

CodeHS Guide to Publishing an App on the App Store or Play Store

So, you made an awesome program on CodeHS and want to try to get it on the App Store or Play Store? In this document we have instructions for how to turn your CodeHS program into an app available for download from the App Store or Play Store. It's a tricky process, but if you're up for the challenge, go for it! This is also relatively new to CodeHS, and we've tried to sort out potential roadblocks, but please get in touch at hello@codehs.com if you run into problems.

Things to Know Before You Start

- We're assuming you have a Mac. We haven't tested this process on a PC. Throughout the process, you will be typing commands from Terminal.
- To submit an app to the App Store, you need an Apple Developer license, which costs \$99/year. You can
- To publish an app to the Play Store, you need a Google Play Developer account, which requires paying a one-time fee of \$25.
- If you are submitting to the App Store, Apple will review your app and decide whether to accept it into the App Store. For more information, see the [App Store Review Guidelines](#) and [Common App Rejections](#).

Want to make an Android App or an iOS App?

[Publishing an Android App to the Play Store](#)

[Install Tools](#)

[Download and Setup App Files](#)

[Build the App](#)

[Test the App](#)

[Upload the App to the Play Store](#)

[Publishing an iOS App to the App Store](#)

[Install Tools](#)

[Download and Setup App Files](#)

[Build the App](#)

[Upload the App to iTunes Connect](#)

[Test the App](#)

[Submit the App to the App Store](#)

Publishing an Android App to the Play Store

Install Tools

1. Node.js
 - a. [This resource](#) has step-by-step instructions for how to install Node onto your Mac. Make sure to update npm also.
2. exp command line tools
 - a. You need an Expo account. Go to expo.io and sign up for an account if you don't have one yet.
 - b. Open Terminal and run `npm install -g exp`. Then login with your Expo account using `exp login`.
3. adb command line tools
 - a. If you want to test your app on an Android device before submitting it to the Play Store, you need adb command line tools installed. There are different ways to install adb, so see [here](#) for instructions.

Download and Setup App Files

1. On CodeHS in your browser, open the graphics or React Native program that you want to publish as an app. Navigate to the Share Tab (on the far right, click More > Share).
2. Click the "Download App Files" button. This will download a zip file. Double-click the zip file to unzip it.
3. You need to add two images to this folder. One will be the icon for your app, and it needs to be a square PNG file named `appIcon.png`. Transparency is acceptable, and often encouraged for this icon. The second image is an image to display while your app is loading. This image needs to be a PNG file (not necessarily square) named `loadingIcon.png`.
4. You shouldn't need to change anything in the three files that were downloaded. If you are attempting to resubmit an app that you have already uploaded to the Google Play Console, then you need to open `app.json` and increment `versionCode` by 1. For example, if `versionCode` of the version you already uploaded was 1, then change it to 2.

Build the App

1. From Terminal, navigate into the folder you unzipped, which should be called `App`. If you've never navigated through directories from the command line before, [this article](#) should be helpful.
2. Run the command `exp build:android`. When asked for keystore information, type "1" to let Expo handle it for you. Run `exp build:status` to monitor the progress of the build.

3. When the build is done, there will be a URL. Open the URL in your web browser. This should download an apk file. Move this apk file into your App folder and rename it something simple, like `app.apk`.

Test the App

1. Connect an Android device to your computer. The connected device must have developer settings enabled, so enable them if not enabled by following [these instructions](#).
2. From Terminal, run `adb install app.apk` (or the different simple name you gave your app).
3. Open the installed app on the device and test.

Upload the App to the Play Store

1. From the [Google Play Console](#), create a new application by clicking “Create Application” in the top right corner.
2. Follow the instructions to fill in the descriptions and pricing, upload the required icons, etc.
3. Go to App Releases > Manage Production, and click “Create Release.” Now upload the apk file that you put into your project folder.
4. Finish filling in the required info for your app and apply a content rating.
5. Once the sidebar has four green checkmarks (next to App Releases, Store Listing, Content Rating, and Pricing & Distribution), return to App Releases and release your app to the Play Store.

Publishing an iOS App to the App Store

Install Tools

1. Node.js
 - a. [This resource](#) has step-by-step instructions for how to install Node onto your Mac. Make sure to update npm also.
2. exp command line tools
 - a. You need an Expo account. Go to expo.io and sign up for an account if you don't have one yet.
 - b. Open Terminal and run `npm install -g exp`. Then login with your Expo account using `exp login`.
3. Xcode
 - a. In order to upload your app to Apple, you need to have Xcode installed. You won't actually do anything from Xcode, but it's necessary to install. You can download [Xcode from the Mac App Store](#).
4. fastlane command line tools
 - a. You will use `fastlane pilot` to upload your app to iTunes Connect. [Here](#) are instructions to install fastlane. Use the Installer Script installation method (unless you have Homebrew).

Download and Setup App Files

1. On CodeHS in your browser, open the graphics or React Native program that you want to publish as an app. Navigate to the Share Tab (on the far right, click More > Share).
2. Click the "Download App Files" button. This will download a zip file. Double-click the zip file to unzip it.
3. You need to add two images to this folder. One will be the icon for your app, and it needs to be a square PNG file named `appIcon.png`. Apple does not allow transparency, so make sure your image has no transparency. The second image is an image to display while your app is loading. This image needs to be a PNG file (not necessarily square) named `loadingIcon.png`.
4. You shouldn't need to change anything in the three files that were downloaded. If you are attempting to resubmit an app that you have already uploaded to iTunes Connect, then you need to open `app.json` and increment `buildNumber` by 1 (but make sure to keep it a string). For example, if `buildNumber` of the version you already uploaded was "1", then change it to "2".

Build the App

1. From Terminal, navigate into the folder you unzipped, which should be called `App`. If you've never navigated through directories from the command line before, [this article](#) should be helpful.

2. Run the command `exp build:ios`. You will need to sign in with your Apple ID. When asked about distribution and push notification certificates, type “1” to let Expo handle those for you. Run `exp build:status` to monitor the progress of the build.
3. When the build is done, there will be a URL. Open the URL in your web browser. This should download an ipa file. Move this ipa file into your App folder.

Upload the App to iTunes Connect

1. From iTunes Connect, add a new app by clicking the “+” button on the left-hand side. Choose the bundle identifier for the app you just built (it should show up as an option).
2. In order to test your app and submit it to the App Store, you first need to upload your app to iTunes Connect, and the `fastlane pilot` command line tool lets you do this. From the command line, run `fastlane pilot upload -u youremail@example.com`. Make sure to use the email associated with your Apple Developer account.
3. It will take a few minutes to upload your build, and some more time for iTunes Connect to process it. Once it’s finished processing and ready for testing, you will receive an email.

Test the App

1. After the app has been processed, you can test it with TestFlight. From the iTunes Connect page for the app, go to TestFlight. Select the build you just uploaded, and add testers (probably yourself).
2. Testers need to have the [TestFlight app](#) installed on their phone. As long as they have TestFlight on their phone, they will be able to install the test version of the app.

Submit the App to the App Store

1. Fill in the App Information, Pricing & Information, and Prepare for Submission required fields. You’ll have to upload an icon and screenshots that are the specified sizes.
2. Click Submit for Review. You’ll be asked “Does this app use the Advertising Identifier (IDFA)?” Check these three boxes:
 - a. Attribute this app installation to a previously served advertisement
 - b. Attribute an action taken within this app to a previously served advertisement
 - c. I, YOUR_NAME, confirm that this app, and any third party...
3. Wait for app to be reviewed (and hopefully accepted into the App Store!). As referenced at the top of this guide, here are the [App Store Review Guidelines](#) and [Common App Rejections](#).