开发环境配置步骤

1. 下载ADT和IDE

<http://developer.android.com/tools/index.html>

1. phonegap run android

Run the "android" command from your adt\sdk\tools folder and install the latest Tools and SDK. Also make sure your PATH has the right variables.

For this you will need ANT to be installed , a JAVA JDK and an Android SDK installed

JAVA\_HOME (C:\Program Files\Java\jdk)  
ANT\_HOME ({ant location}\apache\apache-ant)  
ANDROID\_HOME ({android sdk location}\android-sdk)

Add these to your PATH variable like

%ANT\_HOME%/bin;%ANDROID\_HOME%\tools;%ANDROID\_HOME%\platform-tools;%JAVA\_HOME%\bin

Close and re-open your cmd and run your command again.

Similiar to [PhoneGap/Cordova Android Development](http://stackoverflow.com/questions/20576704/phonegap-cordova-android-development/21613524" \l "21613524)

1. phonegap build android

# Use grunt-angular-phonegap create phone app

## Getting started

This plugin requires Grunt ~0.4.1

**WARNING :** This plugin is still in early alpha. I work on Ubuntu, other operating systems are not tested. Let me know if you need help by writing me at [dsimard@azanka.ca](mailto:dsimard@azanka.ca).

First, make sure that you installed the proper SDK. Supported platforms are : [Android](https://developer.android.com/sdk/index.html), iOS and WindowsPhone

Globally install phonegap and [yeoman/generator-angular](https://github.com/yeoman/generator-angular) :

npm install -g phonegap cordova generator-angular

Create your phonegap project :

phonegap create --name MyApp --id com.yourcompany.myapp myapp && cd myapp

Use the last version of phonegap by adding this line in myapp/www/config.xml :

<preference name="phonegap-version" value="3.1.0" />

Initialize an angular application with yeoman :

yo angular [myapp]

Add this project to your package.json :

npm install grunt-angular-phonegap --save-dev

Add a platform (android, ios, wp7 or wp8) :

cordova platform add [platform]

Check that everything works well :

grunt phonegap:check

## Usage

[platform] supported are : android, ios, wp7 and wp8. I only test with android.

Default is always android

### grunt phonegap:check[:platform]

Check if your computer is ready for PhoneGap development.

### grunt phonegap:build[:platform]

Build the app locally in www.

Use --no-bower to avoid copying the bower\_components directory

Build之后用命令***phonegap install android***进行安装。

grunt phonegap:build:android --no-bower

phonegap install android

### grunt phonegap:emulate[:platform][:emulator]

Start an emulator. If there is one already running, it will be used.

Use :emulator to specify an emulator already running.

### grunt phonegap:send[:platform]

Send the project to remotely build at [http://build.phonegap.com](http://build.phonegap.com/).

NOTE : Before remotely building, you have to login by executing phonegap remote login --username you@gmail.com --password YourPassword

phonegap remote login --username anypossible.w@gmail.com --password String0int

## cordova plugin

不需要手工更改config.xml文件,自动化了.

### install camera plugin

$ cordova plugin add org.apache.cordova.camera

### Look the plugin

$ cordova plugin ls  
    [ 'org.apache.cordova.camera' ]

### Remove the plugin

$ cordova plugin rm org.apache.cordova.camera

### File plugin

    $ cordova plugin add https://git-wip-us.apache.org/repos/asf/cordova-plugin-file.git  
    $ cordova plugin rm org.apache.cordova.core.file

To use the file-transfer plugin you must add that separately.

    $ cordova plugin add https://git-wip-us.apache.org/repos/asf/cordova-plugin-file-transfer.git  
    $ cordova plugin rm org.apache.cordova.core.file-transfer

### Plugins

cordova plugin add org.apache.cordova.camera

cordova plugin add org.apache.cordova.file

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

cordova plugin add org.apache.cordova.geolocation

cordova plugin add <https://github.com/driftyco/ionic-plugins-keyboard.git>

cordova plugin add org.apache.cordova.splashscreen

cordova plugin add org.apache.cordova.network-information

remove some plugin

cordova plugin rm org.apache.cordova.camera

cordova plugin rm com.ionic.keyboard

## Yeoman相关操作

### 起个Node服务器

yeoman内置了Node服务器服务，而且会监听工程目录下的文件的改变，一旦文件发生改变会重新编译文件（sass/less/coffscript）。

下面轮到**grunt**上场，在工程目录下，运行命令：

|  |
| --- |
| grunt server |
| The `server` task has been deprecated. Use `***grunt serve***` to start a server. |

启动成功后会自动打开本地浏览器，默认地址为**http://localhost:9000/**，9000端口号，可能被占用，这时候打开gruntfile.js文件（grunt的任务配置文件），找到：

|  |
| --- |
|  |

修改下**port**，即可。

## Add angular-mobile-nav

|  |
| --- |
| bower install angular-mobile-nav |

并且再把此组件依赖添加到bower.json文件，或者添加后在用bower install来安装。

未使用此组件

## 改用bower install ionic

bower install angular-ui-router

bower install angular-animate

bower install angular-sanitize

bower install angular-ui-utils#bower-event

## debug phonegap on android device

adb logcat Cordova:\* DroidGap:\* CordovaLog:\* \*:S

## 起个node服务器，默认路径为app

grunt serve

## 起个node服务器，路径为dist，压缩后的app目录dist

grunt serve:dist

## 关闭浏览器CORS

### 命令行运行chrome关闭cross domain限制

C:\Users\any>"C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" --args --disable-web-security

### Mozilla FireFox

FireFox的關閉方法也很簡單，在網址列輸入**about:config**後，搜尋**security.fileuri.strict\_origin\_policy**，然後把值value改成「false」即可。

### Microsoft Internet Explorer（IE）disable ACAO

這個很簡單，只要進去IE的網際網路選項的設定中，切到安全性＞自訂等級，然後下拉到雜項區，找到「存取跨網域的資料來源」，點選「啟用」後就可以了。

# iOS

<access origin="yahoo.com" subdomains="true" />

but its not working at all.. every domain is being allowed and i do not know why.

There are 2 config.xml

1. /YourAppName/config.xml <-- check if you add the access here
2. /YourAppName/ww/config.xml

I had the same problem with you (using phongap 3.0), and it is solved by adding access in the 1st file.

# Angularjs

## How directive access controller scope

<https://egghead.io/lessons/angularjs-directives-talking-to-controllers>

## Where Should Model State Be Stored In Angular.js

<http://stackoverflow.com/questions/16695926/cache-dom-in-angular-js>

## How to prevent browser to invoke basic auth popup and handle 401 error

<http://stackoverflow.com/questions/1616277/ajax-authentication-without-letting-browser-pop-up-login-dialog>

## How can I make angular.js post data as form data instead of a request payload?

Need headers:

**{'Content-Type': 'application/x-www-form-urlencoded'}**

and the post data is:

**$.param({key: "value"});**

<http://stackoverflow.com/questions/11442632/how-can-i-make-angular-js-post-data-as-form-data-instead-of-a-request-payload>

# Anuglarjs ui-router

# Ripple Emulator

# Git command

**git add some-file**

**git commit -m "Refactor to simplify"**

**git push**

git remote add origin https://{username}:{password}@github.com/{username}/project.git

**git reset --hard 4a155e5**

返回到之前的一个commit

## tag for github

You can create tags for GitHub by either using:

* the Git command line, or
* GitHub's web interface.

### Creating tags from the command line

You need to explicitly say that you want to push your tags to your remote repo:

git push origin --tags

From the [official Linux Kernel Git documentation for git push](https://www.kernel.org/pub/software/scm/git/docs/git-push.html):

--tags

All refs under refs/tags are pushed, in addition to refspecs explicitly listed on the command line.

Or if you just want to push a single tag:

git push origin <tag>

See also my answer to [Push a tag to a remote repository using Git?](http://stackoverflow.com/questions/5195859/push-a-tag-to-a-remote-repository-using-git/23217431#23217431) for more details about that syntax above.

# NPM

## Npm的配置管理及设置代理

npm全称为Node Packaged Modules。它是一个用于管理基于node.js编写的package的命令行工具。其本身就是基于node.js写的,这有点像gem与ruby的关系。

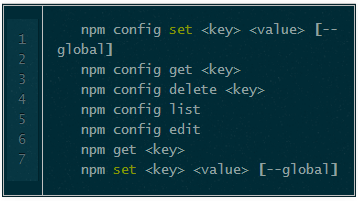
在我们的项目中，需要使用一些基于node.js的javascript库文件，就需要npm对这些依赖库进行方便的管理。由于我们的开发环境由于安全因素在访问一些网站时需要使用代理，其中就包括npm的repositories网站，所以就需要修改npm的配置来加入代理。

下面简要介绍下npm的配置以及如何设置代理。

### npm获取配置有6种方式，优先级由高到底。

1. 命令行参数。 --proxy http://server:port即将proxy的值设为  
   --proxy http://127.0.0.1:8087。
2. 环境变量。 以npm\_config\_为前缀的环境变量将会被认为是npm的配置属性。如设置proxy可以加入这样的环境变量npm\_config\_proxy=http://server:port。
3. 用户配置文件。可以通过npm config get userconfig查看文件路径。如果是mac系统的话默认路径就是$HOME/.npmrc。
4. 全局配置文件。可以通过npm config get globalconfig查看文件路径。mac系统的默认路径是/usr/local/etc/npmrc。
5. 内置配置文件。安装npm的目录下的npmrc文件。
6. 默认配置。 npm本身有默认配置参数，如果以上5条都没设置，则npm会使用默认配置参数。

### 针对npm配置的命令行操作

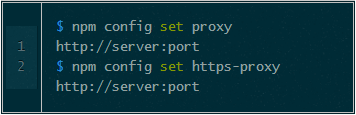


在设置配置属性时属性值默认是被存储于用户配置文件中，如果加上--global，则被存储在全局配置文件中。

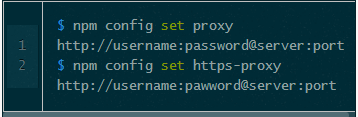
如果要查看npm的所有配置属性（包括默认配置），可以使用npm config ls -l。

如果要查看npm的各种配置的含义，可以使用npm help config。

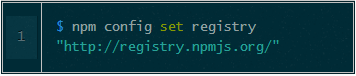
### 为npm设置代理



如果代理需要认证的话可以这样来设置。



如果代理不支持https的话需要修改npm存放package的网站地址。



## NPM国内镜像

镜像使用方法（三种办法任意一种都能解决问题，建议使用第三种，将配置写死，下次用的时候配置还在）:

1.通过config命令

npm config set registry http://registry.cnpmjs.org

npm info underscore （如果上面配置正确这个命令会有字符串response）

2.命令行指定

npm --registry http://registry.cnpmjs.org install

3.编辑 ~/.npmrc 加入下面内容

registry = http://registry.cnpmjs.org

搜索镜像: [http://cnpmjs.org](http://cnpmjs.org/)

建立或使用镜像,参考: <https://github.com/fenmgk2/cnpmjs.org>

# bower

## Registering packages

To register a new package:

* There **must** be a valid manifest JSON in the current working directory.
* Your package should use [semver](http://semver.org/) Git tags.
* Your package **must** be available at a Git endpoint (e.g., GitHub); remember to push your Git tags!

Then use the following command:

bower register <my-package-name> <git-endpoint>

The Bower registry does not have authentication or user management at this point in time. It's on a first come, first served basis. Think of it like a URL shortener. Now anyone can run bower install <my-package-name>, and get your library installed.

There is no direct way to unregister a package yet. For now, you can [request a package be unregistered](https://github.com/bower/bower/issues/120).

Example:

*bower register angular-ui-mapgaode* [*https://github.com/anypossiblew/ui-map-gaode.git*](https://github.com/anypossiblew/ui-map-gaode.git)

*bower register angular-ui-map-qq* [*https://github.com/anypossiblew/ui-map-qq.git*](https://github.com/anypossiblew/ui-map-qq.git)

*bower register angular-ui-map-baidu https://github.com/anypossiblew/ui-map-baidu.git*

## bower-register-new-version

问题：bower ENORESTARGET No tag found that was able to satisfy ~0.1.0

Additional error details:

No versions found in git://github.com/anypossiblew/ui-map-gaode.git

*解决：*[*http://stackoverflow.com/questions/19456091/bower-register-new-version*](http://stackoverflow.com/questions/19456091/bower-register-new-version)

Here's the quick and easy way to solve your problem... from your project's root directory run:

% git tag v1.0.4

If you type:

% git tag

you'll see the version you just created. Now push your Git tags:

% git push origin v1.0.4

Now when you run bower info angular-backstretch-adrr, you'll see the version you just pushed.

## Register your package on Bower

Once you are happy with your package, tag the release with a semver version matching your bower.json file. For example:

git tag -a 1.2.1 -m "Tagging 1.2.1"

You can then register your package on Bower using the bower register command:

bower register [app name] [git endpoint]

## bower proxy

<http://www.html-js.com/article/Methods-Laispace-gitnpmbowergem-mirror-set-or-agent>

修改 .bowerrc 文件(如无则新增):

{

"proxy": "http://proxy.mysite.com:8080",

"https-proxy": "http://proxy.mysite.com:8080"

}

<http://blog.fens.me/nodejs-bower-intro/>

## Update bower

npm update -g bower

# Sass

## Before you start using Sass you will need to install Ruby

<http://www.rubyinstaller.org/>

## ****Install Sass****

gem install sass

check the sass version info

sass -v

## sass project

compass create [project name]

compass watch

# CURL

curl "http://localhost:8080/panor-web/oauth/token" -d grant\_type=implicit -d client\_id=my-trusted-client  
curl "http://localhost:8080/panor-web/oauth/token" -d grant\_type=password -d client\_id=my-trusted-client -d username=user -d password=user  
或者  
curl -X POST -v --data "grant\_type=password&username=user&password=user&client\_id=my-trusted-client" http://localhost:8080/panor-web/oauth/token  
  
token return：  
{"access\_token":"acd5d7b3-71f9-4756-993d-bb55a845feb0","token\_type":"bearer","refresh\_token":"b20b9d4b-8bde-4d6c-a816-faa2a13f3fa9","expires\_in":59,"scope":"read trust write"}  
  
curl "http://localhost:8080/panor-web/oauth/token" -d grant\_type=refresh\_token -d client\_id=my-trusted-client -d refresh\_token=daa35bae-d6d1-4e00-aa87-10996c9df57d  
  
curl "http://localhost:8080/panor-web/me" -H "Authorization: Bearer "  
  
curl "http://localhost:8080/panor-web/me/you" -H "Authorization: Bearer e2dd3748-1f06-4257-afb0-fb97835361a3"  
  
  
curl "http://localhost:8080/panor-web/api/rest/user/1/photos/10/1" -H "Authorization: Bearer 350a5c6c-c755-49f9-a004-38f295263668"  
curl "http://localhost:8080/panor-web/api/rest/user/1" -H "Authorization: Bearer 1ccb4351-86f7-4c09-b9fa-297e2be36afd"

|  |
| --- |
| curl -i -X GET -G -d "swlat=33" -d "swlng=110" -d "nelat=36" -d "nelng=133" -d "level=2" -d "vendor=gaode" -d "width=100" -d "height=100" http://127.0.0.1:8080/panor-web/api/rest/panoramio/photo |

# Nodejs

**REPL** stands for **Read-Eval-Print-Loop**

## New Application

### New package.json

|  |
| --- |
| **{**  "name"**:** "myNodeApp"**,**  "author"**:** "Jaime Developer"**,**  "description"**:** "my test node.js application"**,**  "version"**:** "0.0.1"**,**  "dependencies"**:** **{**  "connect"**:** "1.8.x"**,**  "express"**:** "2.5.x"  **},**  "engine"**:** "0.6.x"**,**  "main"**:** "app.js" **}** |

### npm install

### new app.js

|  |
| --- |
| **var** http **=** require("http"),  *// utilities for working with file paths*  path **=** require("path"),  *// utilities for accessing the file system*  fs **=** require("fs");   http.createServer(**function**(req, res) **{**    *// look for a filename in the URL, default to index.html*  **var** filename **=** path.basename(req.url) **||** "index.html",  ext **=** path.extname(filename),  *// \_\_dirname is a built-in variable containing the path where the code is running*  localPath **=** \_\_dirname **+** "/public/";  **if** (ext **==** ".html") **{**  localPath **+=** filename;  *// verify that this file actually exists and load it, or else return a 404*  path.exists(localPath, **function**(exists) **{**  **if** (exists) **{**  getFile(localPath, res);  **}** **else** **{**  res.writeHead(404);  res.end();  **}**  **}**);  **}**   **}**).listen(8000);  **function** getFile(localPath, res) **{**  *// read the file in and return it, or return a 500 if it can't be read*  fs.readFile(localPath, **function**(err, contents) **{**  **if** (**!**err) **{**  *// use defaults instead of res.writeHead()*  res.end(contents);  **}** **else** **{**  res.writeHead(500);  res.end();  **}**  **}**); **}** |

### Run app.js

**node app.js**

## new cnmap

### add "bower": "1.2.8" into package.json dependencies:

### bower init

bower init will add a bower.json file to the application root dir.

Or add bower.json file manual.

## New cnmap application

<http://twofuckingdevelopers.com/2013/07/yeoman-modern-workflows-for-angularjs/>

<http://newtriks.com/2013/06/11/automating-angularjs-with-yeoman-grunt-and-bower/>

### npm install -g yo grunt-cli bower

### npm install -g generator-angular

### //npm install -g generator-karma

### yo angular

### npm install grunt-karma --save-dev

### grunt karma

test the application

运行grunt serve报错：

TypeError: Cannot read property 'stdout' of undefined

问题解决：

After installing **Ruby**, run:

**gem install compass**

And you should be good to go.

**EDIT:** After you install Ruby, you need to make sure that the ruby runtime files' path is added to your **PATH** variable. (Thanks JagWire!)

## New yo webapp

### npm install yo grunt-cli bower

### npm install generator-webapp

### yo webapp

# CoffeeScript

## install

npm install -g coffee-script

# Jekyll - Transform your plain text into static websites and blogs

~ $ gem install jekyll

~ $ jekyll new my-awesome-site

~ $ cd my-awesome-site

~/my-awesome-site $ jekyll serve

# => Now browse to http://localhost:4000

# 浏览器

浏览器缓存和304小结

# ADB

## Boot loader

adb reboot boot loader

## Reboot enter recovery mode

adb reboot recovery

## debug phonegap app

adb logcat Cordova:\* DroidGap:\* CordovaLog:\* \*:S

## install app

adb install [apk path]

## copy file to device

adb push /path/to/local/file /mnt/sdcard/path/to/file

## TWRP install method

### Download:

[openrecovery-twrp-leo-2.1.1.img](http://techerrata.com/file/twrp2/leo/openrecovery-twrp-2.2.0-leo.img)

### Fastboot Installation Method:

1. Download file to your computer
2. In command line, navigate to the folder you have the recovery
3. Reboot phone into fastboot
4. "**fastboot flash recovery twrp-leo-2.1.1.img**"
5. "**fastboot reboot**"
6. Reboot Phone into recovery as normal

Note that you will need to change the last part to match the name of the file that you just downloaded.

### Android Installation Method:

1. Download file to your computer
2. In command line, navigate to the folder you have the recovery
3. "**adb push recoveryfilename.img /sdcard**"
4. "**adb shell flash\_image recovery /sdcard/recoveryfilename.img**"
5. Reboot Phone into recovery as normal

Note that you will need to change the last part to match the name of the file that you just downloaded.

# Linux command

## 查找指定大小指定文件名模式的文件copy到另一个地方

**find . -iname "\*.jpg" -size +100k -exec cp "{}" ../dest \;**