**Tips on Creating Android App from PWA**

[Progressive Web Apps](https://developers.google.com/web/progressive-web-apps/) (PWA) is a game changer for Mobile app development. We can turn our web page into a mobile compatible app with a little bit more effort. Create a manifesto file and write a service worker logic, then your website is a mobile app. Once you create PWA, you can package it as an Android app and place it in [Google Play](https://play.google.com) store by [using Trusted Web Activities](https://developers.google.com/web/updates/2019/02/using-twa) (TWA). Amazing.

Is it as easy as it sounds? I would say yes. I created an SPA with [Preact](https://preactjs.com/) (you can see [here](https://www.mydatahack.com/sitecore-9-certified-platform-associate-developer-exam-prep-quiz/)) and converted it into a mobile app placed in Google Play (check out the app [here](https://play.google.com/store/apps/details?id=com.mdhsitecorequiz.sitecorecertificationexamquiz)). Because I did no prior knowledge on any of these technologies (PWA, TWA and Android Studio), it took me about a week to convert my Preact app into an actual app appearing in Google Play, including the wait time to get my app approved (which took a few days).

As this activity is fairly straight forwards, there were a few gochas that I wished I had known. I am writing this to share them with you and save you from banging your head against the brick walls to resolve simple issues.

**Prerequisites**

* You have a [PWA app](https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp/) (this post is not about creating PWA app).
* [Android Studio](https://developer.android.com/studio) is installed.
* Google Play account is created (if not, go [here](https://support.google.com/accounts/answer/27441?hl=en)).

**Tips**

Here they come!

**Tip 1: Read documentations**

All you need to follow is this documentation: Using [Trusted Web Activities](https://developers.google.com/web/updates/2019/02/using-twa) although it has some flaws (which will be discussed in the coming tips). To supplement the knowledge, I also recommend you to read [Trusted Web Activities Quick Start Guide](https://developers.google.com/web/updates/2019/08/twas-quickstart). These two documentations supplement each other and fill the gaps for you.

**Tip 2: Be careful about customtabs library version**

In the module level build.gradle, you have to add dependencies for com.github.GoogleChrome.custom-tabs-client:customtabs. Picking the correct version is important. For some reason, the version currently on [Trusted Web Activities](https://developers.google.com/web/updates/2019/02/using-twa) does not work (91b4a1270b). Unless the documentation gets updated, don’t use that one. There are two versions I verified that work (7ab7178ae0 and d08e93fce3). I don’t know the differences. I ended up using 7ab7178ae0.

My setting looks like this in the dependencies section of build.gradle (Module: app):

<pre>

implementation **'com.github.GoogleChrome.custom-tabs-client:customtabs:7ab7178ae0'**

</pre>

**Tip 3: Icons from manifest.json are not used for the App**

App does not use the icons from manifest.json. You need to add icons through [Image Asset Studio](https://developer.android.com/studio/write/image-asset-studio) in Android Studio. Same goes for splash screen. You need to configure it in Android Studio. For my app, I left this as default.

**Tip 4: Create correct assetlink.json**

You do not need to write hacky custom JavaScript to hide the address bar. The reason why your app is still showing the address bar is the association between your site and app is not configured correctly. Read this blog post, [Why Address Bar Is Not Hidden – Android App with TWA](https://www.mydatahack.com/why-address-bar-is-not-hidden-android-app-with-twa/).

**Tip 5: Debugging App on Emulator on full-screen mode.**

Using Trusted Web Activities has a section on this. At first read, it wasn’t clear. I just add some supplementary explanation.

Firstly, go to the Emulator and enable *Enable command line on non-rooted devices* by typing chrom://flags in Chrome’s address bar. This is only needed on the emulator.

From your local machine, you need to run [Android Debug Bridge](https://developer.android.com/studio/command-line/adb) command. On Windows machine this command is already installed in AppData folder in your home directory as C:\Users\yourUserName\AppData\Local\Android\Sdk\platform-tools.

The emulator needs to be running when the command is executed.

<pre>

cd C:\Users\yourUserName\AppData\Local\Android\Sdk\platform-tools.

adb shell "echo '\_ --disable-digital-asset-link-verification-for-url=\"https://airhorner.com\"' > /data/local/tmp/chrome-command-line"

</pre>

**Tip 6: Version number needs to be updated to upload new version**

When you upload a new version to Google Play, you need to update the versionCode and version Name in build.gradle (Module.app) and compile it.

<pre>

defaultConfig {  
 …  
 versionCode 5  
 versionName **"1.5"** …}

</pre>

**Tip 7: Don’t freak out if the pre-launch report has crashes**

If it works on the newer models of phones, don’t worry about your app crashing in older androids. Event though it reports crashes for some phones, your app will be approved for publishing.

Some of the old phone does not have the right chrome versions. Also don’t worry too much about errors and warnings if they are coming from CustomTab module. It will probably work on most of the phones that you and your friends have.

I wasted so much time on fixing every warnings and errors. That’s what developers do, right. We ship products without any warnings or errors. In this case, I stopped worrying about it and decided to let technologies to catch up.

**My TWA configuration**

Finally, let me share my TWA configurations I have done in Android Studio. I have touched 4 files as well as adding icon in res/mipmap. This configuration is the bear minimum to ship your app to the store.

**build.gradle (Project)**

<script src="https://gist.github.com/mydatahack/40259b6152174d24c8f7ce64aa772455.js"></script>

**build.gradle (Module)**

<script src="https://gist.github.com/mydatahack/f9f7290b4beee8adadaa6bc96afc0de0.js"></script>

**AndroidManifest.xml**

<script src="https://gist.github.com/mydatahack/8812939b43ac9ac68f58260013fcad17.js"></script>

**res/values/string.xml**

<script src="https://gist.github.com/mydatahack/bd6ea082c68e62a31527ae32f1f911c9.js"></script>