Ionic框架开发移动 App——自定义登录加密插件

邓慧琴

（安徽电子信息职业技术学院 软件学院，安徽 蚌埠 233000）

摘要：移动 APP 开发是目前最流行的程序开发，很多以前做 Web 应用的公司或个人都在往移动 APP 开发转型。如何在已有的 Web 应用系统之上快速开发出 APP 产品，很多人会选择把前端页面改版成 Html5+CSS3，但是这种开发模式开发出的产品有很多缺点，比如：性能低下、兼容性差、不易维护等。针对这种情况很多移动端 Web 开发框架应运而生，Ionic 就是其中优秀的免费移动端开发框架。本文针对已有 Web 应用后台程序的情况，给出了应用 Ionic 框架直接调用已有的 JAVA 类的方法，从而简化 APP 开发。

关键词: 免费；Ionic；Java；快速 APP 开发；移动端 Web 应用

中图分类号：TP311.1 文献标志码：A 文章编号：1007- 984X(2017)01- 0009- 05

随着移动智能终端的广泛应用和 4G 网络的普及，移动 App 的应用范围越来越广泛，已经成为人们日常活动中不可缺少的一部分。移动 App 开发也已经成为程序开发人员学习研究的主要方向。移动 App 开发模式多种多样，想要做移动 App 开发，选择适合自己的开发模式至关重要。

1 移动 App 的开发模式

目前开发移动 App 有以下几种方式[1] 2：（1）原生/Native：使用原生 SDK 开发 App。优点运行流畅，有足够的开发成本的话，这是最理想的方式；缺点是对不同的平台要分别开发，需要 android 和 IOS 两个开发团队，学习成本高，开发成本高、开发周期长。（2）原生脚本/NativeScript：将原生 API 封装成 JavaScript 接口。NativeScript 方式与原生相比性能损失不大。优点是开发语言统一使用 JavaScript；缺点也是要针对不同的平台分别开发，开发成本高。（3）原生+Web/ Hybrid：主要功能使用原生技术开发，部分扩展功能调用 Web 页面。优点是比纯原生开发周期短，页面更新方便；缺点也是开发成本较高，调用 Web 页面会对 App 的性能和用户体验有很大影响。（4）混合/Hybrid：使用 Web 技术开发 App，使用 Cordova/PhoneGap 之类进行打包封装。优点是采用标准的 Web 技术开发，避免了不同平台原生开发体系的学习，学习成本低，上手快、效率高，一次开发多个平台全部搞定；缺点是在 android 平台性能上有一些损失。

分析完这些开发方式，不难发现，如果是资深的 Web 开发人员想转型做移动 App 开发，最好的选择还是 Hybrid 开发方式。

2 Ionic 开发框架简介

Ionic 开发框架[2]1，号称 Advanced HTML5 Hybrid Mobile App Framework 是 AngularJS 移动端解决方案，可以使用 Web 技术构建出接近原生体验的移动 App。Ionic 主要关注外观和体验，以及应用程序的 UI 交互，特别适合用于基于 Hybird 模式的 HTML5 移动应用程序开发。AngularJS[3]1 是 Google 公司推出的一款优秀的前端 JS 框架，已经被用于 Google 的多款产品当中。Angularjs 主要特性有整体框架符合 MVC 设计思想，模块化和依赖注入以及双向数据绑定等。

3 Ioinc 开发移动 App

3.1 移动 App 开发初体验

收稿日期：2016- 07- 11

基金项目：院级项目：《网页脚本设计》课程改革

作者简介：邓慧琴(1978-)，女，江西九江人，讲师，工程硕士，主要从事 Web 系统开发和 Android 应用开发，171849801@qq.com。

经过一段时间对 Ionic 开发框架的研究，笔者发现其实在 Ionic 框架里通过配置插件的方式可以调用 Java 类里的方法，这种混合开发方式，可以方便地把一些复杂的 Java 代码移植到 App 应用里，这样一来就可以大幅提高 App 运行效率，弥补 Hybrid 开发方式的弊端。

以系统登陆加密常用功能介绍一下在 Ionic 框架里如何调用已有 Java 类里的方法。

Ionic 开发 App 的效率确实惊人，在已有后台 Java 登陆系统的情况下，笔者只用了 1h 左右的时间就可以打包出一个简单 Android 版的 App 应用做登陆测试(此 App 应用其实就是在 Ionic 自带 demo 应用的基础上加入了一个登陆界面，打开应用前先路由到此登录界面)。登录界面如图 1。

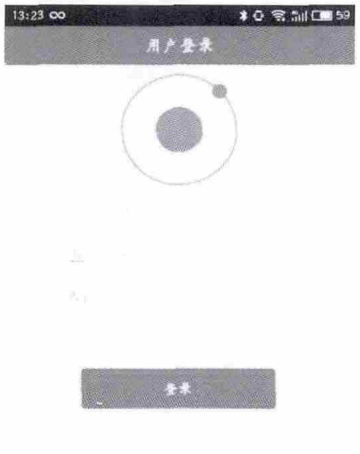

图 1 登录界面

3.2 不做加密的登录不安全

在没有做加密登录时，用户输入的用户名和密码是作为明文直接发送到后台 web 应用服务器，下图是用 Chrome 浏览器调试时截取到的请求信息。

图 2 请求信息

由此可以看出，这样发送用户登录请求是极其不安全的，所以通常都要对用户输入的个人信息做加密处理以后才提交请求给后台，后台再做解密处理来得到用户输入的真正信息。其实在做移动端 App 之前，通常都已经有做好后台服务端的 Web 应用程序，也就说现在已经有写好的处理加密解密的 Java 类，那么如何让 App 能直接使用这个 Java 类呢，这就本文要着重介绍的在 Ionic 框架里通过配置插件的方式来访问已有 Java 类的方法(本文的 Web 服务端使用的是 3DES 加密解密方式，对应的有一个 ThreeDesUtils.java 的类提供了 encrypt 加密方法和 decrypt 解密方法)。

3.3 使用已有的加密算法制作移动端加密插件

首先来看一下使用 Ionic 框架打包生成的 Android 应用的目录结构，如图 3。

这里主要看 3 个目录 www，plugins，platforms。其中 www 目录里是本文设计的展示给用户页面。

plugins 目录里是注册使用的插件，platforms 目录里存放的是最后打包好的 App 项目，这里打包的是 Android 应用，所以里面会生成一个 Android 目录。

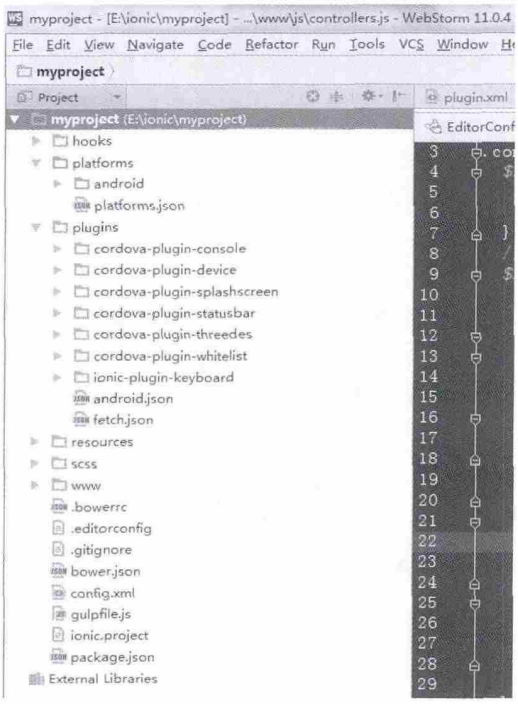

图 3 Android 目录结构

要配置一个 cordova 插件，首先在 plugins 目录里新增一个 cordova- plugin- threedes 目录，目录结构如图 4。

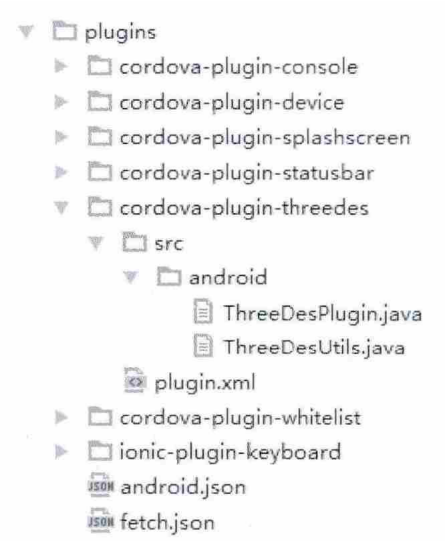

图 4　cordova- plugin- threedes 目录结构

手动新增一个名称为 plugin.xml 的配置文件，文件固定为此名称，因为用 Ionic 打包应用时，系统会自动扫描插件目录里所有 plugin.xml 文件，并进行插件的注册，plugin.xml 配置文件内容如下：

```
<?xml version="1.0" encoding="UTF- 8"?>
<plugin xmlns="http://apache.org/cordova/ns/plugins/1.0"
xmlns:android="http://schemas.android.com/apk/res/android"
id="3des"
version="1.0.0">
<name>3des</name>
<description>Cordova 3des Plugin</description>
<license>Apache 2.0</license>
<keywords>cordova,3des</keywords>
<engines>
<engine name="cordova" version=">=3.0.0" />
</engines>
<platform name="android">
<source- file src="src/android/ThreeDesUtils.java" target- dir="src/cn/edu/ahdy/utils" />
<source- file src="src/android/ThreeDesPlugin.java" target- dir="src/cn/edu/ahdy/utils" />
<config- file target="res/xml/config.xml" parent="/*">
<feature name="3des">
<param name="android- package" value="cn.edu.ahdy.utils.ThreeDesPlugin" />
</feature>
</config- file>
</platform>
</plugin>
```

根据内容可以看出，这里主要配置插件的名称，插件的引擎，使用的平台（只配置了 Android 平台，如果想要支持 IOS 平台，只需要在文件中再增加一个<platform name="ios">对应的配置即可），以及插件代码在平台代码中的生成路径，具体配置方式可以参考目录原有的其它插件。

然后需要在 plugs 目录下的 android.json 文件中再做一下配置，在文件中找到原有插件相对应的位置增加与原有插件类似的配置，具体如下代码：

```
"cordova- plugin- threedes": {
        "PACKAGE_NAME": "com.ionicframework.myproject101076"
}
```

最后就是编写处理加密的 Java 类，其中 ThreeDesUtils.java 就是后台 Web 应用的加密解密类，直接拷贝过来即可。ThreeDesPlugin.java 是需要在前端 AngularJS 中调用的插件类，它必须要继承于 Cordova 插件类来实现，而且必须还要实现 execute 方法，这样前端 AngularJS 才能调用到，ThreeDesPlugin.java 具体代码如下：

```
public class ThreeDesPlugin extends CordovaPlugin {
    /**
  * @param action 外部调用传入需要处理的方法名
  * @paramargs 外部调用传入的参数
  * @paramcallbackContext 回调函数传回处理结果
```

```
 */
@Override
 publicboolean execute(String action, JSONArrayargs,
            CallbackContextcallbackContext) throws JSONException {
        if ("encrypt".equals(action)) {
            JSONObject r = new JSONObject();
r.put("userName", ThreeDesUtils.desValue(args.getString(0)));
r.put("password",ThreeDesUtils.desValue(args.getString(1)));
callbackContext.success(r);
return true; }
return false;}}
```

以上代码其实很简单,就是把传入的用户名和密码做一次 3DES 加密,具体加密方法是调用已有的 java 类,然后拼装一个 json 对象放入 callbackContext 回调函数中返回给前端。

最后只需要在系统登录请求发送前,调用此插件传入用户输入的用户名和密码,取得加密的用户名和密码后,再发送请求去后台 Web 服务端验证登录信息即可实现 App 的加密登录。

下面是截取的部分代码:

AngularJS 控制器 controller.js 中提供的加密登录方法的代码如下:

```
 $scope.login = function(){//对用户登录信息的加密操作
varmyPopup = $dialog.showSuccessMesPopup("正在登录");
cordova.exec(function (result) {
        $data.login(result).success(function(data){
myPopup.close();
varerrorStr = data.errorStr;
if(errorStr ==''){
            $state.go("tab.dash");
}else{
            $dialog.showErrorMesPopup(errorStr);
          }
        }).error(function (data,status) {
myPopup.close();
          $dialog.showErrorMesPopup("登录异常");
        });
      }, function (error) {
myPopup.close();
        $dialog.showErrorMesPopup("登录异常");
      }, "3des", "encrypt", [$scope.user.userName,$scope.user.password]);
  };
```

AngularJS 服务类 services.js 提供发送登录请求的服务,代码如下:

```
.factory('$data', function ($http) {
  return {//发送登录请求
login: function (user) {
      varurl = config.basePath + "/app_login!appLogin.do?user.userName=" + user.userName + "&user.password=" +
                user.password;
url += config.suffix;
        return $http.jsonp(url); }}
```

代码中 cordova.exec 就是 cordova 提供调用插件的方法 ,方法的第 1 参数是插件处理完成后的回调函数，第 2 参数是 plugin.xml 中配置插件的名字，第 3 个参数是插件类里需要传入的方法名，第 4 个参数是插件类里做处理时需要的参数。这里插件处理完成后的回调函数的参数 result 就是 ThreeDesPlugin 类的 execute 方法拼装返回的 json 对象，即已经加密的用户名和密码。

把以上代码应用到之前做的 App 应用中，然后再用 Chrome 浏览器监控一下登陆请求，如图 5，可以看到这时请求发送的用户名和密码就已经是加密后的字符了。

图 5 登陆请求

4　结束语

至此一个通过调用原有 Java 加密类来实现 android 版移动 App 加密登录的功能就完成了。可见在已有 Java 代码的基础上开发 Android App 是非常方便快速的，对于很多以前做 Java Web 应用的开发者来说，想从事 App 开发，Ioinc 开发框架绝对是个不错的选择。

参考文献：

[1]Brad Green，ShyanSeshadri.用 AngularJS 开发下一代 Web 应用[M].大漠穷秋 译.北京:电子工业出版社,2013:77- 108

[2]http://www.ionic.wang Ionic 免费教程全文

[3]http://angularjs.cn/tag/AngularJSAngularJS 中文社区

Ionic framework development mobile App——custom encrypted login plugin

DENG Hui- qin

(Anhui Vocational College of Electronics & Information Technology Software College,Anhui Bengbu 233000, China)

Abstract:Mobile APP development is currently the most popular program development, a lot of companies or individuals previously done web applications are moving to mobile APP development and transformation. So how in the existing web application system, the rapid development of APP products do? Many people may choose to change the front page to Html5+CSS3, but this development model developed products have many shortcomings, such as: low performance, poor compatibility, not easy to maintain, and so on. In view of this situation a lot of mobile terminal web development framework came into being, Ionic is one of the best free mobile client development framework. The content of this article is also aimed at the existing web application background process, how to use the Ionic framework to directly call the existing JAVA class, so as to simplify the development of APP.

Key words：free；Ionic；Java；rapid development mobile app；mobile Web application