**百傲瑞达App开发说明**

维护部门：安防系统事业部

适用范围：百傲瑞达App开发人员

修订记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 日期 | 修订人 | 描述 | 备注 |
| 1.0 | 2017.07.28 | 陈培发 |  | 初稿 |
| 2.0 | 2019.2.19 | 张耀煌 |  | 初稿 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1. 目的 4](#_Toc26953)

[2. 开发环境搭建说明 4](#_Toc18004)

[2.1 安装NodeJS 4](#_Toc24015)

[2.2 安装JDK 4](#_Toc21623)

[2.3 安装TortoiseSVN 5](#_Toc7784)

[2.4 安装Android studio 5](#_Toc18250)

[2.5 安装IDE工具 6](#_Toc18230)

[2.6 安装谷歌浏览器 6](#_Toc27558)

[3. 开发流程说明 6](#_Toc5614)

[3.1 启动App 6](#_Toc963)

[3.2 项目目录结构 6](#_Toc6634)

[3.3 页面组件开发 9](#_Toc1905)

[3.4 页面懒加载配置 10](#_Toc539)

[3.5 使用自定义组件 13](#_Toc3790)

[3.6 使用Ionic4.0插件 14](#_Toc24395)

[3.7 国际化 14](#_Toc1026)

[3.8 自定义矢量图标 14](#_Toc9601)

[3.9 http数据交互 17](#_Toc23434)

[3.10 调试 18](#_Toc19855)

[4. 参考资料 21](#_Toc20495)

# 目的

便于开发人员快速熟悉和搭建ionic跨平台移动开发框架

# 开发环境搭建说明

## 2.1 安装NodeJS

nodejs选择的版本是7.0以上8.0以下，版本太新会有依赖问题。默认安装即可

下载地址: 链接: http://pan.baidu.com/s/1nvIAdIT 密码: wkut

## 2.2 安装JDK

JDK的版本1.7以上，官网下载安装包默认安装即可

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

安装完后按照以下地址说明设置环境变量

<http://jingyan.baidu.com/article/6dad5075d1dc40a123e36ea3.html>

## 2.3 安装TortoiseSVN

从以下地址下载，默认安装即可

链接: http://pan.baidu.com/s/1nvdsAyH 密码: 9wn5

## 2.4 安装Android studio

从官网下载安装包默认安装，此工具自带安装SDK，安装时会让你选择SDK目录，安装后配置SDK环境变量

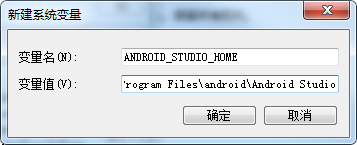
1. 新建环境变量ANDROID\_HOME



1. 编辑path环境变量新增如下路径

;%ANDROID\_HOME%\tools;%ANDROID\_HOME%\platform-tools;

1. 新建环境变量ANDROID\_STUDIO\_HOME



1. 编辑path环境变量新增Grade路径,grade路径位于android studio安装目录下

%ANDROID\_STUDIO\_HOME%\gradle\gradle-3.2\bin;

## 2.5 安装IDE工具

IDE可选择sublime Text，atom等编辑器。默认安装即可

SublimeText : 链接: http://pan.baidu.com/s/1i53PaD3 密码: 6isn

Atom: 链接: http://pan.baidu.com/s/1sl8Yx8x 密码: 6qyf

## 2.6 安装谷歌浏览器

用于开发时使用谷歌浏览器开发模式查看页面效果。网上下载安装包。

# 开发流程说明

## 启动App

1. 拉取git仓库项目

项目地址:http://192.168.200.37/biosecurity-pro-cloud/zkbiosecurity-cloud-att-app

1. 在项目根目录启动命令行窗口
2. 执行npm install -g cnpm安装淘宝镜像cnpm
3. 执行 npm install -g ionic cordova 安装ionic
4. 执行 cnpm install 安装依赖
5. 执行 ionic serve 启动服务

注：-g表示全局安装，只执行一次即可，如果npm执行出错改成使用cnpm

## 项目目录结构

根据启动模块、业务模块、UI控件模块、系统模块和基础模块，分成对应的7个模块工程

根目录，除了UI控件模块和公共模块，其他模块工程都可以单独启动不用依赖。

启动模块：负责引入其他子模块的入口

业务模块：所有业务模块的页面目录

系统模块：系统相关的页面目录

基础模块: 公司信息和个人信息等页面目录

UI控件模块: 封装了常用的UI控件库目录

├─zksecurity-app-att //考勤模块工程

│ ├─ src //工程主要的代码目录

│ │ ├─ app //考勤相关页面目录

├─zksecurity-app-startUp //启动模块工程

│ ├─ src //工程主要的代码目录

│ │ ├─ app //app模块相关

│ │ ├─ ├─app-routing.module.ts //路由配置文件

│ │ ├─ ├─app.component.html //主页面模板文件

│ │ ├─ ├─app.component.spec.ts //单元测试文件

│ │ ├─ ├─app.component.ts //app主类定义文件

│ │ ├─ ├─app.module.ts //apg根模块定义文件

│ │ ├─ ├─app.scss //app样式定义文件

│ │ ├─ assets //静态文件目录相关

│ │ ├─ environments //开发和生产环境配置目录

│ │ ├─ theme //主题样式目录

├─zksecurity-app-common //公共模块工程（主要包含全局的服务和管道等）

│ ├─ src //工程主要的代码目录

│ │ ├─ filter //所有的管道定义目录

│ │ ├─ providers //所有的非页面型功能性服务类定义目录

│ │ ├─├─ http-service.ts //http请求工具类服务

│ │ ├─├─ basecode.ts //创建下拉字典服务

│ │ ├─├─ common-validator.ts //通用表单验证服务

│ │ ├─├─ constants.ts //分页和版本信息服务

│ │ ├─├─ image-utils.ts //图片处理工具服务

│ │ ├─├─ menus.ts //菜单处理工具服务

│ │ ├─├─ message-box.ts //全局消息盒子服务

│ │ ├─├─ properties.ts //获取属性值服务 ？

│ │ ├─├─ settings.ts //ionic3全局缓存Storage服务

│ │ ├─├─ utils.ts //消息提示和字符串处理服务

│ │ ├─├─ zk-message.ts //全局消息提示框服务

│ │ ├─ theme //公共的样式定义目录

├─zksecurity-app-base //基础模块工程

│ ├─ src //工程主要的代码目录

├─zksecurity-app-pers //人事模块工程

│ ├─ src //工程主要的代码目录

├─zksecurity-app-system //系统模块工程

│ ├─ src //工程主要的代码目录

├─zksecurity-app-ui //UI控件工程

│ ├─ src //工程主要的代码目录

│ │ ├─ components //UI组件定义目录

├─src //源目录用户在此目录下开发

│ ├─ app //app模块相关

│ │ ├─ app.component.ts //app主类定义文件

│ │ ├─ app.component.html //app模板文件

│ │ ├─ app.module.ts //app模块定义文件

│ │ ├─ app.scss //app样式定义文件

│ │ ├─ app-routing.module.ts //app路由配置文件

//静态资源文件存放目录

│ │ ├─ fonts //字体图标目录

│ │ ├─ i18n //国际化目录

│ │ ├─ icon //图标存放目录

│ │ ├─ img //图片存放目录

│ ├─ pages //页面组件存放目录

│ │ ├─ acc //门禁页面组件存放目录

│ │ ├─ att //考勤页面组件存放目录

│ │ ├─ base //公共页面组件存放目录，如工作台，登录等

│ │ ├─ park //停车场页面组件存放目录

│ │ ├─ pers //人事页面组件存放目录

│ │ ├─ vis //访客页面组件存放目录

│ ├─ theme // 主题样式文件存放目录

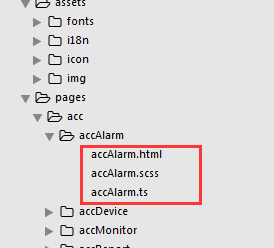
│ │ ├─ \_whole.scss //全局样式定义

│ │ ├─ variables.scss //scss样式变量定义文件,在此文件引入主题目录下的其他样式文件

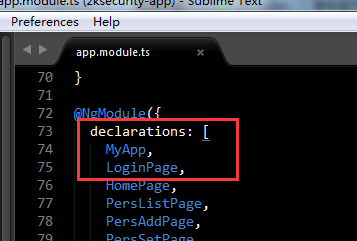
│ ├─ index.html // app索引文件

## 3.3 页面组件开发

1. 在src/pages下对应的模块目录中根据开发的组件功能按照驼峰命名方式，一个组件一个文件夹创建文件夹，每个组件
2. 由xxx.html,xxx.ts,xxx.scss三个文件组成。



1. 在src/app/app.module.ts中导入，进行组件声明。需要声明的有两个，一个是declarations，一个是entryComponents。



1. 如果是工作台功能组件则在src/pages/base/home/home.ts里导入使用声明。在对应的模块节点下添加菜单配置。
2. 页面跳转使用NavController控制器进行跳转。需要进行页面回退的则使用navCtrl.push()。不需要则使用setRoot()

## 3.4 页面懒加载配置

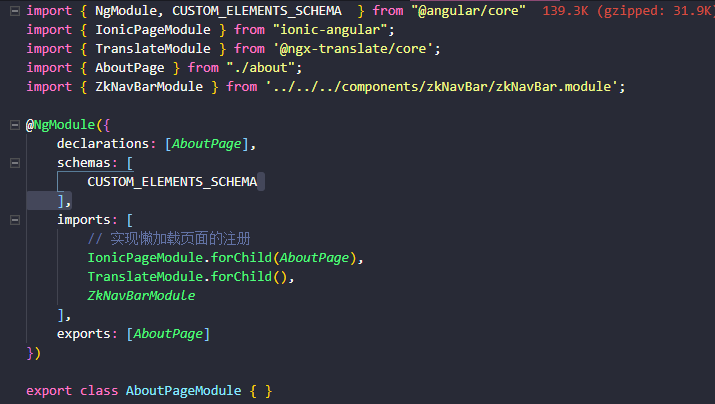
App启动加载介绍:

本应用采用懒加载机制，什么是懒加载呢？当我们第一次进入应用，会加载app.module,如果没有采用懒加载，那么app.module里的所有东西都会被加载，很显然这是不太合适的，因为里面的组件和服务我们在第一次进入应用时并没有全部使用，可能只使用了很少的一部分，那么对于比较大型的应用来说，由于页面和其他的服务，指令，管道比较多，就会造成用户体验不好的影响。 这个时候我们需要把应用分为一个个module，各自引入自己用到的东西，不要管别的模块。那么第一次进入应用，就只加载很少的app.module和root page对应的module，当我们进入其他页面的时候，才会去加载相应的module。这样的好处是项目代码结构清晰，易于维护，易于开发，而且第一次加载速度比较快。 当我们使用Ionic3的CLI来新建页面，默认的会是懒加载的结构，甚至不止页面，所有的component,service,directive都默认的是懒加载。

配置步骤

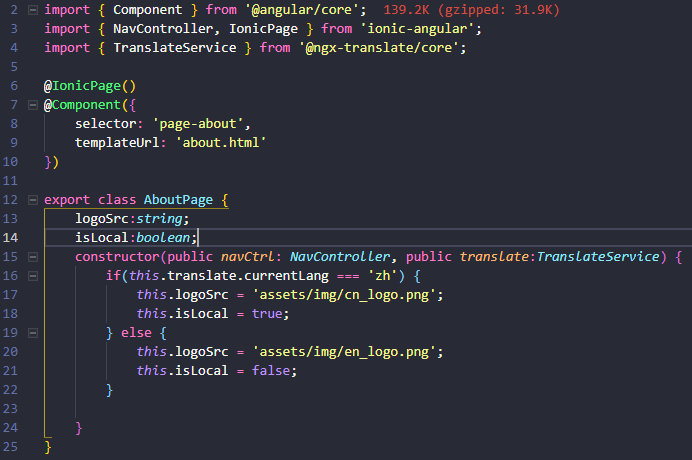
执行命令：

1. 给需要懒加载的页面配置module.ts；



1. 在对应页面的.ts文件里增加@IonicPage()特性；

以about.ts为例，在@Component上方加上@IonicPage()特性(行号5，注意在上面引用IonicPage(见2行)),其他需要懒加载的页面同样操作配置



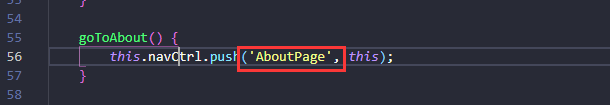
1. 在app.module.ts移除页面引用；

将配置过懒加载的页面，在app.module.ts中进行移除引用

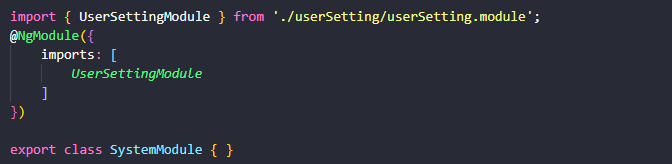
1. 使用懒加载；

使用懒加载，不需要在顶部进行import导入，只需要将之前的页面名字用引号引起来即可：

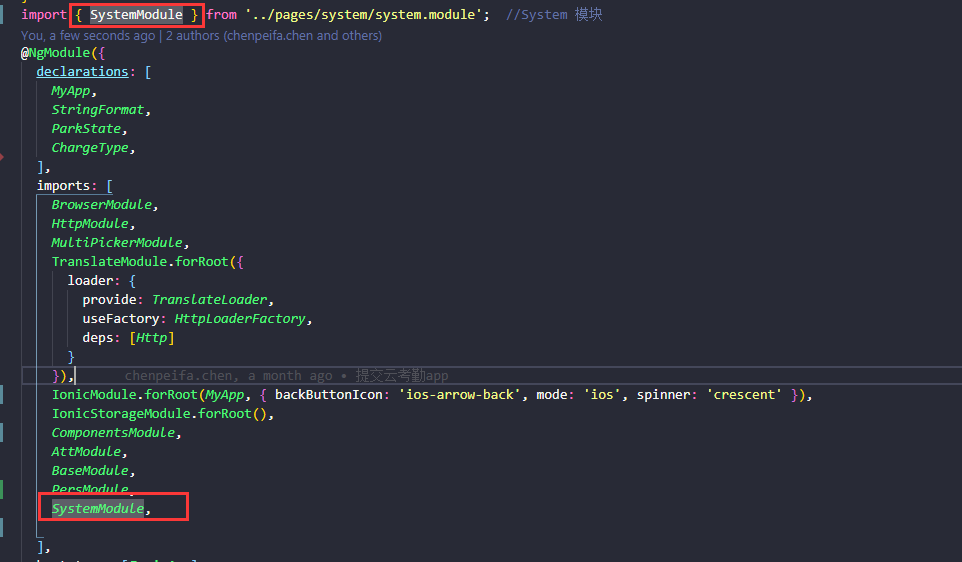
例如 userSetting.ts 代码段：



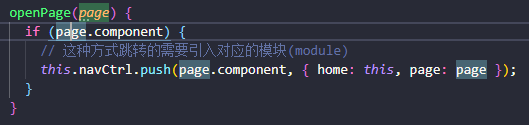
1. 以下方式跳转的需要在 app.module 引入对应的模块
2. System中定义模块



1. appModule引入模块



（3）跳转方式



参考链接：

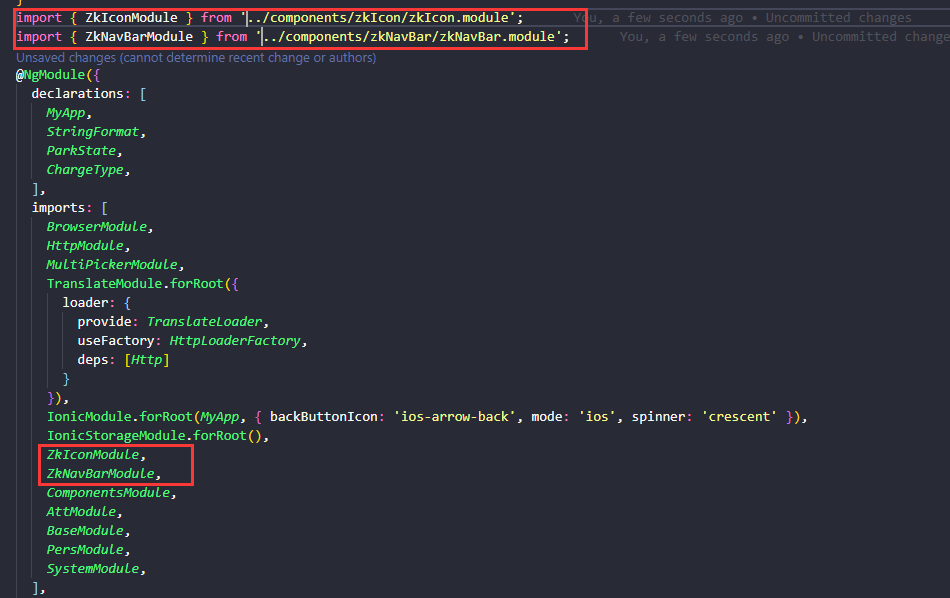
IonicPage：

https://www.angular.cn/guide/ngmodule-faq

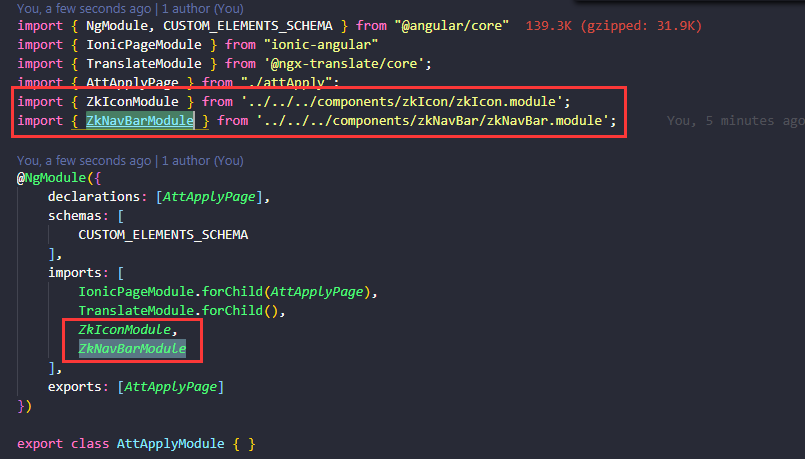
<https://ionicframework.com/docs/api/navigation/IonicPage/>

## 3.5 使用自定义组件

1. 如果调用的页面不是懒加载页面，则在app.module导入



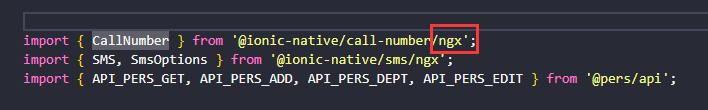
2. 如果调用的页面是懒加载页面，则在页面对应的xxx.module导入



## 3.6 使用Ionic4.0插件

按官网的文档比如安装ionic cordova plugin add cordova-plugin-network-information 默认安装的是4.x.x的，这样导入的时候import { Network } from '@ionic-native/network/ngx';

记得最后面的引入路径加 ngx。



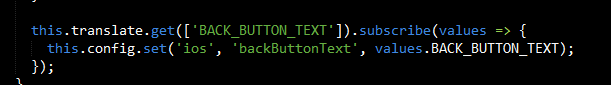
## 3.7 国际化

1. 配置

国际化在assets/i18n/目录下的json文件定义。json文件使用一级结构不建议嵌套。命名全部大写以下划线分隔。

1. 使用

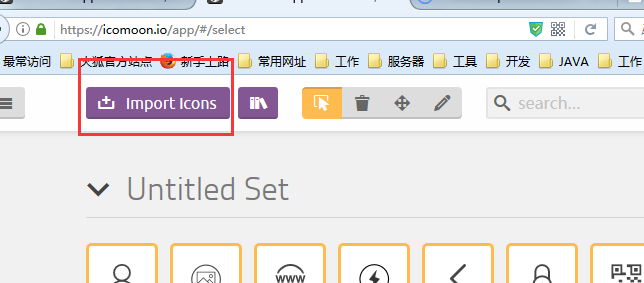
html页面使用过滤器表达式如{{‘COMMON\_OK’|translate}}。ts文件则使用TranslateService注入来使用如:(参考src/app/app.component.ts)



## 3.8 自定义矢量图标

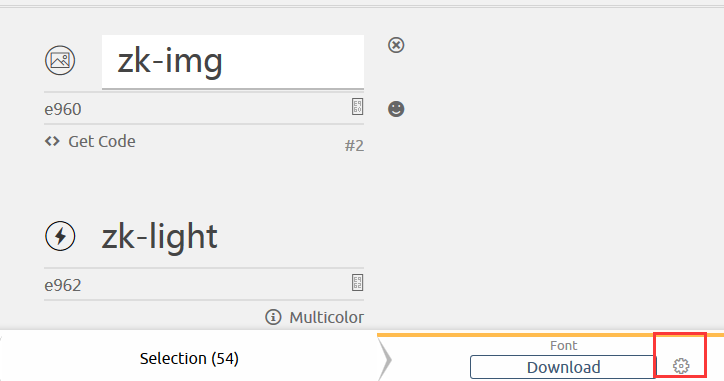
1，打开网站[https://icomoon.io/app/#/select](https://icomoon.io/app/" \l "/select)导入src/assets/fonts/zk-icons.svg导入之前的字体图标

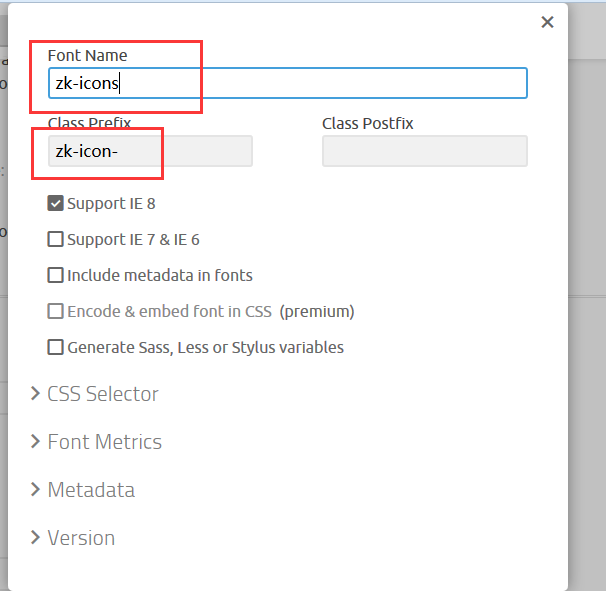
2，找UI人员要相应图标的svg文件再次在图标网站上导入



3，选择步骤1,2导入的图标点击右下角生成字体

4，编辑图标名称并设置字体为zk-icons

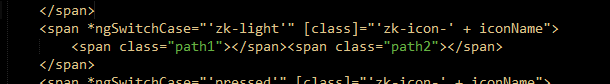






1. 解压字体文件将里面的fonts目录替换到src/assets/fonts里
2. 合并style.css里新增的几个图标样式到src/assets/fonts/zk-icons.scss，切记不要直接用style.css里的内容覆盖zk-icons.scss因为我们有做自己的样式调整。
3. 使用文本编辑器打开demo.html看下新增的图标加载方式，如果是要多个span才能加载出图标，则在src/pages/base/zkIcon/zkIcon.html追加swith案例





## 3.9 http数据交互

使用src/providers/http-service.ts来进行http交互，使用时注入即可。此类是基于angular的http进行封装支持get和post方法。

1、在app.module.ts中引入HttpClientModule 并注入

import {HttpClientModule} from '@angular/common/http';

imports: [

BrowserModule,

HttpClientModule

]

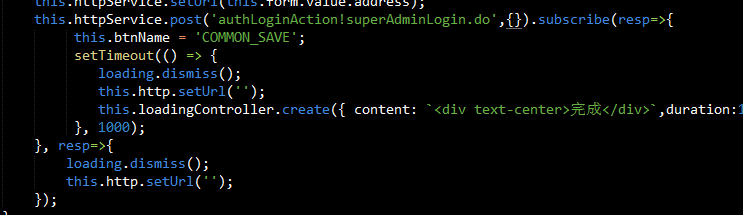
1. 在用到的地方引入HttpClient并在构造函数声明

import {HttpClient} from "@angular/common/http";Error

constructor(public http:HttpClient) { }

3、ionic4.x get请求数据

模式:this.httpService.post(url, params, reqOpt).subscribe(successFun,errorFun);如



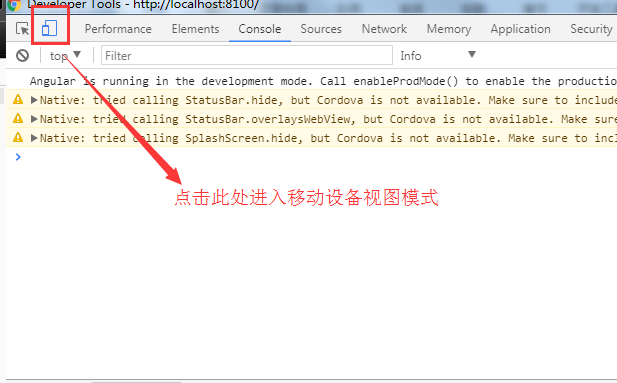
## 3.10 调试

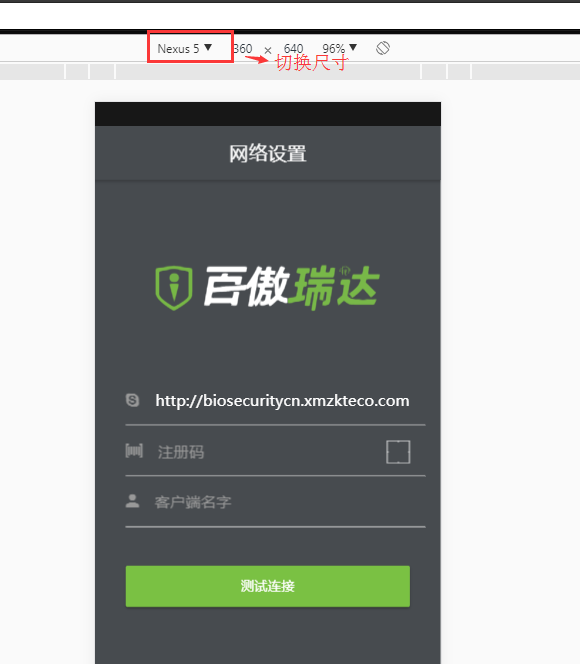
1，android虚拟机调试

安装Android studio后打开安装安卓虚拟机即可。对于大部分页面调试皆可，但对与http数据交互却存在跨域问题。执行命令ionic cordova run android即可

2，谷歌浏览器调试

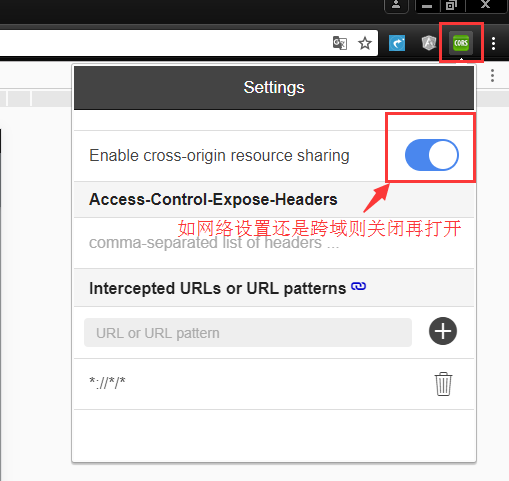
按F12进入开发模式





安装跨域插件，如不能打开则使用翻墙软件

https://chrome.google.com/webstore/detail/allow-control-allow-origi/nlfbmbojpeacfghkpbjhddihlkkiljbi



1. 真机调试

真机不存在跨域问题。手机连接电脑一般是没啥问题，不过有时可能会出现adb端口占用，如果是360手机助手则卸载它使用豌豆荚。在根据以下网上说明处理

http://blog.csdn.net/notzuonotdied/article/details/52404415

# 参考资料

[百傲瑞达APP原型设计](https://modao.cc/app/8lUcoPBowJblzffgSY1t2kaThdO8pxn" \l "screen=sC485135B041496390590876)

[Ionic官方文档](http://ionicframework.com/docs/)

[Angular 4.0文档](https://angular.cn/docs/ts/latest/)

[SCSS风格指南](https://sass-guidelin.es/zh/)

版本要求

Ionic: 3.20.0

Node:7.x