Exploring Expo App Config Properties

Name: Defines the title of the document, defaults to the outer level name

Slug: The friendly URL name for publishing, The appName placed in slug will be the extension of the URL for the project.

Description: provides a general description of what the pinned website does. The owner of the project will be able to pass on a summary of the project to future developers who look at the project.

Owner: The owner is the name of the Expo account that owns the project. This is useful for teams collaborating on a project. If not provided, the owner defaults to the username of the current user.

Privacy: Defaults to unlisted. Unlisted hides the project from search results. Hidden restricts access to the project page to only the owner and other users that have been granted access. Valid Values: public, unlisted, hidden.

- It is important to have privacy values properly defined especially if the project has sensitive information that they don't want to be readily available and access outside of the team.

bundleIdentifier: The bundle identifier for your iOS standalone app, usually made up and needs to be unique on the App Store. Ex. "host.exp.expo" where "exo.host" is domain and "expo" is the app name

Package: This is the package name for the Android standalone app, usually made up but it has to be unique on the Play Store. Ex. "com.example.app" Where "com.example" is domain and "app" is the name of the app.

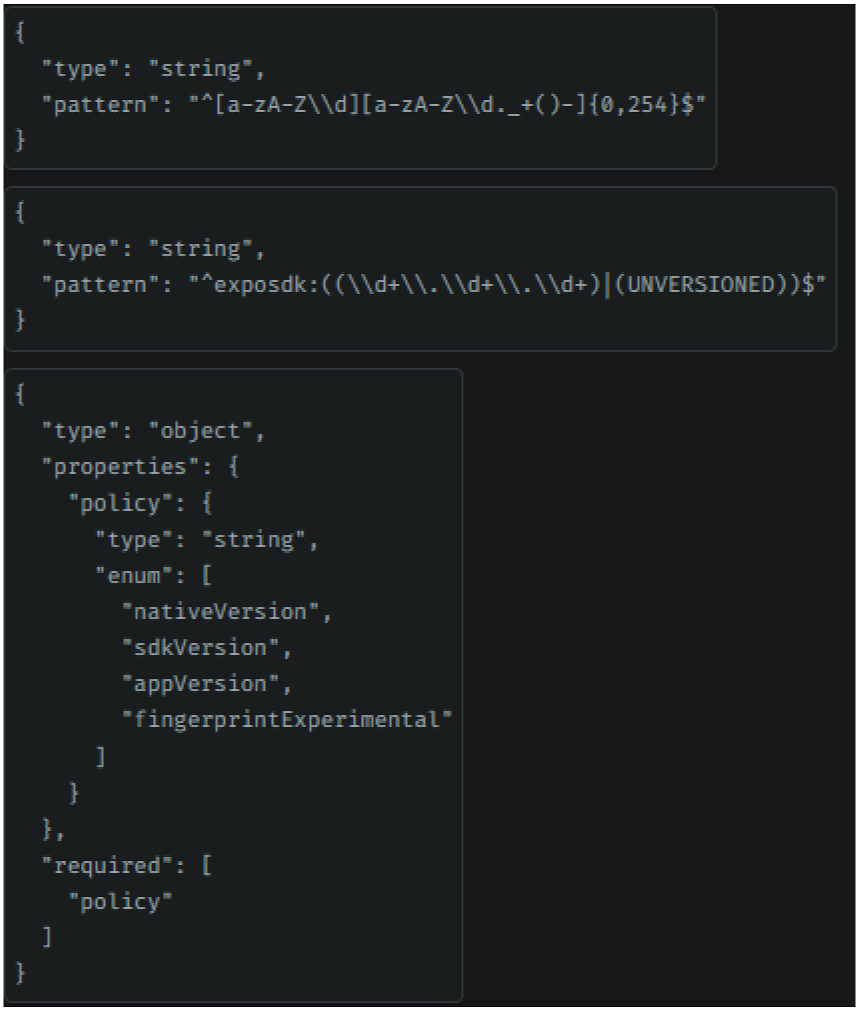
Icon: Local pah or remote URL to an image to use for your app's icon on Android. If specified, this overrides the top-level icon. It is recommended to use a 1024x1024 png file. The icon will appear on the home screen and within the expo app.

- Can be used by both Android and iOS

adaptiveIcon: Settings for an adaptive launcher Icon on android:

- foregroundImage: local path or remote URL to an image to use for the app icon on android.
- monochromeImage: local path or remote URL to image representing the Andoird 13+ monochromatic icon
- backgroundImage: local path or remote URL to a background image for your app's adaptive icon on android
- backgroundColor: Colo to use as the background for your apps adaptive icon on android Defaults to white #FFFFFF. Has no effect if foregroundImage isn't specified.

runtimeVersion: The runtime version is associated with the manifest. Set it to {“policy”: “nativeVersion”} to generate automatically.

- 

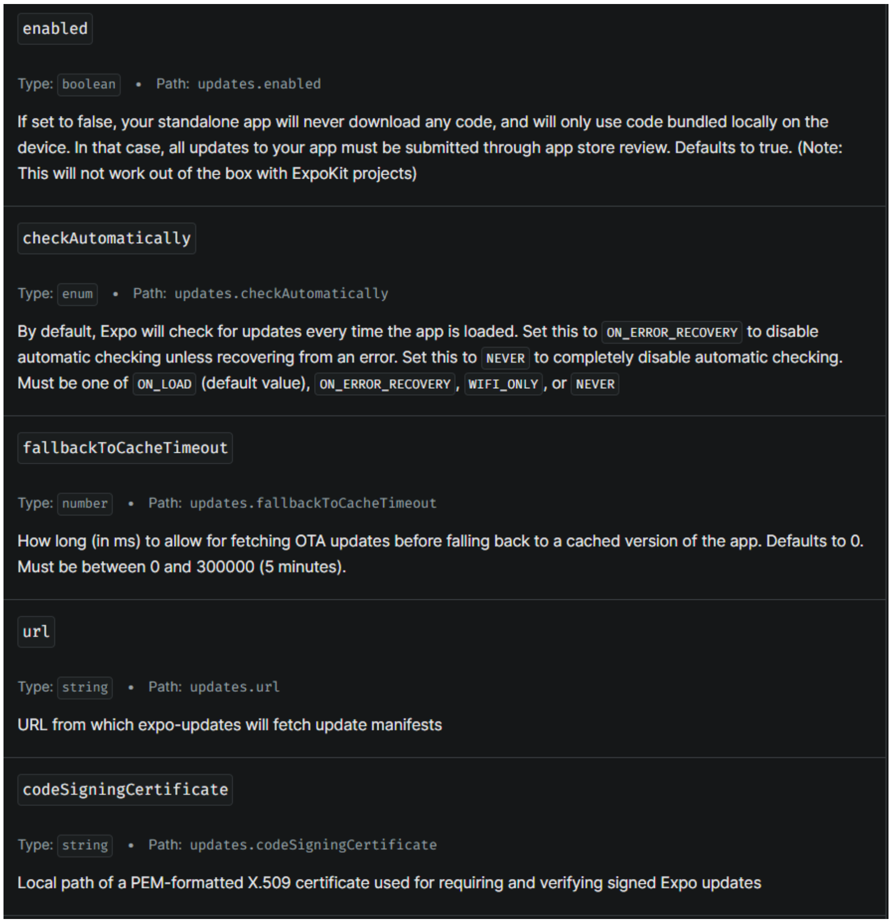
```
{
  "type": "string",
  "pattern": "^[a-zA-Z\\d][a-zA-Z\\d._+()-]{0,254}$"
}
```

```
{
  "type": "string",
  "pattern": "^exposdk:((\\d+\\.\\d+\\.\\d+)|(UNVERSIONED))$"
}
```

```
{
  "type": "object",
  "properties": {
    "policy": {
      "type": "string",
      "enum": [
        "nativeVersion",
        "sdkVersion",
        "appVersion",
        "fingerprintExperimental"
      ]
    }
  },
  "required": [
    "policy"
  ]
}
```

Managing runtime versions and update behavior is very important if you have an application that has a strict security setting that needs to be updated frequently like a banking app. The app probably needs to be updated frequently so that old versions cannot be exploited by hackers trying to accomplish malicious intent.

Updates: configuration for how and when the app should request OTA JavaScript updates



`enabled`

Type: `boolean` • Path: `updates.enabled`

If set to false, your standalone app will never download any code, and will only use code bundled locally on the device. In that case, all updates to your app must be submitted through app store review. Defaults to true. (Note: This will not work out of the box with ExpoKit projects)

`checkAutomatically`

Type: `enum` • Path: `updates.checkAutomatically`

By default, Expo will check for updates every time the app is loaded. Set this to `ON_ERROR_RECOVERY` to disable automatic checking unless recovering from an error. Set this to `NEVER` to completely disable automatic checking. Must be one of `ON_LOAD` (default value), `ON_ERROR_RECOVERY`, `WIFI_ONLY`, or `NEVER`

`fallbackToCacheTimeout`

Type: `number` • Path: `updates.fallbackToCacheTimeout`

How long (in ms) to allow for fetching OTA updates before falling back to a cached version of the app. Defaults to 0. Must be between 0 and 300000 (5 minutes).

`url`

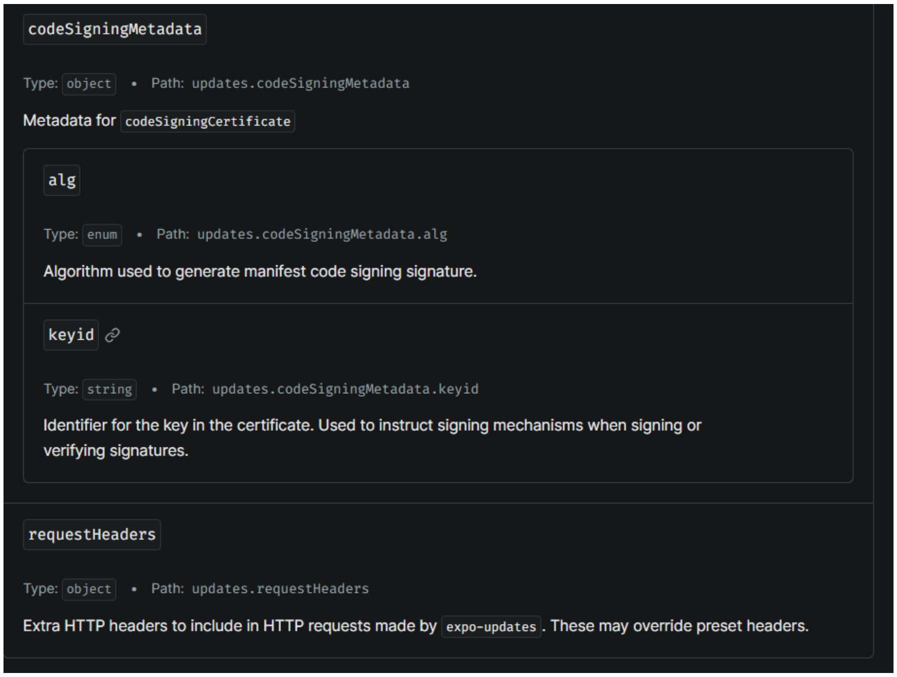
Type: `string` • Path: `updates.url`

URL from which expo-updates will fetch update manifests

`codeSigningCertificate`

Type: `string` • Path: `updates.codeSigningCertificate`

Local path of a PEM-formatted X.509 certificate used for requiring and verifying signed Expo updates



`codeSigningMetadata`

Type: `object` • Path: `updates.codeSigningMetadata`

Metadata for `codeSigningCertificate`

`alg`

Type: `enum` • Path: `updates.codeSigningMetadata.alg`

Algorithm used to generate manifest code signing signature.

`keyid`

Type: `string` • Path: `updates.codeSigningMetadata.keyid`

Identifier for the key in the certificate. Used to instruct signing mechanisms when signing or verifying signatures.

`requestHeaders`

Type: `object` • Path: `updates.requestHeaders`

Extra HTTP headers to include in HTTP requests made by `expo-updates`. These may override preset headers.