Class

ALDSession

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

```
class ALDSession
```

Mentioned in

📄 Licensing alternative distribution apps

📄 Renewing and revoking app licenses

Topics

Initializers

init(request: [UInt8], PASK: [UInt8], encryptionCert: [UInt8], signing Cert: [UInt8], signingKey: [UInt8]?) throws

    Initialize a ALDSession with a license request. Call generateLicense() to create a license.

init(signingCert: [UInt8], signingKey: [UInt8]?, PASK: [UInt8]) throws

    Initialize a static ALDSession without a request. This should only be used when an offline generation of a static license is desired. A static license is a minimal license that is only used to install apps on the device and is not meant to enforce marketplace defined rights. Only generateStaticLicense() should be invoked to generate licenses in this case.

Instance Properties

`var requestAction: ALDLicenseAction`

    action specified in this request

`var requestDeviceID: [UInt8]`

    the device ID of the requested device

`var requestID: UInt64`

    requestID of the session

`var requestTime: UInt64`

    the client time when the request was made

`var requestVersion: UInt32`

    request version number

`var requestedAppleItemIDList: [UInt64]`

    An array of identifiers for apps that iOS requests a license request for in the session.

`var requestedLicenseIDList: [UInt64]`

    the list of license ID the client has requested a renwal for. used only when action is "renew"

`var sessionType: ALDSessionType`

    the current session type

Instance Methods

`func finalizeLicenseResponse(licenseResponse: [UInt8], signature: [UInt8]?) throws -> [UInt8]`

    Returns a signed license in a byte array to send in response to a license request from iOS.

`func generateLicense(attr: ALDLicenseAttribute) throws`

    Generates a license based on the provided ALDLicenseAttribute and add it to the session. Multiple licenses can be generated in this session by callling this function multiple times, they get added to the session response.

`func generateLicenseResponse() throws -> [UInt8]`

    Generates a license response. This method produces a license response, in a bytes array. The response is not yet signed.

```
func generateStaticLicense(licenseID: UInt64, appKey: ALDAppKey) throws
```
Generates a static license based on the provided ALDLicenseAttribute. This method produces a static license, in a bytes array. A static license is a minimal license that is only used to install apps on the device and is not meant to enforce marketplace defined rights.

Enumerations

```
enum ALDLicenseAction
```
The action requested in the license request or provided in the response to one.

```
enum ALDSessionType
```
The type of the license session created. A normalSession is used for a session created with a license request. A staticSession is used for session created without a license request

---

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.

📄 Renewing and revoking app licenses

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

```
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.