

Application Requirements

In order to upload a native application to Apperian, the application must meet certain requirements. This page provides a checklist that you can provide to your app developers to help them ensure that the apps they provide to Apperian administrators are ready to upload. For a detailed description of the requirements listed in the developer checklist, see [Understanding the Requirements](#).

Checklist for Application Developers

| Application Requirements | |
|--------------------------|--|
| <input type="checkbox"/> | Ensure a standard folder structure has been used inside the application file. |
| <input type="checkbox"/> | Ensure that the application is built with icons that are optimized for different screen resolutions. |
| <input type="checkbox"/> | Ensure the application is signed correctly. |
| <input type="checkbox"/> | Provide a list of entitlements/capabilities that are enabled for the application. (for iOS only) |
| <input type="checkbox"/> | Provide a list of the application's extensions. (for iOS only) |



To send the checklist above to your application developer, you can copy and paste it into an email or document.

Understanding the Requirements

Ensure a standard folder structure has been used inside the application file

If an app is built with a non standard folder structure, it will not upload to the Apperian Portal correctly.

When adding an iOS application, Apperian looks in the IPA file for a folder called "Payload." There should be only one `payload` folder, and it should contain only one `.app` file. If the `payload` folder is renamed or there is more than one, Apperian will be unable to recover the contents of an application and the app will not be uploaded. Structuring iOS applications this way is a general industry best practice.

For information on the standard structure of an Android application module that will be built into a `.apk` file, see [Android Application Modules](#).

Ensure that the application is built with icons that are optimized for different screen resolutions

Apperian requires an application to include an app icon in PNG format with at least a resolution of 114 x 114 pixels, but it is a best practice to include multiple icons that are optimized for different screen resolutions.

When adding an Android application, Apperian extracts the highest resolution icon image that it can find to display in both the Apperian Portal and the App Catalog. When adding an iOS application, Apperian extracts the first icon image listed in the `CFBundleIconFiles` array in the `info.plist` file regardless of its resolution. (For more information on the `CFBundleIconFiles` array, see the [iOS Developer Library](#).)

Although an image file is automatically extracted during the upload process, the administrator can upload a different icon through the Apperian Portal (when adding or editing the app) or via the API (using the [POST /applications/app_psk/icon](#) resource). While the icon uploaded to Apperian is what will appear in the Apperian Portal and the App Catalog, one of the icons bundled with the application file will display on the user's mobile device once the app is installed.

Ensure the application is signed correctly

All applications must be signed before they are uploaded to Apperian. Typically, app developers sign applications during the build process, but before administrators can distribute an application to their enterprise's users, they will need to re-sign the app with their company's signing credentials. This can be done using the Admin Portal or a signing package. For more information, see [About Signing](#).

Provide a list of entitlements/capabilities that are enabled for the application (for iOS only)

It is important that the administrator who is uploading and signing an organization's applications has all the relevant information for a particular app's enabled entitlements. When an administrator signs an app, they need to sign it with a distribution provisioning profile that has entitlements (app services) that match the entitlements (capabilities) built into the app. If the entitlements don't match, users will not be able to install the app on their devices. If the app includes app extensions, each extension must be signed with a different provisioning profile that has the correct entitlements.

It is best practice for the Apperian administrator to obtain a list of an app's entitlements from the app's developer before he or she attempt to sign an application. This is helpful to ensure that the administrator is signing with the appropriate distribution provisioning profile(s). If this information is not made available to the administrator, Apperian can help. When uploading a provisioning profile in the **Sign with New Credentials** section of the Signing tab, Apperian compares the entitlements in the profile with the app/extension. If the entitlements don't match, Apperian will warn the administrator and clearly identify what doesn't match. With this level of information, administrators can follow up with the appropriate member of their team to generate the necessary provisioning profile(s) from the Apple Dev Portal.

For more information on entitlements, see [Entitlements](#). For instructions on signing with new credentials, see [Sign an App \(Admin Portal\)](#).

Provide a list of the application's extensions (for iOS only)

It is important for an administrator to know if an app includes extensions. If an app does contain extensions, the administrator will need a separate explicit distribution provisioning profile to sign the app and each extension in the app. If an app contains a watchOS app for Apple Watch, the administrator also needs one distribution provisioning profile for the watchOS app and one for the watchOS app extension.

For more information on extensions, see [App Extensions and watchOS Apps](#).