**成都羿明数字技术有限公司**

**产品开发手册**

目录

[文档历史 2](#_Toc461543673)

[1:概述 2](#_Toc461543674)

[2:准则 3](#_Toc461543675)

[3:结构 4](#_Toc461543676)

[1) 项目结构 4](#_Toc461543677)

[2) 代码结构： 4](#_Toc461543678)

[3) 插件开发 4](#_Toc461543679)

[4:规范 16](#_Toc461543680)

# 文档历史

|  |  |  |  |
| --- | --- | --- | --- |
| 作者 | 时间 | 版本 | 备注 |
| Samson.zeng  (曾夏新) | 2016/9/13 | V0.1 | 初稿，完成项目source结构,Web前端与Android原生关系说明,插件开发说明，编码规范说明 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 1:概述

本文用于为产品开发人员制定移动端(Android )开发准则及相关公用组件、示例代码

# 2:准则

1. 项目基于JDK1.7。
2. 项目开发IDE使用Android Studio 2.0及以上版本。
3. 产品开发分为 Web层(web前端的HTML+JavaScript+CSS)和Android原生层.两者通过EminBridge实现交互.具体功能通过开发插件类(Plugin)实现。
4. Web层的代码与iOS移动端的Web层是同一套代码。
5. Android开发人员的JAVA编码规范参照Emin产品中的Service开发规范。

# 3:结构

## 3.1项目结构：

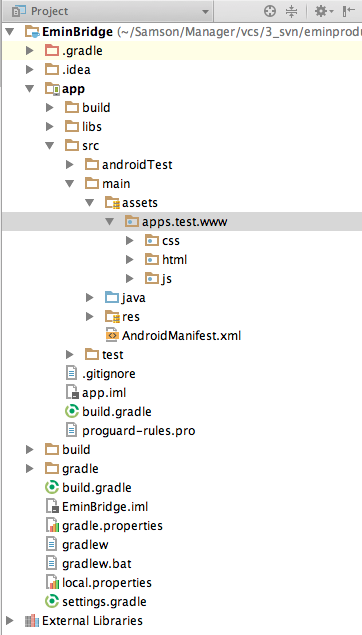
项目分为Web层和Android原生层。

Web层是web前端开发的webapp,包括Html,JavaScript,CSS等, 作为静态资源拷贝到Android项目中(src/main/assets目录下)。为方便调试,该部分可独立于Android Studio开发,如采用HBuilder等前端开发工具开发。

Android原生层为Java部分代码。

## 3.2代码结构：

产品代码结构图：



Web层的代码储存在src/main/assets 目录下,该目录是android项目自动创建。

一个新的项目,需要在该目录下新建apps/<appname>/www 目录,其中appname为webapp的项目名称,html,css,js,img等webapp资源则放在www目录下。

Java原生层的代码则在java目录下。

## 3.3 插件开发：

插件的开发除了开发Java插件类,还需要提供该插件的JavaScript文件。该js是供Web 层调用Android原生层的桥梁。即：Web前端通过调用该js的function实现对Java插件类的调用。

例如:开发Camera插件,则需要PluginCamera.java和pluginCamera.js。该js文件供Web前端引入。

# 4:规范

## 4.1 Android项目规范

包名命名规范：com.emin.<appname>.mobile.xxx

appname 替换为相应的应用名称 例如：meris

xxx可以为plugin、util、等

插件开发说明: 每一个插件对应一个Java插件类和一个JavaScript文件

JAVA部分的插件代码规范

命名规范:PluginXxx,例如PluginCamera，PluginDatabse

Plugin类中的功能方法,有且仅有一个参数,PluginParams

例如:PluginDatabase往数据表中插入数据方法:

**public void** insert(PluginParams params)

其中PluginParams封装了JS传递的参数列表,params.getArguments()返回了该参数数组,通过该数组的index来获取相应的参数。0表示js中function传递进来的第一个参数,1表示第2个,以此类推。

例如JS方法中传递了,insert的sql语句,则通过String sqlString = params.getArguments()[0]来获得该sql语句的字符串。

示例：

**public void** insert(PluginParams params){  
 Context context = params.**webView**.getContext();  
 String sqlString = params.getArguments()[0];  
 DatabaseManager.*getInstance*(context).execSQL(sqlString);  
}

JavaScript插件代码规范

文件命名规范:pluginXxx.js,例如pluginBarcode.js,pluginDatabase.js。首字母小写。

EminBridge是框架封装的对象。Web层调用原生层都得通过该对象。EminBridge下的插件对象统一使用pluginXxx,首字母小写

示例：

;(**function**(w){  
 **var** dbPluginName = **"com.emin.mobile.meris.plugin.PluginDatabase"**;  
 **var** dbPlugin = {  
 initDatabase : **function**(tables){  
 **var** methodName = **"initDatabase"**;  
 **return** EminBridge.execSyncPlugin(dbPluginName,methodName,[tables]);  
 },  
 insert : **function**(sql){  
 **var** methodName = **"insert"**;  
 **return** EminBridge.execSyncPlugin(dbPluginName,methodName,[sql]);  
 },  
 deleteWithSql: **function**(sql){  
 **var** methodName = **"executeSql"**;  
 EminBridge.execSyncPlugin(dbPluginName,methodName,[sql]);  
 },  
 updateWithSql: **function**(sql){  
 **var** methodName = **"executeSql"**;  
 EminBridge.execSyncPlugin(dbPluginName,methodName,[sql]);  
 },   
 queryWithSql : **function**(sql){  
 **var** methodName = **"queryWithSql"**;  
 **return** EminBridge.execSyncPlugin(dbPluginName,methodName,[sql]);  
 }  
 };  
 w.EminBridge.pluginDb = dbPlugin;  
})(window);

Web前端引入该js文件,通过EminBridge.pluginDb.insert (“insert into tbl\_user values(1,’emin’)”);

调用了Android的插件类PluginDatabase中的insert方法