PhoneGap Build

Starter

Painless Mobile Apps Development

Zainul Setyo Pamungkas

PhoneGap Build Starter

Painless Mobile Apps Development

Zainul Setyo Pamungkas

This book is for sale at http://leanpub.com/phonegapbuild

This version was published on 2014-05-26

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

For Almighty Allah, Prophet Muhammad SAW, My Beloved Parents

Contents

So, what is PhoneGap Build?

PhoneGap Build is a cloud service that allow developers to compile HTML5 application into native mobile application. It compile not only for Android and iOS, but also for WebOS, Windows Mobile, Symbian, and Bada. It designed to make mobile application development faster and easier by using existing technology, HTML5. It works like magic. We don't have to know how application compilation done in server. All we have to do is just uploading assets file and we will get market-ready application in different platforms.

PhoneGap Build let us develop for several platforms without even setting up SDK. We don't need to setup ADT and Eclipse for developing Android application, or even download and setup Xcode for developing iOS application. It take care the work in the cloud, we don't need to worry about setting up and maintaining native SDKs. It build applications using latest SDK for targeted platforms. We don't have to manually install and set up new version of SDKs.

PhoneGap Build makes HTML5 application really shine. HTML5 has good cross compatibility across mobile platforms already. And Phonegap Build leverage its capabilities in to the next level by making it native applications that can be installed on real devices. It can be done even without any changes to support multiple platforms using single code base.

PhoneGap Build makes mobile application development cost-effective. Reducing cost for each different platform that want to be supported. Developers can also rapidly prototype app using existing tools to develop web application.They don't need to learn another programming language, juts use existing skill from web development.

PhoneGap Build support 'classic upload' and Git (GitHub) for deploying application. Classic upload can be done by uploading web assets in `*.zip` file. While Git deployment provide us the power of version control, especially using GitHub service. Developers can commit changes to GitHub repositories then pull it from PhoneGap Build to be compiled. This deployment method will significantly increase the speed of application development. One of GitHub power's that can be used is collaboration development. This is great if developers are working on team.

PhoneGap Build gives extra capabilities on application by using plugins. Plugins are used to extend native functionality exposed by the PhoneGap native-app container. They are including child browser, generic push (push notification), barcode scanner, analytics, and FacebookConnect. Developers can include original PhoneGap's plugin to use with Phonegap Build and extend native capabilities of application.

PhoneGap Build allow developers to configure application through single file named `config.xml`. It expand options that can be set up to build application using PhoneGap Build. They are including describing application, configuring device orientation, setting up specific icon to match targeted platform, configuring splash screen, and many more will be described in next section.

Currently, PhoneGap Build offer two different plans, free and paid. Both plan are open source friendly. Which mean we have no limitation to create open source application and how many collaborators are collaborating in the project. The only difference is just how many private application you can build. Free plan allow developers to create only one private application, while paid plan allow 25 applications to be created in private.

Step by Step

In 5 easy steps, you can use PhonGgap Build and get it set up.

Prepare Development Environment

Before using PhoneGap Build service, you will need to check that you have all of the required elements, as listed below:

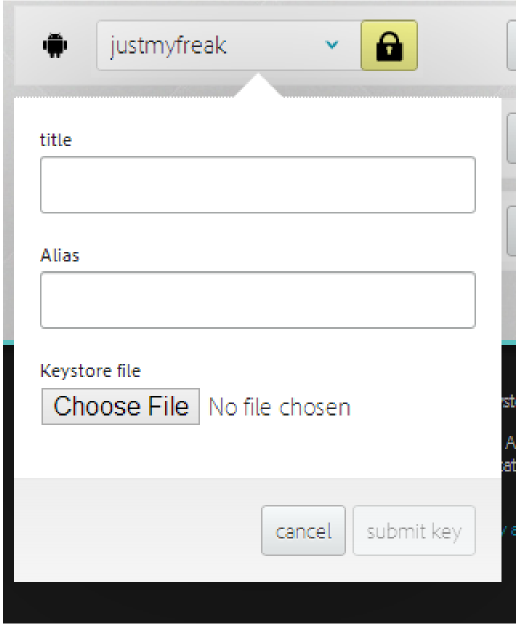
- PC or Mac. This is needed to get certificate for signing application.
- Text editor : Preferably with xml highlighting such as Notepad++ or Sublime Text. You will need this to write `config.xml`.
- Register apple developer program. You will need to enroll on it if you plan to build for iOS devices.
- Google Chrome or other browser. need web browser to use PhoneGap Build.
- GitHub account. You will need it if you want to deploy via Git

Import Signing Keys

Importing signing keys is very important especially if you plan to deploy and publish application on iOS, Android, and Blackberry. For testing purpose, Android and Blackberry require no sign key. But for iOS, you will need to have signing for both development and publishing application.

For Android signing, we first need to generate a signing `keystore` file. Full details are available in http://developer.android.com/guide/publishing/app-signing.html[1]. Please record `alias` as well as `keystore password` and `key password` that you set for your `keystore`. Then go to your account, click add key, and fill all your details.

[1] http://developer.android.com/guide/publishing/app-signing.html



Importing Signing Key

For Blackberry signing, we first need to register on RIM's site at https://www.blackberry.com/SignedKeys/[2] ,then follow the installation process. Once installed, we must find SDK directory when we installed key.

```
C:\SDK_PATH\bin\sigtool.csk
C:\SDK_PATH\bin\sigtool.db
```

Both files are required to sign and build Blackberry application. signtool.csk contain database private key and salt and sigtool.db contain credentials for connecting on RIM's server. Then we need to add both file in BlackBerry key section like we did on Android.

For iOS signing, we have to register Apple's developer program on http://developer.apple.com[3] and pay $99/year for enrolling iOS developer program. We need two files *.p12 and *.mobileprovision to sign iOS application. There are two ways to retrieve those 2 files, first is using standard Apple method documented on http://developer.apple.com[4] and second is using Windows http://community.phonegap.com/nitobi/topics/detailed_guide_for_setting_up_building_ios_apps_without_a_mac[5]

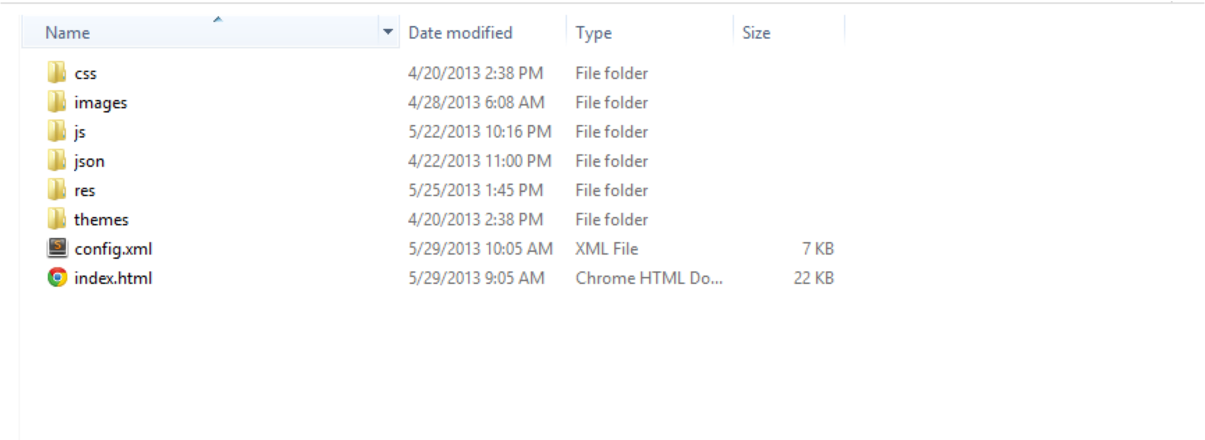
Once we get *.p12 and *.mobileprovison, we can add it to PhoneGap Build sign like we did on Android. Please note that there is 2 version of *.p12 certificate, first is for development and second is for production (distribute on App Store). For testing purpose, we can use development certificate and add our iDevice's UUID in *.mobileprovision.

[2] https://www.blackberry.com/SignedKeys/
[3] http://developer.apple.com
[4] http://developer.apple.com
[5] http://community.phonegap.com/nitobi/topics/detailed_guide_for_setting_up_building_ios_apps_without_a_mac

Preparing Assets

Assets are HTML5 application including html, js, css, and image resources. Actually, Phonegap Build need only one `index.html` and place other resources on servers. But it will not wise for mobile application. Static files such as html, js, css, images, or even json data should be cached offline. It will increase application performance and load time as well as reducing internet connection in application.

When creating application for PhoneGap Build, you have to leave `index.html` at root level of your project directory. Make sure there is no other `*.html` file other than index on root level. This will cause build failed on certain Phonegap version. If you want to add another html file, make sure you add it inside your folder. Here is suggested project folder of Phonegap Build application :



Name	Date modified	Type	Size
css	4/20/2013 2:38 PM	File folder	
images	4/28/2013 6:08 AM	File folder	
js	5/22/2013 10:16 PM	File folder	
json	4/22/2013 11:00 PM	File folder	
res	5/25/2013 1:45 PM	File folder	
themes	4/20/2013 2:38 PM	File folder	
config.xml	5/29/2013 10:05 AM	XML File	7 KB
index.html	5/29/2013 9:05 AM	Chrome HTML Do...	22 KB

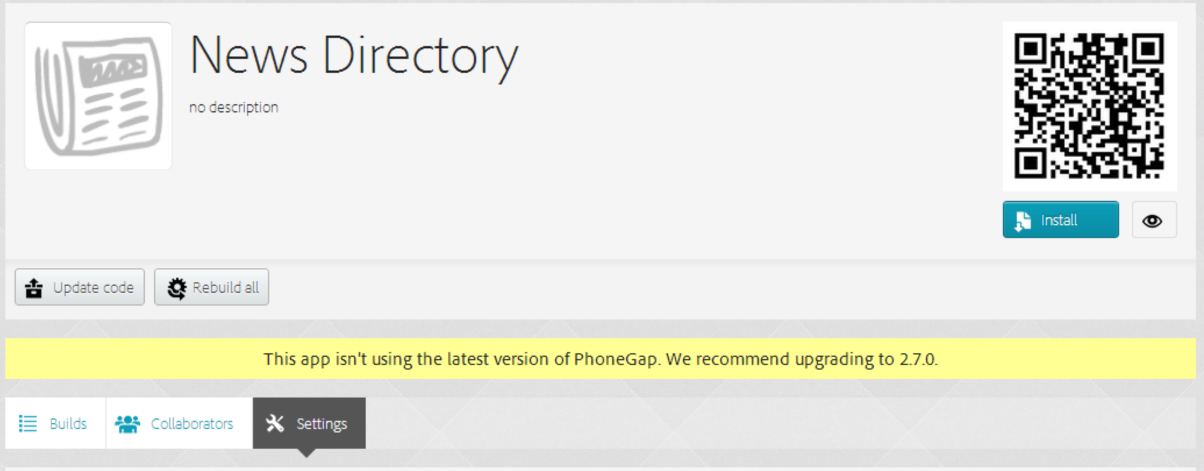
Project Directory Example

Configuring Application

PhoneGap Build allows us to make configuration for building application. Application can be configured trough web interface inside PhoneGap Build account or by using `config.xml`. The `config.xml` file is specified using W3C widget specification (http://www.w3.org/TR/widgets/[6]), which allows developer to specify certain configuration easily through single xml file.

Web configuration can be found under Settings tab of application. By using web configuration we can specify application's icon, name, package, version number, PhoneGap version, as well as its description. Package name must be reverse-domain style (e.g. `com.yourcompany.yourapp`). As for version number, you can use major/minor/patch style version such as `1.0.1`

[6] http://www.w3.org/TR/widgets/



News Directory

no description

This app isn't using the latest version of PhoneGap. We recommend upgrading to 2.7.0.

Another way to configuring application is by using `config.xml` file. This file should be placed under root level of application, same location with `index.html`. The `config.xml` gives us more than we can do in web configuration. Bellow is example of initial `config.xml`.

```xml
<?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.yourcompany.yourapp"
    version   = "1.0.0">

        <name>yourapp</name>
        <description>Your application description</description>
</widget>
```

Mandatory Properties

- `<widget>` : The `<widget>` element must be root of your config.xml document. Attributes. It contain following attributes :
  - `xmlns`. XML's xmlns with value `http://www.w3.org/ns/widgets`
  - `xmlns:gap` with value `http://phonegap.com/ns/1.0`
  - `id`. Application package name.
  - `version`. Application version number
- `<name>` : The name of application
- `<description>` : The description of application

Along with mandatory properties, we can add configuration for :

1. Application Icon

   Application icon can be configured by using `<icon>` element. You can have zero or more of this element. If you don't add `<icon>` element, PhoneGap Build will use default PhoneGap Build icon. The default icon must be named `icon.png` and placed in root of your application.

```
 `<icon src=”icon.png”/>`
```

For iOS icon, PhoneGap Build support classic, retina, and iPad (and retina iPad starting from PhoneGap 2.5.0). The following will define icon for specific screen type of iDevice.

```
 <icon src="icons/ios/icon.png" gap:platform="ios" width="57" height="57" />
 <icon src="icons/ios/icon-72.png" gap:platform="ios" width="72" height="72" />
 <icon src="icons/ios/icon_at_2x.png" gap:platform="ios" width="114" height="1\
14" />

 <!-- retina iPad support: PhoneGap 2.5.0+ only -->
 <icon src="icons/ios/icon-72_at_2x.png" gap:platform="ios" width="144" height\
="144" />
```

As for Android icon, PhoneGap Build support ldpi, mdpi, hdpi, and xhdpi display resolution. The following elements will define icon for each specific screen size.

```
 <icon src="icons/android/ldpi.png" gap:platform="android" gap:density="ldpi" \
/>
 <icon src="icons/android/mdpi.png" gap:platform="android" gap:density="mdpi" \
/>
 <icon src="icons/android/hdpi.png" gap:platform="android" gap:density="hdpi" \
/>
 <icon src="icons/android/xhdpi.png" gap:platform="android" gap:density="xhdpi\
" />
```

2. Application Splash Screens

   You can have zero or more of `<gap:splash>` elements. If you don’t add this element, your application will not have splash sreen. PhoneGap Build support classic, retina, and iPad (and retina iPad starting from Phonegap 2.5.0) for splash screen resolution.

```
 <gap:splash src="splash/ios/Default.png" gap:platform="ios" width="320" heigh\
t="480" />
 <gap:splash src="splash/ios/Default_at_2x.png" gap:platform="ios" width="640"\
 height="960" />
 <gap:splash src="splash/ios/Default_iphone5.png" gap:platform="ios" width="64\
0" height="1136" />
 <gap:splash src="splash/ios/Default-Landscape.png" gap:platform="ios" width="\
1024" height="768" />
 <gap:splash src="splash/ios/Default-Portrait.png" gap:platform="ios" width="7\
68" height="1024" />
 <!-- retina iPad support: PhoneGap 2.5.0+ only -->
 <gap:splash src="splash/ios/Default-Landscape_at_2x.png" gap:platform="ios" w\
idth="2048" height="1496" />
 <gap:splash src="splash/ios/Default-Portrait_at_2x.png" gap:platform="ios" wi\
dth="1536" height="2008" />
```

As for Android, PhoneGap Build support ldpi, mdpi, hdpi, and xhdpi display resolution for splash screen. The following elements will define icon for each specific screen size.

```
 <gap:splash src="splash/android/ldpi.png" gap:platform="android" gap:density=\
"ldpi" />
 <gap:splash src="splash/android/mdpi.png" gap:platform="android" gap:density=\
"mdpi" />
 <gap:splash src="splash/android/hdpi.png" gap:platform="android" gap:density=\
"hdpi" />
 <gap:splash src="splash/android/xhdpi.png" gap:platform="android" gap:density\
="xhdpi" />
```

3. PhoneGap Version Preference

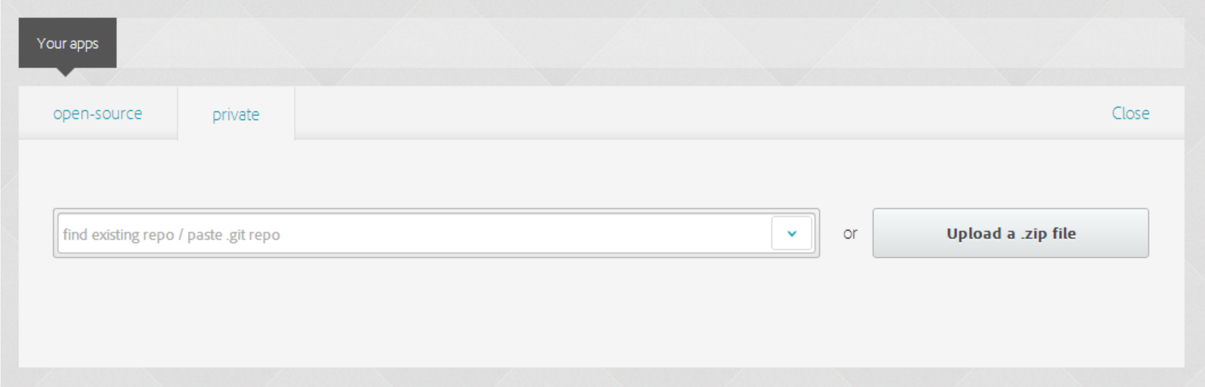
   PhoneGap version can be configured through `<preference>` element. For example `<preference name=”phonegap-version” value=”2.7.0”>`. Phonegap Build support Phonegap versions `2.0.0`, `2.1.0`, `2.2.0`, `2.3.0`, `2.5.0`, and `3.2.0` (default). If you do not specify version, your application will use default version of PhoneGap.
4. Device Orientation Preference

   Device orientation can be configured through `<preference>` element. For example `<preference name="orientation" value="default" />`. Value has three options, they are `default`, `landscape`, and `portrait`. `default` means both `landscape` and `portrait` orientation are supported.
5. Targeting Specific Device Preference

   Target specific preference works for iOS only. You can specify your application support `handset`, `tablet`, or `universal`. Example `<preference name="target-device" value="universal" />`

Zip Upload and Git Deployment

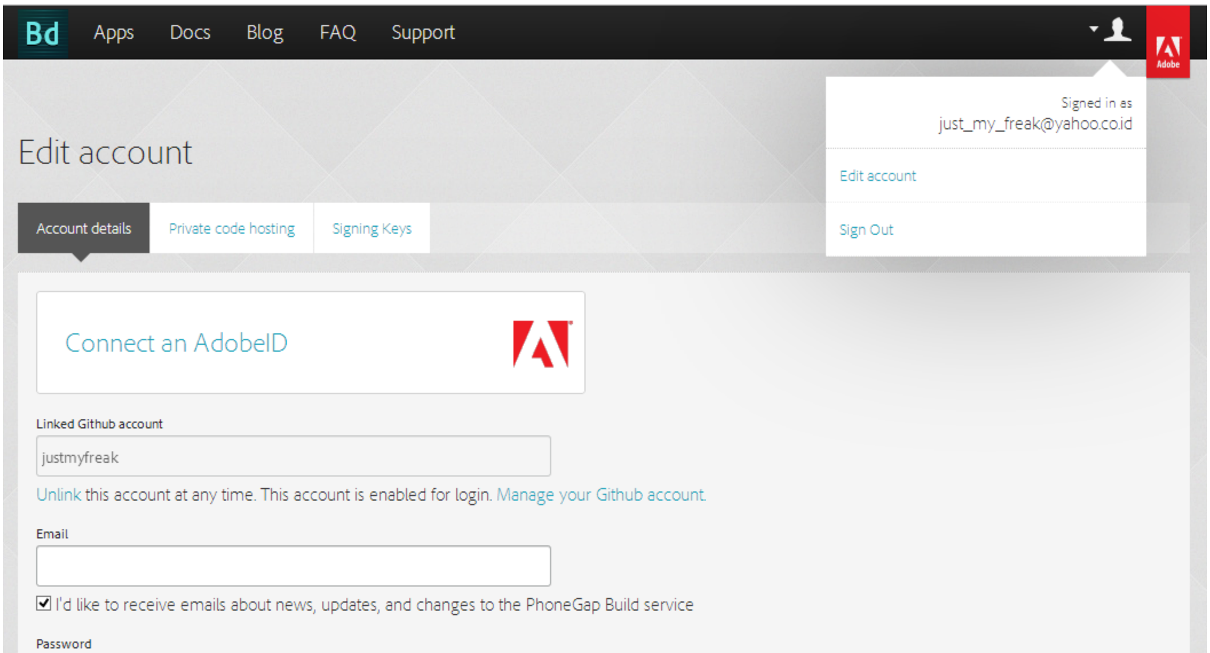
Uploading HTML5 application can be done by using two methods, uploading `*.zip` of your source code or by selecting existing repository in GitHub account. Zip upload only works for private application, while Git works for both private and open source application.



PhoneGap Deployment

Zip uploading is self explanatory. Click on new app then choose ‘Upload a .zip file’. Then you are prompted with dialog in your browser to choose zipped assets of your HTML5 application. Your application will be build once it finished uploading.

To enable GitHub deployment, first you must link your GitHub to PhoneGap Build. To do it go to account, then choose edit account. And then link your GitHub account. If you want to build private application using existing GitHub private repository, you must provide your ssh key. To add your ssh key, simply click on ‘Private code hosting’ tab and add paste ssh key. Your existing application on GitHub now will appear in your ‘new app’ when creating new PhoneGap Build application.



Signed in as
just_my_freak@yahoo.co.id

Edit account

Linked Github account: justmyfreak

Unlink this account at any time. This account is enabled for login. Manage your Github account

Email:

☑ I'd like to receive emails about news, updates, and changes to the PhoneGap Build service

Password

GitHub Deployment

And that’s it!!

By this point, you should ready to use PhoneGap Build and are free to play around and discover more about it.

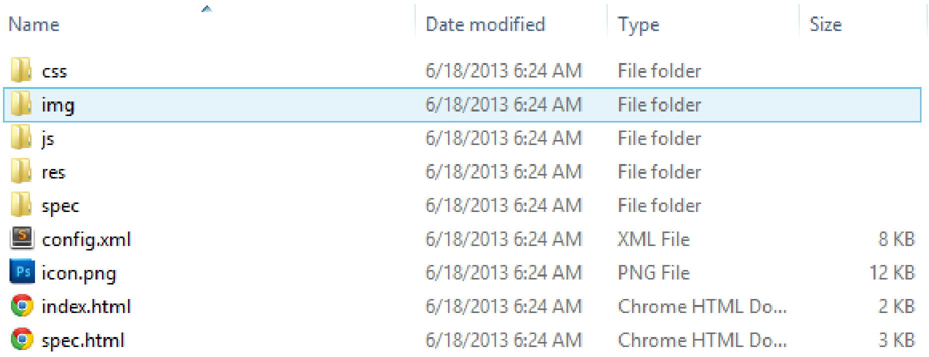
Quick Start - Build Your First Application Using PhoneGap Build

In this section, we will show you how to create your first PhoneGap Build application. From preparation, signing, and getting market ready application.

Step 1 – Preparing Assets

Asset is your HTML5 application including HTML, JS, CSS, and image files. To make it easy, we have prepared starter HTML5 application. You can download or clone it at https://github.com/justmyfreak/Phone Build-Starter/[7]. If you download it, you can extract it to edit source code.

You will find www folder contain many folders and files as the following :



Name	Date modified	Type	Size
css	6/18/2013 6:24 AM	File folder	
img	6/18/2013 6:24 AM	File folder	
js	6/18/2013 6:24 AM	File folder	
res	6/18/2013 6:24 AM	File folder	
spec	6/18/2013 6:24 AM	File folder	
config.xml	6/18/2013 6:24 AM	XML File	8 KB
icon.png	6/18/2013 6:24 AM	PNG File	12 KB
index.html	6/18/2013 6:24 AM	Chrome HTML Do...	2 KB
spec.html	6/18/2013 6:24 AM	Chrome HTML Do...	3 KB

There you go. We have our assets ready. If you notice, it just basic HTML5 application with addition of `config.xml`. Go explore on its source, change HTML file or images as you like. After you are done playing with HTML5 sources, we can step to the next step.

Step 2 – Configure your Application

Configuring application is easy. Open `config.xml` in text editor you like, preferably Notepad++ or Sublime Text. We will change mandatory configuration as following :

- Change application name

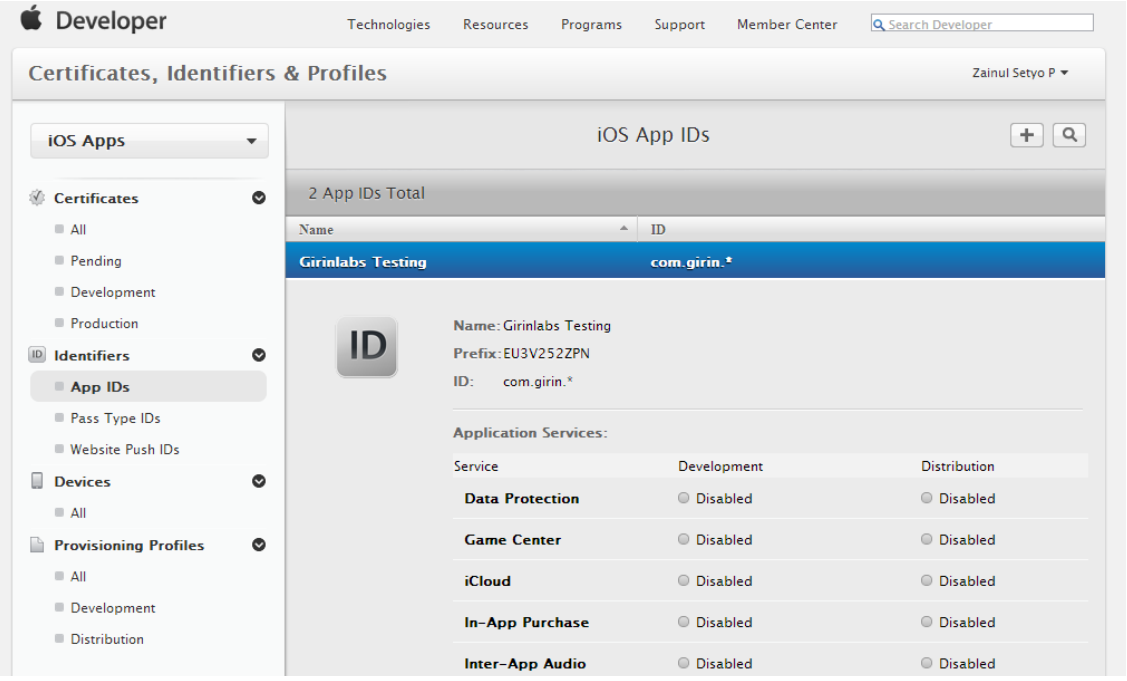
  Application name can be configured by editing
- Change application version

  Changing application version is easy. There is version attribute inside `<widget>` element. Let's change our application version in to `1.1.0` .

[7] https://github.com/justmyfreak/Phonegap-Build-Starter

- Change application id

  Application ID is unique identifier of application. It usually follow this rule : `com.vendor.appname` . Application ID can be edited by changing id attribute inside `<widget>` element. For Android and Blackberry, application ID can be anything as you want. But for iOS, it should be the same with registered Application ID registered in https://developer.apple.com/account/ios/identifiers/b Bellow is example of application ID :



2 App IDs Total

Name	ID
Girinlabs Testing	com.girin.*

Name: Girinlabs Testing
Prefix: EU3V252ZPN
ID: com.girin.*

Application Services:

Service	Development	Distribution
Data Protection	○ Disabled	○ Disabled
Game Center	○ Disabled	○ Disabled
iCloud	○ Disabled	○ Disabled
In-App Purchase	○ Disabled	○ Disabled
Inter-App Audio	○ Disabled	○ Disabled

  If you are using wildcard application ID like me, you can just use `com.vendor.*`. For this application, let's use `com.vendor.starterapp`

- Change application description

  Application description describe about details of your application. It will be displayed inside your PhoneGap Build account. To change your application description, you can edit `<description>` element as you like.

- Change PhoneGap version

  Changing PhoneGap version is required to make your application up to date with latest version of PhoneGap and its benefit. You can edit it in value attribute of `<preference>` element with phonegap-version name attribute. Before changing your PhoneGap version, make sure it is supported by checking it at https://build.phonegap.com/docs/config-xml[9].

Step 3 – Upload and deploy App

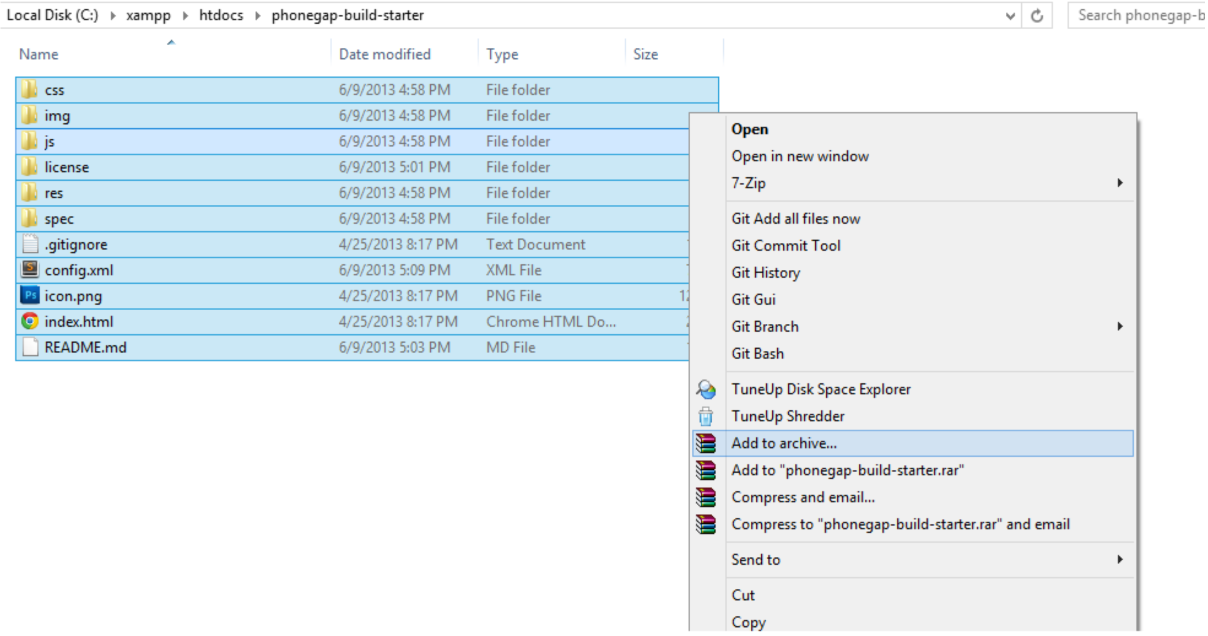
- Zip your application

  After finishing preparing application's assets and its configuration, we just need to archive it into `.zip` file. Make sure you archive from root level of application. If you are using

[8] https://developer.apple.com/account/ios/identifiers/bundle/bundleList.action

[9] https://build.phonegap.com/docs/config-xml

Windows and have Winrar or similar programs, all you need to do is just select all files and folders and then right click and choose Add to archive.

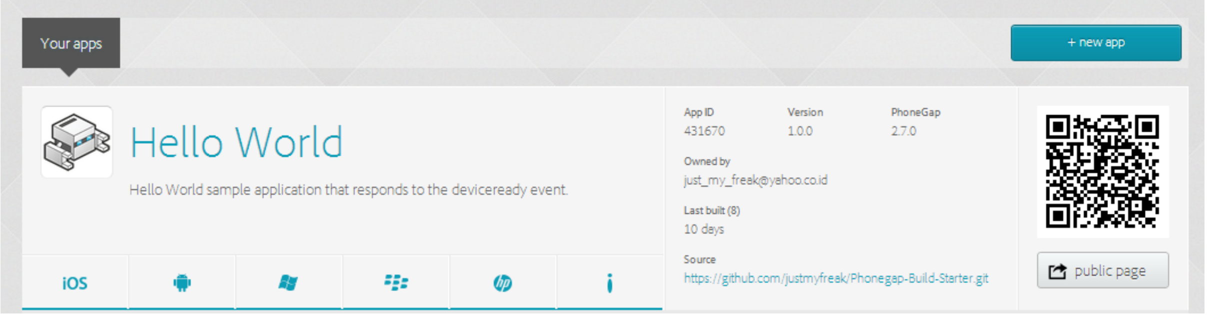


Name	Date modified	Type	Size
css	6/9/2013 4:58 PM	File folder	
img	6/9/2013 4:58 PM	File folder	
js	6/9/2013 4:58 PM	File folder	
license	6/9/2013 5:01 PM	File folder	
res	6/9/2013 4:58 PM	File folder	
spec	6/9/2013 4:58 PM	File folder	
.gitignore	4/25/2013 8:17 PM	Text Document	
config.xml	6/9/2013 5:09 PM	XML File	
icon.png	4/25/2013 8:17 PM	PNG File	
index.html	4/25/2013 8:17 PM	Chrome HTML Do...	
README.md	6/9/2013 5:03 PM	MD File	

image

- Create new application

In this application, we will be using .zip deployment instead of Git. To create new application, simply log in to your PhoneGap Build account. Then choose new app.



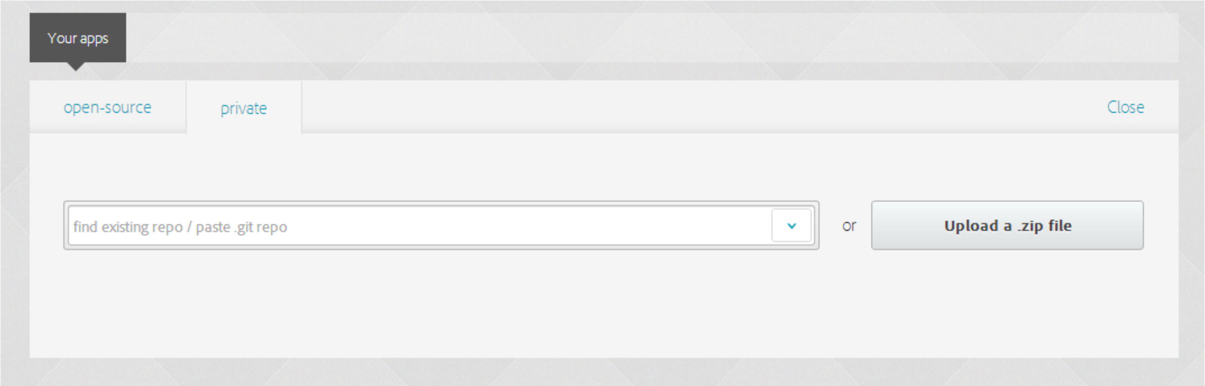
Hello World

Hello World sample application that responds to the deviceready event.

App ID: 431670
Version: 1.0.0
PhoneGap: 2.7.0
Owned by: just_my_freak@yahoo.co.id
Last built (8): 10 days
Source: https://github.com/justmyfreak/Phonegap-Build-Starter.git

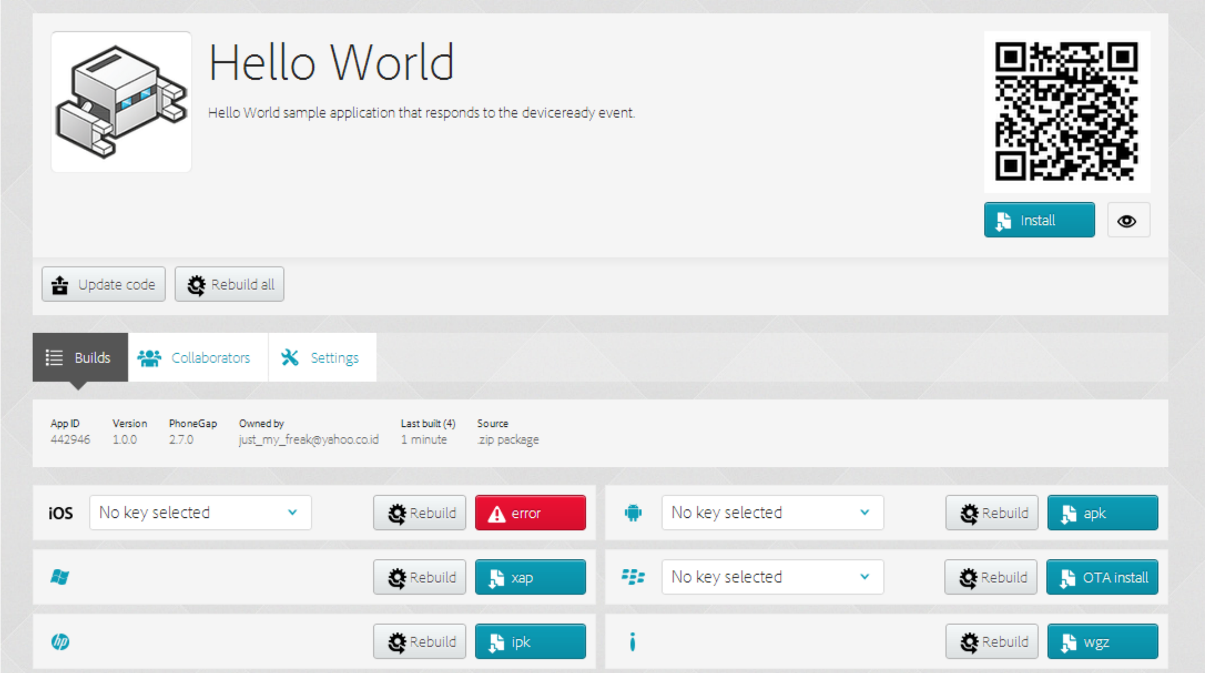
image

Hit `Upload a .zip file` button. Select our `.zip` archive we have created earlier.



image

It will take a while for uploading your archive. After upload completed, click `Ready to build` button then PhoneGap Build will do its magic to compile your application. This process take some times depending on server queue, so be patient.


Hello World

Hello World sample application that responds to the deviceready event.

App ID 442946 | Version 1.0.0 | PhoneGap 2.7.0 | Owned by just_my_freak@yahoo.co.id | Last built (4) 1 minute | Source .zip package

iOS No key selected

image

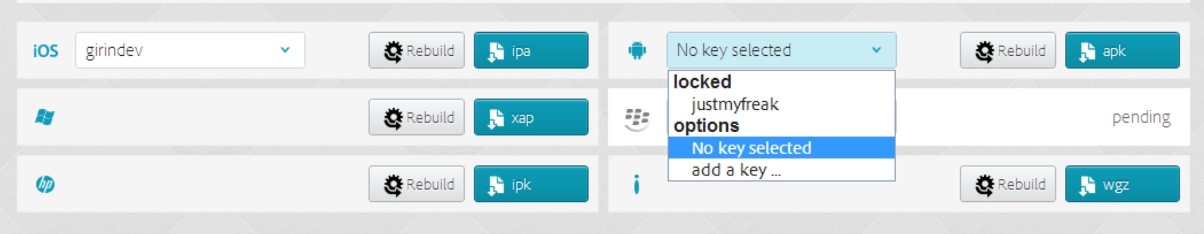
If you notice, everything is built successfully except for iOS. This is normal, so calm down. You have to unlock your sign key to build for iOS. This will be covered in next step.

- Unlock sign key and build

As mentioned earlier, we have to unlock sign key to build application for iOS. iOS need to sign application whether it used for development or production. If you are using development certificate and profile, you will be able to install your application in selected device or jailbroken devices. To distribute your application through Apple App Store, you need to use production certificate and profile.

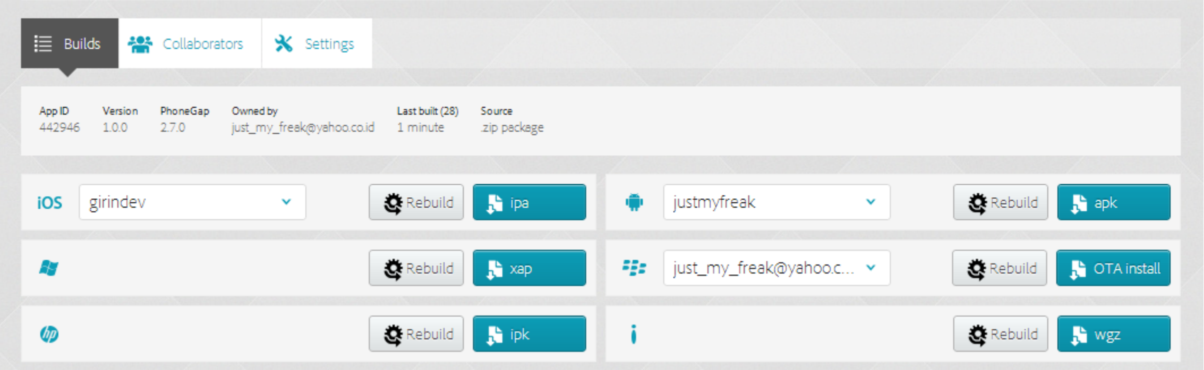
Android and Blackberry require no sign key to be able to build application. But you will need to unlock your sign key to be able distributing application through Android Play Store and Blackberry App World.

We have added our sign keys in chapter 2. So all we need to do is just unlock its password. To do this, you just need to click on dropdown list for platform you want to unlock sign key for.



image

You will see there are locked sign key(s), then choose it. You will see lock icon on right of dropdown list. Click it and insert your password. Right after submitting password, it will rebuild automatically. But if it is not rebuild or having error, you can rebuild manual by clicking `Rebuild` button. Now, you have your application ready for distribution.



App ID	Version	PhoneGap	Owned by	Last built (28)	Source
442946	1.0.0	2.7.0	just_my_freak@yahoo.co.id	1 minute	.zip package

image

- Download application

Now you have all of your application ready to download. All of your applications are ready for distribution to each platform corresponding digital market. Now you can relax and submit your application to market. Be prepared for next great features of PhoneGap Build

Top Features You'll Want to Know About

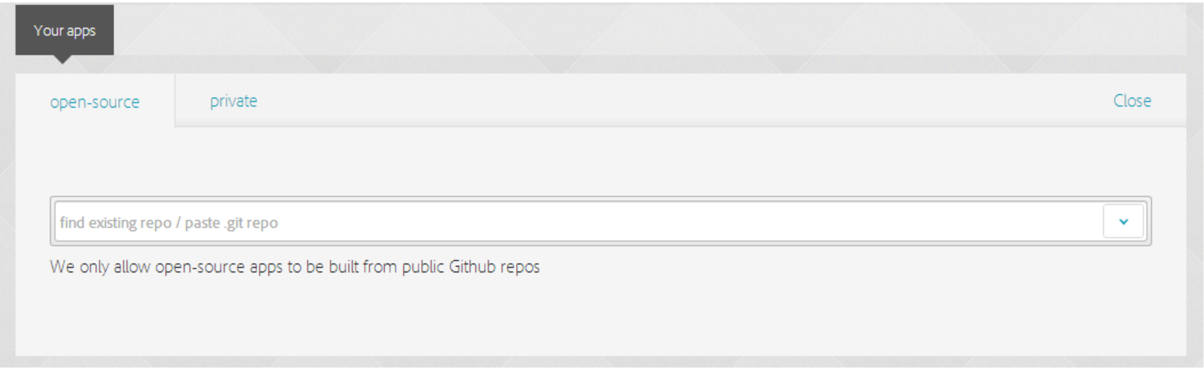
As you start to use PhoneGap Build, you will realize that there are a wide variety of things that you can do with it. This section will teach you all about the most commonly performed tasks and most commonly used features in PhoneGap Build.

1 – Git Deployment

We have talked about Git deployment in previous chapter. It is an alternative way to upload your HTML5 assets other than uploading .zip archive of your application. This is your only way to deploy open source application on PhoneGap Build. We have already knows that free account only allow us to have one private application and unlimited open source application.

By using Git as our deployment method, we get benefit in speed of uploading source code as well as source code management. Source code can be pushed to your GitHub account then latter be pulled in your PhoneGap Build application. If you are not familiar with Git and GitHub, take a look at http://try.github.io[10] for interactive tutorial. Trust me, it is worth to try and learn.

Let's create application using GitHub repositories. Go to your account and click `new app`. Instead of choosing `private`, let's create open source application. Click on `open-source` tab and text field will be shown.



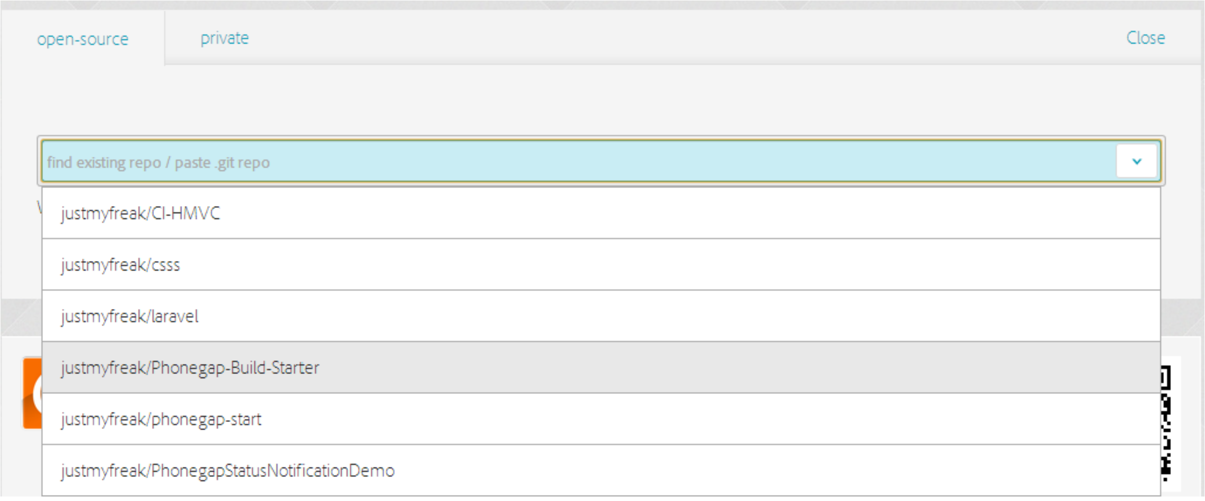
We only allow open-source apps to be built from public Github repos

image

If you have linked your PhoneGap Build account with your GitHub account, you will see that all your open source repos are available. If you cloned my https://github.com/justmyfreak/Phonegap-Build-Starter[11] repos, you can select it. Otherwise, just paste `https://github.com/justmyfreak/Phonegap-Buil` in your text box and PhoneGap Build will automatically fetch your source code.

[10] http://try.github.io

[11] https://github.com/justmyfreak/Phonegap-Build-Starter



justmyfreak/CI-HMVC

justmyfreak/csss

justmyfreak/laravel

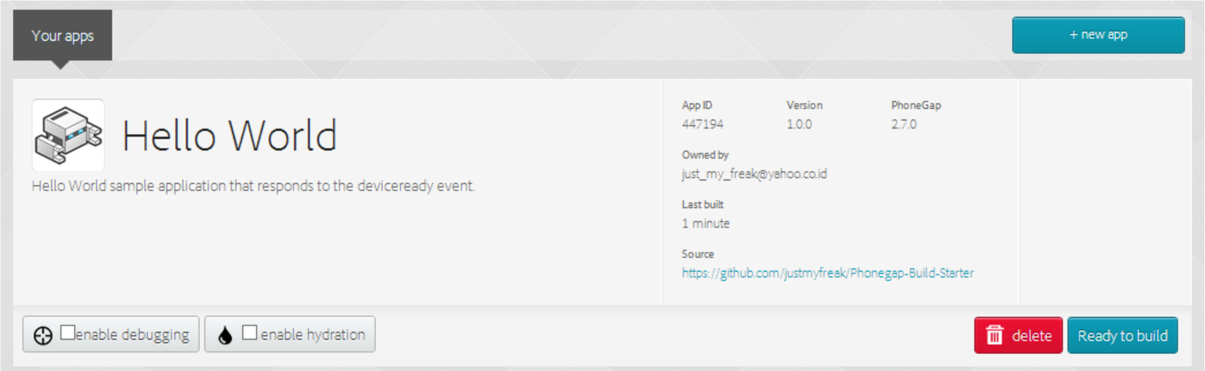
justmyfreak/Phonegap-Build-Starter

justmyfreak/phonegap-start

justmyfreak/PhonegapStatusNotificationDemo

image

After PhoneGap Build done fetching your source code, your application configuration will be fetched from `config.xml` and displayed. You may review it first. If everything is okay, just hit `Ready to build`. Once again, PhoneGap Build will do its magic to build you application. It takes some times to build your application.



Hello World

Hello World sample application that responds to the deviceready event.

App ID: 447194

Version: 1.0.0

PhoneGap: 2.7.0

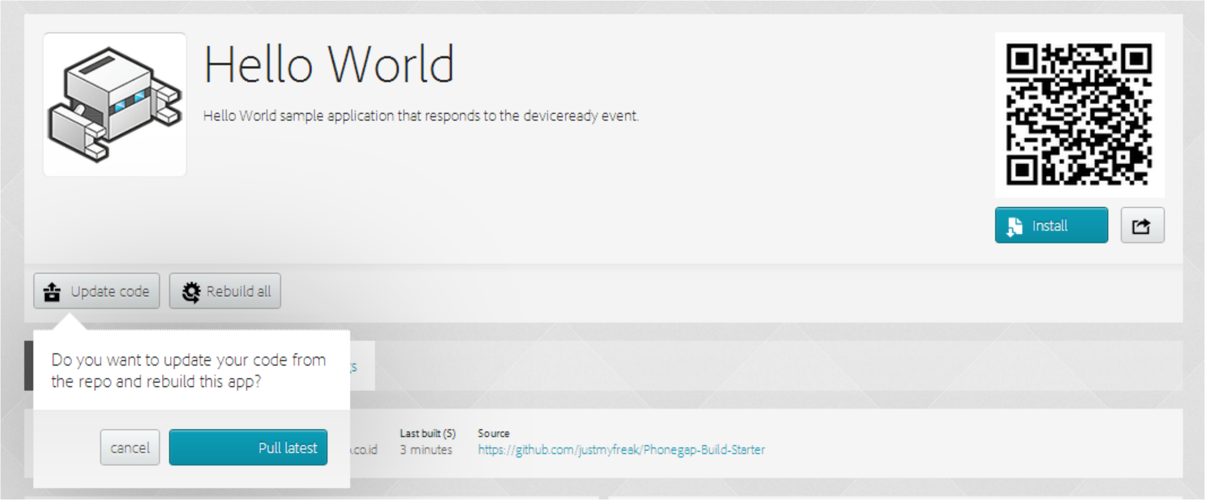
Owned by: just_my_freak@yahoo.co.id

Last built: 1 minute

Source: https://github.com/justmyfreak/Phonegap-Build-Starter

image

Once your application has been build successfully, you can download it and publish it. If you want to update your source code, simply commit change and push to your GitHub repository. Then you can pull latest changes by clicking Update code in your application dashboard.



Hello World

Hello World sample application that responds to the deviceready event.

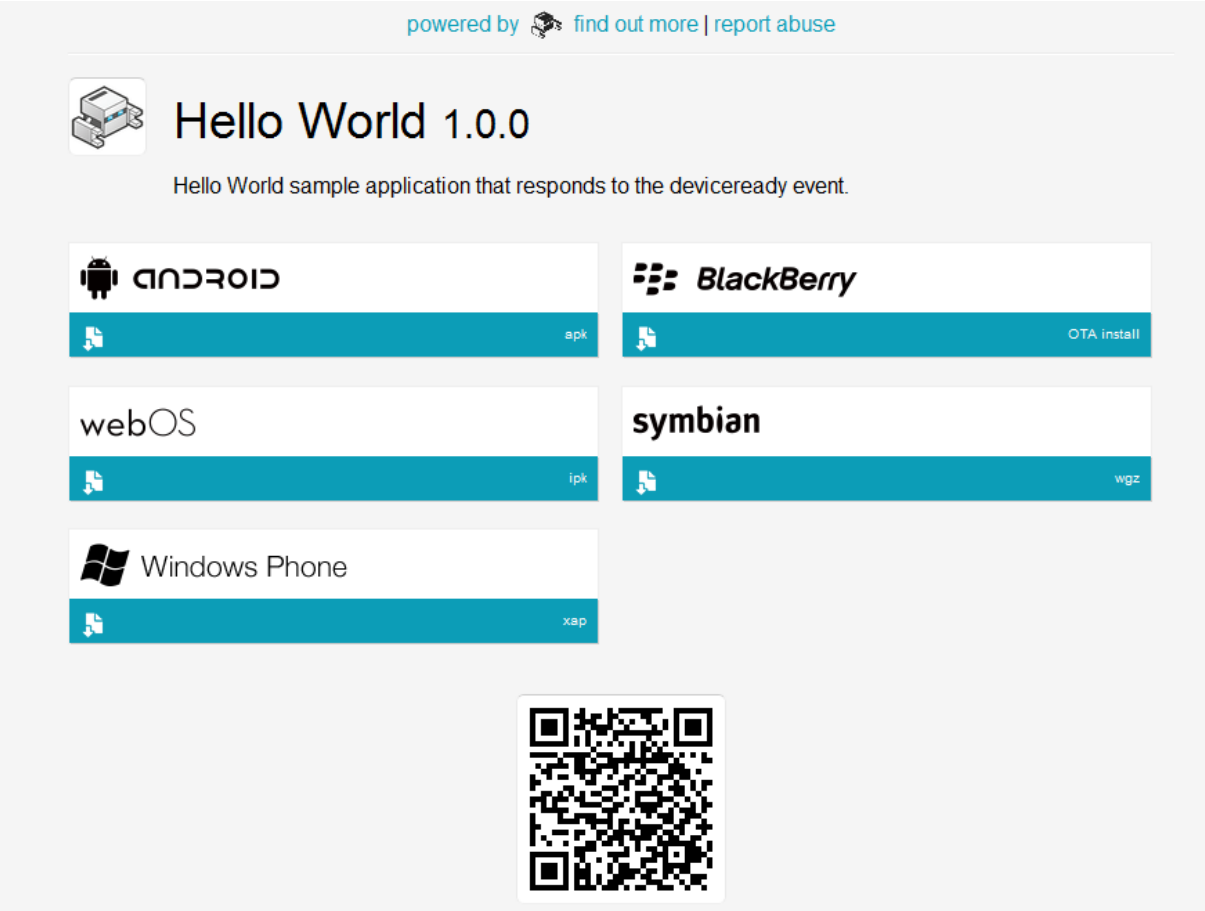
Do you want to update your code from the repo and rebuild this app?

Last built (5) 3 minutes

Source https://github.com/justmyfreak/Phonegap-Build-Starter

image

If you notice, there will be share icon next to install button. Since we are creating open source application, we have a nice landing page for sharing your application to others without submitting to each market. It looks like the following :



Hello World 1.0.0

Hello World sample application that responds to the deviceready event.

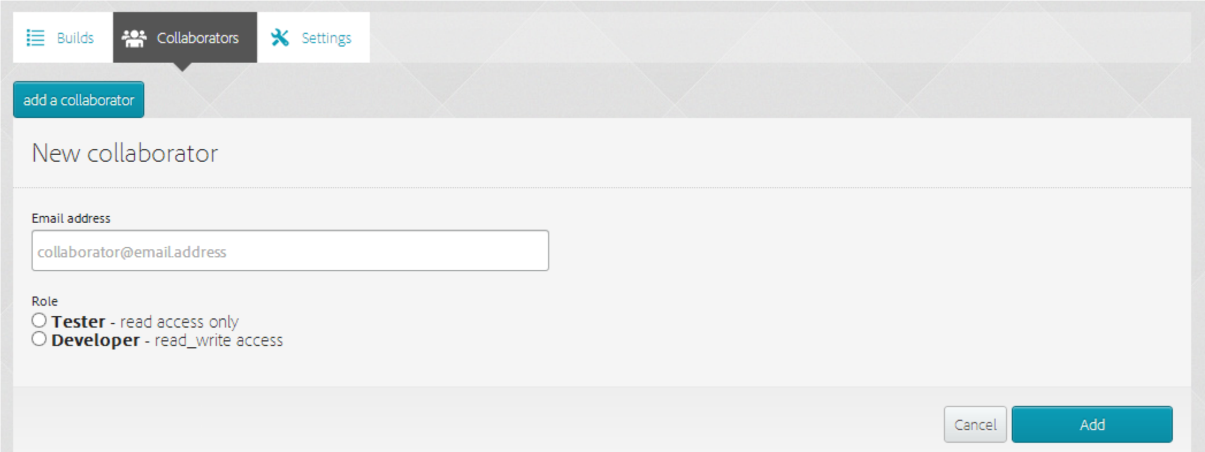
image

2 – Collaboration Development

Another nice thing about PhoneGap Build is collaboration in development. Collaboration not only can be done in writing application, but also in testing application. It will help us when working with others from different place.

There are two types of collaborators, they are Tester and Developer. Tester can only download compiled application, while developer can update source code. Collaboration development can be used in both open source and private application.

To add collaborators, simply click on collaborators tab in your application dashboard. Then you can add collaborator's email address and its role as tester or developer. Collaborator will be notified by email.



image

3 – Native Plugins

Plugins are used to extend your application to have native capabilities. Plugins need to be implemented differently for each platform. Plugins may not be supported in certain platform. Not all PhoneGap plugins can be used in PhoneGap Build. List of supported plugins can be viewed on https://build.phonegap.com/plugins[12].

Plugins can be added to your project by using 2 steps :

1. Importing the native code using the config.xml
2. Referencing the JavaScript code for the pluginReferencing the JavaScript code for the plugin

1 - Importing the native code using the `config.xml`

Importing native code can be done by adding `<gap:plugin>` to your `config.xml`. For example :

[12] https://build.phonegap.com/plugins

```
<gap:plugin name="com.phonegap.plugins.example" version="2.2.1" />
```

For plugins versioning, you can use the tilde ∼ such as :

```
<gap:plugin name="com.phonegap.plugins.example" version="~2" />
```

it will load the latest 2.x version.

```
<gap:plugin name="com.phonegap.plugins.example" version="~2.3" />
```

it will load the latest 2.x version as long as the x is greater or equal to 2

```
<gap:plugin name="com.phonegap.plugins.example" version="~2.3.4" />
```

it will load the latest 2.3.x version as long as the x is greater or equal to 4.

Plugin Parameters

Some plugins may require additional information to work. This can be done by adding some childern to `<gap:plugin>` like this :

```
<gap:plugin name="com.phonegap.plugins.someplugin">
  <param name="APIKey" value="myapikey" />
  <param name="APISecret" value="myapikeyscret" />
</gap:plugin>
```

Example :

Here is an example to add Local Notification to your project :

```
<?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.phonegap.example"
    versionCode = "10"
    version     = "1.0.0" >

    <!-- versionCode is optional and Android only -->

    <name>PhoneGap Example With Local Notification</name>

    <description>
      Your awesome application description
    </description>

    <author href="https://build.phonegap.com" email="mail@domain.com">
      Your Name
    </author>

    <!-- We'll include the Local Notification plugin as an example -->
    <gap:plugin name="de.appplant.cordova.plugin.local-notification" />
</widget>
```

2 - Referencing the JavaScript code

Some plugins utilize the `js-module` element to make cordova load the plugin javascript. Then, no `<script>` refernces in your html is needed. But 3rd party plugin may need to add javascript manually. Please refer to plugin documentation. If you need to add it manually, you can add `<script>` reference like this :

```
<script src="cordova.js"></script>
<script src="localnotification.js"></script>
```

Whether the `script` tag is required or not, you **must not** add actual javascript files of plugin to your repository or your build will fail. These files will be injected automatically by PhoneGap Build.

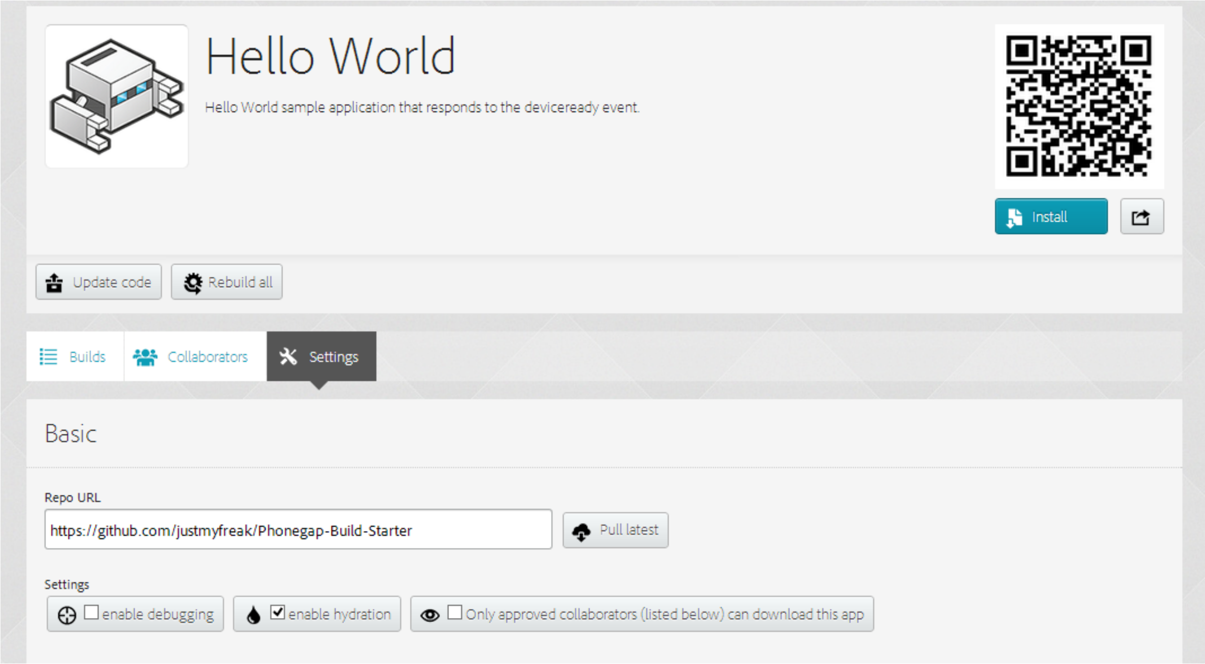
4 – Hydration

Hydration is a tool that has two main benefits for both developers and testers. Improving compilation times and enabling updates pushed directly to application installed on a device. This feature allows us to make instant update without submitting new binary to market. This is very handy for publishing iOS application. Since Apple's review process takes some days to get approved and published to their App Store.

PhoneGap accomplishes this by compiling native binary that acts as container for your mobile application. Once developer uploads a new build, the end user (eg. tester) of the application will be notified upon restart of application. If user decides to run your latest code update, the new code will be automatically fetched and installed in end user device.

To be able to use hydration, you have to configure your application to enable hydration. Remember our last open source application using https://github.com/justmyfreak/Phonegap-Build-Starter[13] as source code? let's add hydration to it. This can be done by clicking settings tab inside your application dashboard. Then check `enable hydration` and save. Your application will be rebuilt

[13] https://github.com/justmyfreak/Phonegap-Build-Starter



Hello World

Hello World sample application that responds to the deviceready event.

Basic

Repo URL

https://github.com/justmyfreak/Phonegap-Build-Starter

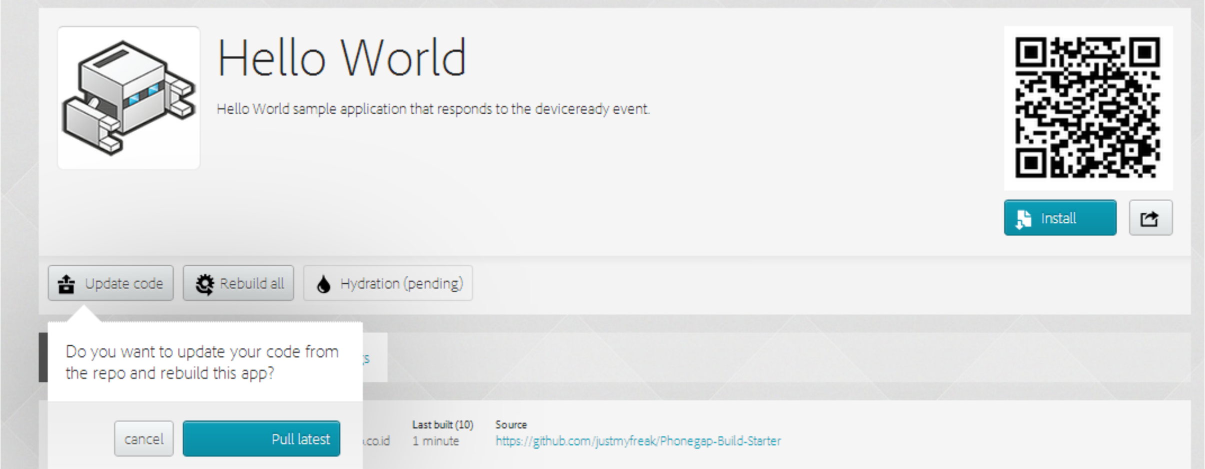
Settings

□ enable debugging ☑ enable hydration □ Only approved collaborators (listed below) can download this app

image

Let's make update to our source code repository by committing change to GitHub. If you want edit your own source code, you may fork my https://github.com/justmyfreak/Phonegap-Build-Starter[14] repo and create new application using your own repo url.

After making changes to your application, be sure to pull latest code by clicking update code and Pull latest. You will see Hydration (pending) which means your code will not be sent to device. After Hydration (complete) changes will be sent to device and user will be prompted to update code.



Hello World

Hello World sample application that responds to the deviceready event.

Do you want to update your code from the repo and rebuild this app?

Last built (10) 1 minute

Source https://github.com/justmyfreak/Phonegap-Build-Starter

image

When using hydration, you may **not** change configuration including name, version, version code, icons, splash screen, preferences, features, and access tags. If you change configuration, PhoneGap Build will generate new binary of your application instead of using hydration.

[14] https://github.com/justmyfreak/Phonegap-Build-Starter

To disable your hydration simply go to application's setting tab and uncheck enable hydration. Don't forget to hit save button.

More Learning Resources

If you need help with Phonegap Build, here are some people and places which will prove invaluable:

Official Site

- Homepage : http://build.phonegap.com[15]
- Manual and Documentation : http://build.phonegap.com/docs[16]
- Blog : http://phonegap.com/blog/phonegap-build/[17]
- PhoneGap Developer Directory : http://people.phonegap.com/[18]

Articles and Tutorials

- PhoneGap and PhoneGap Build in 5 Minutes : http://coenraets.org/blog/2012/12/phonegap-and-phonegap-build-in-5-minutes/[19]
- Push Notification : http://devgirl.org/2013/01/24/push-notifications-plugin-support-added-to-phonegap-build/[20]
- PhoneGap Build Landing Page on Adobe Edge : http://html.adobe.com/edge/phonegap-build/[21]

Community

- Official forums and support : http://community.phonegap.com/nitobi/products/nitobi_-phonegap_build[22]
- Community Portal : http://phonegap.com/community/[23]
- Official IRC channel : `irc.freenode.net #phonegap`
- User FAQ : https://build.phonegap.com/faq[24]

[15] http://build.phonegap.com

[16] http://build.phonegap.com/docs

[17] http://phonegap.com/blog/phonegap-build/

[18] http://people.phonegap.com/

[19] http://coenraets.org/blog/2012/12/phonegap-and-phonegap-build-in-5-minutes/

[20] http://devgirl.org/2013/01/24/push-notifications-plugin-support-added-to-phonegap-build/

[21] http://html.adobe.com/edge/phonegap-build/

[22] http://community.phonegap.com/nitobi/products/nitobi_phonegap_build

[23] http://phonegap.com/community/

[24] https://build.phonegap.com/faq

Blogs

- The blog of core developer of PhoneGap : http://brian.io/[25]
- PhoneGap developerâ€™s blog : http://phonegap.com/blog/[26]
- PhoneGap case study : http://phonegap.com/case/[27]
- Holly Schinsky Blog : http://devgirl.org[28]

Twitter

- @PhoneGapBuild[29] : Official PhoneGap Build Twitter Account
- @PhoneGap[30] : Official PhoneGap Twitter Account. Get updates about PhoneGap that power PhoneGap Build
- @brianleroux[31] : PhoneGap developer.

[25] http://brian.io/
[26] http://phonegap.com/blog/
[27] http://phonegap.com/case/
[28] http://devgirl.org
[29] http://twitter.com/PhoneGapBuild
[30] http://twitter.com/PhoneGap
[31] http://twitter.com/brianleroux

Contributing to this book

This book manuscript is open sourced on GitHub[32]. I decided to make manuscript open source so everybody can contribute and improve this book to be a better book. You can read the details about contribution on my blog post[33]. Credit will be given to every one who contribute to improve this book.

[32] https://github.com/justmyfreak/phonegap-build-starter-book

[33] http://justmyfreak.com/phonegap-build-starter-book-open-sourced/

Final Words

Thanks for reading my book :). If you find something need to be fixed or added feel free to contact me at @justmyfreak[34]. I'll be glad to hear from you.

[34] http://twitter.com/@justmyfreak