

# KaiOS业务简易梳理

---

熟悉业务,夯实基础

## 一.KaiOS系统整体架构

---

### 1.Gaia层

用户接口层,屏幕启动->屏幕渲染均为Gaia层的实现,负责标准应用app的开启与关闭,第三方应用均安装在Gaia层,目前手机端所有apps均是Gaia层的东西

#### 实现方式

完全使用HTML,CSS和JavaScript实现,WEB APIs有Gecko层实现,是Gaia层到底层系统的唯一入口

### 2.Gecko层

应用的运行时环境,提供了对HTML,CSS和JavaScript的支持,确保APIs能够在Gecko支持的系统上良好工作,我的理解是实现了像提供高版本浏览器或者是nodejs这种能使web标准代码运行的基本环境

### 3.Gonk层

更底层,应用到了硬件抽象层部分,暂时不去了解

---

不同的应用采用OOP(Out-Of-Process)多进程运行模型,每个应用都是一个独立的进程,为了保证不同app的独立性与安全性,并且规定不同的进程有不同的权限.在内存的管理上,KaiOS使用和Android相同的技术,如果用户正在操作某个进程,这个进程将超过其他应用程序的内存分配优先级,而那些在后台运行的应用将有可能被系统关闭

## 二.应用打开关闭流程

---

System见听到backspace时间,会关闭当前应用,

调用window.close()主动关闭

终端关闭app:

```
ps -ef | grep Messages // 获取进程PID
adb shell kill -9 PID.
```

## 三.应用manifest

---

### 1.作用(manifest.webapp)

1. 对应用的描述和说明

2. 需要请求的权限
3. 处理其他应用调用其资源的activity请求
4. etc.

## 2.结构

采用JSON数据结构(键值对的形式),描述app的name,icon,locale,permissions等相关信息

### 属性

1. 强制属性
2. 可选属性

## 四.应用js和css文件引用和加载

---

### 1.js,css直接加载与调用

**加载:**应用启动时,index.html中包含的js,css文件就会被加载

**调用:**监听到load事件时,执行初始化操作,或者页面翻译机制准备好以后,在navigator.mozL10n.once的回调函数中启用初始化操作,执行应用代码,一些js代码是自执行,若js文件夹在则立即执行

### 2.延迟加载

#### 作用

提高app运行速度,加快html页面的渲染速度,增强用户体验

在sms app中,在index.html文件中,只引入延迟加载的公共库文件lazy\_loader.js,在startup.js文件中,定义一个延迟加载js及css文件的数组,并使用lazy\_loader加载需要的文件即可

## 五.应用事件处理

---

### 1.事件监听

监听:target.addEventListener(type, listener, options);

type:监听的事件类型

listener:所监听的事件类型触发时,调用listener,这个必须是一个实现了EventListener接口的对象,或者是一个函数

options: (可选)

### 2.事件移除

移除: target.removeEventListener(type, listener[, useCapture])

type:表示需要移除的事件类型

listener:需要移除的 EventListener 函数 (先前使用 addEventListener 方法定义的)

useCapture:(可选)

指定需要移除的 EventListener 函数是否为事件捕获。如果无此参数，默认值为 false。

如果同一个监听事件分别为“事件捕获”和“事件冒泡”注册了一次，一共两次，这两次事件需要分别移除。两者不会互相干扰。

### 3.自定义事件

应用内部可以使用时间构造函数来创建事件,并将事件分发出去,其他地方监听该事件,做出相应处理

```
var event = new Event('build');  
// Listen for the event.  
elem.addEventListener('build', function (e) { ... }, false);  
// Dispatch the event.  
elem.dispatchEvent(event);
```

## 结语

---

暂结于此