Cordova

1. 安装cordova

命令：npm install -g cordova

1. 创建cordova项目

命令：cordova create cordovaApp //cordovaApp: 创建项目名称

cd cordovaApp //转到你创建的目录

cordova platform add android//添加构建app的平台

1. 打包构建

命令：cordova run android//生成apk

路径：platforms\android\app\build\outputs\apk\debug

Vue

1. 安装vue   
   命令; npm install -g vue-cli // 查看版本号：vue -V
2. 创建一个基于webpack的项目

命令: vue init webpack my-vue // my-vue项目名称

可以选择你所需要的依赖

1. 启用项目

命令：cd my-vue //进入项目

命令：npm install //安装依赖

命令：npm run dev //运行（运行时可能会出错，可能是因为安装依赖不完整，根据错误提示安装）

Cordova+vue

修改vue项目配置

1. 修改index.html的head

<meta http-equiv="Content-Security-Policy" content="default-src 'self' data: gap: https://ssl.gstatic.com 'unsafe-eval'; style-src 'self' 'unsafe-inline'; media-src \*; img-src 'self' data: content:;">

*//****这个的时候可能导致页面样式改变，如果改变则不加，否则还是建议加上。****这段主要是防止跨站脚本攻击。*

<meta name="format-detection" content="telephone=no">

<meta name="msapplication-tap-highlight" content="no">

<meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width">

1. 引入cordova.js

<body>

<div id="app"></div>

<script type="text/javascript" src="cordova.js"></script>

<!-- built files will be auto injected -->

</body>

1. 修改src中的main.js

document.addEventListener('deviceready', *function*() {

new Vue({

el: '#app',

template: '<App/>',

components: {

App

}

})

window.navigator.splashscreen.hide()

}, false);

1. 修改config文件夹中的index.js文件

修改build中的

assetsSubDirectory: '',

assetsPublicPath: '',

1. 运行

命令 npm run dev

如果正常说明到这里是没有问题的（注意这里运行的时候需要将document.addEventListener注释，因为在浏览器环境下是找不到cordova.js的也就不能监听到deviceready的事件，打包在真机上才能实现监听）

1. 构建vue项目

命令：npm run build //生成dist文

将dist文件夹中的文件替换到cordova项目中的www文件夹中

Cordova执行

1. 打包

命令：cordova build android // 完成打包

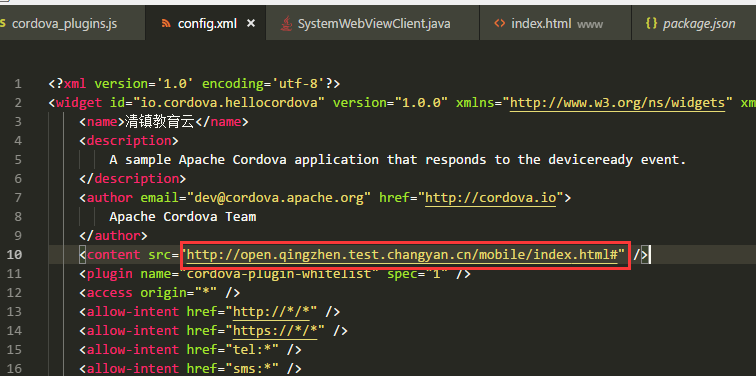
打包之前需要修改cordova文件中的名称图标等项目相关配置

注：查看一些教程需要配置环境变量，这里没有配置，当你打包不成功可能是因为没有配置

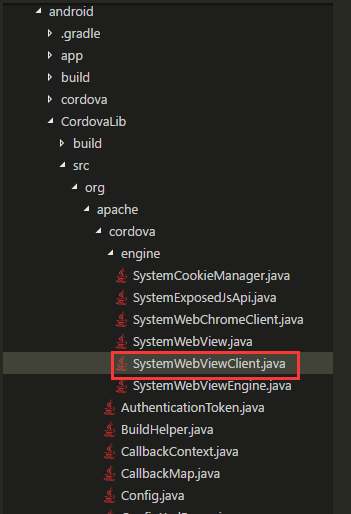
如果你打包的是一个服务器的地址，

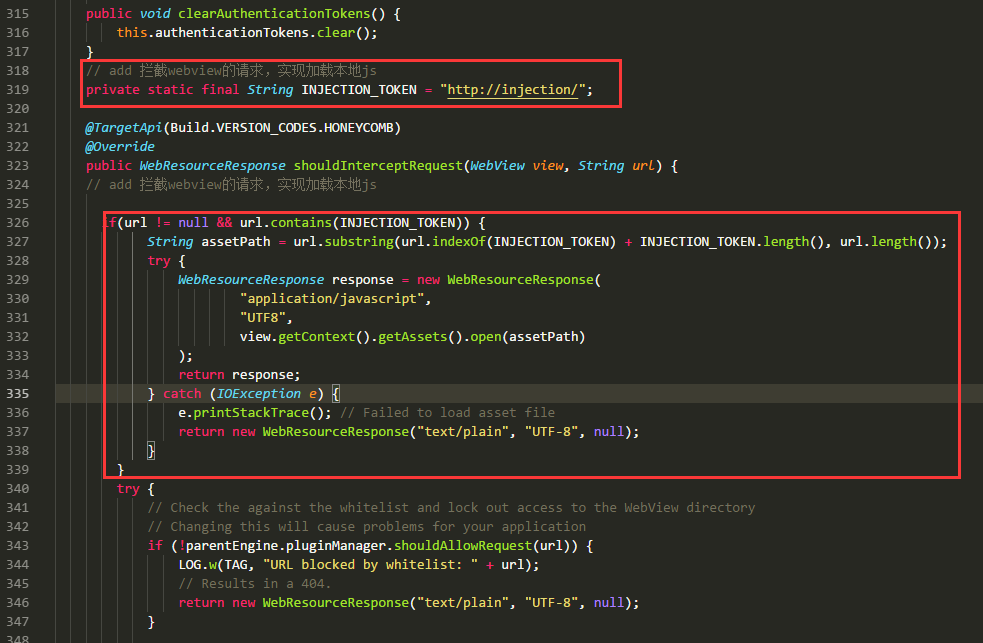
那么你需要以下cordova文件：

1. 将cordova文件中的config.xml的content修改成服务器地址



1. 修改下面这个文件 新增两段代码





// add 拦截webview的请求，实现加载本地js

private static final *String* INJECTION\_TOKEN = "http://injection/";

// add 拦截webview的请求，实现加载本地js

if(url != null && url.contains(INJECTION\_TOKEN)) {

*String* assetPath = url.substring(url.indexOf(INJECTION\_TOKEN) + INJECTION\_TOKEN.length(), url.length());

try {

*WebResourceResponse* response = new WebResourceResponse(

"application/javascript",

"UTF8",

view.getContext().getAssets().open(assetPath)

);

return response;

} catch (*IOException* *e*) {

e.printStackTrace(); // Failed to load asset file

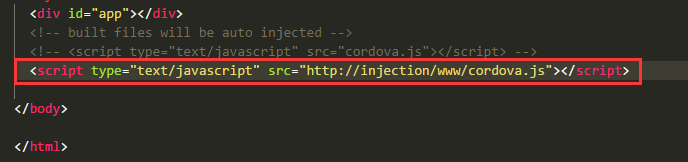
return new WebResourceResponse("text/plain", "UTF-8", null);

}

}

1. 修改你所在代码引入的cordova.js

<script type="text/javascript" src="http://injection/www/cordova.js"></script>



修改安装的背景图标

1. 安装 cordova-plugin -splashscreen

命令：cordova plugin add cordova-plugin-splashscreen

1. 替换原有cordova的图片
2. 在cordova config.xml 文件中引入以下代码

<platform name="android">

<icon density="ldpi" src="res/icon/android/icon-36-ldpi.png" />

<icon density="mdpi" src="res/icon/android/icon-48-mdpi.png" />

<icon density="hdpi" src="res/icon/android/icon-72-hdpi.png" />

<icon density="xhdpi" src="res/icon/android/icon-96-xhdpi.png" />

<splash density="port-hdpi" src="res/screen/android/screen-hdpi-portrait.png" />

<splash density="port-ldpi" src="res/screen/android/screen-ldpi-portrait.png" />

<splash density="port-mdpi" src="res/screen/android/screen-mdpi-portrait.png" />

<splash density="port-xhdpi" src="res/screen/android/screen-xhdpi-portrait.png" />

</platform>

下载文件

1. 安装 cordova-plugin -file-transfer

命令：cordova plugin add cordova-plugin-file-transfer