

第二章 应用程序基础知识

第二章主要讲述了如何创建 webOS 的应用程序，由于 SDK 还没法申请到，我们只能照本宣科的讲述一下这个章节涉及的内容及我们的理解。

webOS 上面的编程模型与普通 web 应用程序有一些不一样的地方就是没有请求/相应的生命周期，几乎所有的代码都是 Javascript 编写的，界面是用 HTML 写的，样式则是 CSS，因而以前做 web 开发的编程人员很容易转型到 webOS 的开发上面，包括开发工具也是类似的，SDK 发布的时候会带一个基于 Eclipse 的 IDE。

创建应用程序

由于 webOS 的程序有一系列约定的目录和文件，所以要用 SDK 带的命令行工具来创建程序，命令为：`C:\Palm\palm-generate AppName`

命令创建完应用程序之后我们可以看看其目录结构，每个 webOS 程序的目录结构都跟这个类似，如下：

```
AppName
-> app
  -> assistants
    -> first-assistant.js
    -> second-assistant.js
    -> ...
  -> models
    -> model1.js
    -> model2.js
    -> ...
  -> views
    -> first
      -> first-scene.html
    -> second
      -> second-scene.html
    -> ...

-> appinfo.json
-> icon.png
-> images
  -> image1_
  -> image2_
  -> ...
-> index.html
-> sources.json
-> stylesheets
  -> AppName.css
  -> ...
```

以上的 AppName 我们可以根据实际来命名，注意 appinfo.json 文件里面的 ID 值和它一致就可以了。

前面我们知道 webOS 是基于 MVC 的模式来设计的：

- assistants 目录就是我们前面第一章提到过的场景助手（scene assistants），主要是提供给框架的控制器用来自定义这个程序的行为，也就是控制器的角色。
- models 目录用于存放应用程序的数据模型，根据 MVC 的模式，如果我们没什么数据访问那就可以不写模型了。
- views 就是用于放的布局，模版，页面等等。

我们温习一下在第一章提到过的一些内容：

- 目录 images 存放图片
- 目录 stylesheets 存放 css 文件
- icon.png 是应用程序图标，具体格式为 64*64 点阵 24 位 RGB 或 8 位 alpha 的 PNG 编码
- Index.html 初始页面，此处不多说，后面详细探讨
- Sources.json 用于延迟加载 javascripts 源码的文件
- Appinfo.json 这是用于加载应用程序的一些参数配置，格式如下：

```
{
  "title": "News",
  "type": "web",
  "main": "index.html",
  "id": "com.palm.app.news",
  "version": "1.0",
  "icon": "icon.png"
}
```

具体含义如下表：

属性	值	必填	描述
title	任何值	是	应用程序名称
type	web	是	普通应用程序
main	index.html	是	应用程序的起始页面，缺省是index.html
id	任何值	是	每个程序必须唯一
version	x.y	是	应用程序版本号，数字的
noWindow	true/false	否	是否无标题程序，缺省是false
icon	文件路径	否	图标的路径，缺省是icon.png
minicon	文件路径	否	在通知栏显示的图标，缺省是minicon.png
category	任何值	否	缺省的编目名称

Index.html 文件

这个文件是程序启动时候第一个加载的，因而要把这段框架的代码放在前面：

```
<script src="/usr/palm/frameworks/mojo/mojo.js"
type="text/javascript" x-mojo-version="1">
</script>
```

Sources.json 文件

我们可以在 index 文件里面把脚本文件都加载进来，但是为了提高性能，我们不建议这样做，正确的做法应该是在 Sources.json 文件中进行延迟加载。

自动生成的模版文件如下，第一个是助手（assistant）文件，第二个是场景（scene）文件：

```
[
{
  "source": "app\assistants\app-assistant.js"
},
{
  "source": "app\assistants\stage-assistant.js"
},
{
  "source": "app\assistants\first-assistant.js",
  "scenes": "first"
},
{
  "source": "app\assistants\second-assistant.js",
  "scenes": "second"
},
]
```

运行调试

象这个那么简单的程序其实是可以直接用浏览器来看的，但是如果你用到了 Mojo 的服务或者界面组件，那就要用仿真器了。仿真器和脚本调试器都集成在 IDE 里面，直接在 Bundles -->Mojo -->Run 菜单运行即可。

至此，完成了第一个程序的创建，第二章还讲述了一个复杂的程序的构造过程（程序名为

News，如下图)，后面将继续介绍。

