

# ECWebView SDK 集成文档

## 集成

使用 gradle 集成方式:

```
compile 'com.fifthera:ecwebview:0.0.9'
```

如果您的项目中, Gradle Plugin 是 3.0 及以上, 可以使用:

```
implementation 'com.fifthera:ecwebview:0.0.9'
```

ECWebView 内部引用了 `appcompat-v7:28.0.0`, 如果你不想使用这个版本, 可以使用 `exclude` 将它排除掉, 再自行引入你选择的版本。示例如下:

```
implementation ('com.fifthera:ecwebview:0.0.9', {  
    exclude group: 'com.android.support'  
})
```

或者在项目的build.gradle中通过如下方式引用.aar 包

```
implementation(name: 'lib_webview-release', ext: 'aar')
```

## 基本使用

使用 ECWebView 就像使用普通的 WebView 控件一样简单。您可以在 xml 布局文件中定义。

```
<com.fifthera.ecwebview.ECWebView  
    android:id="@+id/webview"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" />
```

然后在 Activity 或 Fragment 中对 ECWebView 进行初始化, 以及一些必要的设置, 典型的使用代码如下:

```
// 获取 ECWebView 对象  
ECWebView webView = findViewById(R.id.webview);  
// 创建 JSApi 对象  
JSApi api = new JSApi(this);  
// 将 ECWebView 对象和 JSApi 对象进行绑定  
webView.addJavaScriptObject(api);  
// 加载 URL  
webView.loadUrl(url);
```

```

// 处理 JSApi 的返回结果
api.setOnApiResponseListener(new OnApiResponseListener() {
    @Override
    public void fail(int errorCode) {
        if (errorCode == ErrorCode.TOKEN_FAIL) {
            Toast.makeText(TestActivity.this, "token失效了", Toast.LENGTH_SHORT).show();
        }
        if (errorCode == ErrorCode.COMPOSITE_FAIL) {
            Toast.makeText(TestActivity.this, "图片合成失败", Toast.LENGTH_SHORT).show();
        }
    }

    /**
     * 分享图片到相应的渠道
     * @param bitmap 需要分享出去的图片
     * @param s 需要复制的文案
     * @param shareType 分享渠道: 微信好友 - 0x01 , 微信朋友圈 - 0x02
     */
    @Override
    public void getShareImageBitmap(Bitmap bitmap, String s, int shareType) {
        // 处理合成图片
    }
});

```

ECWebView 已经封装了 `WebChromeClient` 的实现和 `WebViewClient` 的实现，因此不应该再为 ECWebView 设置 `WebChromeClient` 或 `WebViewClient` 了。但是如果有特殊需求，需要处理 `WebChromeClient` 或 `WebViewClient` 的相关回调，我们提供了 `ECWebViewClient` 和 `ECWebChromeClient` 作为代替。

比如：

```

webView.setOnWebViewClientListener(new ECWebViewClient() {
    @Override
    public boolean shouldOverrideUrlLoading(Webview view, String url) {
        // TODO 处理相关业务
        return super.shouldOverrideUrlLoading(view, url);
    }
});

webView.setOnWebChromeClientListener(new ECWebChromeClient() {
    @Override
    public void onReceivedTitle(Webview view, String title) {
        // TODO 处理相关业务
    }
});

```

## 方法说明

### JSApi 返回方法

JSApi 封装了 ECWebView 和 H5 之间的通信，通信的结果由 OnApiResponseListener 监听器返回，该监听器返回的方法如下：

方法签名	含义
fail(int code)	api 调用失败后的回调，code 为错误码。
getShareImgPath(Bitmap bitmap, int shareType)	分享图片 shareType 表示分享类型，TYPE_WECHAT 表示微信，TYPE_WECHAT_MOMENT 表示朋友圈
goBack()	处理H5页面的返回事件

**错误码**

在 JSApi 里面的 fail() 方法中返回的错误码详情如下：

错误码	含义
TOKEN_FAIL	token 失效，需要重新授权。
COMPOSITE_FAIL	分享图片合成失败。

**ECWebView 调用方法**

ECWebView 里面调用的方法基本上都是用来告知 H5 相关信息的，目前有如下的方法：

方法签名	含义
loadUrl(String url)	加载一个 URL
addJavaScriptObject(JSApi jsapi)	绑定 JSApi
showTitle(boolean isShowTitle)	是否显示 title(默认不显示)
refresh()	刷新 H5 界面
setOnWebViewClientListener(ECWebViewClient client)	设置 ECWebViewClient
setOnWebChromeClientListener(ECWebChromeClient client)	设置 ECWebChromeClient