PhoneGap

Tutorial

Simply Easy Learning

www.tutorialspoint.com

SIMPLE EASY LEARNING

About the tutorial

PhoneGap Tutorial

PhoneGap is a software development framework by Adobe System, which is used to develop mobile applications. To develop apps using PhoneGap, the developer does not require to have knowledge of mobile programming language but only web-development languages like, HTML, CSS and JScript. PhoneGap produces apps for all popular mobile OS platforms such as iOS, Android, BlackBerry and Windows Mobile OS etc.

This tutorial concentrates on developing App for Android platform. This tutorial will give you adequate information about how to produce apps quickly using PhoneGap services.

Audience

Programmers who want to put their website on App either offline or online, may take advantage of this tutorial. This tutorial is designed to give a general view of App building using web-technologies. Anybody who is interested to obtain know-how of Apps using web-technologies may use this tutorial.

Prerequisites

It is mandatory that you must have knowledge of HTML, CSS and JScript to create website that you might want to put on App. No other programming language is required to use PhoneGap.

Copyright & Disclaimer

our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

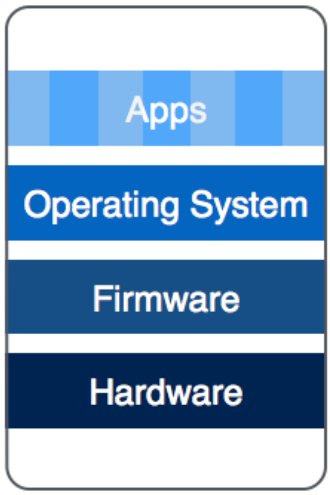
PhoneGap - Overview

1

Introduction

Mobile, handhelds, and easy-to-carry devices have started a new revolution in software engineering. These small but efficient devices are capable to run applications created with high-end programming languages. People who own these devices tend to use them at their maximum as these devices such as mobile phones, are very convenient to use anytime, anywhere.

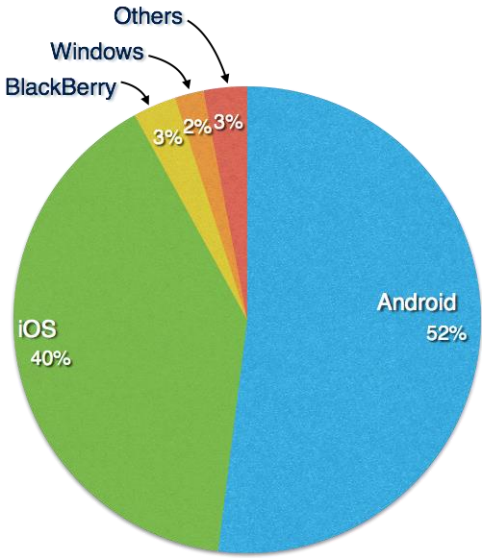
The architecture of a mobile device is similar to that of a computer system. It has custom built hardware, firmware, and operating systems.



These three items are mostly proprietary and are engineered, developed, and assembled under one flagship organization. Apps (Application Software) are developed both by flagship organization and developers from outside the organization.

A number of well-recognized mobile operating systems are available in the market in both proprietary and open-source categories. Most widely used mobile operating systems are:

- Android
- iOS
- BlackBerry
- Windows



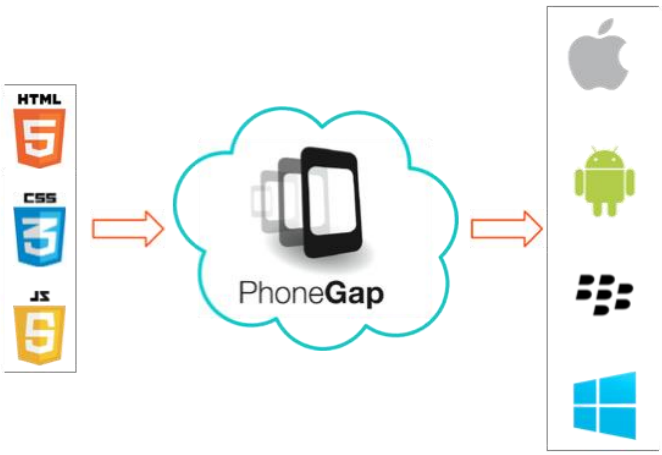
Every mobile operating system provides their own set of tools and environments to develop apps that will run on them. Application made for one operating system cannot run on any other platform as they are entirely different. Developers tend to cover all major mobile operating systems in order to increase reachablity among their users.

Thus it becomes a tedious task to develop an application program that may run on all major OS platforms, keeping its look, feel, and functionality identical on all platforms. For this work, a developer needs to understand all platforms and should have good understanding of major development tools for different OS.

PhoneGap

PhoneGap may be seen as a solution to all problems mentioned earlier. PhoneGap is a framework that makes the developers develop their apps using standard web APIs for all major mobile operating systems. It is open-source and free.

Developers only need to know web development using HTML, CSS and JavaScript. PhoneGap takes care of rest of the work, such as look and feel of the app and portability among various mobile OS.



Using PhoneGap, one can create apps for all major mobile OSl like Apple iOS, Android, BlackBerry, Windows etc. This does not require the developer to have expertise over any of the above mentioned platforms neither the developer is required to know programming to code the app from scratch.

PhoneGap allows its users to upload the data contents on website and it automatically converts it to various App files.

In this tutorial, we shall see how to create app for apple, android, and windows platform online and without using any offline tool.

PhoneGap – Environment Setup

2

We should learn here about how to set up basic environment in order to make apps effortlessly. Though PhoneGap supports offline creation of apps using cordova command line interface and Github repository mechanism but we shall concentrate on minimum effort procedure.

We assume that you are well versed with web technologies and have your web application ready to be shipped as an app. Because PhoneGap supports only HTML, CSS and JavaScript, it is mandatory that the application should be created using these technologies only.

From developers perspective, an app should have the following items included in its package:

- Configuration files
- Icons for app
- Information or content (built using web-technologies)

Configuration

Our web app will need only one configuration file that should be adequate to configure all its necessary settings. Its name is config.xml. This file contains all the necessary information required to compile the app.

Let us see config.xml, for our example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<widget xmlns     = "http://www.w3.org/ns/widgets"
        xmlns:gap = "http://phonegap.com/ns/1.0"
        id        = "com.tutorialspoint.onlineviewer"
        version   = "1.0.0">

    <name>Tutorials Point</name>

    <description>
        Tutorials Point Online Viewer
    </description>

    <author href="http://tutorialspoint.com" email="contact@tutorialspoint.com">
        Tutorials Point
    </author>

    <preference name="permissions" value="none"/>

    <icon src="res/icon/android/drawable-ldpi/tp_icon.png"    gap:platform="android"    gap:qualifier="ldpi"  />
    <icon src="res/icon/android/drawable-mdpi/tp_icon.png"    gap:platform="android"    gap:qualifier="mdpi"  />
    <icon src="res/icon/android/drawable-hdpi/tp_icon.png"    gap:platform="android"    gap:qualifier="hdpi"  />
    <icon src="res/icon/android/drawable-xhdpi/tp_icon.png"   gap:platform="android"    gap:qualifier="xhdpi" />
    <icon src="res/icon/android/drawable-xxhdpi/tp_icon.png"  gap:platform="android"    gap:qualifier="xxhdpi"/>

    <icon src="res/icon/ios/icon-57.png"    gap:platform="ios" width="57"   height="57"     />
    <icon src="res/icon/ios/icon-72.png"    gap:platform="ios" width="72"   height="72"     />
    <icon src="res/icon/ios/icon-57-2x.png" gap:platform="ios" width="114"  height="114"    />
    <icon src="res/icon/ios/icon-72-2x.png" gap:platform="ios" width="144"  height="144"    />

</widget>
```

All configuration contents are wrapped in **<widget>** tag. Brief description of these is as follows:

- <widget id = ”app_id”>

  id is your reserved app-id on various app stores. It is in reverse-domain name style i.e. com.tutorialspoint.onlineviewer etc.

- <widget version = ”x.y.z”>

  This is version number of app in x.y.z format where (x,y,z) are positive integers i.e. 1.0.0, it represents major.minor.patch version system.

- <name> App Name </name>

  This is name of app, which will be displayed below app icon on mobile screen. Your app can be searched using this name.

- <description> My First Web App </description>

  This is brief description of what is the app about, and what it is.

- <author> Author_Name </author>

  This field contains name of the creator or programmer, generally set to the name of organization which is launching this app.

- <preferences name = ”permissions” value = ”none”>

  The preferences tag is used to set various options like FullScreen, BackgroundColor and Orientation for app. These options are in name and value pair. For example: name=”FullScreen” value=”true” etc. Because we do not require any of these advance settings, we just put permissions to none.

- <icon>

Allows us to add icons to our apps. It can be coded in various ways, but we are learning short-cut of everything, so here it is. The src determines the path of icon image. The gap:platform determines for which OS platform this icon is to be used. The gap:qualifier is density that is used by android devices. The iOS devices use width & height parameters.

Icons

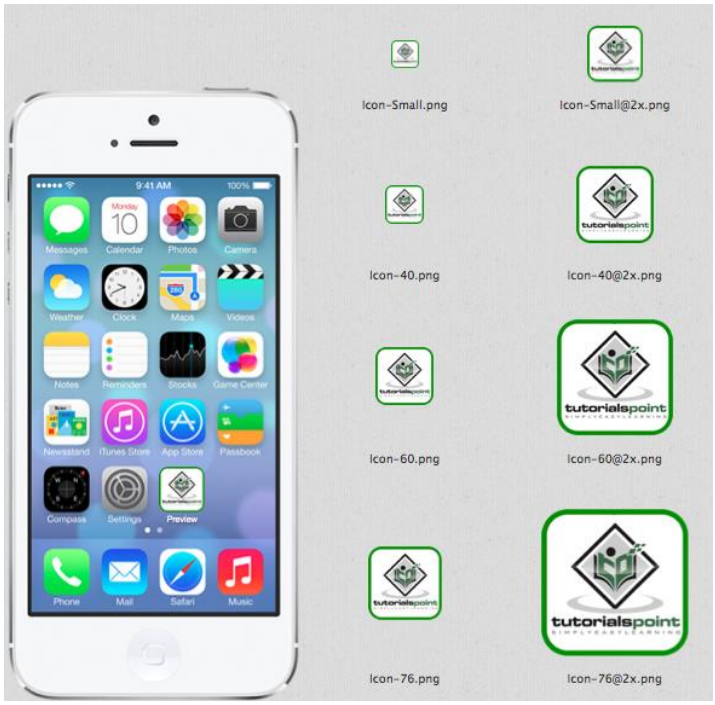
There are devices of various sizes having same mobile OS, so to target an audience of one platform you need to furnish icons of all the mobiles types too. It is important that we prepare icons of exact shapes and sizes as required by particular mobile OS.

Here we are using the folders **res/icon/ios** and **res/icon/android/drawable-xxxx**.

To get this work done fast, you can create a logo of size 1024x1024 and log on to makeappicon.com. This website will help you instantly create logos of all size and for both android and iOS platform.

After providing icon image of size 1024x1024, makeappicon.com should provide the following:

- 1. icons for iOS:



- 2. icons for Android:

This website provides you an option to email all the logos in zip format to your doorstep (a.k.a. email, ofcourse!).

PhoneGap – App Contents

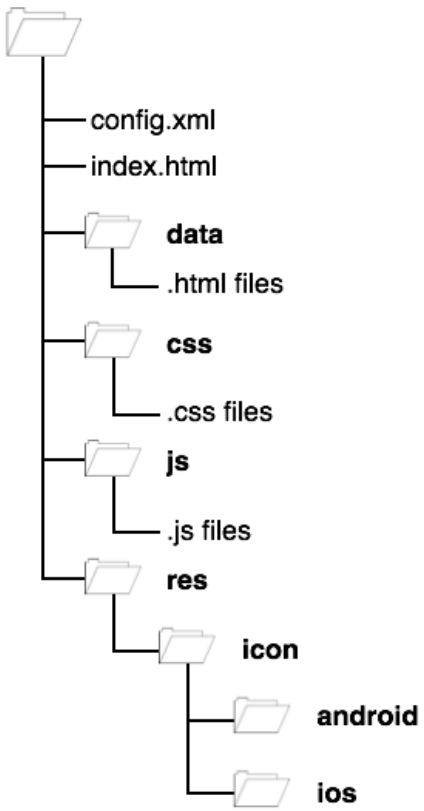
3

Offline websites are copied to local hard-drive and accessed whenever the user needs without any internet connection. Likewise, this offline web app will let you create a web application that is downloaded to its entirety to the mobile devices of user and user can access that offline.

An application for this type of app may include app having collection of stories, short tutorials or any other offline content of users' interest, which he/she can read offline even when internet is not available.

Offline App

The following image represents the folder structure for offline app. At root directory it requires only two files, config.xml and index.xml.

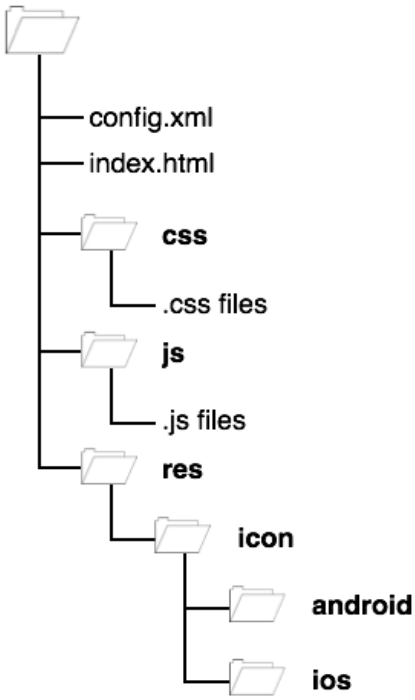


The config.xml contains app configuration settings which we learnt in previous section. The index.html is file that contains homepage of web-contents.

One important thing to learn here is that all links inside all html files should contain relative path only. That is, no absolute path or base href tag should be there.

Online App

The following image shows folder structure for our app to be in online mode. In online mode all web-contents are loaded from internet website.



You may see that **data** folder is missing in online mode app, because all the files reside on actual server and accessible via internet. The index.html file contains actual links as it contains at the web server and all its links are either absolute or used with **base href** tag.

After you have decided the mode of your app and organized its file in the file structure mentioned above, you need to zip your file with any standard zip tool and save it. We shall use this file in next section.

Sign Your App

It is essential for any app to be signed by its developers or developing organization to keep things in order. For this reason you need to sign your app. You may need **keytool** which is a part of standard java distribution.

Execute the following command:

```
keytool -genkey -v -keystore my_keystore.keystore -alias TutorialsPoint -keyalg RSA -
keysize 2048 -validity 10000
```

This should generate my_keystore.keystore file, which we shall need this file in next section.

PhoneGap – App Compilation

4

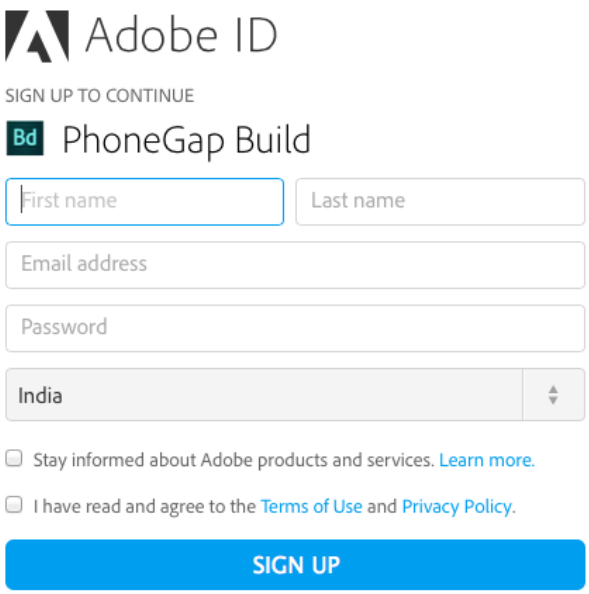
Now we are ready to compile our first web API based quick mode app. In this final segment, we shall learn about the process of transforming our web contents to an app format, which can then be uploaded on online app stores.

PhoneGap accepts user login created on GitHub or using AdobeID. GitHub is a repository service where users can upload their contents and use them providing their URL references. For example, the content we just created can be uploaded to GitHub and then call them directly to PhoneGap.

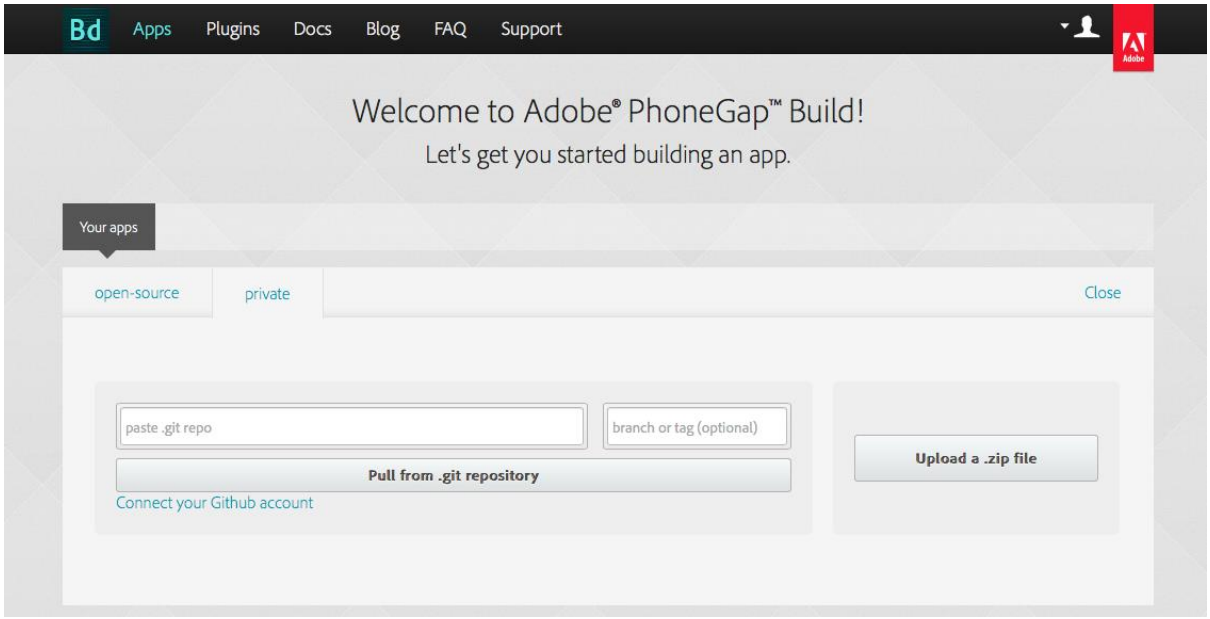
Create Adobe ID

Goto www.build.phonegap.com and click on Register

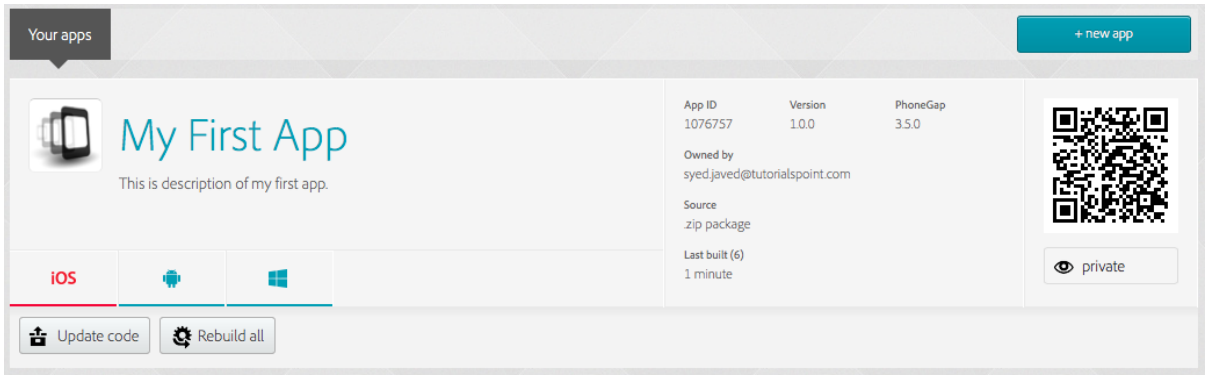
A new window will open as displayed below:



Fill in your details and click on sign up. You can now login with the same user-id to PhoneGap. By default, this page should lead to PhoneGap console as displayed below:

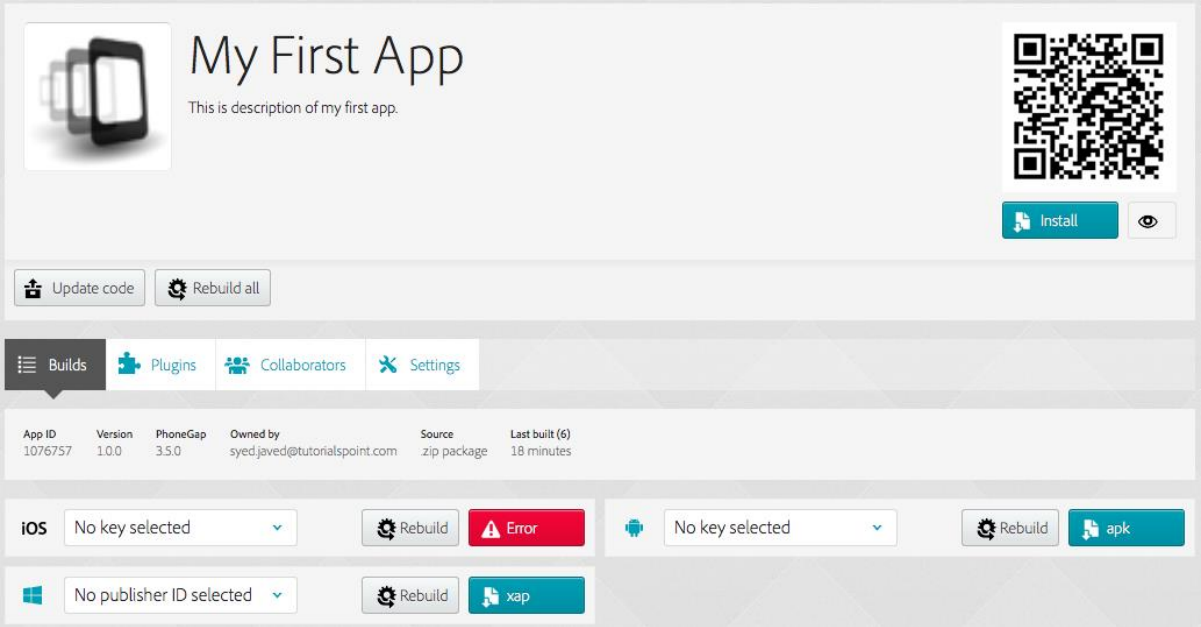


Click 'Upload a .zip file' and upload the .zip file we created, which has all web-content and configurations. You should see the following window after successful upload:

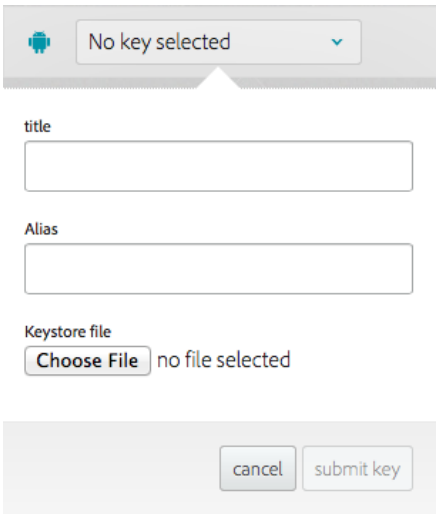


You may instantly see that iOS app has failed its processing as we have not provided any signed key. We are only concentrating on Andoid and you can see that it has been created by PhoneGap. This app can not be uploaded to google store as it is too not signed by key.

Click on the Android icon and the following screen should appear:



Click on drop-down option menu next to Android icon that reads No key selected, click on add a key and the following screen should appear:



Provide title and alias of your choice and click on Keystore file. Provide the keystore file created in last section. Then click on 'Rebuild' button next to it.

The app built by this process can be directly uploaded to Google Play. Click on .apk file and you can download your first web-based free app.

Before uploading, app should be tested in either virtual or real devices.