uml.org.cn 文章 文库 视频 讲座 课程 咨询 认证

QQ群：312010267（ Android开发 ）

火龙果讲堂：
- 一线专家
- 案例回顾
- 经验分享

如何合理设计Hybrid App

About Hybrid App，Ionic Framework，phoneGap，HTML5

西涛　移动互联网资深架构师

Agenda

- Why Hybrid App?
- Is HTML5 Ready?
- Practice of Hybrid App Architect
- What Ionic Framework do?
- PhoneGap/Cordova
- HTML5 in mobile device
- Useful Tools
  - Weinre，ionic，xcode，gulp

Conception

- App：应用程序
  - 完成某项或者几项任务
  - 满足某些需求
- Web：构建于Web技术之上
  - HTML/CSS/JS
  - 服务端
- Web Application

Why Hybrid App？

- Native App， Web App or Hybrid App?

	Device Access	Speed	Development Cost	App Store	Approval Process
Native	Full	Very Fast	Expensive	Available	Mandatory
Hybrid	Full	Native Speed as Necessary	Reasonable	Available	Low Overhead
Web	Partial	Fast	Reasonable	Not Available	None

Why Hybrid App？

- 不是一个纯粹的技术问题，最重要的因素是产品定位和发展策略
- 如果希望尽快发布、跨平台、能快速响应可以预见的迭代，那么混合架构就很值得考虑
- 如果有足够的开发人员覆盖各平台、产品设计成熟度高、产品周期上可接受相对较长的开发时间，那么原生显然是更好的选择
- 如果app里面要实现的功能，很多都没法用 html做，必须用原生代码。例如即时通讯应用，图片处理应用等

HTML5 Ready？

W3C规范在成为正式标准前通常需要历经5个阶段

移动平台主流浏览器都较好的支持HTML5，性能和速度还有待提高

Calculation of support for currently selected criteria

	iOS Safari	Android Browser	Opera Mobile	Blackberry Browser	Chrome for Android	Firefox for Android
		2.1: 24%				
		2.2: 33%				
	3.2: 24%	2.3: 37%				
	4.0-4.1: 33%	3.0: 51%	10.0: 31%			
	4.2-4.3: 38%	4.0: 57%	11.5: 59%			
	5.0-5.1: 63%	4.1: 57%	12.0: 67%	7.0: 59%		
Current	6.0: 66%	4.2: 60%	12.1: 73%	10.0: 83%	25.0: 77%	19.0: 64%

HTML5 Ready？

- 老牌的Sencha Touch、jQuery Mobile、Dojo Mobile
- Cocos2d-html5 EaselJS、Construct2、Impact、GameMaker、limeJS、Crafty等游戏开发引擎或工具
- GroundworkCSS、Skeleton、Gumby、Wirefy、Base等响应式设计框架
- 代表性的HTML5应用包括YouTube、新浪微博、携程、淘宝等

Hybrid App架构设计原则

- 使用phonegap提供跨平台的设备资源访问能力
- 服务器端基于REST + jsonp提供数据
- 大部分WebApp文件（html，css，js）打包在App内，安装到设备上
- Html和js完全分离，html页面模板化，通过js模板引擎渲染服务器端json数据到html模板页面
- Html页面优先采用响应式设计适配不同的屏幕分辨率，特殊页面分别实现多个html页面
- 使用单html页面，html页面模板对应不同的功能模块，降低单个文件的复杂度，通过基于#锚点的Url路由跳转页面

服务器接口设计

- 基于HTTP+json的REST接口，扩展支持jsonp（只需要进行简单改造）
- 隔离APP端和接口，各自独立开发，前期客户端可以使用 mock接口测试数据开发
- 接口可以兼容ios和android，并兼容native版本的 客户端实现
  - 当某些WebApp页面出现难以解决的性能问题时，可以切换到原生界面实现，后面也可以在接口不变的前提下开发native app版本

js框架选择原则

- 需要使用大量UI控件，表单密集型的应用
  - jQuery Mobile
  - Sencha Touch
  - Ionic

- UI较单一，注重性能，注重页面特效的应用（淘宝）
  - 使用多个小型js库
  - zepto.js作为底层库，兼容jquery语法
  - 使用sea.js进行模块的管理和发布
  - 使用backbone.js为基础的MVC架构，用来剥离应用的数据部分
  - 使用underscore.js做为前端模板引擎
  - HTML5游戏开发框架

MVVM Framework

- MVVM的全称是Model View ViewModel，这种架构模式最初是由微软的Martin Fowler作为微软展现层设计模式的规范提出
- MVC模式的衍生物，MVVM模式的关注点在能够支持事件驱动的UI开发平台，例如HTML5, WindowsPresentation Foundation (WPF), Silverlight和ZK framework，Adobe Flex
- ViewModel是一个专门用于数据转换的Controller,它可以把Model中的信息转换为View中的信息，同时从View传递命令给Model
- MVVM JS框架：AngularJs，EmberJs，Avalon，KnockoutJS, Kendo MVVM和Knockback.js

Ionic Framework

- The beautiful, open source front-end framework for developing hybrid mobile apps with HTML5.
- Bootstrap style framework，provide Best Practice for Hybrid App Development
- Ionic is focused on building native/hybrid mobile apps rather than mobile websites.

What Ionic Do?

- AngularJs integrate
- Url routing，use AngularUI Router
  - 单页面应用；可以指定不同的路由，方便开发和集成
- UI components
  - 简单的DOM结构，简洁的界面样式，iOS7扁平风格
- AngularJS Extensions & Directives
  - ion-tab, ion-content, ion-nav-view, ion-header
  - $ionicPopup,$ionicLoading, $ionicModal…
- Ionic Command Line
- CSS / SCSS，gulp

What Ionic Do?


```
.config(function($stateProvider, $urlRouterProvider) {
  $stateProvider

    // setup an abstract state for the tabs directive
    .state('tab', {
      url: "/tab",
      abstract: true,
      templateUrl: "templates/tabs.html"
    })

    // Each tab has its own nav history stack:

    .state('tab.dash', {
      url: '/dash',
      views: {
        'tab-dash': {
          templateUrl: 'templates/tab-dash.html',
          controller: 'DashCtrl'
        }
      }
    })

  // if none of the above states are matched, use this as the fallback
  $urlRouterProvider.otherwise('/tab/dash');
```

What Ionic Do?


```
angular.module('starter.controllers', [])

.controller('DashCtrl', function($scope) {
})

.controller('FriendsCtrl', function($scope, Friends) {
  $scope.friends = Friends.all();
})

.controller('FriendDetailCtrl', function($scope, $stateParams, Friends) {
  $scope.friend = Friends.get($stateParams.friendId);
})

.controller('AccountCtrl', function($scope) {
});
```

```
.factory('Friends', function() {
  // Might use a resource here that returns a JSON array

  // Some fake testing data
  var friends = [...];

  return {
    all: function() {
      return friends;
    },
    get: function(friendId) {...}
```

The Benefit of Ionic

- 使用AngularJS框架，提升开发效率50%以上；JS代码模块化
- 前端开发工程化，不再拼凑各种js插件
- Js开发和HTML Css开发隔离
- 前端自动化测试成为可能，Karma，E2E端到端测试
- 跨平台，跨设备，跨分辨率代码复用

PhoneGap集成框架

开源手机开发框架，使用HTML5, CSS3, JavaScript开发跨平台应用，针对每个平台提供应用包装，提供一致性的多平台硬件设备能力js API

- Cordova命令行
- 设备能力API插件化
- config.xml配置文件

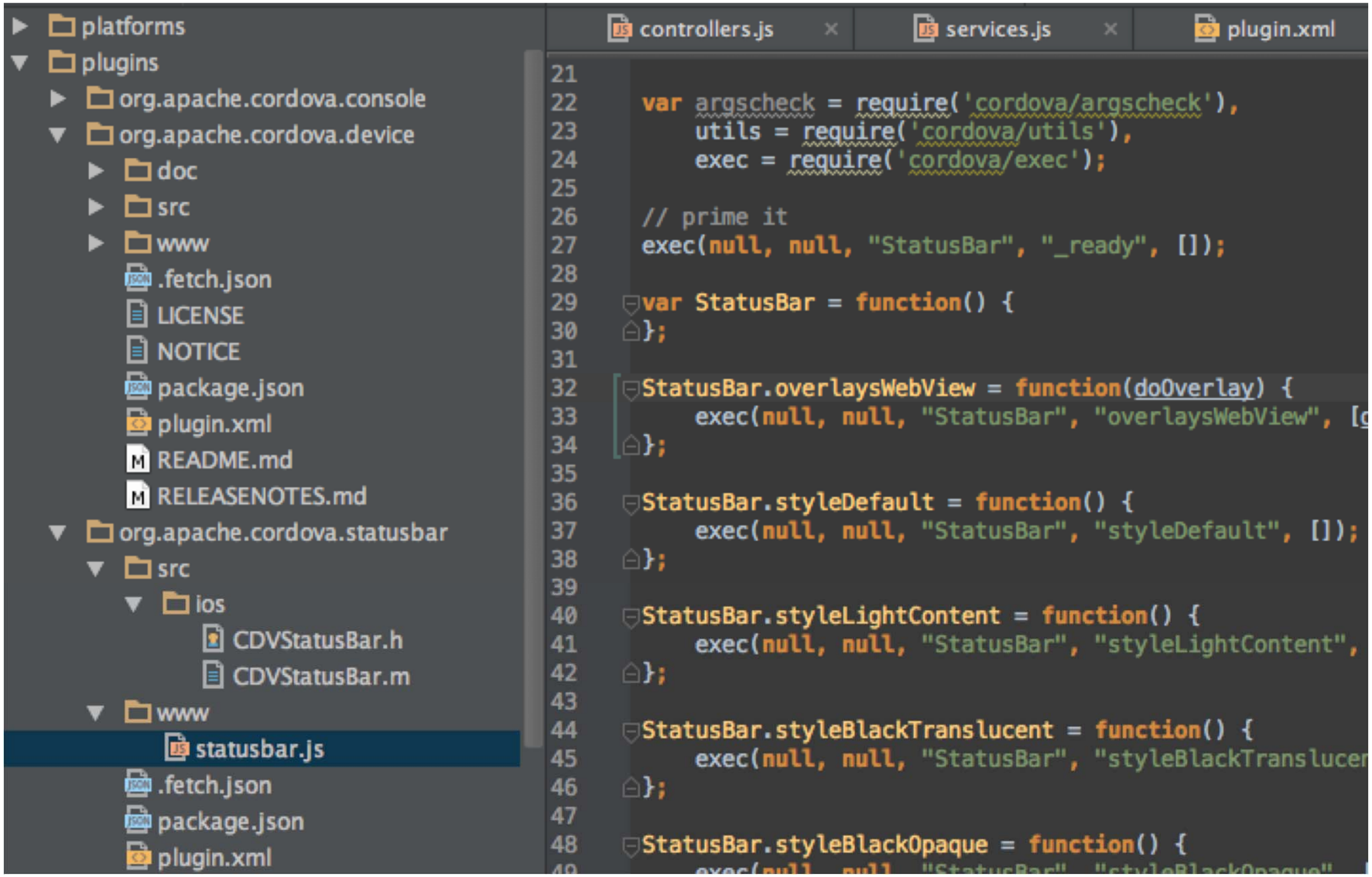
Cordova/PhoneGap

- Cordova CLI命令行工具
  - 提供创建项目，添加目标平台，编译运行项目等功能，自动化
- 设备API插件化
  - From version 3.0 onward, Cordova implements all device APIs as plugins, and leaves them disabled by default
  - Cordova plugin add 和 plugman命令
- 配置文件config.xml
  - <content src="index.html" />
    <preference name="Fullscreen" value="true" />
    <preference name="WebViewBounce" value="true" />
- 跨域问题
  - Cordova adheres to the W3C Widget Access specification, which relies on the <access> element within the app's config.xml
  - 默认<access origin="*" />
  - PhoneGap打包的应用是file本地域

PhoneGap StatusBar Plugin


```
<plugin xmlns="http://apache.org/cordova/ns/plugins/1.0"
    xmlns:rim="http://www.blackberry.com/ns/widgets"
    xmlns:android="http://schemas.android.com/apk/res/android"
    id="org.apache.cordova.statusbar"
    version="0.1.3">
    <name>StatusBar</name>
    <description>Cordova StatusBar Plugin</description>
    <license>Apache 2.0</license>
    <keywords>cordova,statusbar</keywords>

    <engines...>

    <js-module src="www/statusbar.js" name="statusbar">
        <clobbers target="window.StatusBar" />
    </js-module>

    <!-- ios -->
    <platform name="ios">
        <config-file target="config.xml" parent="/*">
            <feature name="StatusBar">
                <param name="ios-package" value="CDVStatusBar" onload="true" />
            </feature>
            <preference name="StatusBarOverlaysWebView" value="true" />
            <preference name="StatusBarBackgroundColor" value="#000000" />
        </config-file>

        <header-file src="src/ios/CDVStatusBar.h" />
        <source-file src="src/ios/CDVStatusBar.m" />

    </platform>
```

PhoneGap StatusBar Plugin



PhoneGap Plugins

- 我们目前用到的插件
  - StatusBar，Console Log，Device，network information
  - 极光推送 Jpush Plugin
- 我们自己也做个插件
  - LockScreen锁屏插件化

HTML5 in mobile device

- 页面Meta控制显示区域各种属性：
- <meta content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0" name="viewport">
  - width - viewport的宽度
  - height – viewport的高度
  - initial-scale - 初始的缩放比例
  - minimum-scale - 允许用户缩放到的最小比例
  - maximum-scale – 允许用户缩放到的最大比例
  - user-scalable – 用户是否可以手动缩放

HTML5 in mobile device

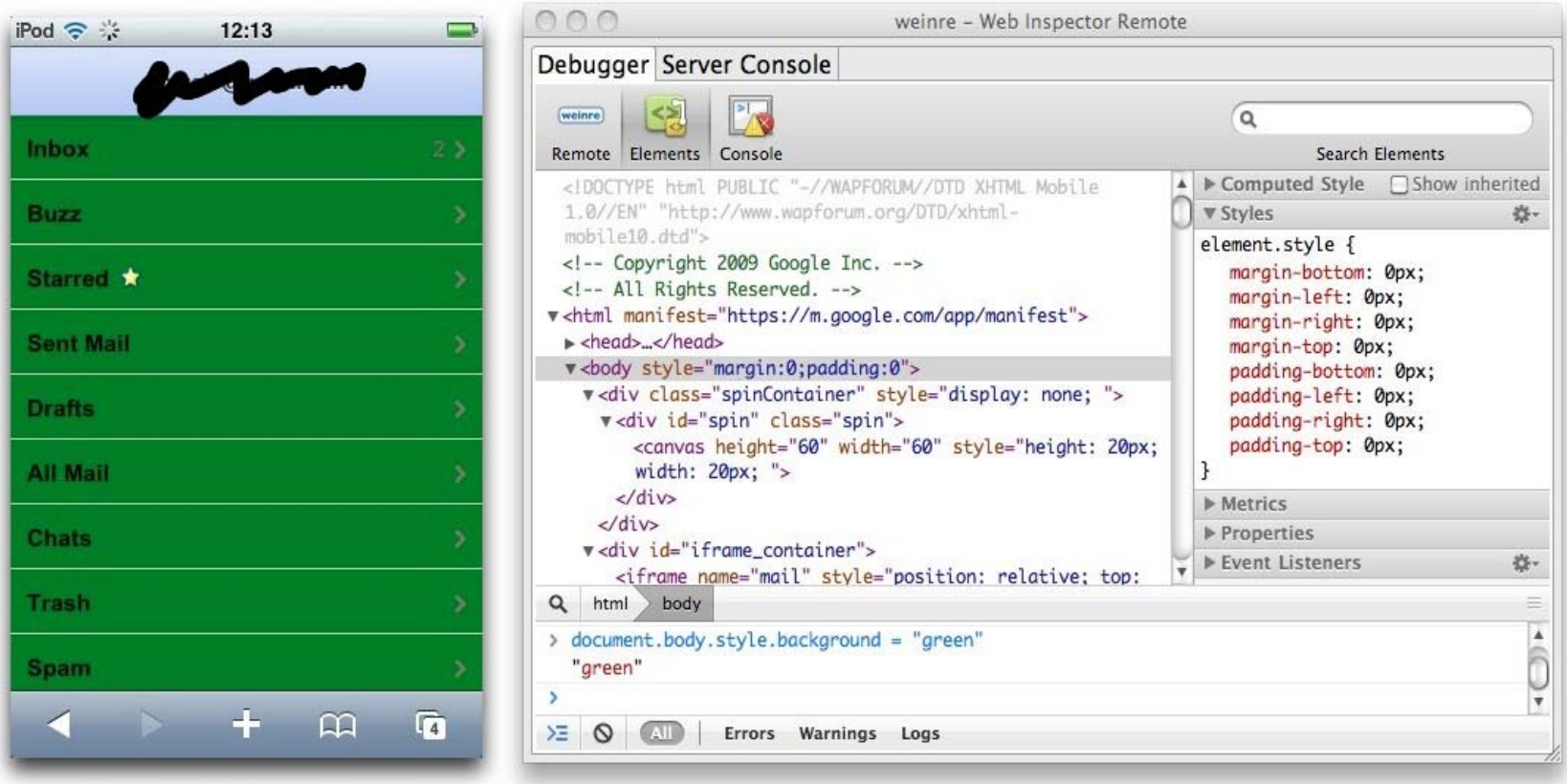
- IOS中Safari允许全屏浏览：
- <meta content="yes" name="apple-mobile-web-app-capable">
- IOS中Safari顶端状态条样式：
- <meta content="black" name="apple-mobile-web-app-status-bar-style">
- 忽略将数字变为电话号码：
- <meta content="telephone=no" name="format-detection">
- 关闭Input键盘默认首字母大写：autocapitalize=”off”

CSS3 in mobile device

- 现在移动端两大阵营ios和android都是基于webkit内核的，而webkit内核对CSS3的支持走在了前面
- CSS3给我们带来了如：文字阴影（text-shadow）、盒子阴影（box-shadow）、圆角（border-radius）、背景渐变（background: linear-gradient(#000, #fff)）、2D变换（transition）、动画（animation）等常用属性
- 还有如-webkit-mask、-webkit-text-stroke、-webkit-nbsp-mode、-webkit-tap-highlight-color、-webkit-box-reflect、-webkit-marquee、-webkit-box等webkit内核私有属性

Mobile Web Debug Tool—Weinre

- Weinre代表**We** b **In** spector **Re** mote，是一种远程调试工具。举个例子，在电脑上可以**即时** 的更改手机上对应网页的页面元素、样式表，或是查看Javascript变量，同时还可以看到手机上页面的错误和警告信息.



Xcode & ionic cmd tool

- ionic命令创建工程流程
  - npm install -g ionic
  - ionic start myApp tabs
  - ionic platform add ios
  - ionic build ios
  - ionic run ios
- Xcode编译，调试和发布

Gulp Aiming to Dethrone Grunt

一个新的构建系统gulp，希望能够取代Grunt，成为最流行的JavaScript任务运行器

- 易于使用：采用代码优于配置策略，gulp让简单的事情继续简单，复杂的任务变得可管理。
- 高效：通过利用node.js强大的流，不需要往磁盘写中间文件，可以更快地完成构建。
- 高质量：gulp严格的插件指导方针，确保插件简单并且按你期望的方式工作。
- 易于学习：通过把API降到最少，你能在很短的时间内学会gulp。构建工作就像你设想的一样：是一系列流管道。

A sample Gulp task


```
var gulp = require('gulp');
var uglify = require('gulp-uglify');
gulp.task('scripts', function() {
  return gulp
    .src(['src/js/**/*.js'])
    .pipe(uglify())
    .pipe(gulp.dest('build/js'));
});
```

交流时间

讲座	5月25日 基于AC的自动化测试集成
讲座	6月14日 重构，我的遗留系统改善之路

QQ群	312010267（ Android开发 ）

更多…	uml.org.cn 文章, 文挡, 课程