

Using PhoneGap Build to Create Apps

Anders Borg, Abiro, 2018-01-18, revision 10 (for PGB Template 0.06, cli-7.1.0)

Introduction

You've decided you want to develop apps, but you find cross-platform development daunting and you can't make use of your long experience in Web technology when developing native apps?

Look no further, PhoneGap Build is for you (there are other options, but for the sake of this tutorial, let's ignore that).

The tutorial is intentionally bare-bones, to highlight that starting and repeating PhoneGap-based projects is quite easy compared to developing native apps.

I hope what is described is independent of what OS you are using, but I can't promise as I only use Windows and macOS.

It assumes you already have an account at PhoneGap Build and that you will make the app for iOS and Android only. Developing for e.g. Windows Phone should work as well without major changes.

Being experienced in HTML, JavaScript and CSS is a reason and prerequisite for using PhoneGap. After the initial configuration of a project, almost every other aspect of the app development is related to Web technologies.

See References for the external sites and services mentioned in the text.

PGB = PhoneGap Build below.

Note: PhoneGap is mostly based on the open source project Cordova, and there's a transition from Adobe's side to even more adhere to Cordova releases, code and documentation. Therefore, there's a tendency that documentation for Cordova is more up-to-date than for PhoneGap. More about that below.

Disclaimer

The documentation for Cordova, PhoneGap and PhoneGap Build is somewhat contradicting taken together. You might find the Cordova documentation more up to date, yet there are some things that relate to specifically PhoneGap Build.

Adobe provides a forum for PhoneGap (see References), so submit questions about PhoneGap and PhoneGap Build there. If you find errors in this tutorial, please let me know.

Publishing an app at any of the stores is very specific to each store, and at times also very frustrating, especially when it comes to iOS. You need to look elsewhere for such information.

During development, you and your test users can install directly from PhoneGap Build. Optionally use TestFlight for iOS to simplify distribution of prototypes to many, otherwise you need to register device UDIDs in the provisioning profile. From a security standpoint, TestFlight is better. To be able to install Android versions from PhoneGap Build you need to enable Unknown Sources in your device.

The project structure

As indicated before, PhoneGap is almost all about HTML, JavaScript and CSS, but there are a few important distinctions, considering a PhoneGap app is a native app from a platform's perspective.

Create a root folder for all your PhoneGap projects, e.g.:

/PhoneGap

Create a sub folder there for each project you create. To be on the safe side, don't use spaces and use only English letters and digits. E.g.:

/PhoneGap/MyApp

Copy the template accompanying this tutorial to your project's folder. The structure is:

.cordova (optional; see Quicker deployment below)

www (the root for your PhoneGap project proper)

www/css (where your stylesheet files go)

www/js (where your JavaScript files go)

www/res (images that are used for app icons and splash screens)

The folders contain files that will get you started on your quest.

You don't have to follow this structure, but this tutorial assumes you do, and **config.xml** heavily relies on it.

Use the same file structure for your projects, to make it easy to use previous projects as a starting point for new ones, and to simplify comparisons and migrations.

Configuration

Edit **config.xml** in an XML and UTF-8 aware editor (preferably).

Replace the following with your information:

- **id** (your app's package name; use reverse domain syntax, a la com.company.project)
- **name** (the app's name that will show up on PGB; you can give it a different name in the stores)
- **description** (the app's description -"-)
- **author** (your e-mail address and site)

Create a ZIP file of everything under **www** inside of that folder.

Upload this file via PGB's **+ new app**.

Build it in PGB.

If it fails, you most likely haven't set up certificates. Android doesn't need any while developing (PGB will apply its own developer certificate, if you don't upload/select any). For iOS you always need one. If you

don't know how that is created leave that for now, provided you at least have an Android phone. See References for more information.

You can also view the build **Log** per platform, if something went wrong in the building process.

Read the displayed QR code from an applicable device, and install and run the app.

After starting the app, it should simply say "Wassup world?" and pop up a Javascript alert. See **index.html** and **applogic.js** for why that happens.

Creating images

To create the icons and splash screens needed for your project you should use **PhoneGap Image Generator** (see References). **config.xml** in the template refers to images according to the structure created by this tool.

At least upload a high-resolution square logo (preferably 512 pixels or better) to be used as the app icon and in the center of splash screens. Possibly also upload a splash screen image. Images will be scaled up or down as needed.

Let the tool do its magic, and download the image pack.

Decompress the ZIP file and move all of it to **www**. Move the **archive** folder outside of the project.

Please don't remove **.pgbomit** under **res**. If it wasn't there, all images would be duplicated for each platform, creating an unnecessarily large app.

Creating an app

index.html, **applogic.js** and **styles.css** should get you started with your project. Leave **index.js** be for now. It secures that **applogic.js** is not called until PhoneGap has been initialized.

As this is all about your creativity and your own and your customers' requirements, here are a few pointers:

- Get the design aspects done early, and use the same styles throughout the app. If you develop this app for customers, this way the design can be signed off early on. Otherwise there's a big risk you will spend most of the project time adjusting the design. Remember, a customer usually has no clue about application logic. They only comment on what they see and can work with on the surface.
- Invoke the plugins you need in **config.xml**. The example file lists a few important ones that I've found are needed in almost all projects. There are tons of plugins available (see References).
- PGB recommends locking what version of plugin to use using **spec**. I recommend you to not do that, but instead use the very latest published versions overall, unless you run into problems.

HTML etc. support in phones

Newer smartphone OSes and browsers tend to be fully compliant with HTML5 and CSS3, but you may still find that some features are not supported. Not the least the plethora of new input field types, overflow options, animations etc., so don't just assume everything in HTML5 and CSS3 will work on all phones. After all, PhoneGap relies on the WebView in the platform.

Also note that in older versions of Android (4.3 and prior), the WebView is not based on Chrome, even if installed, but the older Android browser, which is not altogether HTML5 compliant, and never will be.

Always have phones and possibly tablets on hand running the platforms (and a few of the platform versions) you intend to support. Don't just assume the app will work as expected. Especially, if your app will support both Android and iOS, you need devices supporting both, as plugins have completely different code per platform, even though the JavaScript API is the same.

Options

Syntax checking

Most IDEs and editors have syntax checking and coloring for HTML, Javascript, CSS and XML, like NetBeans and Eclipse, but if you want more elaborate checking (and recommendations) you can use the W3C Markup Validation Service, by uploading what's under **www** to a server. That will not consider the content you create dynamically, unless you somehow bootstrap the JavaScript code, as no plugins are loaded outside of PhoneGap.

Quicker deployment

phonegap remote build makes it possible to deploy projects from the command line, which speeds up frequent test deployments a lot. I always submit new code to PGB that way, except the first time.

To prepare for using it, install Node.js and PhoneGap. See phonegap.com for instructions and links. You also must upload and build the app at least once before you can use this tool. Otherwise there's no PGB project and no App ID.

To have **phonegap remote build** know what PGB project to address, enter the PGB App ID in **.cordova/config.json**. Don't forget: This ID is different for each project. To not hit the limit of 25 private apps, I use a specific App ID for all test apps.

You build the app from the project root. You may use the included **br.bat** (for Windows, but similar scripts can be set up in macOS and Linux).

You always need to state a platform to build for (**phonegap remote build ios or android**). PGB will build for all platforms that are stated in **config.xml** anyway, yet not finish until the app has been built for the stated platform.

References

- Cordova: <https://cordova.apache.org/>
- PhoneGap: <http://phonegap.com/>
- PhoneGap Build: <https://build.phonegap.com/>
- PhoneGap Build community: <https://forums.adobe.com/community/phonegap/build>
- Plugins: <http://cordova.apache.org/plugins/>
- Plugins: <https://www.npmjs.com/>
- Apple Developer: <https://developer.apple.com/>
- Google Developers: <https://developers.google.com/>
- W3C Markup Validation Service: <http://validator.w3.org/>
- PhoneGap Image Generator: <http://pgicons.abiro.com/>