# iOS 端内WebView

目前存在的问题：（已解决）

1. ~~电影票url 加载 时好时坏；京东url 加载ok；他们都会有 -999的error 提示~~
2. ~~加载墨迹商城有几率出现，图片未绘制完毕，等待时间可能会有4s 左右。~~

**为保证质量有2个降级保护**

1. 通过URL 配置 webType=1，2，0 来配置选择那个 webview （优先级最高）
2. 通过服务器配置 0，1，2

**需要服务器人员协同：**

需要：为保证重构后的新控件不影响用户的体验。做降级方案。增加服务器配置项。可根据服务器配置；

需支持系统版本返回比如 10.0+ 系统 返回 webType=2； 返回1 或者 0；

* webType = 0        iOS8+ 使用WKWebView， iOS7 使用 UIWebview
* webType = 1       指定使用 UIWebview
* webType = 2       指定使用 WKWebView （客户端在iOS7中，做容错保护，仅有 UIWebView）

默认使用客户端默认使用 UIWebView。服务器下发使用 0， 或者 ，1 或者 2

目标：

降低web加载的内存；依赖于网页内容。服务器可配置选择web view类型。

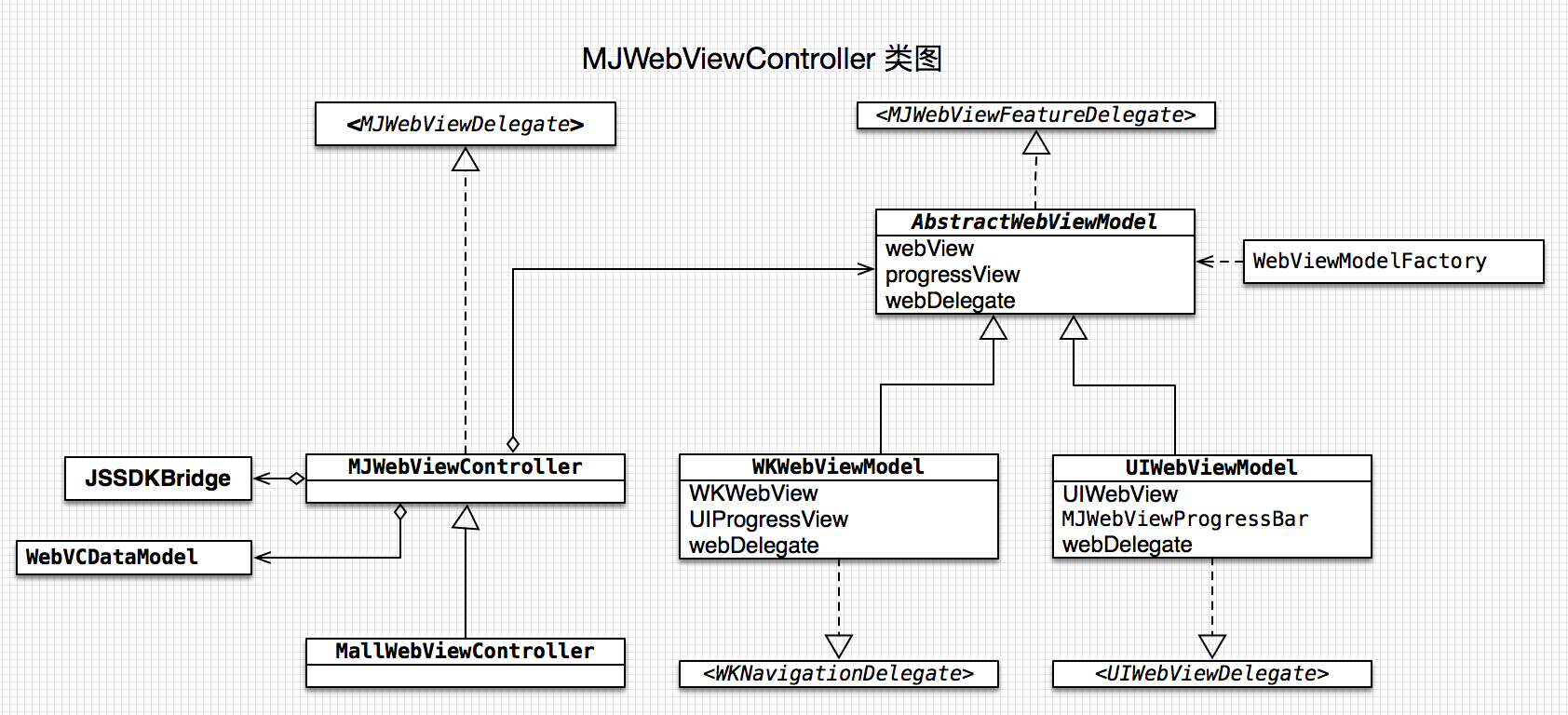
1、背景

由于UIWebView 加载慢，占内存，相应慢。ios8 之后，苹果推出WKWebView 弥补了UIWebView的缺点。占内存小，响应速度快。同时考虑未来web页面加载图片，多媒体等的增多，以及ios8+系统的普及，故做UIWebView，WKWebView 的适配开发，已适应未来的需求变化。

重构 MJWebViewController，MallWebViewController。

2、重构的技术说明

a. 重构后的类图



MJWebViewDelegate 类似之前的 UIWebViewDelegate 的代理功能，适配 WKNavigationDelegate，UIWebviewDelegate。

MJWebViewFeatureDelegate 类似访问WebView控件的 一些特性，如 loadRequest，canGoBack，reload , native

WebViewModeFactory 创建 WKWebViewModel， UIWebViewModel

JSSDKBridge 支持适配 WKWebView， UIWebView 的 js native 相互调用

WebVCDaaModel 数据model 以及 部分的逻辑放在 webvcdataModel 中

以上详情看具体的类。

**JSSDKBridge** 提供支持JSSDK 的功能。 js 与 native 的相互调用，兼容支持 UIWebView，WKWebView。

**WebVCDaaModel** 具体功能包括：组合分享数据；判断某url 是否允许打开。判断展示分享按钮逻辑。

**MJWebViewDelegate** 主要功能提供WebViewModel 的回调。有以下功能：

* 判断是否允许打开url；
* url 开始发起请求；
* url请求成功；
* url 请求失败；

**MJWebViewFeatureDelegate**web 特性有：

* loadURL
* loadRequest:
* canGoBack
* goBack
* reload
* URL
* excuteJSFunction: 执行js

**WebViewModeFactory** 创建Model ，

* UIWebViewModel 创建包含 UIWebview 的model
* WKWebViewModel 创建包含WKWebView 的model

**AbstractWebViewModel** 对webview 的一个封装，支持一些 web的特性。

* loadRequest:
* canGoBack
* goBack
* reload
* URL
* excuteJSFunction://执行 js 方法，同步执行
* webView
* progressView
* showProgressView

**AbstractWebViewModel+ShareInfo** 是分享扩展

**AbstractWebViewModel+HandleMallCookie** 处理商城Cookies

**WebInfoObject** 传递数据类

3、调用说明 参考  MJWebViewController 的使用

//创建Model ，Auto 是根据系统版本来的 ios8+ WKWebView， ios7 UIWebView  
//可以指定 WebViewModelCreateTypeUIWebView 创建 UIWebView  
//WebViewModelCreateTypeWKWebView 创建 WKWebView  
\_webModel = [WebViewModelFactory factoryWithType:WebViewModelCreateTypeAuto   
showProgressView:YES viewController:self];  
\_webModel.webDelegate = self;  
  
// 将 webview 展示  
    UIView \*webView = self.webModel.webView;  
    [self.view addSubview:webView];  
    webView.frame = CGRectMake(0, 0, SCREEN\_WIDTH, self.view.height);  
      
//Class MJWebViewController 设置代理  
// 是否允许发起请求  
- (MJRequestPolicy)shouldAllowRequestWithInfo:(WebInfoObject \*)info  
//开始请求  
- (void)webRequestStartWithInfo:(WebInfoObject \*)info  
//请求成功  
- (void)webRequestFinishLoadWithInfo:(WebInfoObject \*)info  
//请求失败  
- (void)webRequestFailLoadWithInfo:(WebInfoObject \*)info