

ToTokGameSDK Installation

(iOS)

Name	Date	version	remark
Yuanjun	2020.3.18	1.0.0	created

1. Register and create app

1.1 Register a new app in ToTokGame platform

Get application information such as appld, appsecret, gameid.

1.2 If you use Facebook related functions, you need to register with the Facebook development platform to create an application.

<https://developers.facebook.com/apps/>

2. Installation and Runtime Environment

- IDE : Xcode 10 or higher
- iOS : iOS 9.0 or higher

3. Install SDK

SDK Directory Structure :

- ToTokGameSDK
 - Core/ToTokGameSDK.frame
 - SocialLibraries/Facebook/

Information of third-party SDK :

Facebook: use cocoapods

```
'FBSDKCoreKit' '~> 5.13.1'
```

```
'FBSDKLoginKit' '~> 5.13.1'
```

```
'FBSDKShareKit' '~> 5.13.1'
```

3.1 Use Pod Install SDK (Recommend)

It is recommended to use the cocoapods install SDK. Cocoapods provides a simple dependency management system to avoid errors caused by manual import .

first you need to confirm that Cocoapods has been installed, if you have not installed Cocoapods, refer to the official website user guide:

<https://cocoapods.org/>.

1. In the root directory of the Xcode project, create a new Podfile.

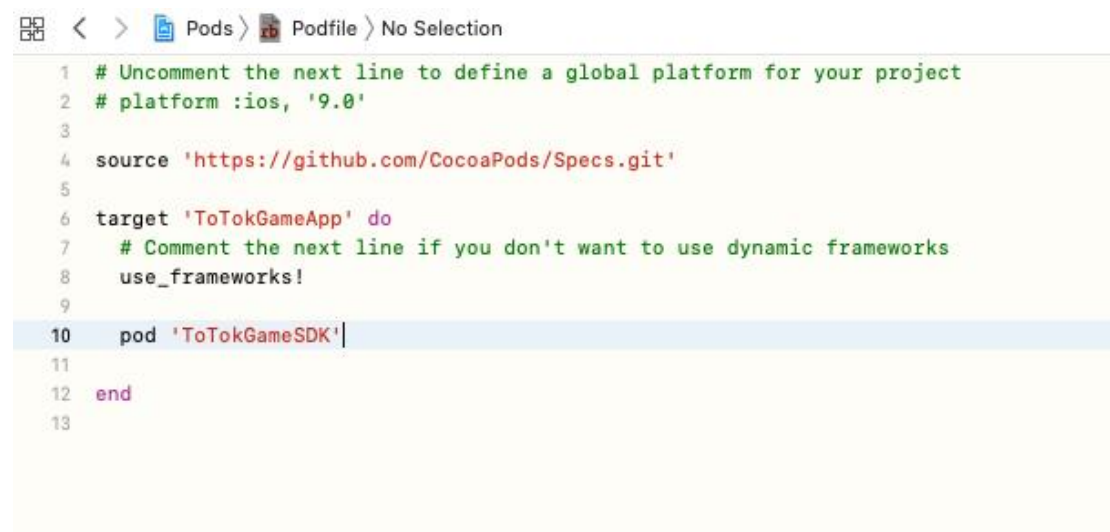
```
$ pod init
```

Add the following to this file :

```
source 'https://github.com/CocoaPods/Specs.git'
```

```
pod 'ToTokGameSDK'
```

Setup as shown below



```
1 # Uncomment the next line to define a global platform for your project
2 # platform :ios, '9.0'
3
4 source 'https://github.com/CocoaPods/Specs.git'
5
6 target 'ToTokGameApp' do
7 # Comment the next line if you don't want to use dynamic frameworks
8 use_frameworks!
9
10 pod 'ToTokGameSDK'
11
12 end
13
```

2. In the Podfile directory, execute the following instructions :

```
$ pod install
```

3. After executing pod install, open the project directory and find the .xcworkspace file to run.

4. Future SDK upgrades can use the update command.

```
$ pod update
```

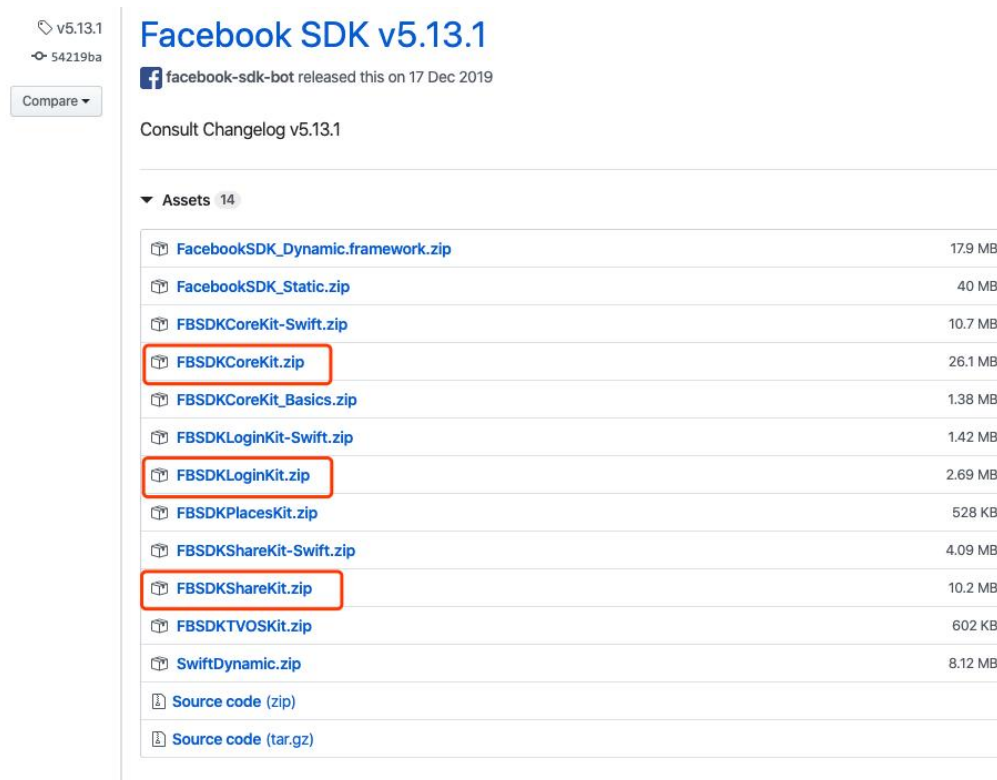
3.2 Install SDK Manually

1. Download ToTokGameSDK related releases on github

<https://github.com/ToTokGames/ToTokGameSDK-iOS/releases>

2. Download related FacebookSDK version 5.3.1 on github

<https://github.com/facebook/facebook-ios-sdk/releases/tag/v5.13.1>



Facebook SDK v5.13.1

facebook-sdk-bot released this on 17 Dec 2019

Consult Changelog v5.13.1

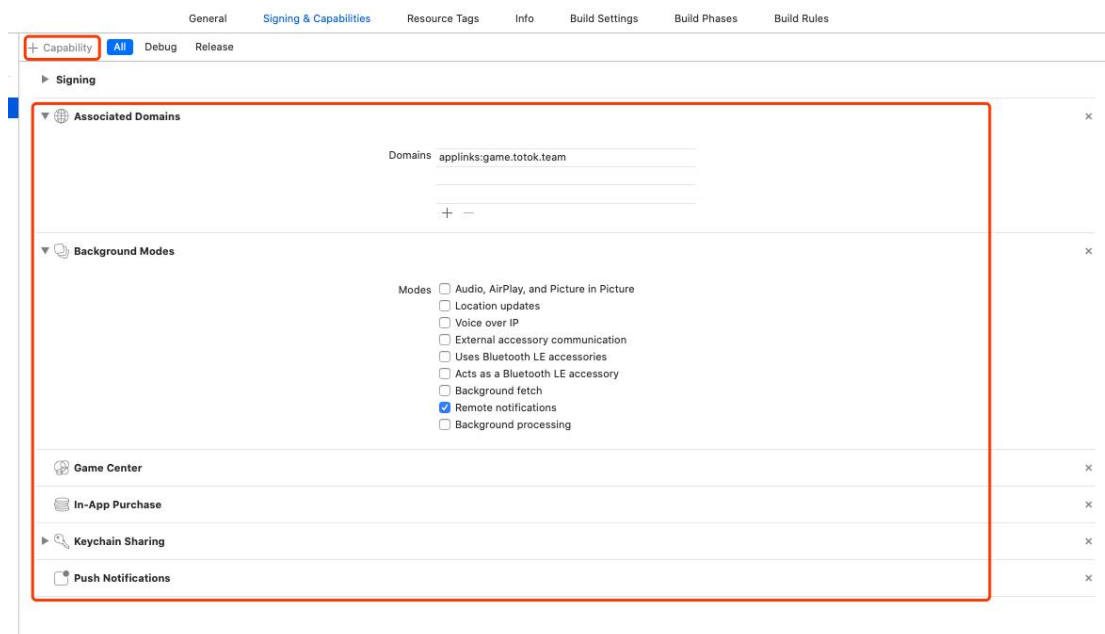
Assets 14

FacebookSDK_Dynamic.framework.zip	17.9 MB
FacebookSDK_Static.zip	40 MB
FBSDKCoreKit-Swift.zip	10.7 MB
FBSDKCoreKit.zip	26.1 MB
FBSDKCoreKit_Basics.zip	1.38 MB
FBSDKLoginKit-Swift.zip	1.42 MB
FBSDKLoginKit.zip	2.69 MB
FBSDKPlacesKit.zip	528 KB
FBSDKShareKit-Swift.zip	4.09 MB
FBSDKShareKit.zip	10.2 MB
FBSDKTVOKit.zip	602 KB
SwiftDynamic.zip	8.12 MB
Source code (zip)	
Source code (tar.gz)	

4. Project Configuration

4.1 Build Configuration

**4.1.1 Apple service configuration, push, keychain, in-app purchase, and Universal Link are the services provided by Apple and need to be configured here .
as shown below**



Domains:

test

applinks:game.totok.team

production

applinks:game.totok.ai

**4.1.2 Set bitcode to NO (not supported by Facebook)
as shown in the below**



4.1.3 Set plist.info

Relevant function configuration: Facebook related (the information marked in red color needs to be obtained from the official Facebook application)

```
<key>NSPhotoLibraryUsageDescription</key>
<string>Photo albums are required to share pictures, please allow access</string>
<key>CFBundleURLTypes</key>
<array>
<dict>
<key>CFBundleTypeRole</key>
<string>Editor</string>
<key>CFBundleURLName</key>
<string>totok</string>
<key>CFBundleURLSchemes</key>
<array>
<string>appid</string>
</array>
</dict>
</dict>
<key>CFBundleURLSchemes</key>
<array>
<string>fbxxxxxxxx</string>
</array>
</dict>
</array>
<key>FacebookAppID</key>
<string>xxxxxxxx</string>
<key>FacebookDisplayName</key>
<string>AppName</string>
<key>LSApplicationQueriesSchemes</key>
```

```
<array>
  <string>whatsapp</string>
  <string>totok</string>
  <string>totoksdk</string>
  <string>fbapi</string>
  <string>fbapi20130214</string>
  <string>fbapi20130410</string>
  <string>fbapi20130702</string>
  <string>fbapi20131010</string>
  <string>fbapi20131219</string>
  <string>fbapi20140410</string>
  <string>fbapi20140116</string>
  <string>fbapi20150313</string>
  <string>fbapi20150629</string>
  <string>fbapi20160328</string>
  <string>fbauth</string>
  <string>fbauth2</string>
  <string>fbshareextension</string>
  <string>fb-messenger-api20140430</string>
  <string>fb-messenger-platform-20150128</string>
  <string>fb-messenger-platform-20150218</string>
  <string>fb-messenger-platform-20150305</string>
</array>

<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <true/>
  <key>NSExceptionDomains</key>
  <dict>
    <key>akamaihd.net</key>
    <dict>
      <key>NSExceptionRequiresForwardSecrecy</key>
      <false/>
      <key>NSIncludesSubdomains</key>
      <true/>
    </dict>
  </dict>
</dict>
```

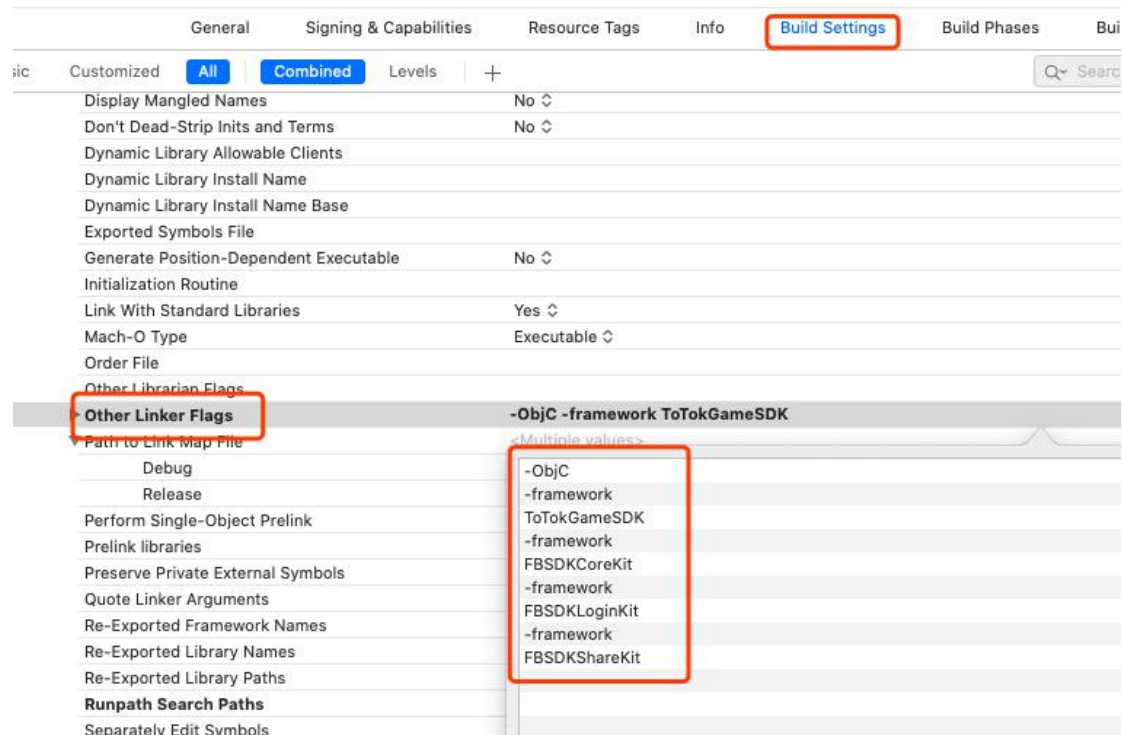
```
<key>facebook.com</key>
<dict>
<key>NSExceptionRequiresForwardSecrecy</key>
<false/>
<key>NSIncludesSubdomains</key>
<true/>
</dict>
<key>fbcdn.net</key>
<dict>
<key>NSExceptionRequiresForwardSecrecy</key>
<false/>
<key>NSIncludesSubdomains</key>
<true/>
</dict>
</dict>
</dict>
```

If it is a pod installation, the configuration is complete here.

If you manually drag in the package, continue the following process.

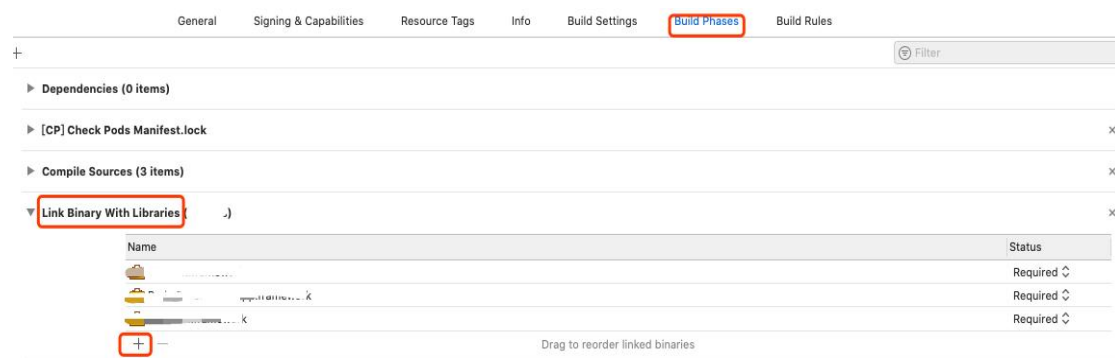
4.1.4 Build configuration

Add Other Linker Flags (Pay attention to case)



4.1.5 The system libraries used by related SDKs need to be configured as follows.

'sqlite3', 'UIKit', 'StoreKit', 'GameKit', 'WebKit', 'UserNotifications', 'Photos', 'ImageIO', 'Foundation', 'CoreFoundation', 'QuartzCore', 'CoreGraphics', 'AVFoundation', 'CoreTelephony', 'Security', 'CoreMotion', 'ExternalAccessory', 'SystemConfiguration', 'Accounts', 'Social', 'AudioToolbox', 'Accelerate'



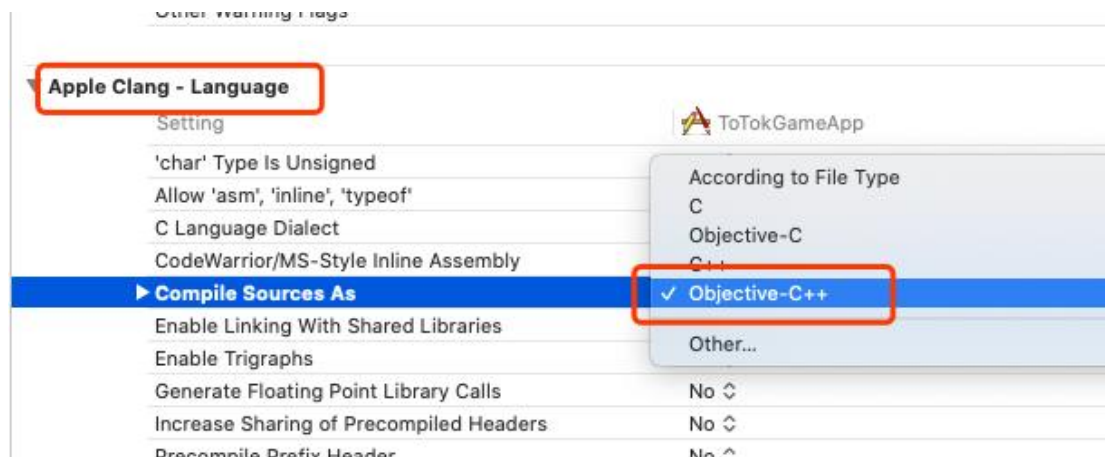
Add the system library required above here.

4.1.6 Compiling FBSDK requires setting compiling C++

Also set in build phases

Set Apple Clang - Language

as shown below

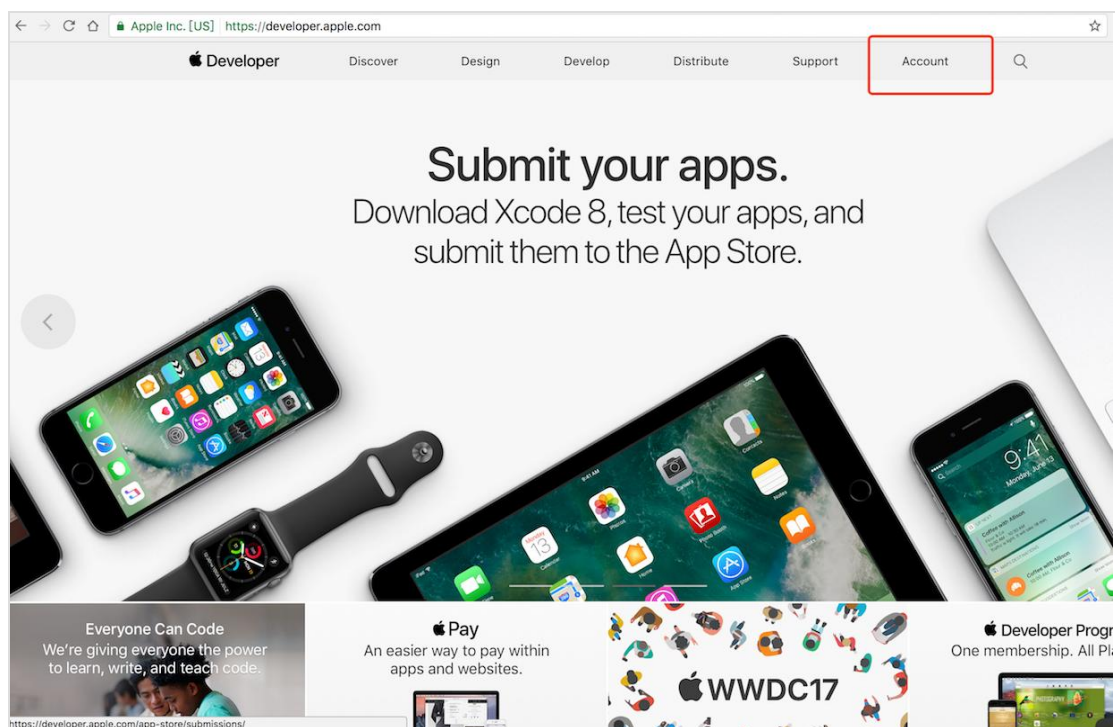


Compile configuration is over here.

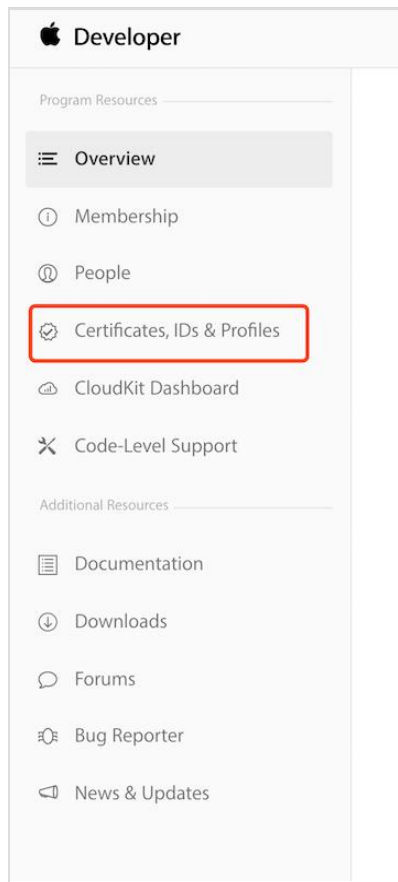
5. Push configuration

5.1 Certificate configuration

- Log in to the Apple Developer Website and enter your developer account.



- Go to the “Certificates, IDs & Profiles” page from the left entry of the developer account page.



- Create an App ID, fill in the name of the App ID and the Bundle ID (skip this step if the ID already exists).



Note: You need to specify the specific bundle ID here. Do not use wildcard characters.

Certificates, Identifiers & Profiles

< All Identifiers

Register an App ID

Back Continue

Platform
☒ iOS, tvOS, watchOS ☐ macOS

Description

You cannot use special characters such as @, &, '., '., *

App ID Prefix
 com.push.demo (Team ID)

Bundle ID
☒ Explicit ☐ Wildcard

We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).

Capabilities

- Enable Push Notification for the app. If the App ID has already been created, you can also enable the Push Notification function through settings.

< All Identifiers

Register an App ID

Back Continue

Capabilities

ENABLED	NAME
<input type="checkbox"/>	Access WiFi Information
<input type="checkbox"/>	App Groups
<input type="checkbox"/>	Apple Pay Payment Processing
<input type="checkbox"/>	Associated Domains
<input type="checkbox"/>	AutoFill Credential Provider
<input type="checkbox"/>	Network Extensions
<input type="checkbox"/>	NFC Tag Reading
<input type="checkbox"/>	Personal VPN
<input checked="" type="checkbox"/>	Push Notifications
<input type="checkbox"/>	Sign In with Apple
<input type="checkbox"/>	SiriKit
<input type="checkbox"/>	System Extension (Supported only on: macOS)

Configure

- After filling in the above attributes, click “Continue” to confirm the correctness of the AppId attribute, and click “Register” to register the AppId successfully.
- If you haven't created a Push certificate before or want to re-create a new one, please create a new one below the certificate list.

Certificates, Identifiers & Profiles

Certificates
Identifiers
Devices
Profiles
Keys
More

Certificates

NAME	TYPE	PLATFORM	CREATED BY	EXPIRATION
Development	iOS Development	iOS	Apple Developer	2020/08/21
Distribution	iOS Distribution	iOS	Apple Developer	2020/08/21
Development	Development	All	Apple Developer	2020/09/22
Distribution	Distribution	All	Apple Developer	2020/09/25
Apple Push	Apple Push Services	iOS	Apple Developer	2020/10/24

- When creating a new certificate, you need to choose the type of APNs certificate. As shown in the figure APNs certificate has two types of development (development) and production (production).

Note: The development certificate is used for development and debugging; the production certificate can be used for both development and debugging and product release. Here we choose the production certificate as an example.

Create a New Certificate

[Continue](#)

Services

- ☐
iOS Apple Push Notification service SSL (Sandbox)
use for develop

Establish connectivity between your notification server and the Apple Push Notification service sandbox environment to deliver remote notifications to your app. A separate certificate is required for each app you develop.
- ☒
Apple Push Notification service SSL (Sandbox & Production)
use for develop and production

Establish connectivity between your notification server, the Apple Push Notification service sandbox, and production environments to deliver remote notifications to your app. When utilizing HTTP/2, the same certificate can be used to deliver app notifications, update ClockKit complication data, and alert background VoIP apps of incoming activity. A separate certificate is required for each app you distribute.
- ☐
macOS Apple Push Notification service SSL (Production)

Establish connectivity between your notification server and the Apple Push Notification service production environment. A separate certificate is required for each app you distribute.
- ☐
Pass Type ID Certificate

Sign and send updates to passes in Wallet.
- ☐
Website Push ID Certificate

Sign and send updates for Websites.
- ☐
WatchKit Services Certificate

Establish connectivity between your notification server, the Apple Push Notification service sandbox, and production environment to update ClockKit complication data. When utilizing HTTP/2, the same certificate can be used to deliver app notifications, update ClockKit complication data, and alert background VoIP apps of incoming activity. A separate certificate is required for each app you distribute.
- ☐
VoIP Services Certificate

Establish connectivity between your notification server, the Apple Push Notification service sandbox, and production environment to alert background VoIP apps of incoming activity. A separate certificate is required for each app you distribute.
- ☐
Apple Pay Payment Processing Certificate

Decrypt app transaction data sent by Apple to a merchant/developer.
- ☐
Apple Pay Merchant Identity Certificate

A client TLS certificate that is used to authenticate you to Apple Pay Payment Processing Servers. You need to accept the agreement 'Apple Pay Platform Web Merchant Terms and Conditions'. [Review Agreement](#)

- Click "Continue", then select the certificate and the AppID to be bound.

Certificates, Identifiers & Profiles

[< All Certificates](#)

Create a New Certificate

[Back](#) [Continue](#)

Select an App ID for your Apple Push Notification service SSL Certificate (Sandbox & Production)

All App IDs that you want to enable for remote notifications require their own Apple Push Notification service SSL certificate. The App ID-specific SSL certificate allows your server to connect to the Apple Push Notification service. Note that only explicit App IDs with a specific Bundle Identifier can be used to create an Apple Push Notification service SSL certificate.

App ID:

- Clicking "Continue" again will let you upload the CSR file. (CSR file will be created in the next step)

Certificates, Identifiers & Profiles

[< All Certificates](#)

Create a New Certificate

[Back](#) [Continue](#)

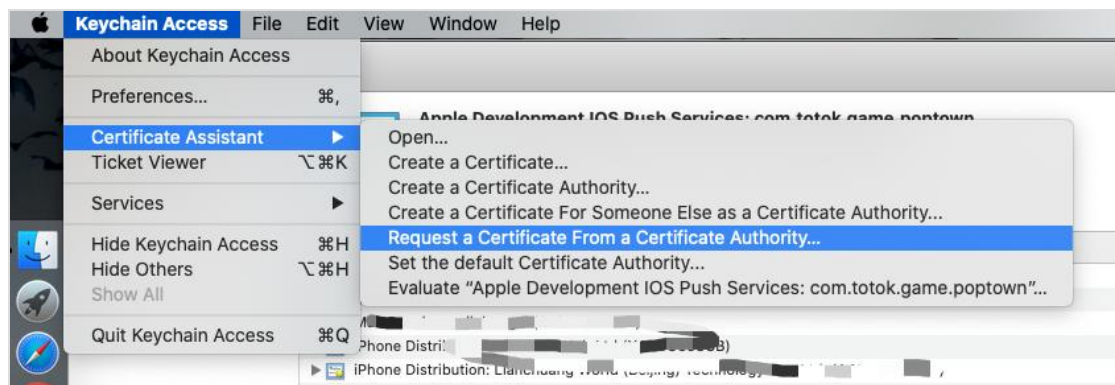
Upload a Certificate Signing Request

To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Mac.
[Learn more >](#)

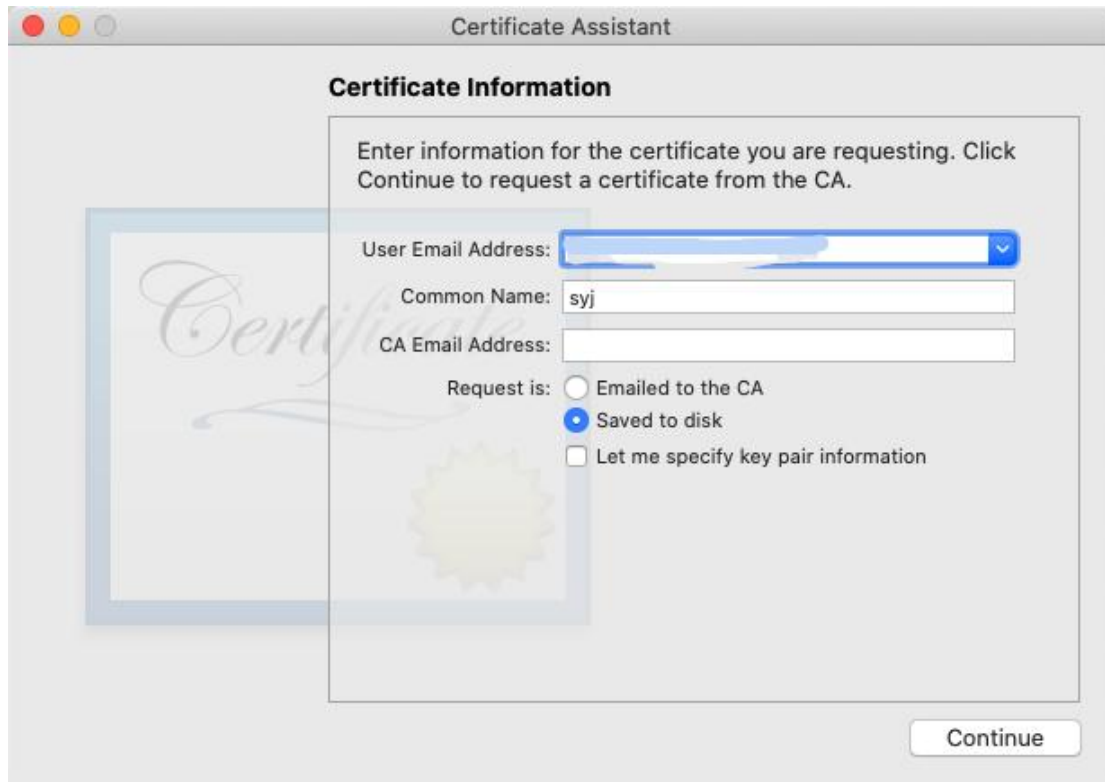
[Choose File](#) CertificateSigningRequest.certSigningRequest

- Open KeychainAccess that comes with the mac system to create a Certificate Signing Request.

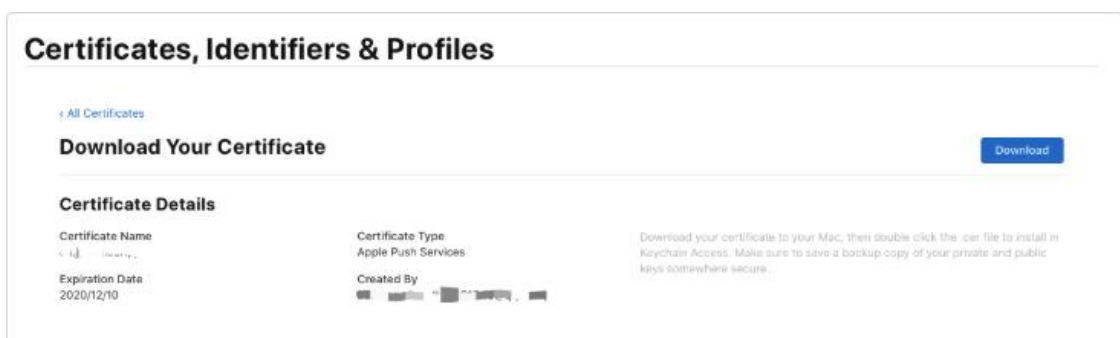
Operate as shown below:



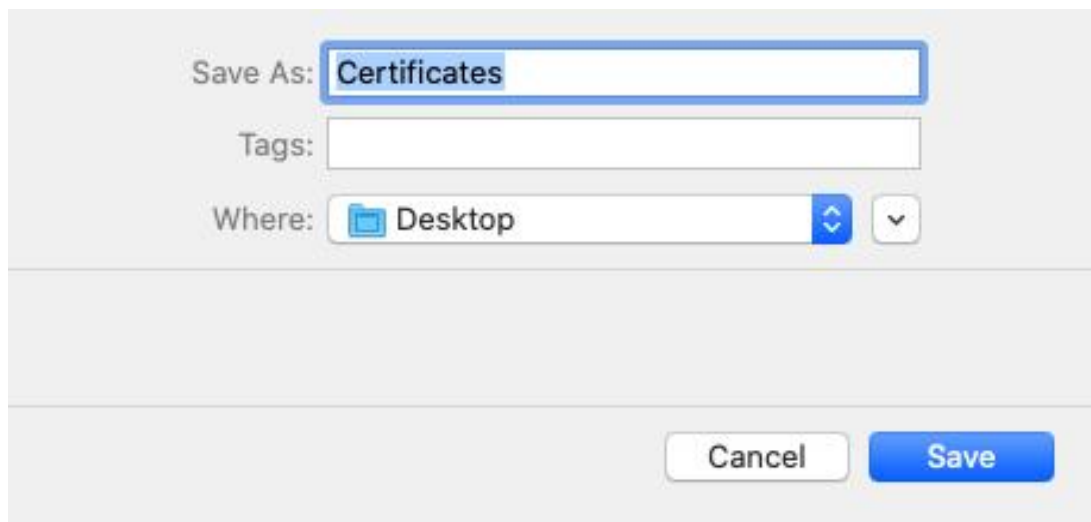
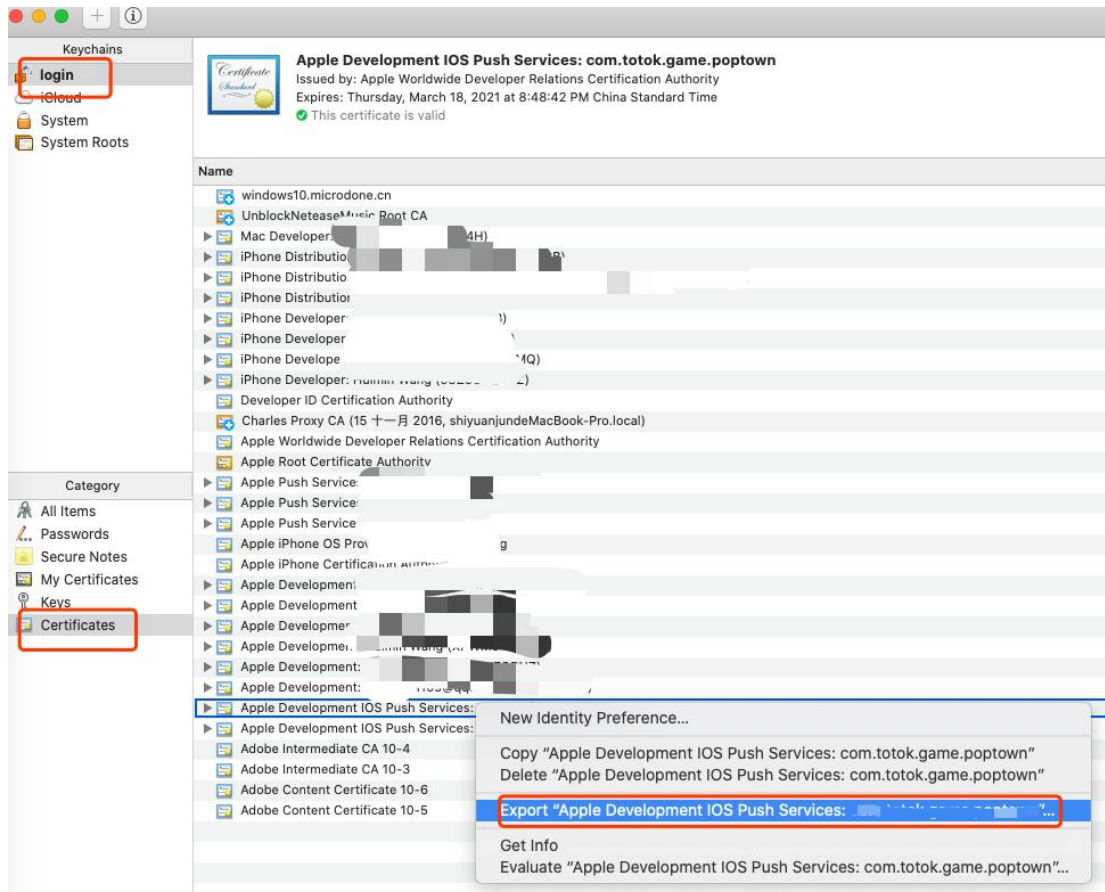
- Fill in "User Email" and "Common Name" and select "Save to Disk". The certificate file suffix is .certSigningRequest.



- Go back to the CSR upload page in your browser and upload the file with the suffix .certSigningRequest just generated.
- After the certificate is generated successfully, click the "Download" button to download the certificate. It is a file with the extension .cer.



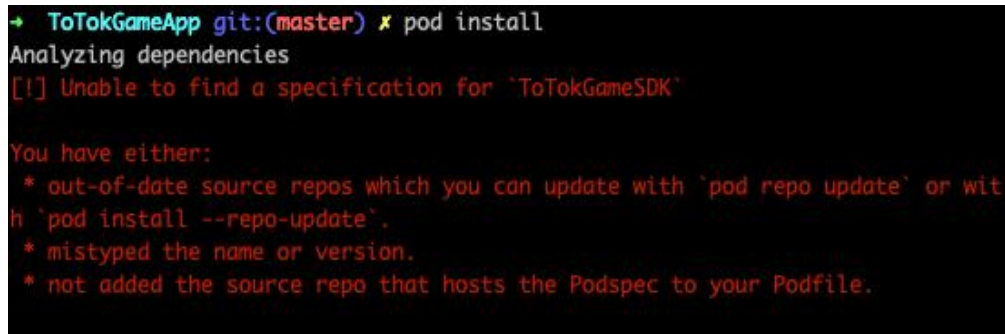
- After double-clicking the certificate, it will open in "KeychainAccess", select "Login" in the "Keychain" list on the left, and "My certificate" in the "Type" list, find the certificate you just downloaded, and export it as a .p12 file. As shown below:



● Common Problem

1. pod related

If you have a pod installed, running pod install directly may cause the following problems

A terminal window with a dark background and light-colored text. The text shows a command being run in a directory named 'ToTokGameApp' on the 'git:(master)' branch. The command is 'pod install'. The output shows 'Analyzing dependencies' followed by an error: '[!] Unable to find a specification for 'ToTokGameSDK''. Below this, a message says 'You have either:' followed by three bullet points: '* out-of-date source repos which you can update with 'pod repo update' or with 'pod install --repo-update'.', '* mistyped the name or version.', and '* not added the source repo that hosts the Podspec to your Podfile.'

```
→ ToTokGameApp git:(master) x pod install
Analyzing dependencies
[!] Unable to find a specification for 'ToTokGameSDK'

You have either:
* out-of-date source repos which you can update with 'pod repo update' or with 'pod install --repo-update'.
* mistyped the name or version.
* not added the source repo that hosts the Podspec to your Podfile.
```

Need to update the pod repo

```
$ pod install --repo-update
```

2. Project related

If you use xcode11 to create a new project, the iOS13 + real machine or simulator has a black screen

Reason: For the multi-window function of the iPad, the life cycle has changed. At present, the SDK does not support the multi-window function, so developer who use the SDK can change the project to the previous structure.

Specific steps are as follows:

1. First delete the Application Scene Manifest column from info.plist

Key	Type	Value
▼ Information Property List	Dictionary	(15 items)
Localization native development re...	String	\$(DEVELOPMENT_LANGUAGE)
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	\$(PRODUCT_BUNDLE_PACKAGE_TYPE)
Bundle versions string, short	String	1.0
Bundle version	String	1
Application requires iPhone enviro...	Boolean	YES
▶ Application Scene Manifest	Dictionary	(2 items)
Launch screen interface file base...	String	LaunchScreen
Main storyboard file base name	String	Main
▶ Required device capabilities	Array	(1 item)
▶ Supported interface orientations	Array	(3 items)
▶ Supported interface orientations (i...	Array	(4 items)

2. In the project directory, delete the Scenedelegate.h and Scenedelegate.m files

3. Then enter AppDelegate.m to comment or delete the content shown below

```
#pragma mark - UISceneSession lifecycle
```

```
- (UISceneConfiguration *)application:(UIApplication *)application
    configurationForConnectingSceneSession:(UISceneSession
    *)connectingSceneSession options:(UISceneConnectionOptions
    *)options {
    // Called when a new scene session is being created.
    // Use this method to select a configuration to create the new
    // scene with.
    return [[UISceneConfiguration alloc] initWithName:@"Default
    Configuration" sessionRole:connectingSceneSession.role];
}

- (void)application:(UIApplication *)application
    didDiscardSceneSessions:(NSSet<UISceneSession *> *)sceneSessions {
    // Called when the user discards a scene session.
    // If any sessions were discarded while the application was not
    // running, this will be called shortly after
    // application:didFinishLaunchingWithOptions.
    // Use this method to release any resources that were specific to
    // the discarded scenes, as they will not return.
}
```

4. The last step is to add the window property in AppDelegate.h

```
@property (strong, nonatomic) UIWindow * window;
```