目录

[1. 项目概述 1](#_Toc419709105)

[1.1基本概述 1](#_Toc419709106)

[1.2编码规范和命名规则 1](#_Toc419709107)

[2. 项目结构 1](#_Toc419709108)

[2.1 Cordova 项目结构 1](#_Toc419709109)

[2.2 Android项目结构 2](#_Toc419709110)

[2.3 Web项目结构 3](#_Toc419709111)

[3. 应用承载页面index.hml 3](#_Toc419709112)

[4. 应用主文件app.js 4](#_Toc419709113)

[5. 应用服务service.js 6](#_Toc419709114)

[6. 应用模块module\_\*.js 7](#_Toc419709115)

# 项目概述

## 1.1基本概述

移动项目基于Cordova构建，使用Onsenui作为前端框架，使用Angularjs处理界面交互样式上使用了BootStrap。

该项目是单页面应用（SPA），所有数据交互通过json进行，由第一个页面承载，即index.html，该页面会将应用中所有要用到的资源一次性载入，之后不会再请求js和css文件。页面的切换也是通过替换div来实现的，而页面的模板放在本地，但需要通过htpp请求，所以运行项目需要服务器环境。

由于BootStrap和Onsenui在样式上有冲突，因此，做了如下调整。

1.屏蔽了Onsenui的网格布局样式

2.屏蔽了BootStrap的popover样式

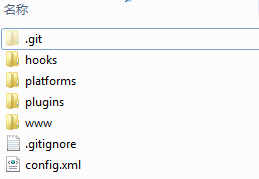
## 1.2编码规范和命名规则

# 项目结构

我们的移动项目是基于Cordova构建的，最顶层是Cordova的项目即由Cordova create 出来的，接着由Cordova build出Android项目，而我们的应用逻辑基本上都集中在Web项目中。

## Cordova 项目结构

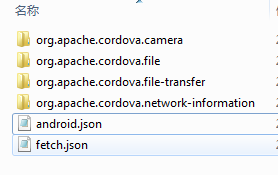
在命令行中执行 cordova create <项目名称> ，创建出来的项目。



Android项目在platforms目录下的android

Web项目在www

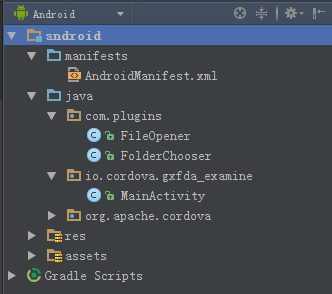
Plugins里面存放着cordova官方提供的插件，以及插件配置, 自己写的插件，记得在android.json中加入配置。



config.xml是cordova项目的基本配置，包括项目包名，应用表现（比如是否可以横屏，是否满屏，即没有最顶上的wifi、电池之类的状态条）。

## Android项目结构

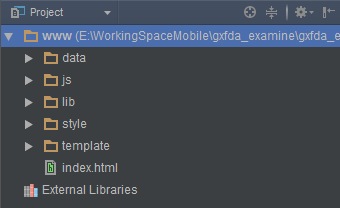
经过cordova platform add android，然后cordova build android之后创建在platform中的项目。这个项目是一个标准的android项目，和一般的android项目没有差异。



Java下的是源码，AndroidMainfest.xml是程序的主配置，com.plugins里面一般放置自己写的插件，MainActivity是程序的主入口。

## Web项目结构

Web项目是存放主要业务逻辑和界面的项目，Cordova build的时候会把该项目打包进Android项目的assets目录下。



data目录是存放离线数据的目录，存放离线数据是为了方便页面调试。

js目录存放我们写的js，包括各个应用模块的逻辑和通用服务等。

lib目录存放引用的框架和插件，比如onsenui和bootstrap等。

style目录存放我们写的页面样式以及字体（包括字体图标）。

template目录存放页面模板

index.html是应用的入口和承载页面。



Js目录下的app.js是应用的主文件

directive.js是指令文件，这个是angularjs的一个概念，可参考angularjs的文档

以module打头的就是系统模块文件，一个模块一个js，包括数据工厂和控制器。

Service.js是服务文件，是一些封装好的共用方法。

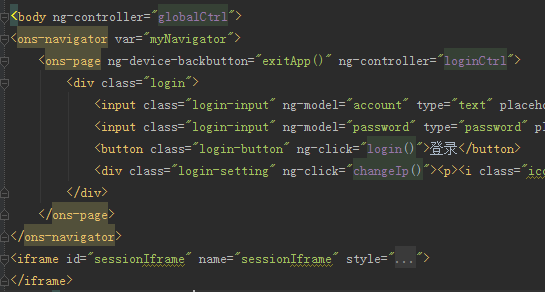
# 应用承载页面index.hml

应用启动后的默认页面，可以在cordova项目下的config.xml中修改。



首先ng-app必须要指定，它告诉Angularjs应用范围，默认叫app。

接着先把引用的js和css全部引入，再引入我们写的js和css。



<ons-navigator>是页面的总导航，var的值是导航的名字，将继承到它下面的所有$scope中，我们应用的页面都是在这个标签中的。而<ons-page>则包含着页面，只有这两个标签嵌套使用才能实现翻页，每个页面模板也都是在<ons-page>中。

在</ons-navigator>下面的iframe是用于让应用取到会话的。

如果看到ng- 或 ons- 这些就是angularjs和onsenui的东西，有看不懂的，可翻阅API

# 应用主文件app.js

这个文件是应用的主要文件，先引入这个文件再进入其他模块的js。只有这里添加angularjs的依赖，其他模块文件是.module(‘app’)，只有一个参数，如果传入第二数组会覆盖该文件的对象。



这里要的appConfig是一个全局的配置文件的作用，配置了默认的服务器地址，端口以及其他全局配置。



.config()是对angularjs的一些默认行为做设置，比如request header等

.run()是应用启动后的初始动作，比如从localstorage取出ip，读本地数据等。

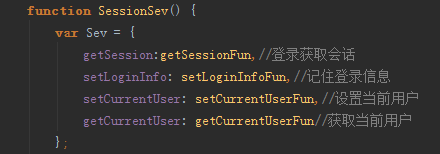
我在这里写了三个控制器，登录控制器(LoginCtrl)是比较重要的。

登录策略是这样的，先发一个ajax请求，传账户密码回服务器去拿回一个用户对象，如果能正确拿到用户对象，则让sessionIframe里面的form再提交一次让应用获得会话。

# 应用服务service.js



现在一共有三个service，封装了三种类型的服务



SessionSev和会话有关，主要是记住当前用户，记住登录成功后的账号密码以及获取会话。

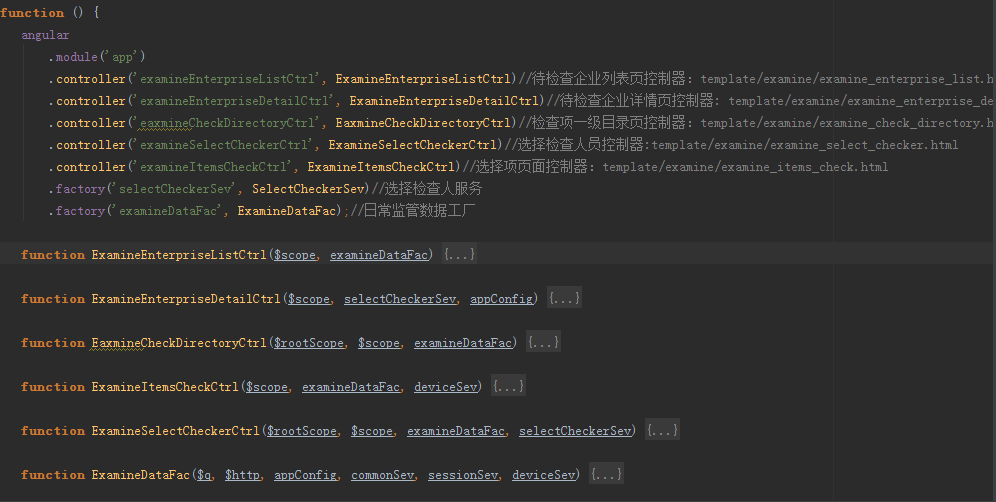


CommonSev是一般的通用方法服务，比如我们封装好的各种弹窗，计算md5码。

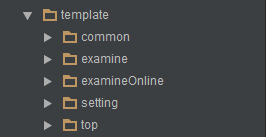


deivceSev是设备相关的服务，是最复杂的的service，它通过cordova反射机制调用设备接口，凡和设备有密切关系的方法都写在这个service中。

# 应用模块module\_\*.js



模块的js是应用的主体，是每个页面的业务逻辑。一个controller对应一个页面，没有特殊情况，不允许多个页面使用一个controller，控制器负责页面的跳转和交互。而数据工厂，负责处理数据和与服务端交互。



每个模块的页面放在tempalte对应的目录下面