |
| // Save\_flag为任务执行标志，用于接收任务执行结果。  ResolveTaskHelper.excuteTask(task, XXXActivity.this, SAVE\_FLAG); |

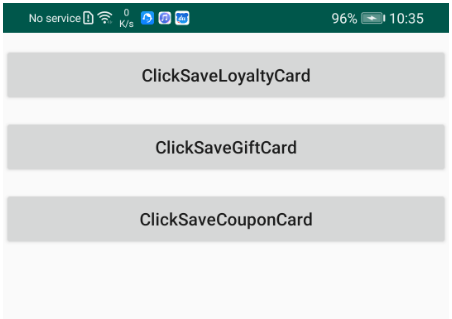
* 接收添加会员卡结果

|  |
| --- |
| public void onActivityResult(int requestCode, int resultCode, Intent data)  {  switch (requestCode) {  case SAVE\_FLAG:  switch (resultCode) {  case Activity.RESULT\_OK: Toast.makeText(this, "Save success",Toast.LENGTH\_LONG).show();  break;  case Activity.RESULT\_CANCELED: Toast.makeText(this, "(Reason：1:User cancel 2: HMS not install or not register)", Toast.LENGTH\_LONG).show();  break;  default:  if (data != null) {  int errorCode = data.getIntExtra(WalletCommonConstants.EXTRA\_ERROR\_CODE, -1);  Toast.makeText(this, "Save failed[" + errorCode + "]：" + analyzeErrorCode(errorCode), Toast.LENGTH\_LONG).show();  } else {  Toast.makeText(this, "保存失败：返回data为空",Toast.LENGTH\_LONG).show();  }  break;  }  }  } |

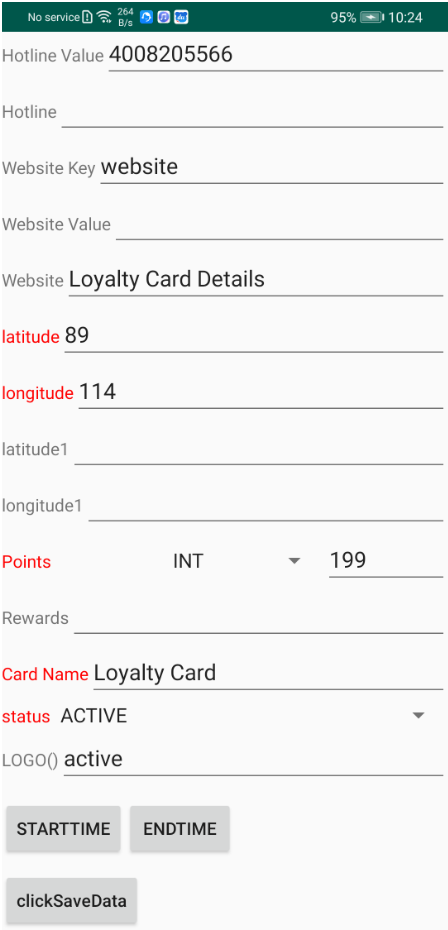
## 打包测试

Duration: 3:00

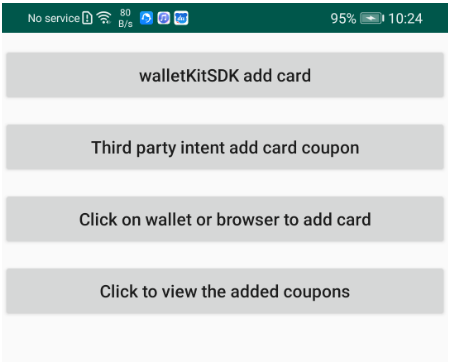
1. 开发完成后，点击图标  ，运行Android Studio工程打包生成APK，并安装在测试手机上。界面如下图所示：



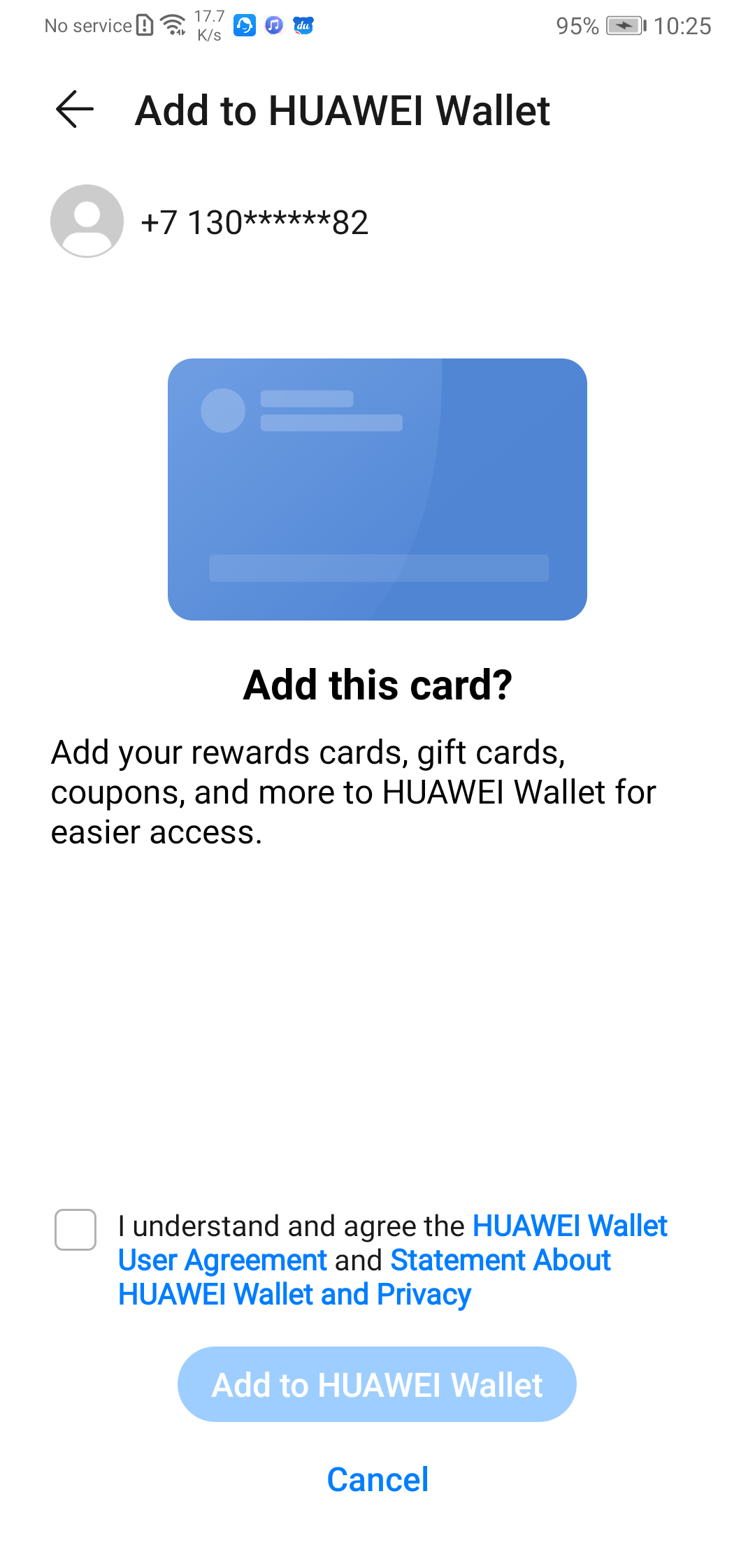
1. 点击第一个按钮，ClickSaveLoyaltyCard，进入会员卡属性配置页面：



1. 输入必填信息，如会员名称，商户名称等。点击“clickSaveData”后跳转如下页面：



1. 点击“walletKitSDK add card”，会拉起WalletKit Adapter确认添卡页面：



1. 首次使用会出现勾选协议复选框，勾选后点击“Add to HUAWEI Wallet”,服务器完成添卡后返回结果码（成功为0），并提示用户保存成功。

## 恭喜你

Duration: 0:00

干得好，你已经成功完成了Codelab并学到了：

* 如何在[华为开发者联盟](https://developer.huawei.com/consumer/cn/service/josp/agc/index.html)上创建应用
* 如何配置开通WalletKit API服务
* 如何导入WalletKit SDK并API调用完成添加卡券

## 参考

[钱包服务相关API介绍](https://developer.huawei.com/consumer/cn/service/hms/catalog/huaweiid_v3.html)

本次Codelab使用的DEMO原码

##### [源码下载](https://obs.cn-north-2.myhwclouds.com/hms-ds-wf/sdk/CodeLab_WalletKit_Demo.zip)