

# HTML5开发技术 在ExMobi中的应用

HTML5简介

开发前的准备

Agile Lite开发框架

HTML5移动应用开发技巧

在ExMobi中使用Agile Lite



# HTML5简介

HTML 5是近十年来Web开发标准最巨大的飞跃。和以前的版本不同，HTML 5并非仅仅用来表示Web内容，它的新使命是将Web带入一个成熟的应用平台，在HTML 5平台上，视频，音频，图象，动画，以及同设备的交互都被标准化。



# HTML5简介

## 语义特性 ( Class : Semantic )

HTML5赋予网页更好的意义和结构。借助 header, footer, section, article 这些标签，我们可以实现更具结构化，语义化的 Web 文档。这样，搜索引擎可以更容易索引 Web 站点，我们也可以搜索到更快，更准确的信息。



# HTML5简介

## 本地存储特性 ( Class: OFFLINE & STORAGE )

基于HTML5开发的网页APP拥有更短的启动时间，更快的联网速度，这些全得益于HTML5 APP Cache，以及本地存储功能。即使在 Internet 连接中断之后也能继续访问HTML5应用。

## 设备兼容特性 (Class: DEVICE ACCESS)

从Geolocation功能的API文档公开以来，HTML5为网页应用开发者们提供了更多功能上的优化选择，带来了更多体验功能的优势。HTML5提供了前所未有的数据与应用接入开放接口。使外部应用可以直接与浏览器内部的数据直接相连，例如视频影音可直接与microphones及摄像头相联。



# HTML5简介

## 连接特性 ( Class: CONNECTIVITY )

更有效的连接工作效率，使得基于页面的实时聊天，更快速的网页游戏体验，更优化的在线交流得到了实现。HTML5拥有更有效的服务器推送技术，**Server-Sent Event**和**WebSockets**就是其中的两个特性，这两个特性能够帮助我们实现服务器将数据“推送”到客户端的功能。

## 网页多媒体特性(Class: MULTIMEDIA)

支持网页端的Audio、Video等多媒体功能，与网站自带的APPS，摄像头，影音功能相得益彰。

## 三维、图形及特效特性 ( Class: 3D, Graphics & Effects )

基于SVG、Canvas、WebGL及CSS3的3D功能，用户会惊叹于在浏览器中，所呈现的惊人视觉效果。

# HTML5简介

## 性能与集成特性 ( Class: Performance & Integration )

没有用户会永远等待你的Loading——HTML5会通过XMLHttpRequest2等技术，解决以前的跨域等问题，帮助您的Web应用和网站在多样化的环境中更快速的工作。

## CSS3特性(Class: CSS3)

在不牺牲性能和语义结构的前提下，CSS3中提供了更多的风格和更强的效果。此外，较之以前的Web排版，Web的开放字体格式（WOFF）也提供了更高的灵活性和控制性。而且CSS3的渲染直接使用的GPU加速，对于支持GPU的设备，其动画运行效率相对于JS实现的动画运行效率要提高3倍以上。

# HTML5优缺点

## 优点

- 语法灵活简单，学习门槛低
- 无需编译，无缝加载
- 逐步增强的本地能力和动画
- 移植性好，维护方便
- 扩展性好，通过JS桥接可与宿主容器进行交互

## 缺点

- 跨平台表现牵强，不同系统和浏览器内核展现有差异
- 性能和效率有待提升
- 成熟框架少，使用成本高
- 存在安全隐患，容易泄露敏感数据

# 开发前的准备

## 几个关键问题

### 1、服务端能力

客户端发送请求在移动web设备上查看某个内容。这个请求通常是一个发送到某台服务器上的HTTP请求。通常，会自动生成用户需要的内容。为了将内容加入到app中通常需要至少两件事情：数据以及怎样使数据变得有意义，或者是我们的HTML5 app。

### 2、你的降级策略如何？

HTML5的跨平台性只是依赖于W3C的标准规范，然而，不同的解析内核对HTML5的处理仍有不同，如何去做到全终端的兼容，特别是Android设备的兼容？

当应用无法按照预想的方式呈现时，能够提供一组隔离代码是最好的撤回策略。换句话说，就是要使代码尽量简洁且不依赖某些特殊的边缘技术——我们称之为优雅的降级。

### 3、如何处理离线数据

15% 的移动apps启动时设备是脱机。所以你的应用程序需要处理脱机的情况。最初的程序设计时页面的访问当然是需要通过Internet的，但是如果连接端掉了，你的程序应该也能够继续正常运行。当设备连接到网络时，应用程序如何处理用户的数据？为了处理脱机访问的情况，是否可以考虑将所有的数据打包返回？



# 开发前的准备

## JS脚本分类

### 1、混合脚本

混合脚本可以将开发人员编写的核心脚本与设备SDK连接起来。如果你想使用本地封装器(如UIWebView和PhoneGap。)将HTML5 app封装起来，那么你需要使用混合脚本。需要针对不同的平台编写不同的混合脚本(phonegap.js是唯一一个支持多个平台的脚本)。

### 2、核心脚本

核心脚本是app在所有平台上的通用部分。如果app是通过浏览器访问的，核心脚本也应该使得app看起来像是一个native SDK。你的app需要负责收集和渲染HTML5页面。这个时候使用jQuery这类完整的框架是非常有用的，但是我们建议还是尽量使用微框架。

### 3、设备脚本

最后，你需要使用设备脚本模拟本地的行为。jQTouch是一个很好的例子，它使用jQuery模拟iPhone的本地行为和动作。但是jQTouch无法区分不同设备间的差异，在Android以及其他的移动平台上，它都仍然使用在iOS上采用的方法。这一点受到用户的诟病。因此，可能需要为不同的设备编写单独的设备脚本。

# 开发前的准备

## CSS分类

### 1、设备主题

这是开发者用来模拟设备风格的CSS代码。它是用户惯用的语言，普通用户看到这些语言后，能够直接开始执行想要处理的任务，而不用花费时间去学习你的界面。据不完全统计，每个平台上有将近100个独立的界面组件。不推荐开发者自己设计一种新的组件，除非你清楚的知道自己在做什么。

### 2、核心主题

核心主题是应用程序中可以重复利用的部分——那些你需要使用，但不一定会显示在界面的上的内容。我习惯将各种部件单独表示(如复位按钮，布局，排版，颜色，图像)，这样就能将这些部件作为我的核心主题。这些主题在不同的平台上都是一样的。比如，你的logo在所有的平台上都是一样的。你的工具栏的颜色总是保持一样的风格。这些组件都可以定义为你核心主题。当你在不同的设备上使用这些核心主题时，只需要修改相应的设备主题即可。。

### 3、App主题

App主题是专属于某个App的风格组件。许多项目旨在将这些组件整合到一个样式表中。但是我建议，还是将你的核心主题和App主题区分开来。这对于debug有很大的帮助。

## 设置屏幕自适应

webkit内核中的一些私有的meta标签，这些meta标签在开发webapp时起到非常重要的作用，它可以设置页面适应屏幕的比例。

1 `<meta content=" width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0;" name=" viewport" />`

2 `<meta content=" yes" name=" apple-mobile-web-app-capable" />`

3 `<meta content=" black" name=" apple-mobile-web-app-status-bar-style" />`

4 `<meta content=" telephone=no" name=" format-detection" />`

第一个meta标签表示：强制让文档的宽度与设备的宽度保持1:1，并且文档最大的宽度比例是1.0，且不允许用户点击屏幕放大浏览；

第二个meta标签是iphone设备中的safari私有meta标签，它表示：允许全屏模式浏览；

第三个meta标签也是iphone的私有标签，它指定的iphone中safari顶端的状态条的样式；

第四个meta标签表示：告诉设备忽略将页面中的数字识别为电话号码

# HTML5移动应用开发技巧

## HTML5标签的使用

在开始编写webapp时，建议前端工程师使用HTML5，而放弃HTML4，因为HTML5可以实现一些HTML4中无法实现的丰富的WEB应用程序 的体验，可以减少开发者很多的工作量，当然了决定使用HTML5前，一定要对此非常熟悉，要知道HTML5的新标签的作用。比如定义一块内容或文章区域 可使用section标签，定义导航条或选项卡可以直接使用nav标签等等。

## 自适应布局模式

在编写CSS时，我不建议前端工程师把容器（不管是外层容器还是内层）的宽度定死。为达到适配各种手持设备，我建议前端工程师使用自适应布局模式（支付宝 采用了自适应布局模式），因为这样做可以让你的页面在ipad、itouch、ipod、iphone、android、web safarik、chrome都能够正常的显示，你无需再次考虑设备的分辨率。

# HTML5移动应用开发技巧

## 块级化a标签

请保证将每条数据都放在一个A标签中，为何这样做？因为在触控手机上，为提升用户体验，尽可能的保证用户的可点击区域较大。而且A可以接收到hover、active等css效果。

## 学会使用webkit-box

前面我们说过自适应布局模式，有些同学可能会问：如何在移动设备上做到完全自适应呢？很感谢webkit为display属性提供了一个webkit-box的值，它可以帮助前端工程师做到盒子模型灵活控制。

## 如何去除Android平台中对邮箱地址的识别

看过iOS webapp API的同学都知道iOS提供了一个meta标签:用于禁用iOS对页面中电话号码的自动识别。在iOS中是不自动识别邮件地址的，但在Android平台，它会自动检测邮件地址，当用户touch到这个邮件地址时，Android会弹出一个框提示用户发送邮件，如果你不想 Android自动识别页面中的邮件地址，你不妨加上这样一句meta标签在head中

```
1 <meta content=" email=no" name=" format-detection" />
```



# HTML5移动应用开发技巧

## iOS中如何禁止用户选中文字

我们通过指定文字标签的-webkit-user-select属性为none便可以禁止iOS用户选中文字。

## 如何解决盒子边框溢出

当你指定了一个块级元素时，并且为其定义了边框，设置了其宽度为100%。在移动设备开发过程中我们通常会对文本框定义为宽度100%，将其定义为块级元素以实现全屏自适应的样式，但此时你会发现，该元素的边框(左右)各1个像素会溢了文档，导致出现横向滚动条，为解决这一问题，我们可以为其添加一个特殊的样式-webkit-box-sizing:border-box;用来指定该盒子的大小包括边框的宽度。

## iOS中如何禁止用户保存图片\复制图片

元素的-webkit-touch-callout属性，同样为一个img标签指定-webkit-touch-callout为none也会禁止设备弹出列表按钮，这样用户就无法保存\复制你的图片了。

# Agile Lite移动应用前端开发框架

Agile Lite是一个基于HTML5+CSS3+JS的移动应用开发框架，在体验上尽量接近Native Like，并且同时支持单页模式和多页模式。

下载地址：<http://www.agilelite.net>

为移动应用开发所有应用场景而设计。

Agile 让HTML5在移动应用开发中充分发挥优势。所有开发者都能快速上手、所有设备都可以适配、所有项目都适用。



## 简单、可扩展

Agile 支持Zepto和jQuery双引擎及相应的扩展；同时Agile均支持单页模式和多页模式的移动应用，可以与ExMobi、PhoneGap等流行的跨平台开发框架一起使用。



## 一个框架、多种设备

你的移动应用能在 Agile 的帮助下通过同一份代码快速、有效适配手机、平板等设备，这一切都是 CSS 媒体查询（Media Query）的功劳。



## 组件齐全

Agile 提供了全面、美观的文档。你能在这里找到关于 HTML 元素、HTML 和 CSS 组件、JavaScript 插件方面的所有详细文档。

## 框架特点

- ① 纯HTML5框架，包括接近原生效果的UI组件和丰富的逻辑处理
- ② 同时支持单页和多页模式，完善的页面交互和参数传递机制
- ③ 继承ExMobi应用的完整生命周期，支持监听各种系统事件
- ④ 无缝调用ExMobi支持的所有本地能力，可与ExMobi原生插件交互
- ⑤ 灵活的扩展性，支持在微信、PhoneGap等不同宿主容器中运行
- ⑥ 定制化的UI风格，可根据需要在线生成框架UI样式

## 单页模式

单页模式（Single-page Application）即在一个HTML5移动应用中只包含一个HTML页面，而不同视图的显示实际是在一个页面中采用动态显隐实现，而其中最重要的技术的就是Ajax，不同视图的获取都是通过Ajax从远程服务器中获取。

也就是说，不同的视图都是一个HTML片段，而不是完整的HTML页面。

在SPA模式中，主页面（完整HTML页面）是可以独立加载、更新和替换的一些可视元素的组合（HTML片段）。通过这种方式，可以不必在每次用户操作后重新加载整个页面。在任何时候，都只显示与应用程序当前阶段相关的可视元素和内容。其他所有内容均被隐藏；但只要应用程序流程中需要用到它，它就会显示出来。



# Agile Lite开发框架

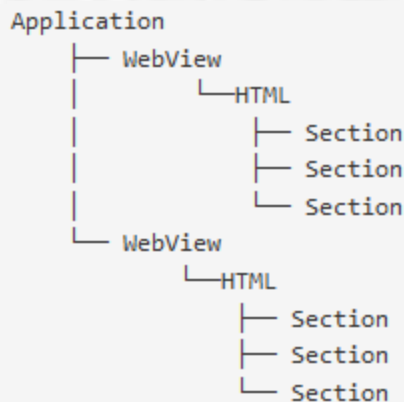
## 多页模式

多页模式 ( Multi-page Application ) 是相对于单页模式而言，应用中的每一个页面都是一个独立HTML页面，而不是HTML片段。

如果是采用传统的WEB思维，在移动应用中的采用多页模式，则每个页面可能都会重复加载一些数据 ( JS、CSS、部分HTML代码等 )，并且用户的前进后退操作都有可能会造成页面的刷新，降低用户体验。

所以，需要采用更加高效的方式将通用的数据缓存，提高每个页面加载速度。

如果一个移动app使用agile开发多页模式的应用，则其页面结构应该是：





# Agile Lite开发框架

## Agile Lite的使用

Agile Lite是一个H5的框架，所以使用Agile Lite必须要引入一些必要的JS和CSS。

## JS的引用

由于JS是顺序执行的语言，所以对于JS文件的应用顺序是非常重要的。

首先是DOM操作引擎，内置了jQuery和Zepto两种，二者取其一，不可混用。

```
<!-- jQuery-->  
<script src="assets/third/jquery/jquery-2.1.3.min.js"></script>  
<script src="assets/third/jquery/jquery.mobile.custom.min.js"></script>
```

```
<!-- ZeptoUI -->  
<script src="assets/third/zepto/zepto.min.js"></script>
```

# Agile Lite开发框架

## JS的引用

其次是使用的第三方工具类，比如artTemplate、iScroll5：

```
<!-- 用于部分与滚动相关的组件-->  
<script src="assets/third/iscroll/iscroll-probe.js"></script>  
<!-- 用于数据注入-->  
<script src="assets/third/arttemplate/template-native.js"></script>
```

最后才是Agile Lite的执行引擎是：

```
<!-- Agile Lite执行引擎-->  
<script type="text/javascript" src="assets/agile/js/agile.js"></script>
```

Agile Lite如果要调用某些宿主容器的本地能力，还需要引入桥接类，比如在ExMobi中：

```
<!-- 宿主容器桥接类-->  
<script type="text/javascript" src="assets/bridge/exmobi.js"></script>  
<script type="text/javascript" src="assets/bridge/agile.exmobi.js"></script>
```

# Agile Lite开发框架

## CSS的引用

Agile Lite内置了Flat UI和Ratchet UI两部分，二者取其一，不可混用。

```
<!-- Flat UI -->
<link rel="stylesheet" href="assets/agile/css/agile.layout.css">
<link rel="stylesheet" href="assets/agile/css/flat/flat.component.css">
<link rel="stylesheet" href="assets/agile/css/flat/flat.color.css">
<link rel="stylesheet" href="assets/agile/css/flat/iconline.css">
```

```
<!-- Ratchet UI -->
<link rel="stylesheet" href="assets/agile/css/ratchet/css/ratchet.min.css">
```

## Agile Lite的启动

Agile Lite默认情况下是不工作的，只有当启动之后才会工作。

Agile Lite的启动方式为：

```
A.launch(options);
```

## Agile Lite的启动

说options之前，我们简单介绍一下宿主容器的概念：H5实际只是一种标准的定义，所以要想把H5渲染成一个网页查看，就需要宿主容器。

最常见的宿主容器就是浏览器，Chrome、火狐、IE等都是，而现在比较火的微信内置浏览器是H5的宿主容器，ExMobi的webview也是H5的宿主容器。区别与一般浏览器，微信、ExMobi作为H5的宿主容器提供了自己特有的扩展能力，这种扩展能力通过桥接的方式在H5可以通过JS来使用。

由于桥接JS的加载需要一定的时间（虽然很短），为了方便开发者判断何时桥接类加载完成，一般宿主容器在加载完桥接类会触发一个事件，开发者只需要监听此事件就可以知道桥接类已经加载完成。

## Agile Lite的启动

options为启动的配置项，主要是初始化一些内部工作的参数。主要包括：

**readyEvent:** AL启动的时机，也就是在什么事件触发AL的初始化。一般在浏览器中就是浏览器的ready事件；如果是需要使用宿主容器的本地能力，则应该为宿主容器桥接类加载完毕的事件，比如在ExMobi中为plusready。也可以设置为空串，意即立即启动。

**backEvent:** 绑定宿主容器返回键事件，如果直接在浏览器中则可以需要设置

**crossDomainHandler:** 跨域请求处理函数，在宿主容器中一般都有自己的http请求类，避免跨域问题，当在具有此能力的宿主容器中可以指向跨域的请求走此函数而不是使用jquery之类的jsonp方式处理。

**usePageParam:** 是否记录url参数，true为记录，false为不记录。此配置需要先了解AL的单页模式的处理：AL对于单页模式实际上是不断往一个HTML页面中加载新的section片段，也就是说页面的URL是不变的，区分这些section是通过URL的hash值来判断的，因为如果URL变了浏览器就会认为是新页面就会整个刷新。假设不记录url参数的话，当用户已经打开第二个页面或者甚至更多页面，这时候用户分享此页面出去，另一个用户点击URL打开的依然为第一个页面，而不是第二个页面，因为第二个页面的构建是通过JS去触发的。而如果设置了true，AL会记录新页面的参数并进行base64编码作为一个hash，从而做到URL不变，根据hash来记录参数的效果。



# Agile Lite开发框架

## Agile Lite的启动

下面就是一个典型的启动写法：

```
A.launch({  
    readyEvent: 'ready'  
});
```

```
$(document).on('ready', function(){  
    A.launch({  
        readyEvent: "  
    });  
});
```

以上写法为一般浏览器中的启动，后面会介绍在ExMobi中如何启动。

当启动结束，Agile Lite会触发document的**agileready**自定义事件，开发者可在此事件  
中做必要的逻辑，比如调用宿主容器的本地能力等。

## 控制器与组件

当一个元素设置了data-toggle，则此元素即为控制器；  
当一个元素设置了data-role，则此元素即为组件；  
组件的切换是用过控制器来完成，页面也可以认为是一个组件。

## 控制器工作原理

满足以下条件的组件和控制器可以正常工作：

- 1、控制器的data-toggle的值与组件的data-role的值一致；
- 2、控制器的href值与组件的id“同名”。

## 控制器工作原理

所谓同名，控制器的href值只能有两种形式：

1、#id形式，这时候此id必须跟组件的id一致；比如：

```
<!-- 控制器 --->  
<a href="#tab1" data-toggle="article">tab</a>  
<!-- 组件 --->  
<article id="tab1" data-role="article">  
</article>
```

2、url形式，这时候url的文件名（不含后缀）必须与组件的id一致。比如：

```
<!-- 控制器 --->  
<a href="tab2.html?a=1" data-toggle="article">tab</a>  
<!-- 组件 --->  
<article id="tab2" data-role="article">  
</article>
```

# Agile Lite开发框架

## 组件生命周期

控制器的作用就是控制组件的加载和显隐，所以一般组件都包括创建（load）、显示（show）、隐藏（hide）这三个生命周期。

当组件处于某个生命周期时都会触发相应的事件。

事件的名称规则为：组件名+生命周期名称。

比如section组件第一次加载的时候会触发sectionload和sectionshow事件，当某个section被隐藏（打开新的section，上一个section则被隐藏）会触发sectionhide事件。

## 组件的销毁

Agile Lite中的组件不能随意的通过\$的remove函数删掉，而是建议通过设置data-cache= "false" 来设置组件不缓存。因为AL对已经加载过一次的组件再次打开的时候不重新加载的，也就是默认是缓存的，当开发场景不需要缓存的时候设置此属性即可

# Agile Lite开发框架

## 控制器传参

当控制器需要给打开的组件传参时，控制器的href只能为url形式，而不能为#id形式。而组件获取参数的方式则是通过A.Component.params(selector)来调用，此方法得到的是一个参数的json对象。并且获取参数通常是在组件load或者show的时候获取。其中selector为控制器控制的组件的选择器。比如下面的article控制器：

```
<a class="menu" data-role="tab" href="second_article.html?id=1&title=Agile框架实战" data-toggle="article">  
  <span class="menu-icon iconfont iconline-notebook"></span>  
  <span class="menu-text">Article</span>  
</a>
```

```
$('#second_article').on('articleload', function(){  
  var params = A.Component.params(this);//获取所有参数，此处为{'id':'1',  
  'title':'Agile框架实战'}  
  A.showToast('参数id的值为: '+params.id);  
});
```



# Agile Lite开发框架

## 数据注入

Agile Lite的数据注入是基于artTemplate，默认使用其Native语法。

详细参考：<https://github.com/aui/artTemplate/wiki/syntax:native>

```
<script id="template" type="text/html">
<% for (i = 0; i < list.length; i ++) { %>
  <li>条目内容 <%= i + 1 %> : <%= list[i] %></li>
<% } %>
</script>
```

其特点是：

- 1、模板必须位于script标签内，并且必须设置id和type，其中type必须为text/html
- 2、Agile Lite对其做了扩展，模板内容可以独立于html之外而使用src引用，这时候script标签内有内容责备忽略，而使用src指向的模板内容

# Agile Lite开发框架

## 渲染模板

Agile Lite对artTemplate进行了基础的渲染模板封装，即：将一个JSON对象注入到artTemplate的模板中，返回渲染后的html代码片段。

基本用法：A.template('#templateId').render(json);

其中templateId为模板的id，json为要注入到模板中的数据对象

```
var html = A.template('#template').render({list:['Agile Lite移动前端框架',  
'ExMobi三大引擎完美融合']});
```

```
$('#content').append(html);
```

## 参考

[http://www.agilelite.net/extend.html#render\\_html](http://www.agilelite.net/extend.html#render_html)

## 占位模板

占位模板即是將某个模板放置于要显示的位置，当往模板注入数据的时候，渲染后的数据即在模板所在的位置进行展示，不需要手动放置于某个dom中。

用法：A.template('#id').renderReplace|renderBefore|renderAfter(data, callback)

其中，id为artTemplate模板的id，data为要注入的JSON数据对象或者为一个url，此url的响应为要注入的JSON数据字符串

Callback为注入后的回调函数，其接收三个内置参数，第一个参数为渲染后的dom对象，第二个参数为模板的内容字符串，第三个参数为要注入的数据JSON对象。

每次注入的新数据，根据函数renderReplace替换原来已经注入的数据；renderBefore为在原有数据的顶部追加；renderAfter为在原有数据的尾部追加。

```
A.template('#template').renderAfter(  
    {list:['Agile Lite移动前端框架', 'ExMobi三大引擎完美融合']},  
    function(){  
        refresh.refresh();  
    });
```

## 占位模板

假设页面中有占位模板如下：

```
<article data-role="article" id="main_article" data-scroll="pull" class="active"
style="top:44px;bottom:0px;">
  <div class="scroller">
    <ul id="content" class="listitem">
      <li class="sliver">介绍</li>
      <script id="template" type="text/html">
        <% for (i = 0; i < list.length; i ++) { %>
          <li>条目内容 <%= i + 1 %> : <%= list[i] %></li>
        <% } %>
      </script>
    </ul>
  </div>
</article>
```

# Agile Lite开发框架

## 占位模板

可以在下拉刷新组件中使用占位模板动态加载数据：

```
$('#main_article').on('refreshInit', function(){  
    //首次进来加载使用replace  
    A.template('#template').renderReplace('json/template.json');  
    var refresh = A.Refresh('#main_article');  
    refresh.on('pulldown', function(){  
        //下拉刷新加载到前面用before  
        A.template('#template').renderBefore({list:['Agile Lite移动前端框架',  
'ExMobi三大引擎完美融合']},function(h){  
            //h为渲染后的dom对象  
            refresh.refresh();  
        });  
    });  
});
```

## 参考

[http://www.agilelite.net/extend.html#render\\_positoin](http://www.agilelite.net/extend.html#render_positoin)



# Agile Lite开发框架

## Agile Lite与iScroll5

Agile Lite的组件与iScroll5由于千丝万缕的关系，其中的scroll、refresh和slider组件均是通过iScroll5实现的（具体为其中的iscroll-probe.js细分类），所以这几个组件必须符合iScroll的三层架构，即wrapper、scroller、container三部分。

关于iScroll5，推荐大家阅读《[iScroll5中文API建议参考手册](#)》

## iScroll5组件的刷新

我们知道iScroll5实现区域的滚动不是使用系统自带的滚动条，而是自己去计算滚动区域来模拟滚动的。

所以，用了iScroll5的Agile Lite组件必须要遵循iScroll5的刷新要求。即：

当组件的内部dom发生变化，导致组件的空间大小有变化，这时候需要调用iScroll5的刷新方法（refresh）以使组件内部的滚动区域重新进行计算。

比如：img的src加载图片异步可能导致滚动区域空间大小变化；ajax异步加载数据也可能导致滚动区域空间大小变化。这时候都需要调用iScroll5的刷新方法。

# Agile Lite开发框架

## 在Agile Lite中使用刷新

iScroll5的刷新函数在其实例对象中即可调用，方法为：

```
var myScroller = new IScroll();  
myScroller.refresh();
```

使用了iScroll5的Agile Lite组件的实例化对象本身就是iScroll5的实例对象。

所以，此组件的实例对象.refresh()即可调用iScroll5的刷新函数。比如scroll滚动组件：

```
var scroll = A.Scroll('#id');  
scroll.refresh();
```

而一般组件的变化通常是跟数据注入有关系的，因为数据注入实际就是对组件内容的修改，所以这里就需要重新提到占位模板的回调函数，此函数的最大作用就是做组件的刷新。

# 在ExMobi中使用Agile Lite

## webview设置

```
<html isbridge="true">
  <head>
    <title show="false">Agile Lite</title>
    <script>
      <![CDATA[

    ]]>
    </script>
  </head>
  <body style="padding:0px;margin:0px;"
    <webview id="browser" url="res:page/www/index.html"
backmonitor="true"></webview>
  </body>
</html>
```

- 1、isbridge设置为true，即可在webview中使用ExMobi的本地能力，否则不可调用
- 2、backmonitor设置为true，即可拦截uixml页面的关闭，而透传到ExMobi的webview中

# 在ExMobi中使用Agile Lite

## 引用桥接类

```
<!-- 宿主容器桥接类-->  
<script type="text/javascript" src="assets/bridge/exmobi.js"></script>  
<script type="text/javascript" src="assets/bridge/agile.exmobi.js"></script>
```

## 启动Agile Lite

```
A.launch({  
    readyEvent: 'plusready', //触发ready的事件，在ExMobi中为plusready  
    backEvent: 'backmonitor', //绑定ExMobi的后退事件  
    crossDomainHandler: function(opts) {  
        $util.server(opts); //使用ExMobi的http请求类处理跨域  
    }  
});
```

# 在ExMobi中使用Agile Lite

## 标准JS封装的ajax

用法：A.util.ajax( options )

说明：此方法除了是对标准JS里的ajax封装，同时与启动项crossDomainHandler相互作用，在跨域的时候智能选择

## ExMobi中的ajax






函数	参数	描述
\$util.ajax	opts 为json对象 {url,type,data,success,error,isBlock}	该请求实际调用的是ExMobi的直连DirectAjax，与ExMobi服务端无关
\$util.server	opts 为json对象 {url,type,data,success,error,isBlock}	该请求实际调用的是ExMobi的Ajax服务端路由处理，可以使用ExMobi服务端的路由拦截处理
\$util.form	opts 为json对象 {url,type,fileElementId,data,success,error,isBlock}	该请求实际调用的是ExMobi的DirectFormSubmit，与ExMobi服务端无关，主要用于带文件的表单提交。 fileElementId用于设置存储了文件信息的对象id
\$util.serverForm	opts 为json对象 {url,type,fileElementId,data,success,error,isBlock}	该请求实际调用的是ExMobi的FormSubmit服务端路由处理，主要用于带文件的表单提交。 fileElementId用于设置存储了文件信息的对象id



# 在ExMobi中使用Agile Lite

## 本地能力的使用

### File对象属性

属性	说明及描述
<a href="#">objName</a>	<p>【说明】返回该对象所属类名，字符串全小写，只读。返回file</p> <p>【适用平台】    </p> <p>【基座版本】4.0<sup>+</sup>，wp8在4.6<sup>+</sup>支持，win8在4.10<sup>+</sup>支持</p>

手册中标注了地球仪图标  的对象和方法均支持在HTML5中直接使用。

如果实在页面加载完成就使用请在plusready事件中使用；如果在Agile Lite中使用请在Agile Lite初始化结束事件agileready后使用，或者在某个组件的show或者load事件中使用。

# 支撑渠道一览

**Agile Lite官网**

<http://www.agilelite.net/>

**YY在线培训**

频道：73690765

**远程协助**

QQ群：498752673

**支撑电话**

025-66777333

**支撑邮箱**

[support@exmobi.cn](mailto:support@exmobi.cn)



**EXMOBi®**

[www.exmobi.cn](http://www.exmobi.cn)