

# DeviceWeb模块接入流程

版本	说明	日期
1.0.0	初稿	2019.07.02
1.0.1	修改文档结构区分场景	2019.07.03

## DeviceWeb模块接入流程

### 1 使用场景

#### 1.1 技能授权

#### 1.2 UI模版

### 2 设备要求

### 3 工程配置流程

### 4 代码接入流程

#### 4.1 技能授权

#### 4.2 UI模版

### 5 API与接口说明

#### 5.1 技能授权

##### 5.1.1 DeviceWebController

##### 5.1.2 TVSWebView

##### 5.1.3 TVSWebDevice

#### 5.2 UI模版

##### 5.2.1 OfflineWebManager

## 1 使用场景

针对有屏设备的部分VUI使用场景而定制的一套在Android设备上集成运行的webSDK，目前使用场景有两大类：

### 1.1 技能授权

通过使用SDK拉起指定CP授权URL的页面进行技能授权登录，主要包括以下场景

- QQ音乐登录
- FM技能登录（包括阅文、喜马拉雅等）
- 各类IoT厂商的登录

### 1.2 UI模版

语音交互需要有不同的VUI显示来承载不同的技能，为了能够减少厂商自己实现有屏端的UI模版开发成本，可以通过接入web模版方案，能够快速使用我们已经实现好的标准UI模版，接入厂商只需要按照文档的描述接入，不需要开发相关UI。接入技能模版ui默认没有特殊的定制需求，能够接受技能模版参考设计上的规则，例如自动退出逻辑。

## 2 设备要求

- Android智能有屏设备
- 设备支持WebView
- 使用TVSSDK方式接入

## 3 工程配置流程

在工程中添加websdk的aar及okhttp内部引用

```
repositories {
    flatDir {
        dirs 'libs'
    }
}
...
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation (name: "tvswb-without-core-[VersionCode]-release", ext:
'aar')
    implementation 'com.squareup.okhttp3:okhttp:3.8.1'
}
```

## 4 代码接入流程

具体流程代码可以参考demo

### 4.1 技能授权

- 初始化TVSWebDevice来设置设备信息
- 设置TVSToken，从TVSSDK中获取
- 设置WebUI相关回调事件处理（包括加载状态等）
- 设置叮当内置JS回调
- 打开URL（是否需要缓存、是否带上tvsToken）

### 4.2 UI模版

- 接入方需要将web模版的资源包内置到初始化时设置的目录，web模版资源包为 webtemplate.zip
- 初始化web模版，在activity或者application的onCreate方法中调用OfflineWebManager的init方法

- 根据tvssdk返回的ui数据调用OfflineWebManager的isCanOpenByWebTemplate方法判断该数据是否能用web模版打开
- 如果可以用web模版打开，自己实现装载webview界面的窗口，并启动窗口
- 在实现的窗口中，调用OfflineWebManager的getLoadWebView方法获取webview，并将其绑定到窗口上
- 调用OfflineWebManager的notifyMediaIdAndPlayStatus和notifyTTSPPlayStatus方法，将音频和tts的播放状态传递给web模版
- 根据更新需要，找相应时机调用OfflineWebManager的checkNativeWebPkgUpdate接口检查更新web模版资源

## 5 API与接口说明

### 5.1 技能授权

#### 5.1.1 DeviceWebController

HTML5 WebView模块中TVSWebView对象的控制类，用于控制WebView的内容展示等。

##### setWebDeviceInfo

设置当前设备信息。

```
public void setWebDeviceInfo(TVSWebDevice device)
```

名称	说明
device	设备信息

##### loadURL

打开指定的URL，不带cache

```
public void loadURL(String url, boolean withTVSToken)
```

名称	说明
url	指定url地址
withTVSToken	是否携带TVSToken

##### goForward

前进到下一个网页，返回值表示是否可以前进。

```
public boolean goForward()
```

##### goBack

后退到前一个网页，返回值表示是否能够后退

```
public boolean goBack()
```

## stop

停止加载。

```
public void stop()
```

## reload

刷新（重新加载）当前URL

```
public void reload()
```

## execJS

执行指定的JS函数。

```
public void execJS(String funcName, String funcParam, String h5Id)
```

名称	说明
funcName	函数名
funcParam	函数参数，字符串类型，不需要转义，也不需要引号扩起来
h5Id	该次请求的ID，不需要转义，也不需要引号扩起来

## execJSV2

执行指定的JS函数，增加JS请求返回码。

```
public void execJSV2(String funcName, int retCode, String funcParam, String h5Id)
```

名称	说明
funcName	函数名
retCode	JS请求返回码
funcParam	函数参数，字符串类型，不需要转义，也不需要引号扩起来
h5Id	该次请求的ID，不需要转义，也不需要引号扩起来

## setUIEventListener

设置UI相关事件的回调。

```
public void setUIEventListener(WebController.UIEventListener uiEventListener)
```

名称	说明
uiEventListener	UI相关事件的回调

### setSimpleBusinessEventListener

设置业务相关事件的回调。

```
public void setSimpleBusinessEventListener(WebController.SimpleBusinessEventListener businessEventListener)
```

名称	说明
businessEventListener	业务相关事件的回调

### onDestroy

在容器Activity的onDestroy中需要调用该方法，回收相关资源。

```
public void onDestroy()
```

## 5.1.2 TVSWebView

WebView类，封装了TVS的业务逻辑，可以通过getController()获取控制类进行操作。

### 构造方法

```
public TVSWebView(@NonNull android.content.Context context)
```

```
public TVSWebView(@NonNull android.content.Context context, @Nullable android.util.AttributeSet attrs)
```

```
public TVSWebView(@NonNull android.content.Context context, @Nullable android.util.AttributeSet attrs, int defStyleAttr)
```

### getController

获取用于控制当前WebView内容展示相关接口的控制类实例，返回控制类实例。

```
public WebController getController()
```

## 5.1.3 TVSWebDevice

用来表示设备信息的简单类。

字段列表

名称	说明
productID	后台申请的Product ID。
dsn	设备序列号。
pushID	Push ID。
pushIDExtra	Push ID Extra。
guid	GUID。
deviceId	设备ID。
deviceName	设备名称。
deviceType	设备类型。
deviceSerial	设备系列。
deviceOEM	设备厂商。
deviceOEMURL	设备品牌标志URL。
deviceOEMToken	设备厂商Token（用于PushKit）。
deviceRemark	设备备注。
imei	IMEI。
qua	QUA。
lc	LC。
mac	MAC地址。
qimei	QIMEI。
enrollTime	注册时间。
bindTime	绑定时间。
extra	扩展信息。
businessExtra	业务扩展信息。

构造函数

```
public TVSDevice()
```

toString

```
public String toString()
```

## 5.2 UI模版

### 5.2.1 OfflineWebManager

UI模版的管理类，详见demo

#### init

初始化UI模版

```
public void init(Context inContext, String sLocalPkgPath, OfflineWebCallback  
inCallback)
```

名称	说明
inContext	Context上下文
sLocalPkgPath	指定的离线资源存放位置
inCallback	注册的回调接口，返回初始化是否成功以及需要厂商实现的接口

#### isCanOpenByWebTemplate

判断是否能够用Web化模版打开UI

```
public boolean isCanOpenByWebTemplate(String jsonData, String jsonSemantic,  
boolean reportEnd, String dialogRequestId)
```

名称	说明
jsonData	服务返回的v3 json UI相关数据
jsonSemantic	服务返回的semantic json数据
reportEnd	模版界面退出时是否要调用上报，组合技能使用逻辑
dialogRequestId	界面的会话id

#### notifyMediaIdAndPlayStatus

同步本地的媒体正在播放的媒体的id和播放状态信息

```
public void notifyMediaIdAndPlayStatus(String sMediaID, String sParentId,  
String sStatus)
```

名称	说明
sMediaID	媒体的id服务数据返回的id
sParentId	媒体所属的父id(例如属于哪个专辑)
sStatus	媒体的播放状态

## notifyTTSPlayStatus

同步本地TTS播放的状态

```
public void notifyTTSPlayStatus(String dialogRequestId, String sStatus)
```

名称	说明
dialogRequestId	TTS播报的唯一id（用来确认跟打开的页面是否是同一轮会话）
sStatus	TTS状态

## checkNativeWebPkgUpdate

检查离线web模版是否有更新的接口

```
public void checkNativeWebPkgUpdate()
```

## getLoadWebView

获取加载url的webview对象

```
public TMSWebView getLoadWebView()
```

## destroyWebView

清除webview的加载历史记录

名称	说明
inWebView	释放临时加载使用的webview对象

```
public void destroyWebView(TMSWebView inWebView)
```