Documentation

Language: Swift

App License Delivery SDK / Renewing and revoking app licenses

Article

Renewing and revoking app licenses

Determine whether an app for which you issue a license launches.

Overview

An app license expires when its expiry date passes. The expiry date is the `duration` after the `issuedTime`. Your license server determines the value of those properties at the time of issuing a license (see Licensing alternative distribution apps).

When an app’s license expires, the device’s operating system doesn’t allow the app to launch. You determine the criteria for renewal, expiration, and revocation of licenses for apps that you distribute. Depending on the circumstances, you can renew a license before it expires, let it expire, or revoke the license deliberately in advance.

Handle a license renewal request from the system

For app license renewal or revocation, the system checks with your server by sending a request. The communication is similar to a dynamic license request for licenses for new app versions, except that the system calls a different endpoint on your server — the `licenseRenewalURL` in your `marketplace-kit` configuration file:

```
{
    "license": {
        "dynamicLicenseURL" : "<licensing endpoint>",
        "licenseRenewalURL" : "<renewal endpoint>",
        ...
    } ...
}
```

Once a day, the system checks for app licenses set to expire within the next 24 hours and includes each in a license renewal request. Depending on how your license server responds, the system either allows the app to continue to run or prevents it from running after expiration.

The renewal request is a POST with the following payload:

```
{
    "renewalRequest": "BASE_64_ENCODED_LICENSE_REQUEST",
    "licenseKey": : "BASE_64_ENCODED_LICENSE_KEY",
}
```

To handle the request, decode the `"renewalRequest"` value using the `"licenseKey"` as shown in Licensing alternative distribution apps.

Each POST requests renewal for one or more license IDs. On your license server, retrieve the list of license IDs in the request using the `ALDSession` property's `requestedLicenseIDList`.

The format you supply in response is:

```
{
    "license" : "BASE_64_ENCODED_LICENSE",
    "ineligibleLicenses": [
        "LICENSE_ID1",
        "LICENSE_ID2"
    ]
}
```

Renewal response key	Description
`"license"`	A signed, `base64`-encoded license, as done for dynamic license responses. See Licensing alternative distribution apps.
`"ineligible Licenses"`	An array of `string` license IDs from the request that you don't intend to renew.

Renew an app license from your app

To renew a license, call `requestLicenseRenewal(appleItemIDs:)` from within your marketplace app, or other app you distribute from your website. The system then calls your license renewal endpoint and provides the license ID for the given `AppleItemID`. Your license server responds by providing a new license with the `issuedTime` set to today, and a `duration` set at your discretion.

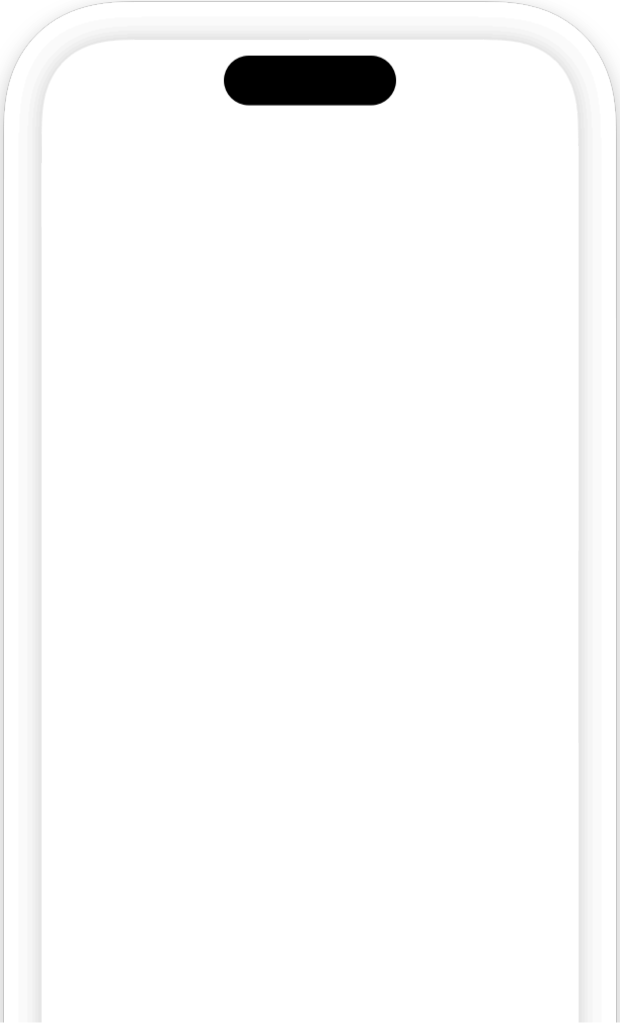
If any logic running on your web server determines that an app license needs adjusting, record the state and periodically check it from your marketplace app. For example, your marketplace app can call a custom endpoint you implement just for that purpose. When the state indicates the license needs adjusting, your app calls `requestLicenseRenewal(appleItemIDs:)`, which intructs the system to send the renewal request to your license server.

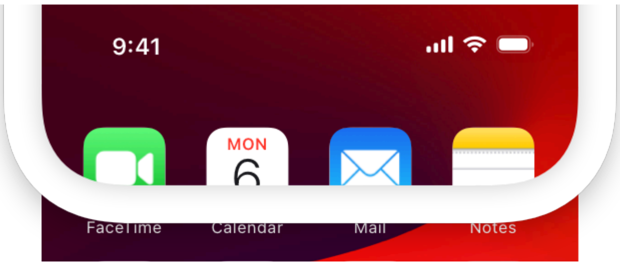
Revoke an app license from your app

To revoke an app license, call `requestLicenseRenewal(appleItemIDs:)` passing the app's `AppleItemID` from your app. The system then sends a request to your license renewal endpoint with that app's license ID. In response, your endpoint includes the license ID in the `ineligibleLicenses` array, which instructs the system to prevent the app from running.

Resolve an expired license from the Home Screen

When a license expires for your marketplace app or an app that your marketplace distributes and a person taps the app icon on the Home Screen, the system presents the alert:





A Resolve button follows that, after a person taps it, presents a web view that loads a page you specify in your server's `marketplace-kit` configuration file as the `licenseResolutionURL`:

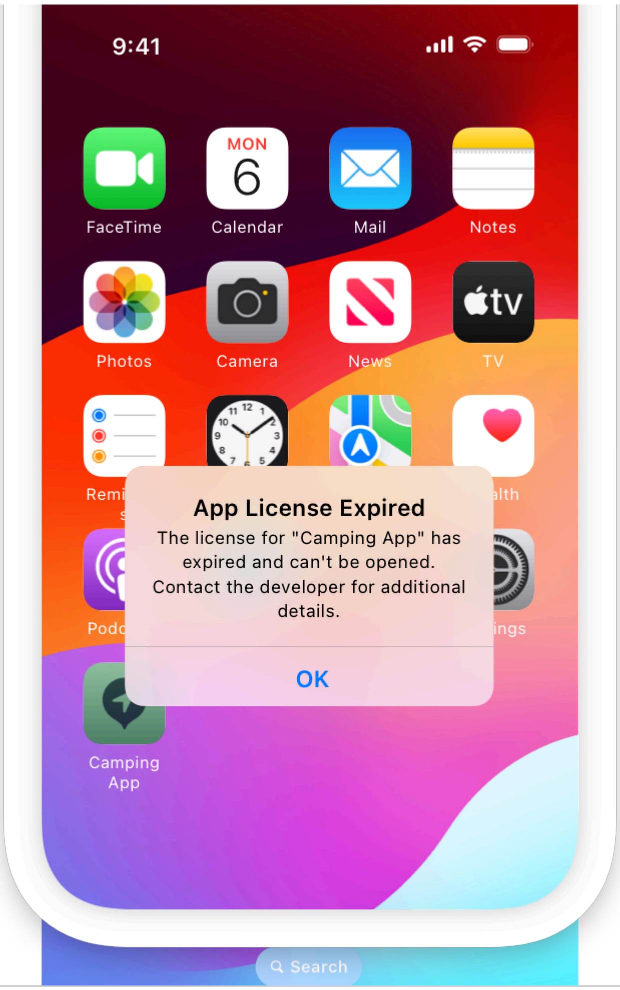
```
"license": {
    "licenseRenewalURL" : "<renewal endpoint>",
    "licenseResolutionURL" : "<license resolution webpage>",
    ...
} ...
```

When the system issues a GET request for the page, it includes the following parameters:

License resolution parameter	Description
`account`	The account you associate with the person in the intitial `MarketplaceKitURIScheme` to install the app.
`appleItemID`	An identifier for a requested app. For more information, see `AppleItemID`.
`appleVersionID`	A version number that corresponds to the alternative distribution package. For more information, see `AppleVersionID`.

Format the web page in your response to the person's unique situation so they can resolve the issue in line. If they can't resolve the issue immediately, your webpage provides the necessary context on the expiration or status of the problem.

If your configuration file lacks `licenseResolutionURL`, then the system presents the alert:



App License Expired

The license for "Camping App" has expired and can't be opened. Contact the developer for additional details.

See Also

App licensing

Licensing alternative distribution apps

Build a license server that supports the installation of your apps and the apps available in your marketplace.

`struct ALDAppKey`

A structure that identifies an app and a key that’s required to decrypt the app’s license request.

`struct ALDLicenseAttribute`

A structure that defines the requested license type for the session.

`class ALDProvider`

An object that creates a session with the alternative app marketplace’s signing assets.

`class ALDSession`

A structure that contains the details of a license request and methods to generate license responses.