如何开发移动端页面

前言

现在，移动互联网大热，我们公司也有越来越多的移动互联网产品，而其中很多产品都是使用 native code+web HTML 技术架构来开发的。native code 为 HTML 提供运行环境，并以 Javascript 接口为 HTML 提供本地功能接口（如访问文件系统、打开摄像头、读取通讯录等等）。当从头开发一个 App 时，通常有以下几个步骤：

1. 在 Eclipse 中新建一个 Android 工程（iphone 开发类似），工程完整文件夹通常由技术中心提供（已引入必要的 java 文件、jar 包、配置文件等）。
2. 修改工程名称、包路径、类名、配置文件等，使此工程变成你们自己的工程。
3. 在 www 文件下开发 html、js 和 css，完成相应的业务功能，html 通常放置在/www/pages/中,js 和 css 通常放置在/www/resources/中。

本文所讨论的内容只限于以上步骤中的第 3 步，前两步骤的相关问题以及本地功能的接口调用不在本文讨论的范围内。

背景

如果你正在进行移动端页面开发，那么无论你喜欢或不喜欢，你已经是一个前端工程师了。熟悉甚至精通 HTML、Javascript 和 CSS 对于开发一个优秀的 App 是必不可少的，三者缺一不可。如果你说“我是一名程序员，CSS 是美工活，我不会”，那现在就开始学习 CSS 吧，使用 CSS 是一名前端工程师的必备技能，了解和熟悉 CSS 对于高效开发 HTML 和 Javascript 是大有帮助的。

我们选用 jQuery 和 jQuery mobile 作为开发最基础的框架，jQuery 用于操作 DOM 和 ajax 调用等，jQuery mobile 用于管理页面之间的导航以及提供 UI 组件等。在写这篇文章时，框架使用的 jQuery 版本是 1.6.1，jQuery mobile 版本是 1.0 b1。熟悉 jQuery，尤其是 jQuery 选择器，对于开发有极大帮助。jQuery mobile 提供的 UI 组件，使用起来非常方便，若开发一个功能和用户交互比较简单、对用户界面要求不多的应用，使用这些 UI 组件，开发可以非常的快速，但由于其样式和交互的局限性，往往不能适应移动互联网产品的需求（UIUE 的设计人员可能会设计出很美观的界面和很炫目的交互），所以这些 UI 组件使用的不多。关于 jQuery 和 jQuery mobile 的 UI 组件，请参考文档，本文不会再讨论，下文会涉及到 jQuery mobile 管理页面之间的导航的内容。

除了使用以上两个开源框架外，我们也提供了一些框架和组件，这些是我们在开发 App 或者支撑开发的过程中总结提炼出来的。具体到每个组件，我们都进行了封装，有的以类的方式提供，有的以函数方式提供，有的以 html 片段的方式提供，不同的方式有着不同的扩展性，宗旨是尽量方便开发人员调用并且提供方法满足未知的需求。有的组件以 js+css 文件提供，有的被包含在 mobile-commom.js 中，有的需要开发人员拷贝 demo 中的代码段。每个组件都尽量提供 demo 和使用说明文档，所以每个组件如何使用，本文不再赘述，请参考 demo 和使用说明文档。本文重点讨论如何组合这些组件，开发完整的功能页面。

第一个页面

1. 先搭建页面框架代码，如下图所示（为了文档的美观和可读性，代码以图片方式贴出，若需要复制代码，请从 demo 中复制）。

```
<html>
<head>
    <meta name="viewport" content="width=device-width"/>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />

    <!-- 加载jQuery及jQuey mobile -->
    <link rel="stylesheet" type="text/css" href="../resources/jquery_mobile_1.0b1/jquery.mobile-1.0b1.min.css" />
    <script type="text/javascript" src="../resources/jquery-1.6.1.min.js"></script>
    <script>
        $(document).bind("mobileinit", function(){
            $.mobile.defaultPageTransition="none"; //关闭默认的页面切换动画
            $.mobile.loadingMessage=false; //关闭默认的加载提示框
        });
    </script>
    <script type="text/javascript" src="../resources/jquery_mobile_1.0b1/jquery.mobile-1.0b1.min.js"></script>

    <!-- 加载框架 -->
    <link rel="stylesheet" type="text/css" href="../resources/mobile-common.css" />
    <script type="text/javascript" src="../resources/mobile-common.js"></script>

    <!-- 加载框架 -->
    <script type="text/javascript" src="../resources/cordova-2.0.0.js"></script>
    <script type="text/javascript" src="../resources/jsict_extend.js"></script>
</head>
<body>
    <div data-role="page" id="firstPage">

    </div>
</body>
</html>
```

解释：和普通 html 页面一样，<html>、<head>和<body>标签都是必不可少的。

2. <head>标签中先引入两个<meta>标签：

```
<meta name="viewport" content="width=device-width"/>
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
```

解释：这两个<meta>标签分别控制 viewport 的尺寸和页面的编码。

3. 然后引入 jQuery 和 jQuery mobile 框架，要注意引入的顺序：

```
<link rel="stylesheet" type="text/css" href="../resources/jquery_mobile_1.0b1/jquery.mobile-1.0b1.min.css" />
<script type="text/javascript" src="../resources/jquery-1.6.1.min.js"></script>
<script>
    $(document).bind("mobileinit", function(){
        $.mobile.defaultPageTransition="none"; //关闭默认的页面切换动画
        $.mobile.loadingMessage=false; //关闭默认的加载提示框
    });
</script>
<script type="text/javascript" src="../resources/jquery_mobile_1.0b1/jquery.mobile-1.0b1.min.js"></script>
```

解释：若要对 jQuery mobile 进行配置，需要在引入 jQuery 之后，引入 jQuery mobile 之前进行配置，更多的配置项请参考手册。

4. 接着需要引入技术中心提供的框架代码：

```
<!-- 加载框架 -->
<link rel="stylesheet" type="text/css" href="../resources/mobile-common.css" />
<script type="text/javascript" src="../resources/mobile-common.js"></script>

<!-- 加载框架 -->
<script type="text/javascript" src="../resources/cordova-2.0.0.js"></script>
<script type="text/javascript" src="../resources/jsict extend.js"></script>
```

解释：mobile-common.css 和 mobile-common.js 提供页面开发中常用的组件、方法等，cordova.js（phonegap.js）和 jsict_extend.js 提供调用本地功能的接口。

5. 最后引入项目中需要的其他的 css 和 js 文件以及你自己的 javascript 代码：

```
<link rel="stylesheet" type="text/css" href="../resources/dynamicPic.css" />
<script type="text/javascript" src="../resources/dynamicPic.js"></script>

<script>
    function doMyJob(){
        alert("hello");
    }
</script>
```

6. 在<body>标签中，我们需要构建一个 ，其属性 data-role="page"，如下：

```
<div data-role="page" id="firstPage">

</div>
```

解释：jQuery mobile 以这样的 来标识一个页面单元。虽然可以在一个 html 文件中添加多个 ，但我们建议一个 html 文件对应一个 。除了为实现特殊的效果，其他所有的 html 代码都写在这样的 中。

接下来你可能会通过 javascript 来完成各种功能，这些 js 代码可以写在上面第 5 项所指示的位置。但是在<script>标签之间的语句，函数定义（不执行函数体），事件绑定等，都是在 载入前执行的。当然你也可以在 ready 和 loaded 事件响应中执行语句，但仍然满足不了一些需求，并且 ready 事件和 loaded 事件只有在第一个 html 加载时会被触发一次，将来跳转到其他的 html 时 ready 和 loaded 事件不会再被触发。jQuery mobile 为 扩展了四个事件：pagebeforecreate、pagecreate、pagebeforeshow 和 pageshow。可以用如下代码来绑定事件：

```
$("#firstPage").live("pageshow",function(){

});
```

要解释着四个事件，就不得不说一下 jQuery mobile 的页面管理机制。jQuery mobile 将 代码片段载入到 DOM 树后，需要这个 进行配置和渲染，配置和渲染前触发 pagebeforecreate 事件，配置和渲染完成后触发 pagecreate 事件，然后要显示这个 ，显示前触发 pagebeforeshow 事件，显示完成后触发 pageshow 事件。pagebeforecreate 和 pagecreate 事件只会被触发一次，而 pagebeforeshow 和 pageshow 事件会被触发多次。这一点很重要，下面来解释。当加载第一个 html 时，浏览器会构建一棵 DOM 树，第一个页面中 会依次触发上述四个事件。如果使用超链接或者 $.mobile.changePage 方法跳转到其他 html，不会清空当前 DOM 树，而是从目标 html 中截取 片段添加到当前 DOM 树，然后再依次触发上述四个事件，以后的情况都是如此。所以 DOM 树会不断增大，所有之前定义的变量、函数、样式等都会一直存在。如果目标 html 之前已经加载过（回退到上一个页面也属于这种情况），那不会再去截取目标 html 中的 ，而是直接将 DOM 树中 显示出来即可，由于之前已经配置和渲染过，所以不会触发 pagebeforecreate 和 pagecreate，但仍然会触发 pagebeforeshow 和 pageshow 事件。

在第一个页面中常常需要读写本地系统配置等，这就需要调用本地功能接口。所有的本地功能接口调用都必须在 deviceready 事件触发后，所以请看下面代码：

```
document.addEventListener("deviceready",onDeviceReady,false);
```

本地功能接口调用必须写在 onDeviceReady 函数及之后执行的函数中。

第一个页面通常进行初始化的工作，这些工作需要耗费一些时间，所以页面常常只呈现一张精美图片和一个加载提示。待初始化完成后，跳转到下一个页面即可。代码非常简单：

```
<img src="../resources/images/loading.jpg" style="position:absolute;width:100%;height:100%;">
<img src="../resources/images/loading2.gif" style="position:absolute;width:10%;left:45%;top:15%;">
```

效果图如下：

功能页面

接下来，我们需要将产品的功能展示给用户了。和第一个页面一样，先要搭建页面框架代码，如下图所示：

```
<html>
<head>
</head>
<body>
    <div data-role="page" id="mainPage">
        <script>
            $("#mainPage").live("pageshow",function(){

            })
        </script>
    </div>
</body>
</html>
```

看出有什么不同了吗？<head>标签中没有引用任何 js 和 css 文件。前面我们介绍过，从一个

页面跳转到另一个页面时，DOM 树不会被替换，而是截取目标页面中的 片段添加到当前 DOM 树。所以 之外的任何代码都是没有意义的，这里<head>标签中即使引用 js 和 css 文件也是无效的。所有的 js、css 文件引用尽量放在第一个页面或者 中。需要再次强调的是，DOM 不会被清空，之前页面定义的所有 js 全局变量、函数、css 样式甚至是 的 id，都会被保留，所以请不要重复定义全局变量、函数、样式和 id。

以后的工作，如果用一句话概括就是：根据设计需求，完成编码。

现在我们得到一张设计图，需要我们将首页面做成下面的样子：

设计将页面分成了三个部分：头部标题、底部导航栏、中间的内容部分（之所以将滚动图片和列表都归为中间部分，是因为设计需要可以滚动中间部分的内容）。大部分的页面也都是这么划分的。因此我们在 中添加以下代码：

```
<div data-role="page" id="mainPage">
    <script>
        $("#mainPage").live("pageshow",function(){

        })
    </script>

    <div class="page_header">

    </div>

    <div class="page_content">

    </div>

    <div class="page_footer">

    </div>
</div>
```

在 添加标题，按钮等：

```
<div class="page_header">
    <div class="page_header_title">天翼3G</div>
    <img class="left_img" src="../resources/images/main_setting.png">
    <div class="right_button">搜索</div>
</div>
```

按钮有 left_img、right_img、left_button 和 right_button 四种可以选择，用以确定位置。效果图如下：

在 添加导航条，由于导航条设计比较多样，难以对各种样式、各种交互进行统一封装，所以我们只提供最简的 html 片段，以方便开发人员进行任意的修改。最简单的导航条代码如下：

```
<div class="page_footer">
    <div class="jszx-bar-icon-up jszx-bar-item-4" id="mainFooter">
        <div><img src="../resources/images/icon1.png"><span class="jszx-bar-text">精品推荐</span></div>
        <div><img src="../resources/images/icon2.png"><span class="jszx-bar-text">应用分类</span></div>
        <div><img src="../resources/images/icon3.png"><span class="jszx-bar-text">数字企业</span></div>
        <div><img src="../resources/images/icon4.png"><span class="jszx-bar-text">搜索</span></div>
    </div>
</div>
```

jszx-bar-icon-up 表明每一项中的图片在文字上方，还有 jszx-bar-icon-left（图片在文字左边）和 jszx-bar-icon-none（没有图片）可以选择。jszx-bar-item-4 表明导航条中有 4 项，可以选择的范围是 1~5。效果图如下：

通常选中的某个导航按钮应该区别与其他项，我们默认为第一项添加选中样式，代码如下：

```
<div class="page_footer">
    <div class="jszx-bar-icon-up jszx-bar-item-4" id="mainFooter">
        <div class="jszx-bar-item-active"><img src="../resources/images/icon1.png"><span class="jszx-bar-text">精品推荐</span></div>
        <div><img src="../resources/images/icon2.png"><span class="jszx-bar-text">应用分类</span></div>
        <div><img src="../resources/images/icon3.png"><span class="jszx-bar-text">数字企业</span></div>
        <div><img src="../resources/images/icon4.png"><span class="jszx-bar-text">搜索</span></div>
    </div>
</div>
```

效果图如下：

有时你可能需要每项之间有分割线，代码如下：

```
<div class="jszx-bar-icon-up jszx-bar-item-4" id="mainFooter">
    <div><img src="../resources/images/icon1.png"><span class="jszx-bar-text">精品推荐</span><div class="jszx-bar-split"></div></div>
    <div><img src="../resources/images/icon2.png"><span class="jszx-bar-text">应用分类</span><div class="jszx-bar-split"></div></div>
    <div><img src="../resources/images/icon3.png"><span class="jszx-bar-text">数字企业</span><div class="jszx-bar-split"></div></div>
    <div><img src="../resources/images/icon4.png"><span class="jszx-bar-text">搜索</span></div>
</div>
```

效果图如下：

接着你需要为导航条响应点击事件，代码如下：

```
$("#mainFooter>div").addEventListener("quickClick",function(e){
    $("#mainFooter>div").removeClass("jszx-bar-item-active");
    $(e.currentTarget).addClass("jszx-bar-item-active");

    alert($(e.currentTarget).find(".jszx-bar-text").text());
});
```

代码很简单，去除已有的 jszx-bar-item-active 样式，为当前被点击项添加 jszx-bar-item-active 样式，告诉用户当前点击的是哪一项。

中的内容需要滚动，所以添加如下 html 片段：

```
<div class="page_content">
    <div class="jszx-wrapper" id="mainWrapper"><div class="jszx-scroller" id="mainScroller">

    </div></div>
</div>
```

在对滚动组件进行初始化，由于初始化工作只需要做一次，可以放在用全局变量进行控制。代码如下：

```
if(!window.mainInit){
    window.mainInit=true;

    $("#mainWrapper")[0].style.height=$("#mainPage .page_content").height()+"px";
    initScroll({
        "wrapper":"mainWrapper",
        "dir":"y",
        "bounce":true
    });
}
```

也许你会问为什么不将这段代码放在 pagecreate 事件中，这样不需要全局变量，也可以保证

初始化只进行一次。因为在页面未显示时，.height()方法取值为 0，所以必须在 pageshow 以后才可以进行初始化。

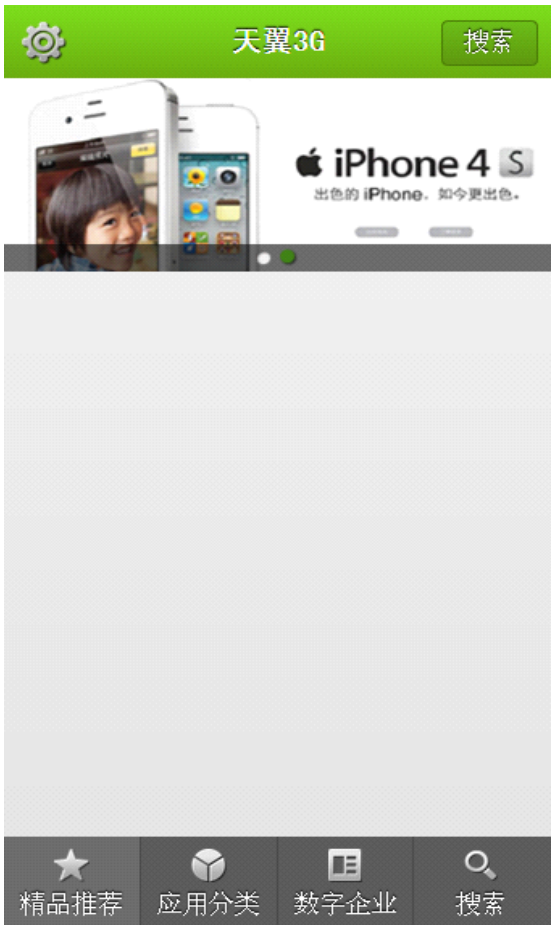
先添加动态图册组件（焦点图、广告等），在页面中添加 容器：

```
<div class="page_content">
    <div class="jszx-wrapper" id="mainWrapper"><div class="jszx-scroller" id="mainScroller">
        <div id="dynamicPic1"></div>

    </div></div>
</div>
```

初始化代码同样由全局变量控制，只运行一次，如下：

```
var pics=["../resources/images/j1.jpg","../resources/images/j2.jpg"];
var markers=["../resources/images/d_gray.png","../resources/images/d_green.png"]
var dp=new DynamicPic({
    "id": "dynamicPic1", //id，必须
    "pics": pics, //图片URL的数组，必须
    "markers": markers, //标识URL的数组，元素0为非当前图片标识，元素1为当前图片标识，默认UR
    "width": window.innerWidth, //默认为window.innerWidth
    "height": window.innerWidth*3/8, //默认为window.innerWidth*9/16
    "timer": "3000", //默认为5000
    "onClick": function(index){
        alert("这是第"+(index+1)+"张图片");
    },
    "onChange": function(index){
        //alert("这是第"+(index+1)+"张图片");
    }
});
```



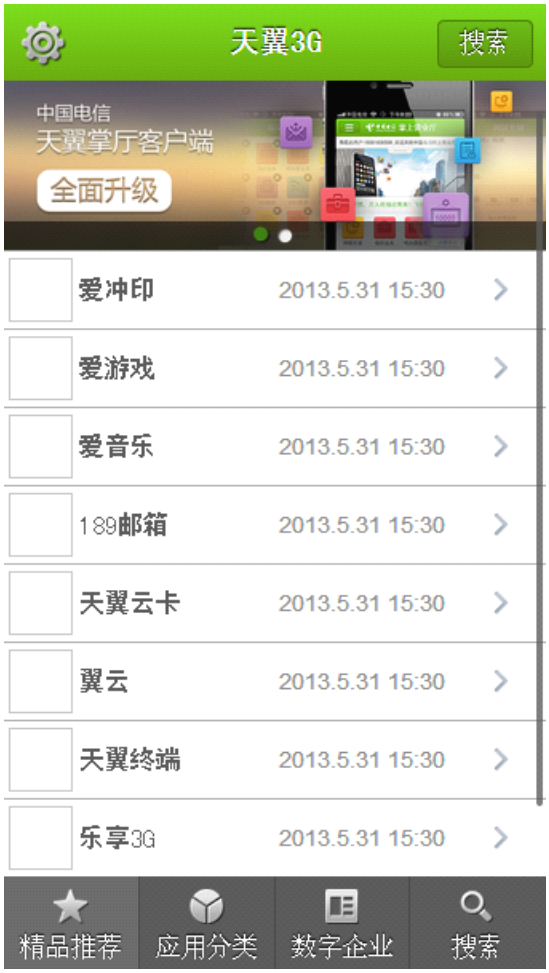
最后添加列表，同样需要在页面中添加 容器：

```
<div class="jszx-wrapper" id="mainWrapper"><div class="jszx-scroller" id="mainScroller">
    <div id="dynamicPic1"></div>
    <div id="mainList"></div>
</div></div>
```

创建列表行需要使用 createListRow 方法，代码如下：

```
var data=["爱冲印","爱游戏","爱音乐","189邮箱","天翼云卡","翼云","天翼终端","乐享3G","高速互联"];
for(var i=0;i<data.length;i++){
    var html="<table width='100%'><tr>"
            +   "<td width='15%'><img src='' style='height:2.5em;width:2.5em;'></td>"
            +   "<td width='45%'><div style='font-family:黑体;color:#555;'>"+data[i]+"</div></td>"
            +   "<td width='40%'><div style='font-size:0.85em;color:#888;'>2013.5.31 15:30</div></td>"
            +"</tr></table>";
    var $row=createListRow(html,"arrow-r");
    $row.appendTo($("#mainList"));
}
```

假设 data 是通过 ajax 从服务器获取来的数据，列表行需要循环读取 data 数据并且调用 createListRow 的方法来创建。你可以将每条数据中的不同字段（demo 中 data 是一维数组，实际情况下，data 可能是个对象数组或多维数组，每条数据有多个字段），利用<table>或者 进行排版，添加适合的样式 style 或者 class。效果图如下：



这样，首页面就完成了。可以看到整个开发过程还是比较简单的，由于页面元素都是已有的组件，所以只需要组合这些组件就可以了。每一个组件都有自己的 demo 和使用手册，个性化的修改可以参照这些 demo 和使用手册。我们已经封装了部分常用的组件供各个项目

使用。将来也会继续扩充我们的组件库。

总结一下，读到这里，你应该了解：

- ✓ native code+web HTML 技术架构来开发 app 是怎么一回事；
- ✓ web HTML 开发使用了哪些框架；
- ✓ jQuery mobile 是怎么一回事；
- ✓ 如何搭建页面框架；
- ✓ html 和 javascript 代码应该写在什么地方；
- ✓ 如何利用几个组件来“组装”成一个页面；

如果你对以上提到的几个问题还有疑问，你可能还需要仔细阅读相关文档和以上的内容。如果你觉得已经入门或者可以上手开发了，那继续来完善首页面吧，因为之前只是按照设计“画”出了这个页面，它离实际可运行还有以下几个工作：

1. 为顶部标题栏按钮添加事件响应；
2. 修改底部导航样式，使它更加美观；
3. 完善底部导航事件响应函数，点击每一项都可以切换到正确的页面；
4. 向服务器获取真实的列表数据，然后排版、美化、添加事件响应等；
5. 向服务器获取真实的焦点图数据；
6. 也许你的首页面会有更加复杂的交互，将它们添加进来；

完成了以上的步骤，首页面的开发才能基本完成。

关于调试

上面开发完成的 html 页面可以在手机上运行，但为了调试方便，开发的过程中我们也常常用 PC 浏览器直接打开 html 页面运行调试。使用 PC 浏览器调试有以下几个注意点：

- ✓ 如果页面中调用了手机本地功能接口，无法在 PC 浏览器上运行；
- ✓ 使用 webkit 内核浏览器调试，如 chrome 或者 safari，不要用 IE 调试；
- ✓ chrome 的开发者工具非常强大，对于提高开发效率，解决问题大有帮助；
- ✓ 如果用 chrome 进行调试，由于 chrome 本身的安全机制，$.mobile.changePage 方法和超链接跳转页面都不起作用。解决方法：新建 chrome 快捷方式=》右击=》选择属性=》在“目标（T）”中的内容修改成"C:\Documents and Settings\Administrator\Local Settings\Application Data\Google\Chrome\Application\chrome.exe" --allow-file-access-from-files=》确认=》打开此快捷方式，拖入页面即可；