

# Node.js、Ionic、Cordova、AngularJS 安装指南

## 目录

一. Node.js .....	3
1. Node.js .....	3
1.1 Node.js 简介 .....	3
1.2 Windows 下安装 Node.js .....	4
2. NPM.....	4
2.1 npm 简介 .....	4
2.2 安装 npm .....	4
3. Express .....	4
3.1 Express 简介 .....	4
3.2 安装 express .....	5
4. 检验是否安装成功.....	5
5. 创建 hellworld 工程 .....	5
6. 用 npm 进行添加和启动项目 .....	6
7. 访问.....	7
二. Ionic.....	8
1. Ionic.....	8
1.1 Ionic 简介 .....	8
1.2 安装 Ionic.....	8
1.2 开始一个 Ionic APP .....	9
1.3 如何使用模板.....	9
1.4 在浏览器中测试.....	9
1.5 添加平台.....	10
2. ANT.....	10
2.1 ANT 简介 .....	10
2.2 安装 ant .....	10
3. build 项目.....	11
4 配置 android AVD .....	12
5 安装 APP 到模拟器 .....	13
6 安装 APP 到手机 .....	14
三. Cordova.....	16
1. Android SDK .....	16
1.1 简介.....	16
1.2 安装 Android SDK .....	16
2. ADT.....	17
2.1 ADT 简介 .....	17
2.2 安装 ADT.....	18
3. GIT .....	18
3.1 GIT 简介 .....	18
3.2 特点.....	18
3.3 安装 GIT .....	20

4. Cordova .....	20
4.1 Cordova 简介 .....	20
4.2 安装 Cordova .....	20
5. Bower .....	21
5.1 Bower 简介 .....	21
5.2 安装 bower .....	21
6. 创建 APP .....	22
7. 添加平台支持.....	23
8. Build APP .....	24
9. 准备和编译 APP .....	24
10. 安装 APP 到模拟器 .....	24
11. 安装 APP 到手机 .....	25
12. 添加插件功能.....	26
13.查看已安装插件.....	28
14.移除插件.....	28
15.高级插件选项.....	29
四. Bootstrap.....	29
1. Bootstrap 简介.....	29
1.1 Bootstrap 特点.....	29
1.2 Bootstrap 组件.....	30
1.3 Javascript 插件 .....	30
2.安装 Bootstrap.....	30
2.1 通过 bower 安装 .....	30
2.2 通过 npm 进行安装 .....	30
3. 目录结构.....	31
3.1 预编译版.....	31
3.2 Bootstrap 源码.....	32
4. GRUNT.....	32
4.1 Grunt 简介 .....	32
4.2 Grunt 安装 .....	32
4.3 可用的 Grunt 命令 .....	33
五. AngularJS.....	34
1. AngularJS 简介 .....	34
2. AngularJS 特性 .....	34
2.1 特性一：数据绑定.....	34
2.2 特性二：模板.....	34
2.3 特性三：MVC.....	35
2.4 特性四：依赖注入（Dependency Injection，即 DI） .....	35
2.5 特性五：Directives（指令） .....	35

日期	版本	说明	作者
2015-01-29	1.0	初稿	李金柱
2015-02-03	1.1	修订	李金柱

# 一. Node.js

## 1. Node.js

### 1.1 Node.js 简介

Node.js® is a platform built on [Chrome's JavaScript runtime](#) for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Node.js 是一个基于 Chrome JavaScript 运行时建立的一个平台， 用来方便地搭建快速的 易于扩展的网络应用· Node.js 借助[事件驱动](#)， 非阻塞 [I/O](#) 模型变得轻量和高效， 非常适合运行在分布式设备 的 数据密集型 的实时应用

## 1.2 Windows 下安装 Node.js

下载地址: <http://nodejs.org/download/>

根据您的操作系统选择相应的 32-bit 或者 64-bit 的.msi 或者.exe 文件并下载

以下两种方式任选其一:

1. .msi 文件下载成功后双击选择 “Next” 完成安装
2. 在 E 盘创建文件夹 nodejs, 拷贝 node.exe 到 nodejs 文件夹下, 并将"E:\nodejs"加入系统环境变量 PATH 中

## 2. NPM

### 2.1 npm 简介

A package manager for node

NPM 的全称是 Node Package Manager, 是一个 NodeJS 包管理和分发工具, 已经成为了非官方的发布 Node 模块(包)的标准。

Nodejs 自身提供了基本的模块, 但是开发实际应用过程中仅仅依靠这些基本模块则还需要较多的工作。幸运的是, Nodejs 库和框架为我们提供了帮助, 让我们减少工作量。但是成百上千的库或者框架管理起来又很麻烦, 有了 NPM, 可以很快的找到特定服务要使用的包, 进行下载、安装以及管理已经安装的包

### 2.2 安装 npm

下载地址: <https://github.com/npm/npm/tags>

将 npm 源代码解压到 E:\npmjs 目录中。

在命令提示符窗口中执行下面的操作, 完成 npm 的安装:

```
E:\>cd npmjs
```

```
E:\npmjs>node cli.js install npm -gf
```

NPM 安装完成后, 将"E:\nodejs\node\_modules"加入系统环境变量 NODE\_PATH 中

## 3. Express

### 3.1 Express 简介

Express is a minimal and flexible Node.js web application framework that provides a robust set of

features for web and mobile applications.

Express 是一个极简的，灵活的 Node.js web 应用程序框架，针对 web 和移动应用程序提供了一套强健的功能。

## 3.2 安装 express

在命令提示符窗口中执行下面的操作：

```
npm install express -g  
npm install express-generator -g
```

安装成功后，目录如下：

我的电脑 > 电脑磁盘 (E:) > nodejs > node\_modules

名称	修改日期	类型
express	2015/1/29 18:35	文件夹
express-generator	2015/1/29 18:39	文件夹
npm	2015/1/29 18:33	文件夹

## 4. 检验是否安装成功

在命令提示符窗口中执行下面的操作：

```
E:\>node -v  
v0.10.36  
  
E:\>npm -v  
2.4.1  
  
E:\>express -U  
4.11.2
```

注：express -V，V 需要大写

## 5. 创建 hellworld 工程

在命令提示符窗口中执行下面的操作：

Express helloworld

```

E:\>express helloworld

create : helloworld
create : helloworld/package.json
create : helloworld/app.js
create : helloworld/public
create : helloworld/views
create : helloworld/views/index.jade
create : helloworld/views/layout.jade
create : helloworld/views/error.jade
create : helloworld/routes
create : helloworld/routes/index.js
create : helloworld/routes/users.js
create : helloworld/bin
create : helloworld/bin/www
create : helloworld/public/stylesheets
create : helloworld/public/stylesheets/style.css

install dependencies:
$ cd helloworld && npm install

run the app:
$ DEBUG=helloworld:* ./bin/www

create : helloworld/public/javascripts
create : helloworld/public/images

```

工程目录创建成功:

我的电脑 > 电脑磁盘 (E:) > helloworld				▼	↺	搜索
名称	修改日期	类型	大小			
bin	2015/1/29 20:37	文件夹				
public	2015/1/29 20:37	文件夹				
routes	2015/1/29 20:37	文件夹				
views	2015/1/29 20:37	文件夹				
app.js	2015/1/29 20:37	JavaScript 文件	2 KB			
package.json	2015/1/29 20:37	JSON 文件	1 KB			

## 6. 用 npm 进行添加和启动项目

在命令提示符窗口中执行下面的操作:

```
cd helloworld
```

```
npm install //下载依赖包
```

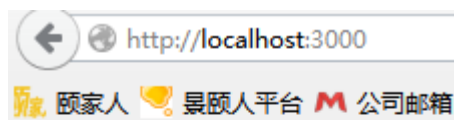
```
npm start //启动项目
```

我的电脑 > 电脑磁盘 (E:) > helloworld		
名称	修改日期	类型
bin	2015/1/29 20:57	文件夹
node_modules	2015/1/29 20:58	文件夹
public	2015/1/29 20:57	文件夹
routes	2015/1/29 20:57	文件夹
views	2015/1/29 20:57	文件夹
app.js	2015/1/29 20:57	JavaScript 文件
package.json	2015/1/29 20:57	JSON 文件

### 目录介绍:

- node\_modules, 存放所有的项目依赖库。(每个项目管理自己的依赖)
- package.json, 项目依赖配置及开发者信息
- app.js, 程序启动文件
- public, 静态文件(css, js, img)
- routes, 路由文件(MVC 中的 C, controller)
- Views, 页面文件(Ejs 模板)

## 7. 访问



# Express

Welcome to Express

## 二. Ionic

### 1. Ionic

#### 1.1 Ionic 简介

The beautiful, open source front-end SDK for developing hybrid mobile apps with HTML5.

Ionic offers a library of mobile-optimized HTML, CSS and JS components for building highly interactive apps. It is built with Sass and optimized for AngularJS. It's in and it looks like a very promising framework for developing hybrid mobile apps in HTML5.

漂亮的，开源的前端 SDK，使用 HTML5 开发 hybrid 手机应用

Ionic 是一个用 HTML, CSS 跟 JS 开发的一个用于移动设备的混合 APP 开发框架，采用 Sass 与 AngularJS 开发

#### 1.2 安装 Ionic

在命令提示符窗口中执行下面的操作：

```
$ npm install -g ionic
```

```
E:\>cd npmjs
E:\npmjs>npm install -g ionic
```

安装成功后：

我的电脑 > 电脑磁盘 (E:) > nodejs > node\_modules

名称	修改日期	类型
angular-touch	2015/1/30 17:17	文件夹
bower	2015/1/30 16:52	文件夹
cordova	2015/1/30 14:58	文件夹
express	2015/1/29 18:35	文件夹
express-generator	2015/1/29 18:39	文件夹
ionic	2015/1/30 15:22	文件夹
jquery	2015/1/30 17:22	文件夹
npm	2015/1/29 18:33	文件夹



## 1.2 开始一个 Ionic APP












在命令提示符窗口中执行下面的操作：

```
$ ionic start myapp [template]
```

Template 有如下三种，默认是 tabs project：

- [tabs](#) (默认)
- [sidemenu](#)
- [blank](#)

命令执行成功后：

我的电脑 > 电脑磁盘 (E:) > myapp		
名称	修改日期	类型
 hooks	2015/2/2 9:00	文件夹
 plugins	2015/2/2 9:00	文件夹
 scss	2015/2/2 9:00	文件夹
 www	2015/2/2 9:00	文件夹
 .bowerrc	2015/2/2 9:00	BOWERRC 文件
 .gitignore	2015/2/2 9:00	文本文档
 bower.json	2015/2/2 9:00	JSON 文件
 config.xml	2015/2/2 9:00	XML 文件
 gulpfile.js	2015/2/2 9:00	JavaScript 文件
 package.json	2015/2/2 9:00	JSON 文件
 README.md	2015/2/2 9:00	MD 文件

## 1.3 如何使用模板

Tabs、sidemenu、blank 模板不能自己运行，因为少了 ionic 的 library 和 AngularJS

有两种方法使用模板：

1. 使用 ionic Node.js 创建一个新的 ionic 项目
2. 拷贝粘贴此模板到已经存在的 Cordova 项目

## 1.4 在浏览器中测试

在命令提示符窗口中执行下面的操作：

```
$ cd myapp
```

```
$ ionic serve
```

在浏览器地址栏输入：<http://localhost:8100>

就可以在浏览器中以 web 方式访问程序了















## 15 添加平台

在命令提示符窗口中执行下面的操作：

```
$ cd myapp
```

```
$ ionic platform add android
```

命令执行成功后：

我的电脑 > 电脑磁盘 (E:) > myapp > platforms > android		
名称	修改日期	类型
 assets	2015/2/2 9:23	文件夹
 cordova	2015/2/2 9:23	文件夹
 CordovaLib	2015/2/2 9:23	文件夹
 libs	2015/2/2 9:23	文件夹
 platform_www	2015/2/2 9:23	文件夹
 res	2015/2/2 9:23	文件夹
 src	2015/2/2 9:23	文件夹
 .gitignore	2015/2/2 9:23	文本文档
 .project	2015/2/2 9:23	PROJECT 文件
 AndroidManifest.xml	2015/2/2 9:23	XML 文件
 build.gradle	2015/2/2 9:23	GRADLE 文件
 custom_rules.xml	2015/2/2 9:23	XML 文件
 project.properties	2015/2/2 9:23	PROPERTIES 文件
 settings.gradle	2015/2/2 9:23	GRADLE 文件

## 2. ANT

### 2.1 ANT 简介

Apache Ant,是一个将[软件](#)编译、测试、部署等步骤联系在一起加以自动化的一个工具，大多用于 Java 环境中的软件开发。由 Apache [软件](#)基金会所提供。

### 2.2 安装 ant

下载地址：<http://ant.apache.org/bindownload.cgi>

把下载到的 apache-ant-1.9.4-bin.zip 解压到 E:\apache-ant-1.9.4

window 中设置 ant 环境变量:

ANT\_HOME: E:\apache-ant-1.9.4

PATH: E:\apache-ant-1.9.4\bin

CLASSPATH: E:\apache-ant-1.9.4\lib

**检验是否安装成功:**

在命令提示符窗口中执行下面的操作:

```
$ ant -version
```

### 3. build 项目

在命令提示符窗口中执行下面的操作:

```
$ cd myapp
```

```
$ ionic build android
```

Build 成功之后, E:\myapp\platforms\android 生成一个 android 项目目录

我的电脑 > 电脑磁盘 (E:) > myapp > platforms > android		
名称	修改日期	类型
ant-build	2015/2/2 11:26	文件夹
ant-gen	2015/2/2 11:26	文件夹
assets	2015/2/2 11:25	文件夹
cordova	2015/2/2 9:23	文件夹
CordovaLib	2015/2/2 11:25	文件夹
libs	2015/2/2 9:23	文件夹
platform_www	2015/2/2 9:23	文件夹
res	2015/2/2 9:23	文件夹
src	2015/2/2 9:23	文件夹
.gitignore	2015/2/2 9:23	文本文档
.project	2015/2/2 9:23	PROJECT 文件
AndroidManifest.xml	2015/2/2 11:25	XML 文件
build.gradle	2015/2/2 9:23	GRADLE 文件
build.xml	2015/2/2 11:25	XML 文件
custom_rules.xml	2015/2/2 9:23	XML 文件
local.properties	2015/2/2 11:25	PROPERTIES 文件
project.properties	2015/2/2 9:23	PROPERTIES 文件
settings.gradle	2015/2/2 9:23	GRADLE 文件

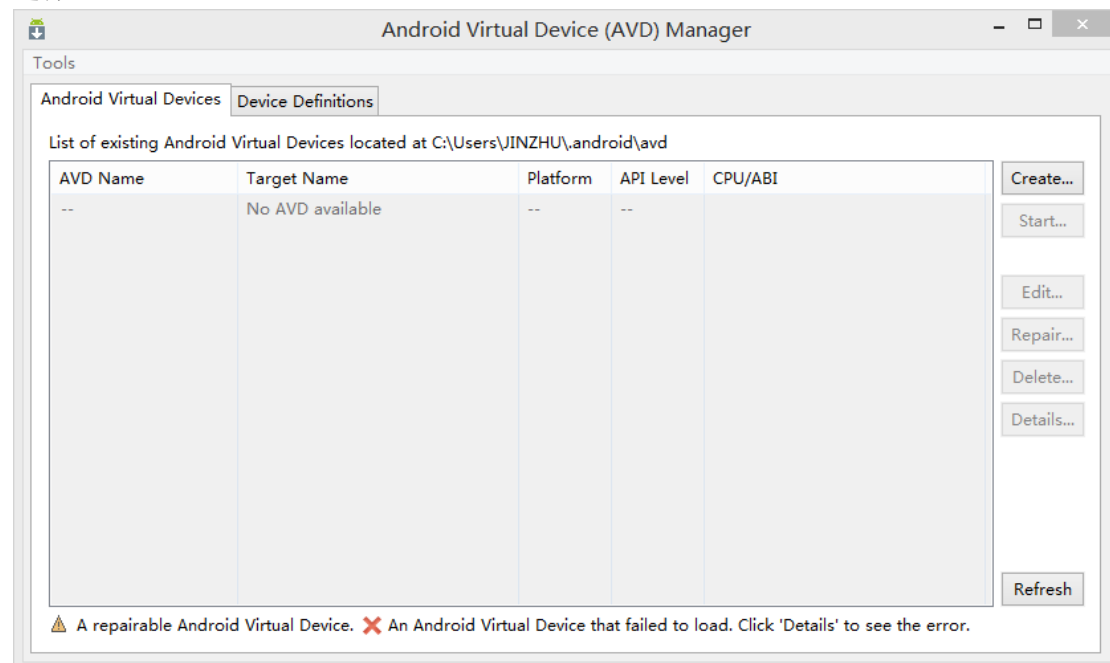
## 4 配置 android AVD

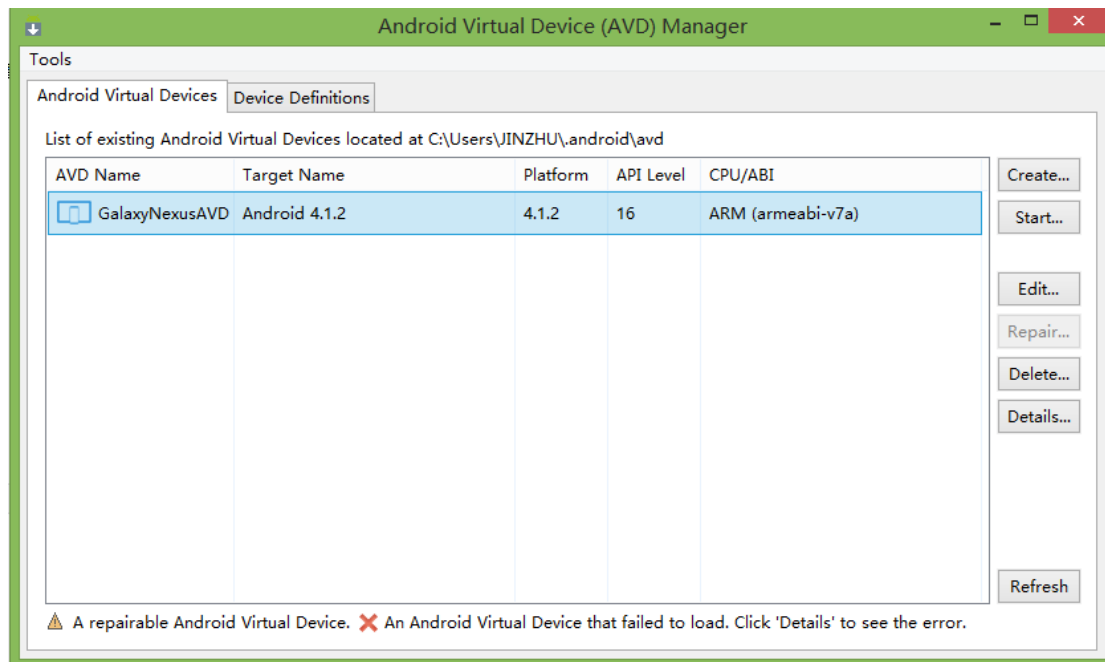
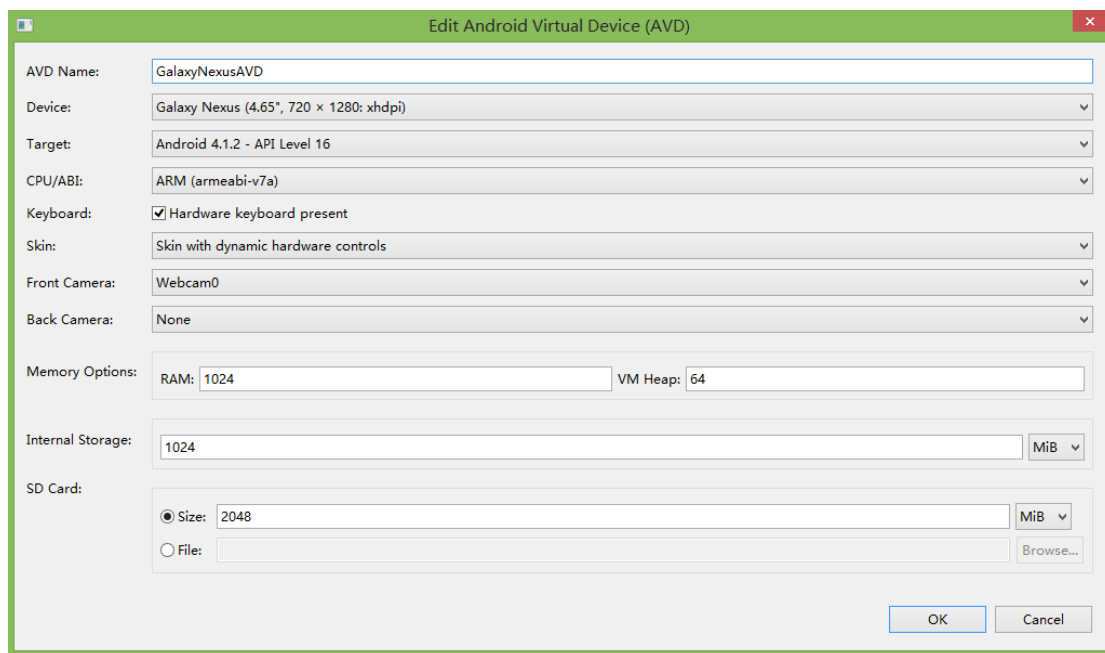
在 SDK 安装目录，打开 AVD Manager.exe

我的电脑 > 电脑磁盘 (E:) > android-sdk-windows

名称	修改日期	类型
add-ons	2015/1/31 22:22	文件夹
build-tools	2015/1/30 18:21	文件夹
docs	2015/1/30 17:31	文件夹
extras	2015/1/31 9:03	文件夹
platforms	2015/1/31 19:11	文件夹
platform-tools	2015/1/30 14:27	文件夹
samples	2015/1/31 19:13	文件夹
sources	2015/1/31 19:29	文件夹
system-images	2015/1/31 20:19	文件夹
temp	2015/1/31 22:22	文件夹
tools	2015/1/30 14:06	文件夹
AVD Manager.exe	2014/12/18 8:05	应用程序
SDK Manager.exe	2014/12/18 8:05	应用程序
SDK Readme.txt	2014/12/18 8:05	文本文档

选择 Create a new AVD



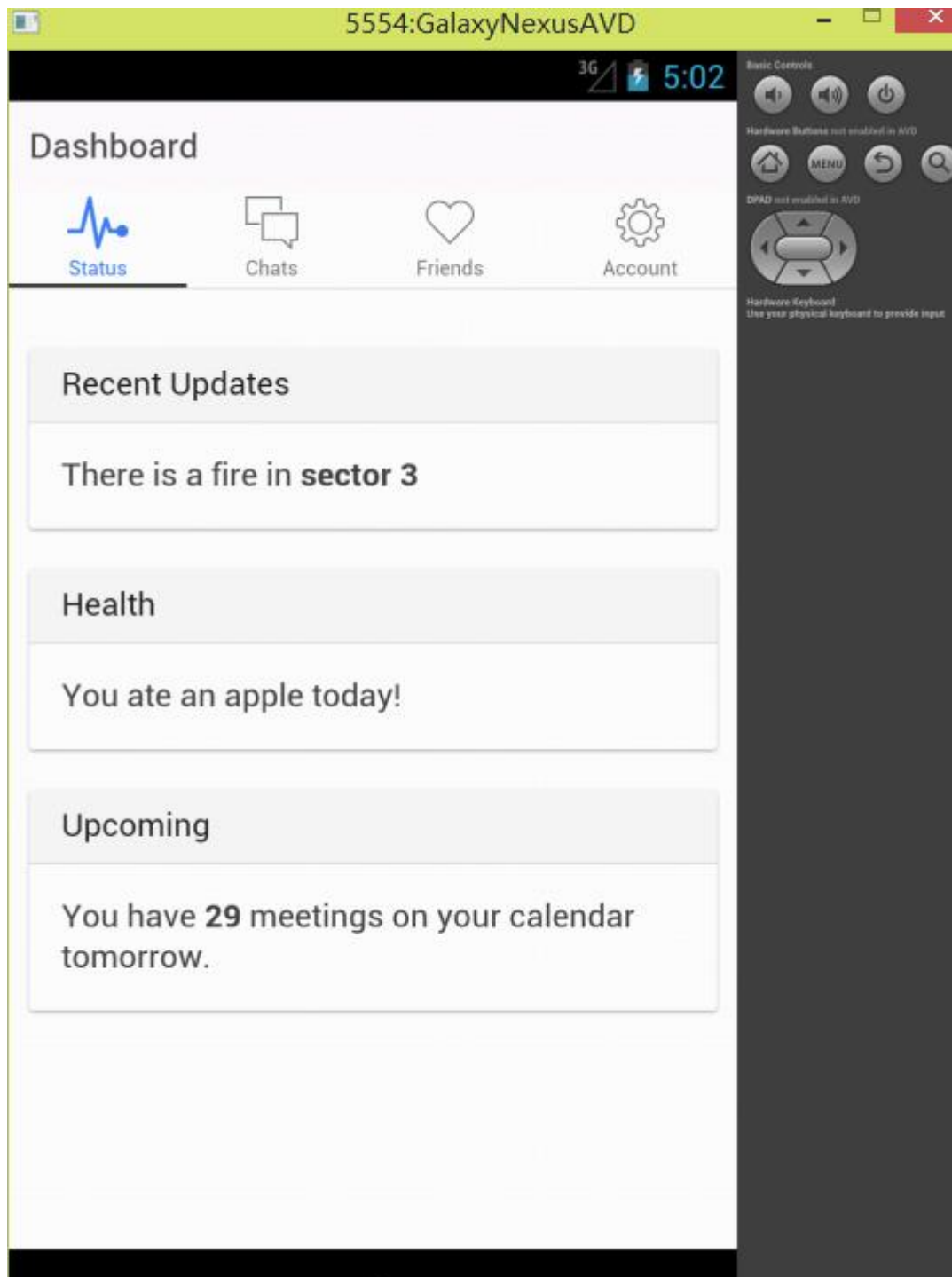


## 5 安装 APP 到模拟器

在命令提示符窗口中执行下面的操作：

```
$ cd myapp
```

```
$ ionic emulate android
```



## 6 安装 APP 到手机

```
$ cd myapp  
$ ionic run android
```

## 7 更改 app name

修改项目根目录下 config.xml 的<name>夕阳红</name>

如果是中文，必须把 config.xml 文件的编码格式设置为 utf-8



## 7 更改 app icon

```
<platform name="android">
  <icon src="res/android/ldpi.png" density="ldpi" />
  <icon src="res/android/mdpi.png" density="mdpi" />
  <icon src="res/android/hdpi.png" density="hdpi" />
  <icon src="res/android/xhdpi.png" density="xhdpi" />
</platform>
```

```
<platform name="android">
  <!-- you can use any density that exists in the Android project -->
  <splash src="res/screen/android/splash-land-hdpi.png" density="land-hdpi"/>
  <splash src="res/screen/android/splash-land-ldpi.png" density="land-ldpi"/>
  <splash src="res/screen/android/splash-land-mdpi.png" density="land-mdpi"/>
  <splash src="res/screen/android/splash-land-xhdpi.png" density="land-xhdpi"/>

  <splash src="res/screen/android/splash-port-hdpi.png" density="port-hdpi"/>
  <splash src="res/screen/android/splash-port-ldpi.png" density="port-ldpi"/>
  <splash src="res/screen/android/splash-port-mdpi.png" density="port-mdpi"/>
  <splash src="res/screen/android/splash-port-xhdpi.png" density="port-xhdpi"/>
</platform>
```

```
<preference name="SplashScreenDelay" value="10000" />
```

### 三. Cordova

## 1. Android SDK

### 1.1 简介







SDK: (software development kit) 软件开发工具包。被软件开发工程师用于为特定的软件包、[软件框架](#)、硬件平台、操作系统等建立应用软件的开发工具的集合。

因此, Android SDK 指的是 Android 专属的软件开发工具包。

### 1.2 安装 Android SDK

下载地址: <http://developer.android.com/sdk/index.html>

解压下载到的 zip 到 E:\android-sdk-windows

我的电脑 > 电脑磁盘 (E:) > android-sdk-windows		
名称	修改日期	类型
 add-ons	2014/12/18 8:05	文件夹
 platforms	2014/12/18 8:05	文件夹
 tools	2015/1/30 14:06	文件夹
 AVD Manager.exe	2014/12/18 8:05	应用程序
 SDK Manager.exe	2014/12/18 8:05	应用程序
 SDK Readme.txt	2014/12/18 8:05	文本文档

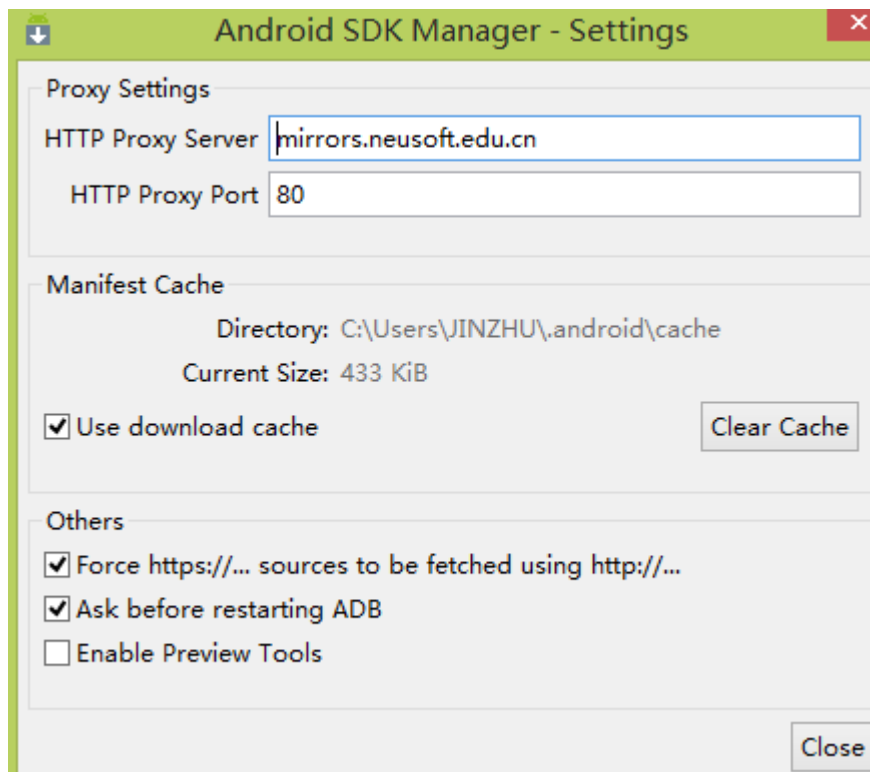
把 E:\android-sdk-windows\tools 加入到 环境变量→系统变量→PATH 中

双击 SDK Manager.exe

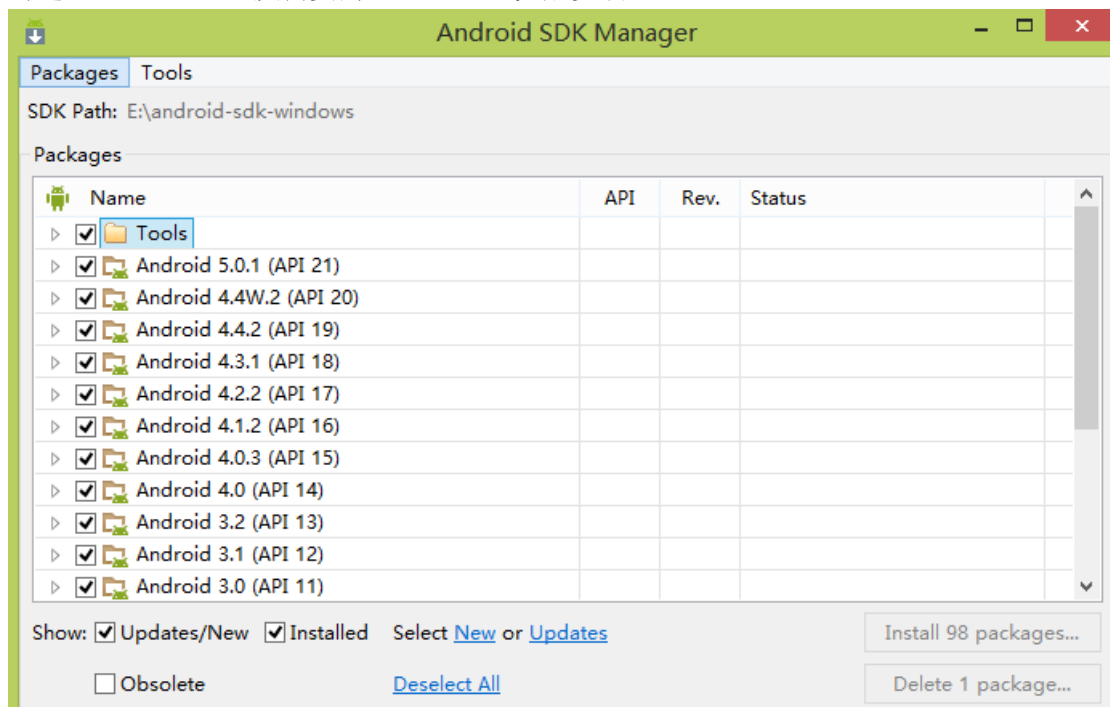
选择 Tools→Options

填写代理





勾选 Tools、Extras、及需要的 Android API 完成安装



## 2. ADT

### 2.1 ADT 简介

Android Development Tools

在 [Eclipse](#) 编译 IDE 环境中,需安装 [ADT](#)(Android Developer Tools)Plug-in,这是 Android 在 Eclipse 上的开发工具。

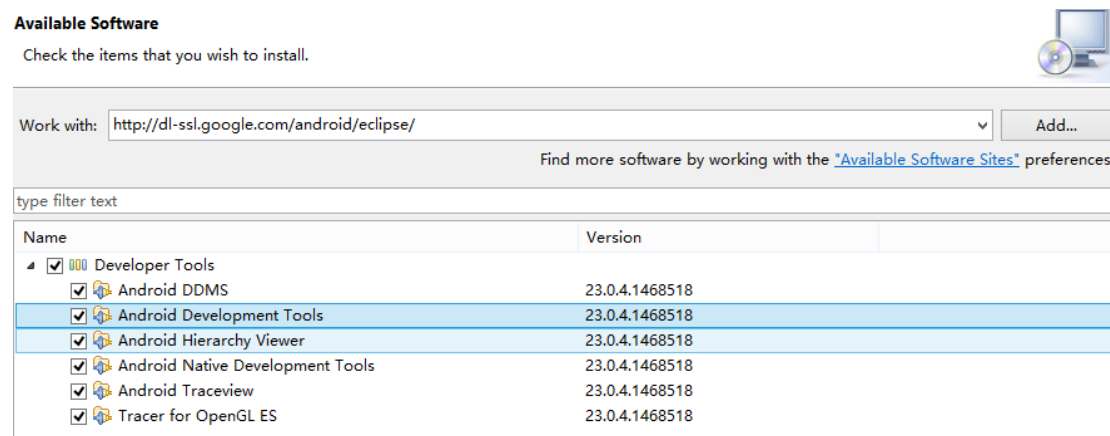
下载以及安装可以参照:Google Android ,Eclipse ,JDK.<sup>[1]</sup>

目前 [Android](#) 开发所用的开发工具是 [Eclipse](#),在 [Eclipse](#) 编译 IDE 环境中,安装 ADT,为 Android 开发提供开发工具的升级或者变更,简单理解为在 Eclipse 下开发工具的升级下载工具。

## 2.2 安装 ADT

启动 Eclipse, 点击 Help 菜单 -> Install New Software

输入地址: <http://dl-ssl.google.com/android/eclipse/>, 完成安装



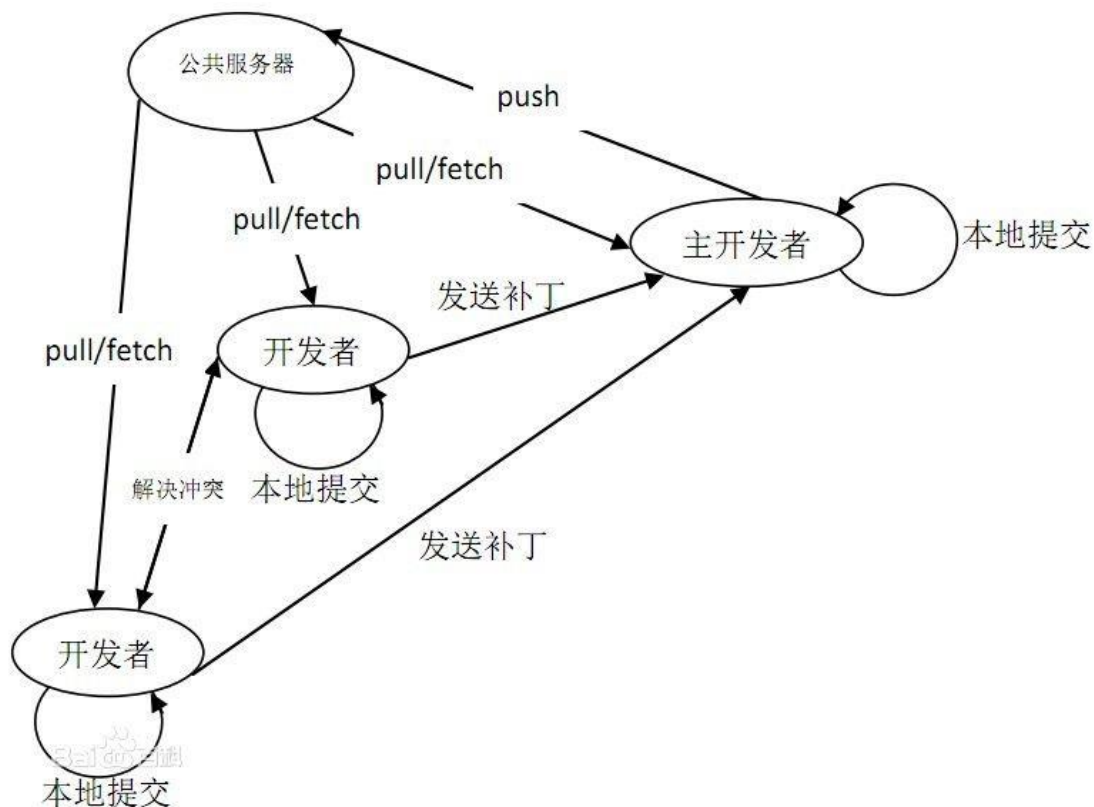
## 3. GIT

### 3.1 GIT 简介

Git 是一个开源的分布式版本控制系统,用以有效、高速的处理从很小到非常大的项目版本管理

### 3.2 特点

分布式相比于集中式的最大区别在于开发者可以提交到本地,每个开发者机器上都是一个完整的数据库。



### 3.2.1 Git 的功能特性

从一般开发者的角度来看，git 有以下功能：

- 1、从服务器上克隆数据库（包括代码和版本信息）到单机上。
- 2、在自己的机器上创建分支，修改代码。
- 3、在单机上自己创建的分支上提交代码。
- 4、在单机上合并分支。
- 5、新建一个分支，把服务器上最新版的代码 **fetch** 下来，然后跟自己的主分支合并。
- 6、生成补丁（**patch**），把补丁发送给主开发者。
- 7、看主开发者的反馈，如果主开发者发现两个一般开发者之间有冲突（他们之间可以合作解决的冲突），就会要求他们先解决冲突，然后再由其中一个人提交。如果主开发者可以自己解决，或者没有冲突，就通过。
- 8、一般开发者之间解决冲突的方法，开发者之间可以使用 **pull** 命令解决冲突，解决完冲突之后再向主开发者提交补丁。

从主开发者的角度（假设主开发者不用开发代码）看，git 有以下功能：

- 1、查看邮件或者通过其它方式查看一般开发者的提交状态。
- 2、打上补丁，解决冲突（可以自己解决，也可以要求开发者之间解决以后再重新提交，如果是开源项目，还要决定哪些补丁有用，哪些不用）。
- 3、向公共服务器提交结果，然后通知所有开发人员。

### 3.2.2 优点

适合[分布式开发](#)，强调个体。

公共服务器压力和数据量都不会太大。

速度快、灵活。

任意两个开发者之间可以很容易的解决冲突。

离线工作。

### 3.2.3 缺点

资料少（起码中文资料很少）。

学习周期相对而言比较长。

不符合常规思维。

代码保密性差，一旦开发者把整个库克隆下来就可以完全公开所有代码和版本信息。

## 3.3 安装 GIT

下载地址：<http://git-scm.com/download/win>

双击下载的 Git-1.9.5-preview20141217.exe 完成安装

将 E:\Git\bin;E:\Git\libexec\git-core 加入到 环境变量→系统变量→PATH 中

## 4. Cordova

### 4.1 Cordova 简介

Cordova 提供了一组设备相关的 API，通过这组 API，移动应用能够以 JavaScript 访问原生的设备功能，如摄像头、麦克风等。

Cordova 还提供了一组统一的 JavaScript 类库，以及为这些类库所用的设备相关的原生后台代码。





Cordova 支持如下移动操作系统：iOS, Android, ubuntu phone os, Blackberry, Windows Phone, Palm WebOS, Bada 和 Symbian。

### 4.2 安装 Cordova

在命令提示符窗口中执行下面的操作：

```
$ E:\>cd nodejs
$ E:\nodejs>npm install -g cordova
```

我的电脑 > 电脑磁盘 (E:) > nodejs > node\_modules

名称	修改日期	类型
 cordova	2015/1/30 14:58	文件夹
 express	2015/1/29 18:35	文件夹
 express-generator	2015/1/29 18:39	文件夹
 npm	2015/1/29 18:33	文件夹

## 5. Bower

### 5.1 Bower 简介

Bower 是一个 Web 开发的包管理软件。前端开发中，或多或少，都会依赖于现成的 framework 或者 lib，如 bootstrap、jquery 等。每次编写代码之前，都需要从网上下载，或者从硬盘深处找出珍藏的，项目所依赖的框架或者库文件。同时，所下载的文件，又可能依赖于另外的库文件。

Bower 的诞生，正是为了解决此繁琐的依赖关系的。

需要注意的是，Bower 的运行，依赖于版本控制工具 git，从 github 拉取依赖信息。

github 地址：<https://github.com/bower/bower>

### 5.2 安装 bower

很多前端工具，都是由 Node.js 所编写的，Bower 也不例外。因此，安装 Bower 之前，请确保已经安装好 Node.js 环境。由于 Bower 运行，需调版本控制工具 git，因此也需要确保环境已经安装了 git。

在命令提示符窗口中执行下面的操作：

```
$ E:\>cd nodejs
$ E:\nodejs>npm install -g bower
```

```
E:\>cd nodejs
E:\nodejs>npm install -g bower
```

我的电脑 > 电脑磁盘 (E:) > nodejs > node\_modules

名称	修改日期	类型
bower	2015/1/30 16:52	文件夹
cordova	2015/1/30 14:58	文件夹
express	2015/1/29 18:35	文件夹
express-generator	2015/1/29 18:39	文件夹
ionic	2015/1/30 15:22	文件夹
npm	2015/1/29 18:33	文件夹

## 6. 创建 APP

在命令提示符窗口中执行下面的操作：

```
$ F:\>cordova create helloworld com.helloworld.hello HelloWorld
```

```
F:\>cordova create helloworld com.helloworld.hello HelloWorld
Creating a new cordova project with name "HelloWorld" and id "com.helloworld.hello" at location "F:\helloworld"
Downloading cordova library for www...
Download complete
```

成功之后将在 F 盘生成 helloworld 目录：

我的电脑 > 电脑磁盘 (F:) > helloworld

名称	修改日期	类型
hooks	2015/1/30 15:23	文件夹
platforms	2015/1/30 15:23	文件夹
plugins	2015/1/30 15:23	文件夹
www	2015/1/30 15:23	文件夹
config.xml	2015/1/30 15:23	XML 文件

参数说明：

第一个参数 “helloworld”：

项目目录名称，此目录不可以事先存在，指令会自动创建此目录

第二个参数 “com.helloworld.hello”：

给项目提供一个反向域标识符，类似于 java 项目中的包名称。

如果您不写第三个参数，那么第二个参数是可选的，默认值 “io.cordova.hellocordova”，您也可以在 config.xml 中修改它

第三个参数 “HelloWorld”：

应用程序显示的 title，此参数是可选的，默认值 “HelloCordova”，

您可以在 config.xml 中修改它

```
<?xml version='1.0' encoding='utf-8'?>
<widget id="com.helloworld.hello" version="0.0.1" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/ns/1.0">
  <name>HelloWorld</name>
  <description>
    A sample Apache Cordova application that responds to the deviceready event.
  </description>
  <author email="dev@cordova.apache.org" href="http://cordova.io">
    Apache Cordova Team
  </author>
  <content src="index.html" />
  <access origin="*" />
</widget>
```

## 7. 添加平台支持















所有的命令需要在项目目录，或者子目录下执行  
在命令提示符窗口中执行下面的操作：

```
$ F:\>cd helloworld
```

在您 build 项目之前，您需要设置目标平台  
在执行命令之前，首先确认您已经安装相应 SDK

```
F:\>cd helloworld
F:\helloworld>cordova platform add android
```

命令执行成功：

我的电脑 > 电脑磁盘 (F:) > helloworld > platforms > android		
名称	修改日期	类型
 assets	2015/2/2 13:12	文件夹
 cordova	2015/2/2 13:12	文件夹
 CordovaLib	2015/2/2 13:12	文件夹
 libs	2015/2/2 13:12	文件夹
 platform_www	2015/2/2 13:12	文件夹
 res	2015/2/2 13:12	文件夹
 src	2015/2/2 13:12	文件夹
 .gitignore	2015/2/2 13:12	文本文档
 .project	2015/2/2 13:12	PROJECT 文件
 AndroidManifest.xml	2015/2/2 13:12	XML 文件
 build.gradle	2015/2/2 13:12	GRADLE 文件
 custom_rules.xml	2015/2/2 13:12	XML 文件
 project.properties	2015/2/2 13:12	PROPERTIES 文件
 settings.gradle	2015/2/2 13:12	GRADLE 文件

Mac 系统：

```
$ cordova platform add ios
$ cordova platform add amazon-fireos
$ cordova platform add android
$ cordova platform add blackberry10
$ cordova platform add firefoxos
```

Windows 系统:

```
$ cordova platform add wp8
$ cordova platform add windows
$ cordova platform add amazon-fireos
$ cordova platform add android
$ cordova platform add blackberry10
$ cordova platform add firefoxos
```

检查当前平台

在命令提示符窗口中执行下面的操作:

```
$ cordova platforms is
```

```
F:\helloworld>cordova platforms is
Installed platforms:
Available platforms: amazon-fireos, android, blackberry10, browser, firefoxos, w
indows, windows8, wp8
```

移除平台

```
$ cordova platform remove blackberry10
$ cordova platform rm amazon-fireos
$ cordova platform rm android
```

## 8. Build APP

在命令提示符窗口中执行下面的操作:

```
$ cd helloworld
```

```
$ cordova build android
```

同样的你可以 build 提前平台:

```
$ cordova build ios
```

## 9. 准备和编译 APP

在命令提示符窗口中执行下面的操作:

```
$ cd helloworld
```

```
$ cordova prepare android
```

```
$ cordova compile android
```

同样的你可以编译其他平台:

```
$ cordova prepare ios
$ cordova compile ios
```

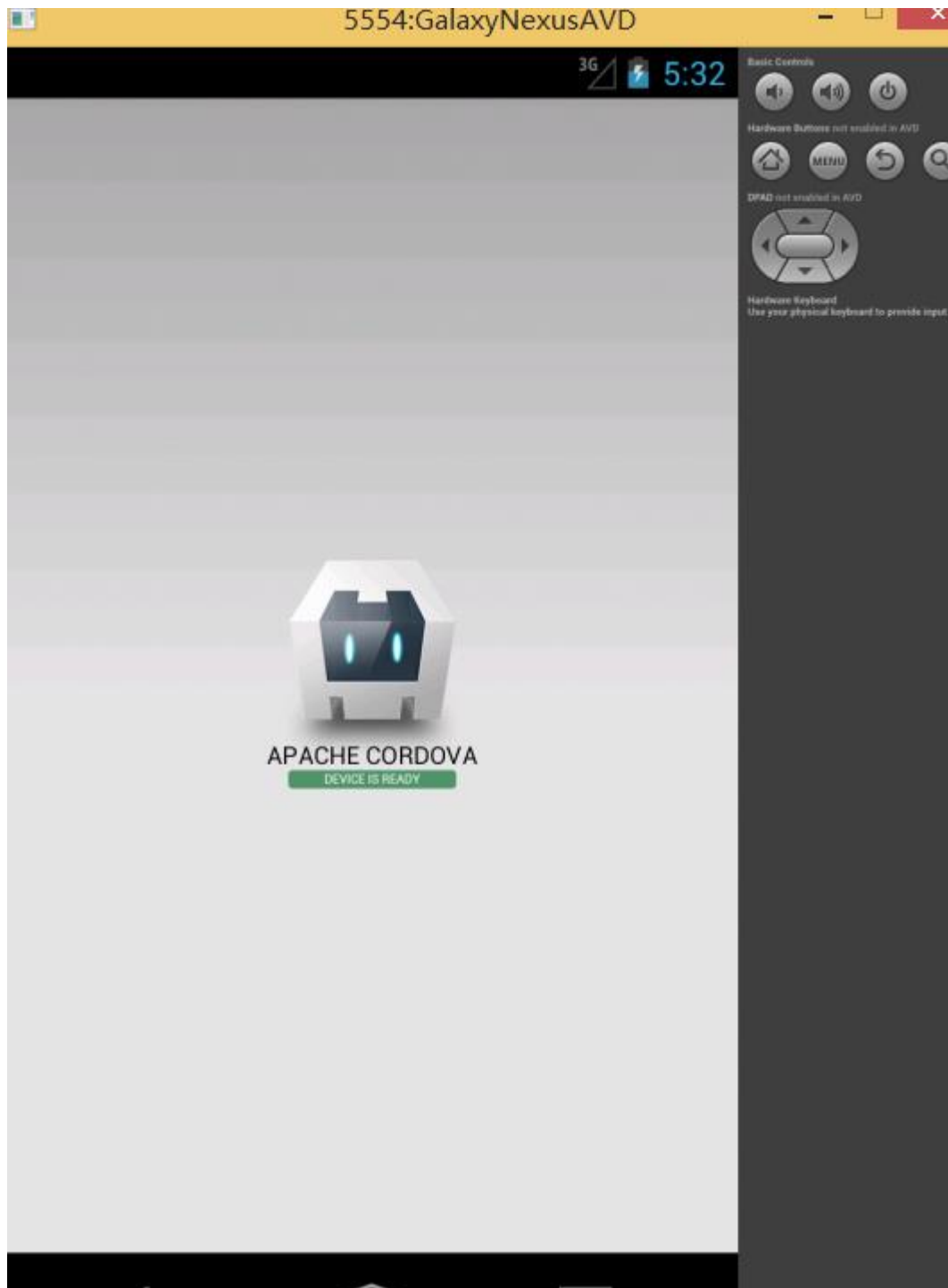
## 10. 安装 APP 到模拟器

在命令提示符窗口中执行下面的操作:



```
$ cd helloworld  
$ cordova emulate android
```

模拟器启动成功：



## 11. 安装 APP 到手机

用 USB 线连接手机和电脑，同时手机设置允许 **USB debugging**  
在命令提示符窗口中执行下面的操作：

```
$ cd helloworld
```

```
$ cordova run android
```

## 12. 添加插件功能

查找已注册的 bar code 插件 (registry.cordova.io)

在命令提示符窗口中执行下面的操作：

```
$ cd helloworld
```

```
$ cordova plugin search bar code
```

```
F:\helloworld>cordova plugin search bar code
com.blackberry.community.barcode scanner - BlackBerry 10 Community Contributed AP
I to scan barcodes
com.gpdi.plugin.myplugin.myplugin - You can use the BarcodeScanner plugin to sca
n different types of barcodes (using the device's camera) and get the metadata e
ncoded in them for processing within your application.
com.manateeworks.barcode scanner - Manateeworks Barcode Scanner Plugin
com.mirasense.scanditsdk.plugin - BarcodeScannerPlugin =====
com.phonegap.plugins.barcode scanner - You can use the BarcodeScanner plugin to s
can different types of barcodes (using the device's camera) and get the metadata
encoded in them for processing within your application.
com.virtualama.bs - Scans Barcodes.
org.bloxlab.barcode scanner - Scans barcodes using the camera for WP8 devices.
org.cloudsky.cordova plugins.zbar - Plugin to integrate with the ZBar barcode sca
nning library.
org.pluginporo.barcode_scan_plugin - Barcode Scan Plugin for Cordova
org.pluginporo.honeywell_scanner_plugin - Honeywell Barcode Scanner Plugin
```

安装 bar code 插件

在命令提示符窗口中执行下面的操作：

```
$ cd helloworld
```

```
$ cordova plugin add com.phonegap.plugins.barcode scanner
```

安装成功后：

我的电脑 > 电脑磁盘 (F:) > helloworld > plugins		
名称	修改日期	类型
com.phonegap.plugins.barcode scanner	2015/2/2 15:32	文件夹
org.apache.cordova.device	2015/2/2 15:23	文件夹
org.apache.cordova.statusbar	2015/2/2 15:26	文件夹
android.json	2015/2/2 15:32	JSON 文件

根据需要进行安装如下插件：

- Basic device information (Device API):

```
$ cordova plugin add org.apache.cordova.device
```

- Network Connection and Battery **Events**:

```
$ cordova plugin add org.apache.cordova.network-information
$ cordova plugin add org.apache.cordova.battery-status
```

- Accelerometer, Compass, and Geolocation:

```
$ cordova plugin add org.apache.cordova.device-motion
$ cordova plugin add org.apache.cordova.device-orientation
$ cordova plugin add org.apache.cordova.geolocation
```

- Camera, Media playback and Capture:

```
$ cordova plugin add org.apache.cordova.camera
$ cordova plugin add org.apache.cordova.media-capture
$ cordova plugin add org.apache.cordova.media
```

- Access files on device or network (File API):

```
$ cordova plugin add org.apache.cordova.file
$ cordova plugin add org.apache.cordova.file-transfer
```

- Notification via dialog box or vibration:

```
$ cordova plugin add org.apache.cordova.dialogs
$ cordova plugin add org.apache.cordova.vibration
```

- Contacts:

```
$ cordova plugin add org.apache.cordova.contacts
```

- Globalization:

```
$ cordova plugin add org.apache.cordova.globalization
```

- Splashscreen:

```
$ cordova plugin add org.apache.cordova.splashscreen
```

- Open new browser windows (InAppBrowser):

```
$ cordova plugin add org.apache.cordova.inappbrowser
```

- Debug console:

```
$ cordova plugin add org.apache.cordova.console
```

## 13.查看已安装插件

在命令提示符窗口中执行下面的操作：

```
$ cordova plugin is
```

或者

```
$ cordova plugin list
```

```
F:\helloworld>cordova plugin is
com.phonegap.plugins.barcodescanner 2.0.1 "BarcodeScanner"
org.apache.cordova.device 0.2.13 "Device"
org.apache.cordova.statusbar 0.1.9 "StatusBar"

F:\helloworld>cordova plugin list
com.phonegap.plugins.barcodescanner 2.0.1 "BarcodeScanner"
org.apache.cordova.device 0.2.13 "Device"
org.apache.cordova.statusbar 0.1.9 "StatusBar"
```

## 14.移除插件

移除单个插件

在命令提示符窗口中执行下面的操作：

```
$ cd helloworld
```

```
$ cordova plugin rm org.apache.cordova.console
```

或者

```
$ cordova plugin remove org.apache.cordova.console
```

批量移除插件

在命令提示符窗口中执行下面的操作：

```
$ cd hellworld
$ cordova plugin remove org.apache.cordova.console org.apache.cordova.device
```

## 15. 高级插件选项

通过在 `registry.cordova.io` 注册的已知 ID 来安装插件：

```
$ cordova plugin add org.apache.cordova.console
```

批量安装插件：

```
$ cordova plugin add org.apache.cordova.console org.apache.cordova.device
```

安装指定版本的插件，格式：ID@VERSION

```
$ cordova plugin add org.apache.cordova.console@latest
$ cordova plugin add org.apache.cordova.console@0.2.1
```

通过 URL 安装 GIT 资源库中主分支插件：

```
$ cordova plugin add https://github.com/apache/cordova-plugin-console.git
```

URL#字符串，可以安装 GIT 资源库某个标记或者分支插件：

```
$ cordova plugin add https://github.com/apache/cordova-plugin-console.git#r0.2.0
```

如果插件在 GIT 资源库的子目录，可以通过#字符串 来访问

```
$ cordova plugin add https://github.com/someone/aplugin.git#/my/sub/dir
```

也可以安装某个标记或者分支下的子目录插件

```
$ cordova plugin add https://github.com/someone/aplugin.git#r0.0.1:/my/sub/dir
```

也可以安装本地目录插件

```
$ cordova plugin add ../my_plugin_dir
```

## 四. Bootstrap

### 1. Bootstrap 简介

Bootstrap 是最受欢迎的 HTML、CSS 和 JS 框架，用于开发响应式布局、移动设备优先的 WEB 项目

#### 1.1 Bootstrap 特点

Bootstrap 是基于 [HTML5](#) 和 [CSS3](#) 开发的，它在 [jQuery](#) 的基础上进行了更为个性化和人性化

的完善，形成一套自己独有的网站风格，并兼容大部分 jQuery 插件。

## 1.2 Bootstrap 组件

Bootstrap 中包含了丰富的 Web 组件，根据这些组件，可以快速的搭建一个漂亮、功能完备的网站。其中包括以下组件：

下拉菜单、按钮组、按钮下拉菜单、导航、导航条、面包屑、分页、排版、缩略图、警告对话框、进度条、媒体对象等

## 1.3 Javascript 插件

Bootstrap 自带了 13 个 [jQuery](#) 插件，这些插件为 Bootstrap 中的组件赋予了“生命”。

其中包括：

模式对话框、标签页、滚动条、弹出框等。

## 2. 安装 Bootstrap




用于生产环境的 bootstrap: <http://d.bootcss.com/bootstrap-3.3.2-dist.zip>

Bootstrap 源码: <http://d.bootcss.com/bootstrap-3.3.2.zip>

参考: <http://v3.bootcss.com/getting-started/>

下载压缩包之后，将其解压到 E 盘：

我的电脑 > 电脑磁盘 (E:) > bootstrap-3.3.2-dist

名称	修改日期	类型
 css	2015/2/3 15:31	文件夹
 fonts	2015/2/3 15:31	文件夹
 js	2015/2/3 15:31	文件夹

### 2.1 通过 bower 安装










```
$ cd nodejs
```

```
$ bower install bootstrap
```

### 2.2 通过 npm 进行安装

```
$ cd nodejs
```

```
$ npm install bootstrap
```

我的电脑 > 电脑磁盘 (E:) > nodejs > node_modules		
名称	修改日期	类型
 angular-touch	2015/1/30 17:17	文件夹
 bootstrap	2015/2/3 15:53	文件夹
 bower	2015/1/30 16:52	文件夹
 cordova	2015/1/30 14:58	文件夹
 express	2015/1/29 18:35	文件夹
 express-generator	2015/1/29 18:39	文件夹
 ionic	2015/1/30 15:22	文件夹
 jquery	2015/1/30 17:22	文件夹
 npm	2015/1/29 18:33	文件夹

### 3. 目录结构

#### 3.1 预编译版

```
bootstrap/
├── css/
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap-theme.css
│   ├── bootstrap-theme.css.map
│   └── bootstrap-theme.min.css
├── js/
│   ├── bootstrap.js
│   └── bootstrap.min.js
└── fonts/
    ├── glyphicons-halflings-regular.eot
    ├── glyphicons-halflings-regular.svg
    ├── glyphicons-halflings-regular.ttf
    ├── glyphicons-halflings-regular.woff
    └── glyphicons-halflings-regular.woff2
```

预编译文件可以直接使用到任何 **web** 项目中。我们提供了编译好的 **CSS** 和 **JS** (`bootstrap.*`) 文件，还有经过压缩的 **CSS** 和 **JS** (`bootstrap.min.*`) 文件。同时还提供了 **CSS 源码映射表** (`bootstrap.*.map`)，可以在某些浏览器的开发工具中使用。同时还包含了来自 **Glyphicons** 的图标字体，在附带的 **Bootstrap** 主题中使用到了这些图标。

## 3.2 Bootstrap 源码

Bootstrap 源码包含了预先编译的 CSS、JavaScript 和图标字体文件，并且还有 LESS、JavaScript 和文档的源码。具体来说，主要文件组织结构如下：

```
bootstrap/
├── less/
├── js/
├── fonts/
├── dist/
│   ├── css/
│   ├── js/
│   └── fonts/
├── docs/
└── examples/
```

less/、js/ 和 fonts/ 目录分别包含了 CSS、JS 和字体图标的源码。dist/ 目录包含了上面所说的预编译 Bootstrap 包内的所有文件。docs/ 包含了所有文档的源码文件，examples/ 目录是 Bootstrap 官方提供的实例工程。除了这些，其他文件还包含 Bootstrap 安装包的定义文件、许可证文件和编译脚本等。

## 4. GRUNT

### 4.1 Grunt 简介

JavaScript 世界的构建工具

为何要用构建工具？

一句话：自动化。对于需要反复重复的任务，例如压缩（minification）、编译、单元测试、linting 等，自动化工具可以减轻你的劳动，简化你的工作。当你正确配置好了任务，任务运行器就会自动帮你或你的小组完成大部分无聊的工作。

参考：<http://www.gruntjs.net/>

### 4.2 Grunt 安装

在命令行中输入以下命令：

```
$ cd nodejs
```



```
$ npm install -g grunt-cli
```

```
E:\nodejs>npm install -g grunt-cli
E:\nodejs\grunt -> E:\nodejs\node_modules\grunt-cli\bin\grunt
grunt-cli@0.1.13 E:\nodejs\node_modules\grunt-cli
├── resolve@0.3.1
├── nopt@1.0.10 <abbrev@1.0.5>
├── findup-sync@0.1.3 <lodash@2.4.1, glob@3.2.11>
```

进入 /bootstrap/ 根目录，然后执行 npm install 命令：

```
$ cd node_modules
```

```
$ cd bootstrap
```

```
$ npm install
```

## 4.3 可用的 Grunt 命令

**grunt dist** （仅编译 CSS 和 JavaScript 文件）

重新生成 /dist/ 目录，并将编译压缩后的 CSS 和 JavaScript 文件放入这个目录中。作为一名 Bootstrap 用户，大部分情况下你只需要执行这一个命令。

**grunt watch** （监测文件的改变，并运行指定的 Grunt 任务）

监测 Less 源码文件的改变，并自动重新将其编译为 CSS 文件。

**grunt test** （运行测试用例）

在 [PhantomJS](#) 环境中运行 [JSHint](#) 和 [QUnit](#) 自动化测试用例。

**grunt docs** （编译并测试文档中的资源文件）

编译并测试 CSS、JavaScript 和其他资源文件。在本地环境下通过 `jeekyll serve` 运行 Bootstrap 文档时需要用到这些资源文件。

**grunt** （重新构建所有内容并运行测试用例）

编译并压缩 CSS 和 JavaScript 文件、构建文档站点、对文档做 HTML5 校验、重新生成定制工具所需的资源文件等，都需要 [Jeekyll](#) 工具。这些只有在你对 Bootstrap 深度研究时才有用。

## 除错

如果你在安装依赖包或者运行 Grunt 命令时遇到了问题，请首先删除 npm 自动生成的 `/node_modules/` 目录，然后，再次运行 `npm install` 命令。

# 五. AngularJS

## 1. AngularJS 简介

Angular JS (AngularJS) 是一组用来开发 Web 页面的框架、模板以及数据绑定和丰富 UI 组件。它支持整个开发进程，提供 web 应用的架构，无需进行手工 DOM 操作。AngularJS 很小，只有 60K，兼容主流浏览器，与 jQuery 配合良好。

## 2. AngularJS 特性

### 2.1 特性一：数据绑定

数据绑定可能是 AngularJS 最酷最实用的特性。它能够帮助你避免书写大量的初始代码从而节约开发时间。一个典型的 web 应用可能包含了 80%的代码用来处理，查询和监听 DOM。数据绑定使得代码更少，你可以专注于你的应用。

传统来说，当 model 变化了。开发人员需要手动处理 DOM 元素并且将属性反映到这些变化中，这是一个双向的过程。一方面，model 变化驱动了 DOM 中元素变化，另一方面，DOM 元素的变化也会影响到 Model。这个在用户互动中更加复杂，因为开发人员需要处理和解析这些互动，然后融合到一个 model 中，并且更新 View。这是一个手动的复杂过程，当一个应用非常庞大的时候，将会是一件非常费劲的事情。

### 2.2 特性二：模板

在 AngularJS 中，一个模板就是一个 HTML 文件。但是 HTML 的内容扩展了，包含了很多帮助你映射 model 到 view 的内容。

HTML 模板将会被浏览器解析到 DOM 中。DOM 然后成为 AngularJS 编译器的输入。AngularJS 将会遍历 DOM 模板来生成一些指导，即，directive（指令）。所有的指令都负责针对 view 来设置数据绑定。

我们要理解 AngularJS 并不把模板当做 String 来操作。输入 AngularJS 的是 DOM 而非 string。数据绑定是 DOM 变化，不是字符串的连接或者 innerHTML 变化。使用 DOM 作为输入，而不是字符串，是 AngularJS 区别于其它的框架的最大原因。使用 DOM 允许你扩展指令词汇并且可以创建你自己的指令，甚至开发可重用的组件。

## 2.3 特性三：MVC

针对客户端应用开发 AngularJS 吸收了传统的 MVC 基本原则。MVC 或者 Model-View-Controll 设计模式针对不同的人可能意味不同的东西。AngularJS 并不执行传统意义上的 MVC，更接近于 MVVM (Model-View-ViewModel)。

## 2.4 特性四：依赖注入 (Dependency Injection, 即 DI)

AngularJS 拥有内建的依赖注入子系统，可以帮助开发人员更容易的开发，理解和测试应用。

DI 允许你请求你的依赖，而不是自己找寻它们。比如，我们需要一个东西，DI 负责找创建并且提供给我们。

## 2.5 特性五：Directives (指令)

指令是我们最喜欢的特性。你是不是也希望浏览器可以做点儿有意思的事情？那么 AngularJS 可以做到。

指令可以用来创建自定义的标签。它们可以用来装饰元素或者操作 DOM 属性。