淘北研彩票业务web无线开发实战

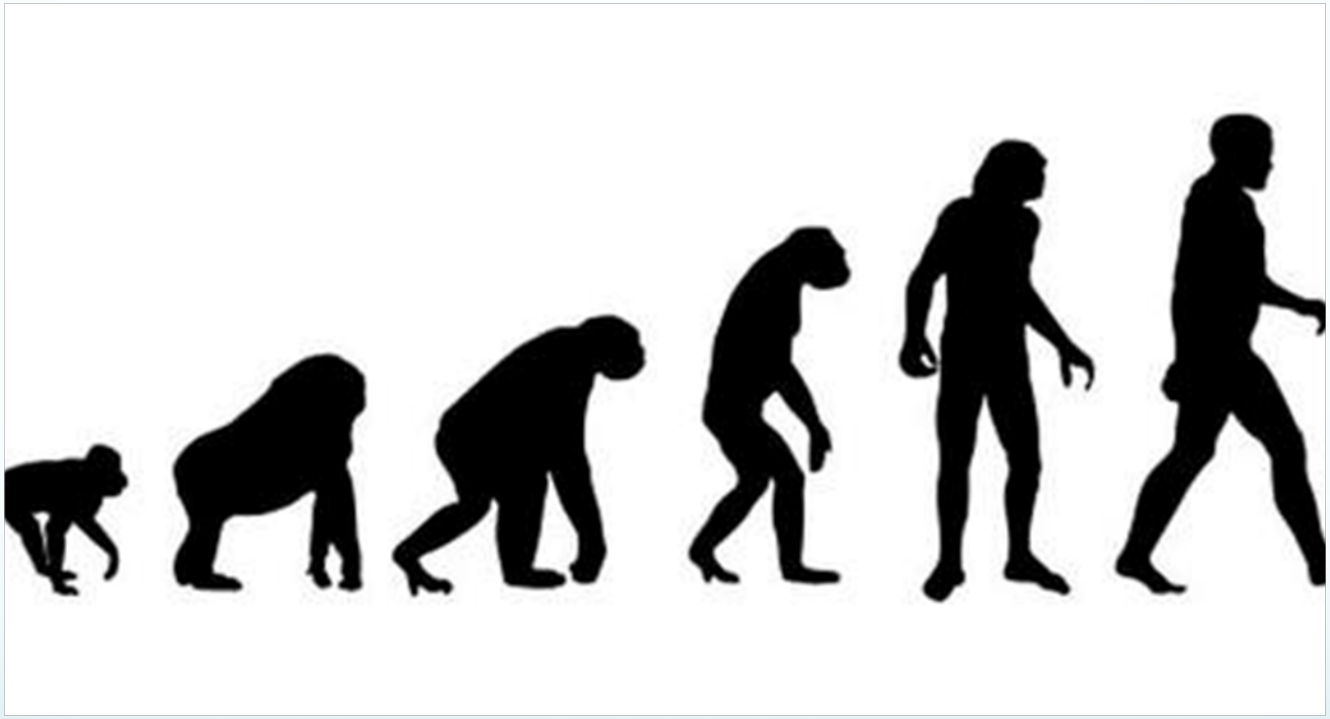
彩票订单系统项目总结

函谷

AGENDA

- 彩票H5订单的历史变迁
- 开始移动开发
- 前事不忘，后事之师

Part 1




iPod
21:20
我的订单
全部订单
中奖订单
双色球
第11138期
参与合买
20000.00元
300000000.00元
2012-02-27 17:25
已发奖
双色球
第11137期
代购
80.00元
5000000.00元
2012-02-27 17:25
已发奖
双色球
第11135期
代购
2.00元
立即付款
2012-02-27 17:25
足球单场
第11134期
参与合买
刚刚正好七..
30.00元
2012-02-27 17:25
已付款
购彩大厅
幸运选号
开奖公告
我的投注
更多


iPod
21:20
返回
订单详情
继续投注
双色球 第11138期
订单详情
订单号 623196500
下单时间 2011-12-01 15:23
中奖金额 100.00元
投注金额 2480.00元
订单状态 下单成功
投注内容
标识码 200100000101FFF111410057D0
投注内容 03 07 13 28 30 33 | 08
倍数 15倍
注数 150注
开奖号码
06 07 15 26 27 30 09
购彩大厅
幸运选号
开奖公告
我的投注
更多

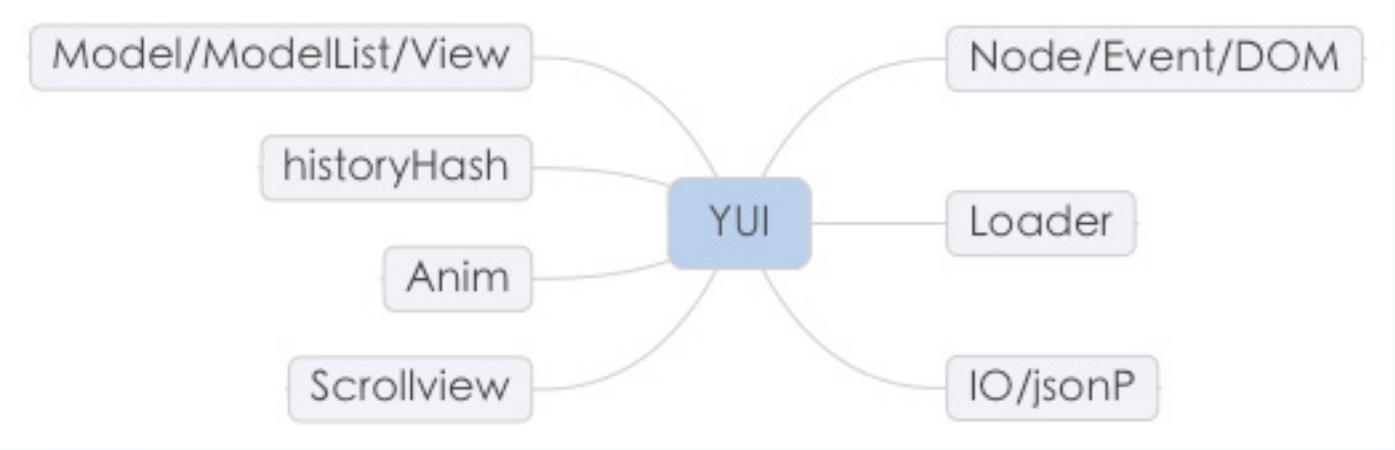
phone


iPad
9:49
4%
返回
我的订单
全部订单
中奖订单
第00038666期 参与合买
20000.00元 300000000.00元
双色球 2012-02-27 17:25 已发奖
第11137期 代购
80.00元 5000000.00元
双色球 2012-02-27 17:25 已发奖
第11136期 发起合买
50.00元
大乐透 2012-02-27 17:25 未中奖
第11135期 代购
2.00元 立即付款
双色球 2012-02-27 17:25
第11134期 参与合买 刚刚正好..
30.00元
足球单场 2012-02-27 17:25 已付款
第11133期 参与合买 必中之仙
600.00元
竞彩足球 2012-02-27 17:25 参与失败(合买未满员)
第11132期 参与合买 枫榜1688
90.00元
竞彩足球 2012-02-27 17:25 参与失败(合买已撤单)
订单详情
订单号 644187832
下单时间 2011-12-19 11:43
玩法 胜平负
投注金额 2.00元
中奖金额 1980.00元
订单状态 下单成功
选号方案
场次 主队 对阵 客队 投注 胆 比分 让球(出票)
周一005 塞维利亚 VS 西布罗姆维奇 胜(1.52) × 2:1 0
周一005 里斯本竞技 VS 比兰尼塞斯 胜(1.52) √ 3:2 -1
选号详情
彩票标识码: 2019796779727073****
序号 场次和出票赔率 过关方式 中奖金额 注数 倍数 投注金额
01 周五020(胜:1.52)x周六056(胜:1.42) 2串1 8000.00元 1 10 20.00元

pad

彩票H5订单 V1.0

- 2012年4月初 — 2012年5月底
- 基于YUI 3.3.0的单页应用
- 兼容pad和phone，阀值为640px
- 应用场景：
  - iPhone客户端 V1.4.0
  - iPhone客户端 V1.4.1
  - iPad 竞彩足球

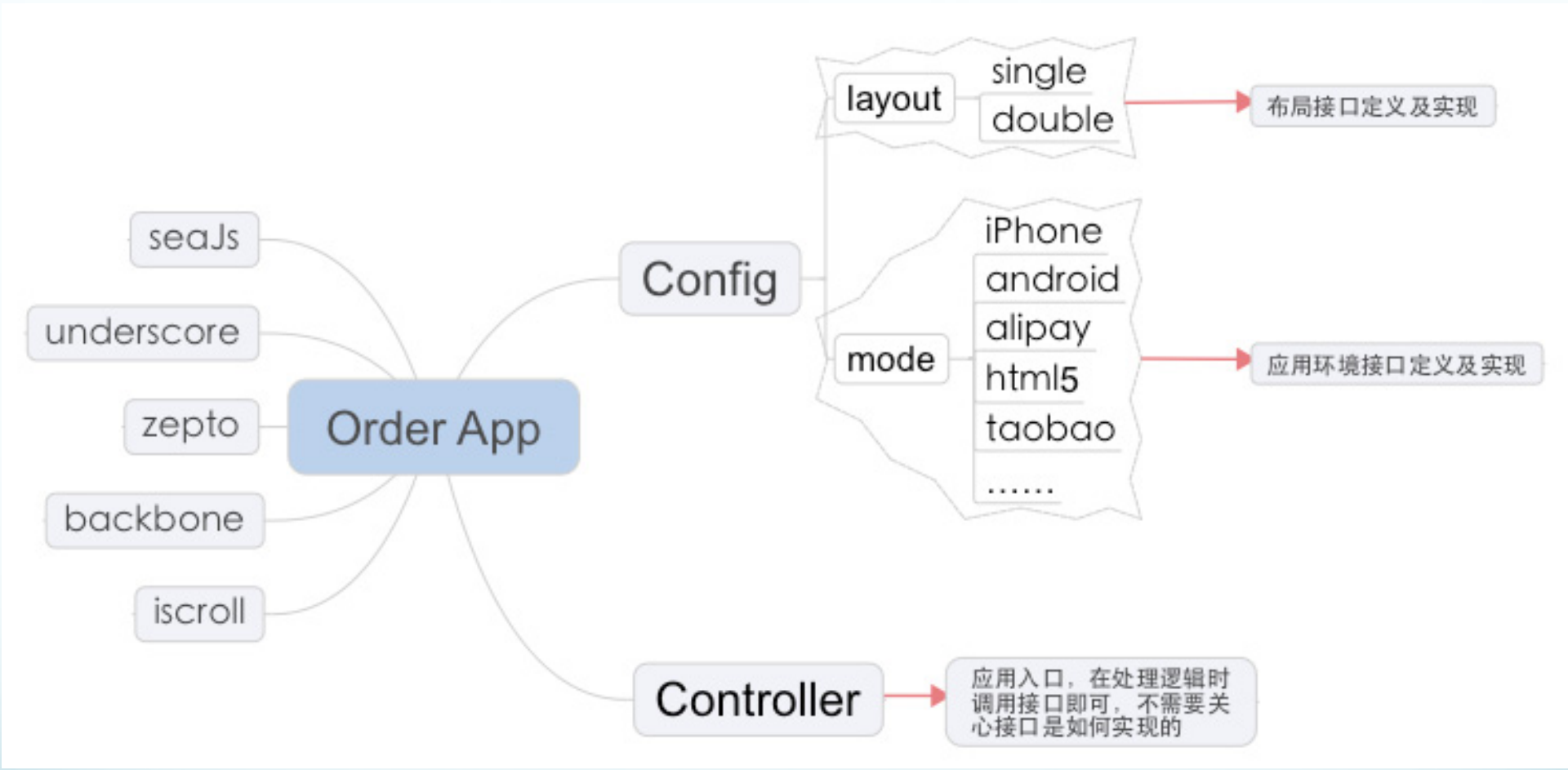

Model/ModelList/View
historyHash
Anim
Scrollview
YUI
Node/Event/DOM
Loader
IO/jsonP

发现的问题

- YUI 相对于移动端太过于臃肿和庞大
- 项目初期代码设计不够完善，导致逻辑混乱
  - 路由设计（一套路由，完全依赖响应式）
  - 应用场景设计（内部判断，难以扩展）
- 速度和性能
- 缺乏完整的开发和测试环境

彩票H5订单 V2.0

- 2012年6月
- 去除对YUI的依赖，改用seaJS、backbone、zepto...
- 代码重构，路由拆分
- 应用场景：
  - iPhone客户端 V1.4.2
  - iPhone客户端 V1.5.0
  - 支付宝客户端
  - 淘宝主站iPhone客户端
  - Wap彩票高端版


seaJs
underscore
zepto
backbone
iscroll
Order App
Config
layout
single
double
布局接口定义及实现
mode
iPhone
android
alipay
html5
taobao
.....
应用环境接口定义及实现
Controller
应用入口，在处理逻辑时调用接口即可，不需要关心接口是如何实现的

改进的问题

- 灵活，方便扩展
- 各个应用场景相隔离
- 摸索出一套开发、发布的流程
  - 使用前端开发机作为测试机
  - 尽早将页面发布上线，提前暴露问题
  - 对于Application Cache的使用更加熟练
  - 形成Change Log，规范发布流程

依然存在的问题

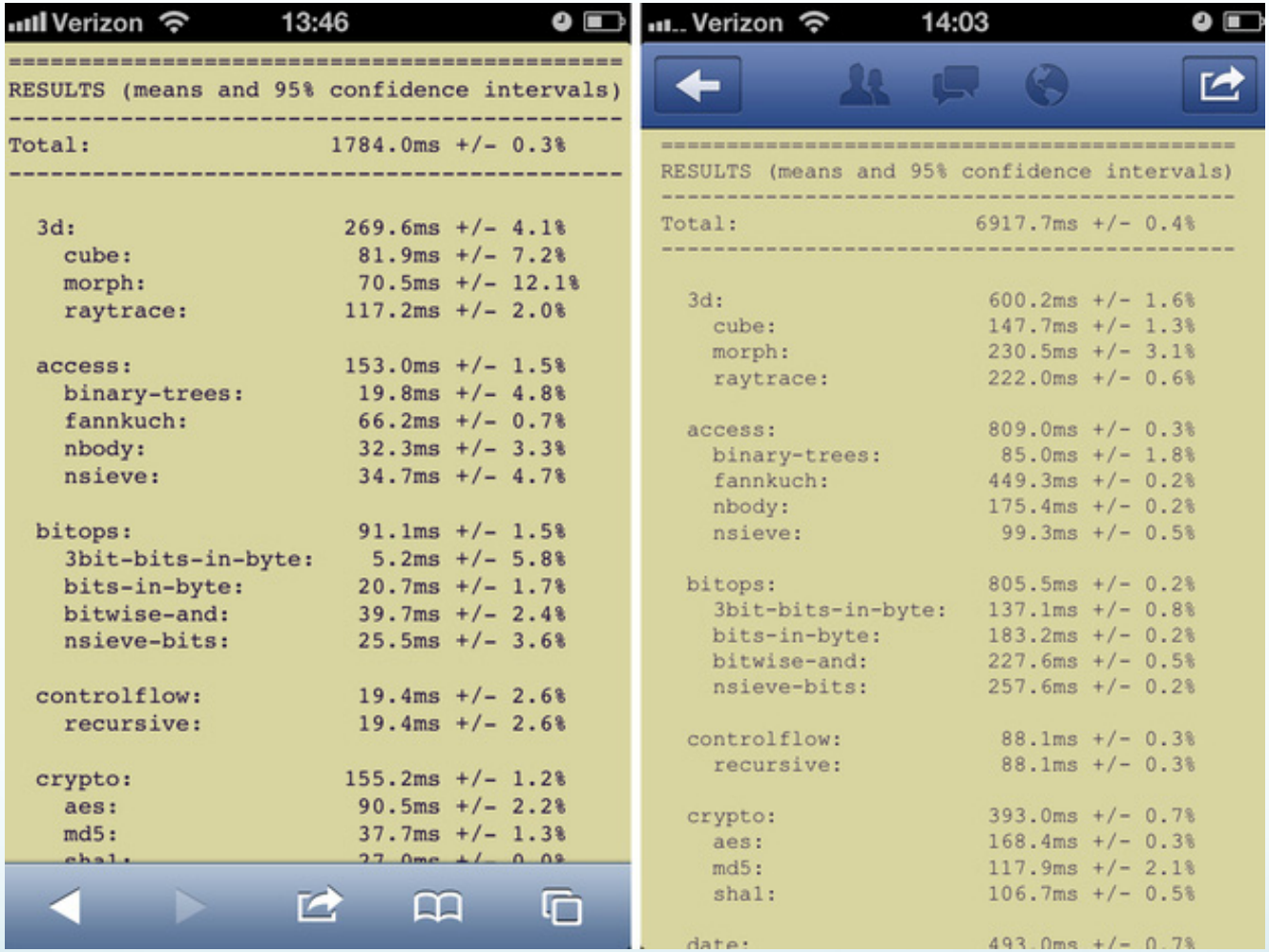
- 尽管速度、性能略有提升，但仍是一个难题
- 部分Android手机由于硬件问题不支持带参数URL
- 根据屏幕尺寸的适配不再重要
  - 主要应用场景为客户端应用内部，而不是直接访问的网站

彩票H5订单 V3.0

- 2012年10月
- 本地化，打包到客户端内部；业务包更新不依赖客户端
- 多页面应用，使用Local Storage进行页面间通信
- 个性化定制，不再支持pad和phone的兼容
- 应用场景：
  - iPhone客户端 V1.6.0
  - Android客户端 V1.6.0
  - 支付宝客户端
  - 淘宝主站iPhone客户端
  - 淘宝主站Android客户端

总结

- 优势
  - 快速响应用户需求
  - 复用性
- 劣势
  - 速度、性能
  - 体验



Mobile Safari/UIWebView JavaScript效率对比

Part 2

相对于桌面端的开发

- 一些特殊的设置
- 一些不同的选择
- 整体方向大同小异
- 性能问题更加严峻
- 兼容性问题更加棘手

Meta设置 —— viewport

- 由于客户端一般不需要考虑横竖屏切换的问题，因此推荐的配置是
  - width=device-width:以窗口大小为基准
  - initial-scale=1:页面加载时为原始尺寸
  - maximum-scale=1:无论如何缩放比例均为1
  - user-scalable=0:禁止页面缩放，更重要的作用是在Android下激活 position:fixed 属性

Meta设置 —— format-detection

- 屏蔽移动设备对HTML代码中格式化信息的探测，例如电话号码，邮箱地址等。通常为了让页面正常显示，需要将自动检测关闭
  - telephone=no
  - email=no
  - 在iOS 5.0以下的系统中，webview中的format-detection设置会部分失效，最保险的方案是在客户端代码中进行设置

Meta设置 —— 其他

- apple-mobile-web-app-capable：允许web app以全屏方式打开
- apple-touch-startup-image：程序启动画面，必须符合iOS宽高标准
- apple-touch-icon：桌面图标的图片，要针对iPhone和iPad设置两个值，分别为57*57和72*72

- 以上设置仅适用于iOS，Android系统无法从桌面启用web app

HTML5

- 语义化标签：
  - <header>
  - <section>
  - <footer>
  - <article>
  - <nav>
  - ……
- Web Forms —— input
  - 调起英文键盘: text
  - 调起数字键盘：number，iOS超过3位数字会自动添加分隔符
  - 调起纯数字键盘（常用）：tel
  - 调出系统日期选择：date（仅限iOS）

HTML5

- Canvas
- Application Cache
- Local Storage
- Post Message
- Geolocation
- Touch Event
- Gesture: 手势动作，仅限iOS
- Devicemotion：设备运动，仅限iOS
- ……

http://mobilehtml5.org/

CSS3

- background:-webkit-gradient
- transition/transform/animation：android下慎用，且3.0以下不支持3D
- border-radius
- box-shadow：android需要使用前缀-webkit，并且在position：fixed的元素上使用会产生位置偏移
- text-shadow
- box-flex：注意flex-basis的作用
- background-size：设计师一般都是按照960*640的尺寸进行设计的，于是background-size就非常非常有用了

JS library

- 轻量级，高性能
  - seaJS：模块加载器，遵循CMD规范
  - zepto：与jQuery类似的语法，常用触屏事件的封装
  - iScroll：为移动终端提供模拟滚动的解决方案
  - juicer：高效、轻量的JavaScript模板引擎
  - backbone：一个成熟而强大的MVC类库，重度依赖underscore，DOM处理依赖于jQuery/Zepto；如果数据结构不是特别复杂的话不推荐使用
  - underscore：一个完备的工具集
- 具体情况具体分析，根据应用场景选择合适的library

navigator

- userAgent
- onLine：判断是否有网络连接
- standalone：iOS中判断是否从桌面进入web app
- geolocation：地理位置

Event

- touchstart (touches/targetTouches/changedTouches)
- touchmove
- touchend
- touchcancel
- orientationchange：iOS，Android下用window.resize()
- devicemotion：iOS
- gesturestart：iOS
- gesturechange：iOS
- gestureend：iOS

图片

- 尽可能少的使用图片：CSS3
- 尽量避免使用gif图片：CSS3
- icon/图标使用高清大图（PNG32），以适应iOS的高精度视网膜屏

调试

- 在PC chrome下进行开发和调试
- 强大的safari 6
  - 请升级你的iPhone/iPad到iOS 6.0+
  - 与MAC连接后，打开safari，在偏好设置-高级中勾选在菜单栏显示“开发”菜单 （safari 6 没有Windows版）
  - 你就可以在开发菜单下发现你的设备，然后就可以和在PC端一样进行debug，查看网络请求……
- Android请善用alert和console，jsconsole可以帮你查看页面上的日志信息
- 有时候，客户端开发同学的日志能帮你起到事半功倍的效果

Part 3


保护南京难民的“活菩萨”
“可以宽恕，但不可以忘却”
—— 约翰·拉贝
“Forgivable, but Unforgettable”
— John.Rabe
「許すことはできても、忘れてはいけない」
— ジョン.ラーベ

Android Webview 设置

- 默认不可使用网络，禁用JavaScript和local Storage

  - 设置许可

  ```
  <uses-permission android:name="android.permission.INTERNET" />
  ```

  - 启用JavaScript

  ```
  Webview.getSettings().setJavaScriptEnabled(true);
  ```

  - 启用local Storage

  ```
  Webview.getSettings().setJavaScriptEnabled(true);
  ```

Android Webview 设置

- 在webview中响应链接

```
mWebView.setWebViewClient(new WebViewClient(){
        public boolean shouldOverrideUrlLoading(WebView view, String url) {
                view.loadUrl(url);
                return true;
        }
});
```

- 处理默认的后退事件

```
public boolean onKeyDown(int keyCode, KeyEvent event) {
        if ((keyCode == KeyEvent.KEYCODE_BACK) && mWebView.canGoBack()) {
                mWebView.goBack();
                return true;
        }
        return super.onKeyDown(keyCode, event);
}
```

移动端区域滚动

- 现状
  - iframe：移动终端无法滚动
  - position:fixed：iOS 5 + 、Android 2.2 +
  - overflow:scroll：iOS 5 +
  - iScroll：多节点下Android性能影响较大
- 现有方案
  - 滚动开始时隐藏，结束后计算位置并显示。例如jQuery Moblie 1.0
  - 使用position:fixed，低版本降级处理，显示为static。例如jQuery Moblie 1.1 +
  - 禁止原生滚动事件，使用JS计算滚动。例如iScroll、Sencha touch
  - 多webview，需要app支持，并提供页面间通信方法。

移动端区域滚动

- 推荐解决方案
  - 页面小部分区域滚动
    - 一般情况下节点较少，可以使用iScroll
  - 需要对滚动进行编程，如下拉刷新等
    - 说服产品放弃下拉刷新，改用按键触发
    - 使用iScroll（或流火同学改进版本）
  - mobile标准三栏或两栏布局
    - Android下直接使用position:fixed
    - iOS下使用iScroll
    - 兼顾性能和体验

iScroll 使用推荐

- 使用iScroll-lite
- 如果可以，请设置hScrollbar:false, vScrollBar:false
- 设置useTransition:true
- iScroll外部的元素(尤其是form元素)不要定义在iframe的高度之内

超链接打开应用程序或应用商店

- 淘宝彩票
  - “taobaocaipiao”为协议名，在客户端内部定义
  - iOS：在iOS程序的Info.plist中添加一个URL types节点九可以很方便的注册URL Scheme
  - Android：在androidManifest.xml文件的启动action中定义scheme <data android:scheme=“...” />
- 淘宝彩票
- 淘宝彩票

Native 与 Web 的通信 -- Android

```
//将JAVA对象绑定到JavaScript中
mWebView.addJavascriptInterface(new JsToJava(), 'stub');
//在JavaScript中调用Java方法
window.stub();


function invokedByJava(data){
    //do something
}
//在Java中调用JavaScript方法
public void onClick(View v) {
    mWebView.loadUrl("javascript:invokedByJava('java_data')");
}
//打开webview，调用页面
mWebView.loadUrl("file:///xxx.html");
```

Native 与 Web 的通信 -- iOS

```
//Objective-C
- (BOOL)webView:(UIWebView *)webView
    shouldStartLoadWithRequest:(NSURLRequest *)request
    navigationType:(UIWebViewNavigationType)navigationType {
        NSURL * url = [request URL];
        if ([[url scheme] isEqualToString:@"gap"]) {
            //在这里做js调native的事
            //...
            //完成之后回调js
            //[webView stringByEvaluatingJavaScriptFromString:
                    @"alert('done')"];
            return NO;
        }
        return YES;
}
```

Native 与 Web 的通信 -- iOS

```
//通知iPhone UIWebView加载url对应的资源，url格式为gap:something
function loadURL(url) {
    var iFrame;
    iFrame = document.createElement('iframe');
    iFrame.setAttribute('src', url);
    iFrame.setAttribute('style', 'display:none;');
    iFrame.setAttribute('height', '0px');
    iFrame.setAttribute('width', '0px');
    iFrame.setAttribute('frameborder', '0');
    document.body.appendChild(iFrame);
    //发起请求后将其从DOM移除
    iFrame.parentNode.removeChild(iFrame);
    iFrame = null;
};
```

Application Cache

- 定义了manifest文件后，所有网络请求都会经过该配置，如果没有设置缓存，也没有在白名单里配置，则访问不到资源文件：将NETWORK设置为*
- 如果manifest文件或者其内部列举的某一个文件不能正常下载，整个更新过程将视为失败，浏览器继续全部使用老的缓存
- 引用manifest的html必须与manifest文件同源，在同一个域下
- 在manifest中使用的相对路径，相对参照物为manifest文件
- FALLBACK中的资源必须和manifest文件同源
- 当一个资源被缓存后，该浏览器直接请求这个绝对路径也会访问缓存中的资源
- 站点中的其他页面即使没有设置manifest属性，请求的资源如果在缓存中也从缓存中访问

Application Cache

- 在使用手动更新swapCache()之后并不会马上载入新的资源文件，而是在下次加载时才载入，可以通过reload使页面重新加载
- Cache目录下的文件可以设置Etag/Last-Modified等，但不能有hash或query值
- 如果在TMS中发布manifest文件，推荐使用Firefox，Chrome可能会使manifest文件失效
- TMS设置页面信息-高级，请去除勾选php页面是否头部自动附加回车（正常情况由于安全方面原因请勾选此项，防止utf7 XSS）
- 一般情况下只需要修改版本号即可更新Application Cache，但有时也会出现意外，可尝试修改Cache中的某个文件名

诡异的Android input

- 在Android设备上，当一个input获得焦点时，我们所看到的并不是原生的Webkit input，而是浏览器创建的一个新的text组件，它会尝试覆盖在原来的input上。因此，在很多时候，我们会发现input的样式错误，或者位置错误，甚至会出现随着输入内容上下乱跑的情况。
- 这种情况通常出现的条件有：
  - Android 2.2/2.3
  - input的祖先元素设置了transform:translate3d()或position:fixed
- 因此，我们应当尽量避免将form表单放在设置了fixed的元素内部。

其他

- 当有些时候对DOM的更新操作不生效时，请尝试使用延时器（Android 4+）
- 由于移动网络下载速度的限制，请不要在HEAD中加载过多的网络请求，防止长时间白屏
- 移动终端在手指按下时会同时触发touchstart和click事件，但click事件会延迟0.3s执行，为此，zepto提供了tap和longtap事件

Reference

- http://stackoverflow.com/questions/10542525/how-can-i-prevent-wild-scrolling-when-a-fixed-position-text-input-form-field-gai
- http://code.google.com/p/android/issues/detail?id=14295
- https://github.com/scottjehl/Device-Bugs/issues/3
- https://developers.google.com/mobile/articles/webapp_fixed_ui?hl=zh-CN

END