1. **登录验证流程**

TVS（Tencent Voice Service）开放平台的帐号验证和Token传递流程仿照AVS（Alexa Voice Service）的验证流程来进行的。

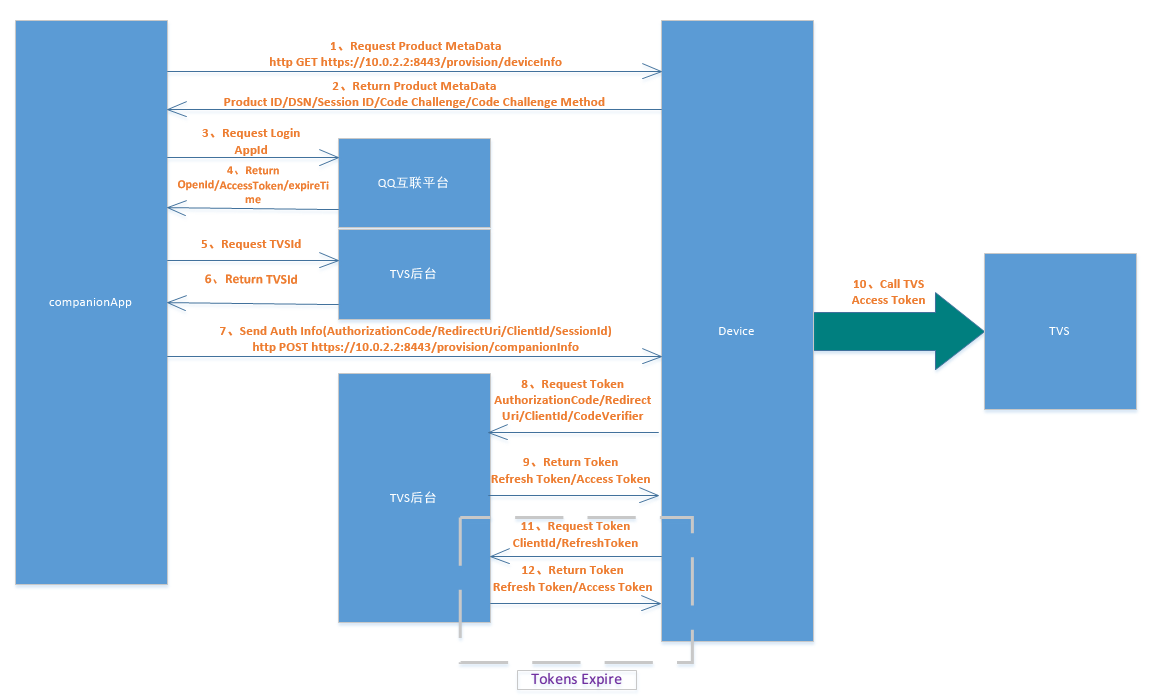
1. 厂商APP-LWA（Login With Amazon）-AVS音箱-AVS验证流程



1. 厂商APP-微信开放平台-TVS后台-TVS音箱-TVS验证流程



3、厂商APP-QQ互联平台-TVS后台-TVS音箱-TVS验证流程



**二、厂商APP接入TVS登录系统配置步骤**

1. 注册AppId

在微信开放平台和QQ互联平台注册AppId。微信开放平台注册成功后提供AppId、AppSecret、应用主包名；QQ互联平台注册成功后提供AppId。

1. gradle配置

打开工程app目录下的build.gradle

（1）确保defaultConfig下的applicationId为微信开放平台下注册的包名。

（2）配置dependencies

compile (name: tvsloginlib-XXX-release, ext:'aar')

compile 'com.android.support:design:xx.x.0'

compile 'com.tencent.mm.opensdk:wechat-sdk-android-with-mta:+'

compile 'com.google.code.gson:gson:2.8.1'

compile 'com.squareup.okhttp3:okhttp:3.8.1'

（3）配置flatDir

repositories {

flatDir {

dirs 'libs'

}

}

（4）确保minSdkVersion大于等于15。

（5）如有别的module需要引用这个aar所在的module，需要在另一个module的gradle里配置如下

repositories {

flatDir {

dirs project(':[aarmodulename]').file('libs')

}

}

3、aar配置

将tvsloginlib-XXX-release.aar放入app\libs目录下。

4、Manifest配置

打开工程AndroidManifest.xml

（1）确保package名称与微信开放平台下注册的包名一致。

（2）将application的name改为com.tencent.ai.tvs.LoginApplication，或自定义的继承自LoginApplication的Application类，并添加android:exported=”true”属性。

1. **TVSSDK接口及使用说明**
2. **LoginProxy**

TVS登录功能代理类

**getInstance**

获取LoginProxy实例

static LoginProxy getInstance(String appIdWx, String appIdQQOpen, Context context)

参数

appIdWx：传入微信开放平台申请的appId

appIdQQOpen：传入QQ互联平台申请的appId

context：传入Application Context

**getWebInstance**

获取LoginProxy实例（适用于扫码登录场景）

static LoginProxy getWebInstance(String appidWxWeb, String appIdQQOpenWeb, Context context)

参数

appidWxWeb：传入微信开放平台申请的网站应用的appId，传入null或””默认为腾讯叮当Web版应用appId

appIdQQOpenWeb：传入QQ互联平台申请的网站应用的appId，传入null或””默认为腾讯叮当Web版应用appId

context：传入Application Context

**getInfoManager**

获取LoginInfoManager实例

LoginInfoManager getInfoManager(ELoginPlatform platform)

参数

platform：传入需要获取哪个帐号平台的登录信息

**requestLogin**

请求TVS登录

void requestLogin(ELoginPlatform platform, String productId, String dsn, Activity activity)

参数

platform：传入需要登录哪个帐号平台

productId：设备的productId

dsn：设备的序列号

activity：登录Activity实例

**requestTokenVerify**

请求验票刷票

void requestTokenVerify(ELoginPlatform platform, String productId, String dsn)

参数

platform：传入在哪个帐号平台验票刷票

productId：设备的productId

dsn：设备的序列号

**isTokenExist**

判断Token信息存在与否

boolean isTokenExist(ELoginPlatform platform, Context context)

参数

platform：传入需要判断哪个帐号平台票据信息是否存在

context：当前应用上下文

**clearToken**

清除Token信息，用于注销

void clearToken(ELoginPlatform platform, Context context)

参数

platform：传入需要清除哪个帐号平台的票据信息

context：当前应用上下文

**setOwnActivity**

设置Wup接口调用所属的Activity，用于WUPResp与UI线程进行同步

void setOwnActivity(Activity activity)

参数

activity：传入用于授权后回调的Activity

**setAuthorizeListener**

设置票据验证监听器

void setAuthorizeListener (AuthorizeListener listener)

参数

listener：AuthorizeListener实例

**handleQQOpenIntent**

QQ登录帐号信息回调

void handleQQOpenIntent(int requestCode, int resultCode, Intent data)

参数

requestCode：Activity跳转请求Code，确认返回数据是哪个Activity的

resultCode：子Activity通过setResult返回的code

data：Intent对象，带有返回的数据

**handleQQOpenUserInfo**

QQ登录用户信息回调

void handleQQOpenIntent()

**getClientId**

获取用于对齐AVS厂商信息的ClientId

String getClientId(ELoginPlatform platform)

参数

platform：获取哪个登录平台的信息

**getUserId**

获取用户Id

String getUserId()

**setLoginEnv**

设置帐号后台环境

void setLoginEnv(ELoginEnv env)

参数

env：帐号后台环境类型

**setUserCenterEnv**

设置用户中心网络环境

void setUserCenterEnv(ELoginEnv env)

参数

env：用户中心网络环境类型

**requestGetCaptcha**

获取手机号短信验证码

void requestGetCaptcha(ELoginPlatform platform, String phoneNumber)

参数

platform：帐号类型

phoneNumber：需要绑定的手机号

**requestBindPhoneNumber**

绑定手机号

void requestBindPhoneNumber(ELoginPlatform platform, String phoneNumber, String captcha)

参数

platform：帐号类型

phoneNumber：需要绑定的手机号

captcha：手机验证码

**isLocationOpValid**

地址操作是否合法

boolean isLocationOpValid(ELoginPlatform platform)

参数

platform：帐号类型

返回值

true：合法，可以进行地址操作

false：非法，不可进行地址操作

**requestBindLocation**

常用地址绑定操作

void requestBindLocation(ELoginPlatform platform, LocationInfo homeLocationInfo, LocationInfo companyLocationInfo)

参数

platform：帐号类型

homeLocationInfo：常用地址（家庭）信息

companyLocationInfo：常用地址（公司）信息

**requestQueryLocation**

常用地址查询操作

void requestQueryLocation(ELoginPlatform platform)

参数

platform：帐号类型

**setBindingListener**

设置绑定信息监听器

void setBindingListener(BindingListener listener)

参数

listener：绑定信息监听器

**toUserCenter**

跳转用户中心页面（不带回调）

void toUserCenter(EUserAttrType type, DeviceManager deviceManager)

参数

type：H5页面类型

deviceManager：设备信息类型

**toUserCenter**

跳转用户中心页面（带回调）

void toUserCenter(EUserAttrType type, DeviceManager deviceManager, UserCenterStateListener listener)

参数

type：H5页面类型

deviceManager：设备信息类型

listener：用户中心帐号状态回调

**requestSetPushMapInfoEx**

设置推送绑定接口

void requestSetPushMapInfoEx(ELoginPlatform platform, PushInfoManager pushInfoManager, DeviceManager deviceManager)

参数

platform：帐号类型

pushInfoManager：推送信息管理器

deviceManager：设备信息管理器

**requestDelPushMapInfo**

解除设备推送绑定接口

void requestDelPushMapInfo(ELoginPlatform platform, PushInfoManager pushInfoManager, DeviceManager deviceManager)

参数

platform：帐号类型

pushInfoManager：推送信息管理器

deviceManager：设备信息管理器

**requestGetPushDeviceInfo**

void requestGetPushDeviceInfo(ELoginPlatform platform)

参数

platform：帐号类型

**getMemberStatus**

获取设备领取会员信息接口

void getMemberStatus(ELoginPlatform platform, DeviceManager deviceManager)

参数

platform：帐号类型

deviceManager：设备信息管理器

**isWXAppInstalled**

判断手机是否安装微信

boolean isWXAppInstalled()

**isWXAppSupportAPI**

判断手机微信App版本是否支持登录

boolean isWXAppSupportAPI()

**requestQRLogin**

请求扫码登录

void requestQRLogin(Activity activity, QRStateListener stateListener, QRCustomViewListener customViewListener, boolean isTestEnv, boolean isDebugEnabled)

参数

activity：传入需要接受扫码回调的Activity实例

stateListener：扫码登录回调监听器

customViewListener：自定义View监听回调

isTestEnv：H5是否测试环境

isDebugEnabled：是否打开H5调试模式

**tvsAuth**

请求授权到TVS叮当平台（包括微信授权、刷票；QQ授权）

void tvsAuth(ELoginPlatform platform, String acctRet);

参数：

platform：帐号类型

acctRet：官方授权返回Json字段

**tvsQQOpenVerify**

请求验证QQ互联登录票据

void tvsQQOpenVerify(String appId, String openID, String accessToken);

参数：

appid：QQ互联Appid

openID：QQ互联登录的OpenID

accessToken：QQ互联登录的票据信息

**tvsSetUser**

设置登录帐号的用户信息

void tvsSetUser(ELoginPlatform platform, String userRet);

参数：

platform：帐号类型

userRet：官方请求用户信息返回Json字段

**tvsOpenMiniProgram**

打开指定小程序，并接收小程序跳回extMsg数据

void tvsOpenMiniProgram(MiniProgManager manager, MiniProgCallback callback);

参数：

manager：小程序信息管理类

callback：接收小程序跳回数据的回调

1. **LoginInfoManager**

微信登录信息管理类，用户信息基类，被WxInfoManager和QQOpenInfoManager继承。

**accessToken**

String授权AccessToken

**refreshToken**

String 授权RefreshToken

**openID**

String 用户OpenID

**expireTime**

long 刷票过期时间

**tvsID**

String 授权后请求到的tvsID

1. **AuthorizeListener**

微信票据验证监听器

void **onSuccess**(int type, CommOpInfo commOpInfo);

微信接口调用成功回调，用于UI更新

void **onError**(int type, CommOpInfo commOpInfo)

微信接口调用失败回调，用于UI更新

参数

type:调用类型

AUTH\_TYPE：授权票据类型

REFRESH\_TYPE：刷新票据类型

WX\_TVSIDRECV\_TYPE：微信TVSID接收类型

QQOPEN\_TVSIDRECV\_TYPE：QQ TVSID接收类型

TOKENVERIFY\_TYPE：QQ帐号验证类型

USERINFORECV\_TYPE：用户信息获取类型

WX\_VALID\_LOGIN\_TYPE：微信合法登录类型

QQOPEN\_VALID\_LOGIN\_TYPE：QQ合法登录类型

MANAGEACCT\_TYPE：帐号操作类型

REPORTENDSTATE\_TYPE：上报端状态类型

GETPUSHCARDMSG\_TYPE：推送卡片消息类型

1. **EloginPlatform**

登录帐号类型枚举

WX：微信帐号

QQOpen：QQ互联帐号

QQ：QQ帐号（内部）

1. **EloginEnv**

帐号后台环境配置

FORMAL：正式环境

TEST：测试环境

1. **UserInfoManager**

用户信息管理类，包含

idType 帐号类型 0：微信 1：QQ

nickName 昵称

headImgUrl 头像URL地址

sex 性别

unionId 微信UnionId

**getInstance**

获取UserInfoManager实例

static UserInfoManager getInstance()

1. **LocationInfo**

地址信息结构体类，包含

longitube：经度信息

latitube：纬度信息

addr：详细地址

name：地点名称

cabLongitube：打车点经度信息

cabLatitube：打车点纬度信息

cabAddr：打车点详细地址

cabName：打车点地点名称

1. **LocManager**

地址信息管理类

**getInstance**

获取LocManager实例

static LocManager getInstance()

**getLocation**

获取地址信息结构体

LocationInfo getLocation(ELocationType type)

参数

type：地址信息类型

1. **ElocationType**

常用地址类型枚举

HOME：家庭类型

COMPANY：公司类型

1. **BindingListener**

用户信息绑定监听器

void onSuccess(int type, CommOpInfo commOpInfo);

用户信息绑定接口调用成功回调，用于UI更新

void onError(int type, CommOpInfo commOpInfo)

用户信息绑定接口调用失败回调，用于UI更新

参数

type:调用类型

GET\_CAPTCHA\_TYPE：获取短信验证码类型

BIND\_PHONENUMBER\_TYPE：绑定手机号类型

BIND\_LOCATION\_TYPE：绑定常用地址类型

QUERY\_LOCATION\_TYPE：查询常用地址类型

SET\_PUSH\_MAP\_INFOEX\_TYPE：设备绑定类型

DEL\_PUSH\_MAP\_INFO\_TYPE：设备解绑类型

GET\_PUSH\_DEVICE\_INFO\_TYPE：获取绑定设备列表类型

BIND\_GET\_MEMBER\_STATUS\_TYPE：绑定设备领取会员状态查询类型

REPORT\_RELATION\_TYPE：上报QBGuid类型

1. **PushInfoManager**

推送信息管理类

**getInstance**

获取PushInfoManager实例

static PushInfoManager getInstance()

id：推送ID

idExtra：推送额外信息

idType：推送类型

1. **DeviceManager**

设备信息管理类

DeviceManager构造函数

public **DeviceManager**()：用于构造手机端设备信息

public **DeviceManager**(String ip, String data)：用于构造设备端设备信息

参数

ip：设备目标

data：设备信息JSON数据

qua：设备QUA

guid：设备GUID

imei：设备IMEI

licence：设备licence

mac：设备mac地址

qimei：设备qimei

ip：设备ip

os：设备操作系统

deviceOEMUrl：品牌Icon

deviceOEM：品牌名称

deviceType：设备类型

deviceSerial：设备序列号

deviceMark：设备备注

deviceId：设备ID

deviceName：设备名称

manufacturer：设备制造商

packageName：设备主包名

isBinded：是否绑定过

bindOpenId：绑定的帐号OpenID

bindAccountType：绑定的帐号类型

bindTime：绑定时间

enrollTime：注册时间

productId：TVS标准信息设备ID

dsn：TVS标准信息设备序列号

1. **EUserAttrType**

用户中心类型枚举

HOMEPAGE：主页

CPOPERATION：运营页面

INFOSETTING：常用信息设置页面

AGREEMENT：声明页面

PRIVACY：隐私协议页面

FEEDBACK：反馈页面

1. **UserCenterStateListener**

用户中心帐号状态类型回调

void onSuccess(int type, CommOpInfo commOpInfo);

成功回调，用于UI更新

void onError(int type, CommOpInfo commOpInfo);

失败回调，用于UI更新

void onCancel(int type, CommOpInfo commOpInfo);

取消回调，用于UI更新

LOGIN\_TYPE：登录状态

LOGOUT\_TYPE：注销状态

1. **ProductManager**

**getInstance**

获取ProductManager实例

static ProductManager getInstance()

pushDeviceInfos

帐号下绑定设备的列表

aiAcctInfo

返回设备对应的帐号信息

1. **QRStateListener**

LOGIN\_TYPE：登录类型

扫码登录状态回调

void onSuccess(ELoginPlatform platform, int type, CommOpInfo commOpInfo);

成功回调，用于UI更新

void onError(int type, CommOpInfo commOpInfo);

失败回调，用于UI更新

void onCancel(int type, CommOpInfo commOpInfo);

取消回调，用于UI更新

1. **MiniProgManager**

小程序信息管理类

userName：小程序原始id

path：小程序内指定的路径，默认为首页

miniprogramType：小程序版本类型

1. **EMiniProgType**

小程序版本类型枚举

RELEASE：正式版

TEST：开发版

PREVIEW：体验版

1. **MiniProgCallback**

小程序跳回app数据回调

void onReceiveExtMsg(String msg)