**PHONE GAP TUTORIAL**

*Contents*

[Phone Gap Setup Guide](#h.viu8bzkwg93q)

[Steps to install phone gap](#h.oqlfv8g8zbe1)

[Set Up New Project:](#h.zbf4d8lud3sd)

[Important distinctions between a project created with and without phone gap:](#h.6vp5dn8md0pv)

[Launching without Eclipse:](#h.sp4sk7guuv6u)

[Launching with Eclipse:](#h.gzetridjrt2x)

[Starting with project created from phone gap:](#h.6hfik5evc575)

[Eg1:](#h.3u6sqcudq46k)

[example.java](#h.gz48hyz8hfos)

[Index.html](#h.6ku03jhpt1so)

[Output on Emulator:](#h.4rwiudhn7f7b)

# 

# 

# 

# 

# 

# 

# 

# 

# Phone Gap Setup Guide

## **Steps to install phone gap**

1. Go to <http://phonegap.com/> and click on Download Phone Gap and extract its contents.
2. Install Java and set java path in JAVA\_HOME in environment variables.

* Additionally, you may need to include %JAVA\_HOME%\bin to your PATH as well. To check to see if this is required, run a command prompt and type java. If the program can not be found add %JAVA\_HOME%\bin to the PATH. You may need to specify the full path instead of using the %JAVA\_HOME% environment variable.

1. Install Android SDK from <http://developer.android.com/sdk/index.html>
2. Set up the paths <android sdk path>\tools and <android sdk path>\platform-tools in Path of Environment variables after installing Android SDK.
3. Download Apache Ant version 1.9 from <http://ant.apache.org/>.

* Finally, you may need to include %ANT\_HOME%\bin to your PATH as well. To check to see if this is required, run a command prompt and type ant. If the program can not be found add %ANT\_HOME%\bin to the PATH. You may need to specify the full path instead of using the %ANT\_HOME% environment variable.

## **Set Up New Project**:

After extracting the installed phoneGap.

1. Navigate to bin/ directory in android/ directory of phone gap.
2. Type create <project folder path> <package name> <project name>

Eg:

Project folder path: C:\users\karthik kamarapu\workspace/android1\sample

Package name: com.example.sample

Project name :sample.

The above will create a directory with the project name at the path mentioned in the project folder path.The project created will have all the directories required to build an application using phone gap.

## Important distinctions between a project created with and without phone gap:

1. A project created through phone gap will have the directory www/ under
2. Assets/ directory.
3. The www/ directory will have index.html,cordova-x.x.x.js,main.js,master.css and spec.html files.
4. A coredava/ directory is also present which has appinfo.jar
5. The lib/ directory will have cordova-x.x.x.jar present in it.
6. AndroidManifest.xml will have all the permission automaticall added.

## Launching without Eclipse:

1)Go to the project directory from command prompt and give command:

Ant debug

The above command will build the project-debug.apk file and place it in the

Directory of the project.If the release mode is chosen to build,it has to be signed manually.

2)In command prompt type,

Android avd

The above command will list all the avd's installed and asks us to choose one to launch.If only one avd is installed it will automatically select the same.

3)Once the emulator is open,Go to the androidsdk/platform-tools/ directory in the command prompt.

Type the command

Adb install <path of apk>/project-debug.apk

The above command will install the apk file on the emulator.

## **Launching with Eclipse**:

1. Create New Project->Android Project from Existing Code and select the directory created at Set Up New Project.
2. If the project has any errors from AndroidManifest.xml.Select the target version of android API level as the highest.Then from project menu bar select clean.

## **Starting with project created from phone gap**:

1. The project already has a www\ directory under assets\ directory.

The index. Html in www\ will have the display logic and gets called from the main activity.

### Eg1:

#### example.java

**package** org.apache.cordova.example;

**import** android.os.Bundle;

**import** org.apache.cordova.\*;

Public class example extends DroidGap {

@override

Public void onCreate(Bundle savedInstanceState){

super.onCreate(savedInstanceState);

super.loadUrl("file:///android\_asset/www/index.html");

}

}

#### Index.html

<!DOCTYPE HTML>

<html>

<head>

<title>Device Properties Example</title>

<script type="text/javascript" charset="utf-8" src="cordova-2.7.0.js"></script>

<script type="text/javascript" charset="utf-8">

// Wait for Cordova to load

//

document.addEventListener("deviceready", onDeviceReady, false);

// Cordova is ready

//

function onDeviceReady() {

navigator.geolocation.getCurrentPosition(onSuccess, onError,{enableHighAccuracy:true});

}

// onSuccess Geolocation

//

function onSuccess(position) {

var element = document.getElementById('geolocation');

element.innerHTML = 'Latitude: ' + position.coords.latitude + '<br />' +

'Longitude: ' + position.coords.longitude + '<br />' +

'Altitude: ' + position.coords.altitude + '<br />' +

'Accuracy: ' + position.coords.accuracy + '<br />' +

'Altitude Accuracy: ' + position.coords.altitudeAccuracy + '<br />' +

'Heading: ' + position.coords.heading + '<br />' +

'Speed: ' + position.coords.speed + '<br />' +

'Timestamp: ' + position.timestamp + '<br />';

}

// onError Callback receives a PositionError object

//

function onError(error) {

alert('code: ' + error.code + '\n' +

'message: ' + error.message + '\n');

}

</script>

</head>

<body>

<p id="geolocation">Finding geolocation...</p>

</body>

</html>

#### Output on Emulator:

