

iOS Softphone Build and Publish

MizuTech VoIP App Builder

Contents

<i>Introduction</i>	1
<i>Softphone Customization</i>	2
<i>Required Resources</i>	2
<i>The Build and Publish Process</i>	2
<i>1. Apple Developer Account</i>	3
<i>2. Developer Profile</i>	3
A.) CSR	3
B.) Certificate	6
<i>3. App Profile</i>	12
A.) App ID	13
B.) Profile	14
<i>4. PushKit</i>	15
<i>5. VoIP App Builder</i>	18
<i>6. Publish</i>	19
<i>7. Ad-Hoc</i>	20

Introduction

This document will guide you through the steps necessary for building and publishing your custom [iOS softphone](#) app to AppStore.

Starting from February 27, 2019 Apple introduced [two-factor authentication](#), so only you will be able to upload your iOS softphone to AppStore. Until two-factor authentication was introduced, we were able to login with your Apple Developer account and create/generate all the necessary Certificates and Provisioning Profiles for publishing iOS softphone. This is no longer possible, so you will have to do these required steps and upload the necessary resources via the web interface.

Disclaimer:

Mizutech provides the iOS softphone “as-is”, without extra support included and with no guarantee on the App Store submission or your business success. We recommend testing it before to purchase. We offer full refund for the iOS softphone cost in case if you are not satisfied with our software.

Softphone Customization

The [VoIP app builder website](#) is capable to generate a turnkey customized iOS softphone, which you can publish on the Apple App Store under your company or personal account.

You can find a description about our iOS softphone [here](#)

The following customizations can be applied for the softphone:

- Preconfigured server settings, so the users will have to type only their username and password to begin the usage
- Branding: brand name/app name, company name, icon, logo
- All common codec for SIP (including G.729 and wideband OPUS)
- Tunneling/encryption (optional, only if you use the Mizutech tunneling solution)
- Balance, rating display and other API integrations (if your server has HTTP API for these)
- Callback, phone to phone, SMS and recharge (if your server has a HTTP API for these)
- Links to various webpages such as to user account page or company homepage
- Other optional customizations enabled by the softphone builder website

Required Resources

The following iOS specific resources will be required by the [VoIP app builder service](#):

1. Basic mandatory settings such as app name, company name and SIP server address (SIP listener domain name or IP:port)
2. Icon (1024 x 1024 pixels PNG) and optionally a logo image (250 pixels width rectangle PNG)
3. Your iOS Distribution certificate
4. App ID and developer Team ID
5. Apple/iOS Developer Distribution certificate and app provisioning profile
6. Other optional settings and extra customizations

A Mac machine or any macOS device is required to get your developer certificate and publish the softphone on the Apple AppStore.

If you don't have an Apple Mac device, then you can also use a VMWare/VirtualBox/other [macOS virtual machine](#) installed on Windows or Linux.

Below is a step by step guide on how to create/generate the necessary certificates, App ID and Provisioning profiles.

The Build and Publish Process

1. [Apple Developer account](#): Create an Apple Developer account, if you don't already have one, and enroll in the iOS Developer program
2. [Create developer account](#) (if you don't have it already)

3. [Create developer and app profile](#): create Distribution certificate
4. [App profile](#): create App ID and Distribution Provisioning profile
5. [VoIP App builder](#): Set/upload the details (obtained by following the above steps) on the VoIP app builder website which will generate your custom softphone
6. [Publish](#): Upload the received iOS softphone to App Store Connect and publish it
7. [Ad-Hoc](#): Instructions for Ad-Hoc distribution for test only (optional; only if you requested ad-hoc test distribution)

All these steps are explained in details below:

1. Apple Developer Account

An Apple developer profile is mandatory to be able to create or publish any iOS app.

Create an Apple Developer account if you don't already have and enroll in the iOS Developer program:

Create an Apple developer account if you don't already have one: <https://developer.apple.com/account/>

Register to the Apple developer program: <http://developer.apple.com/programs/start/standard/>

This will cost \$99 and might take a few days to be ready. Apple might ask your ID or might contact you by phone during the approval process.

Make sure to carefully complete all profile details, otherwise your request might be rejected.

Your profile will get a team id which will become the `developerteamid` parameter (Membership -> Team ID)

Once you are done we can move to the next step.

2. Developer Profile

Create a Distribution Provisioning Profile for the application.

This consists of two steps:

A.) CSR: Create a Certificate Signing Request

Applications > Utilities -> Keychain Access app > Certificate Assistant > Request a Certificate

B.) Certificate: Create a Distribution certificate on the developer portal

App Store and Ad Hoc -> distribution.cer (iosdistributioncertificatename)

A.) CSR

A CSR is required to be able to obtain your apple developer certificate (which is discussed at the next point).

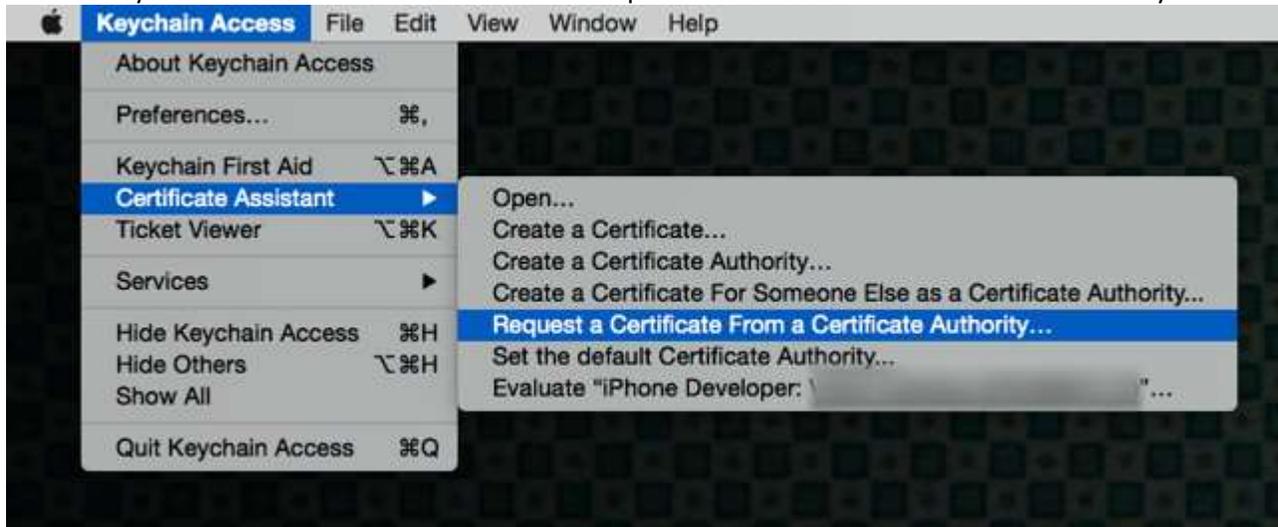
To create a CSR (Certificate Signing Request), follow the steps below:

1. On your Mac go to the folder Applications > Utilities and open Keychain Access.



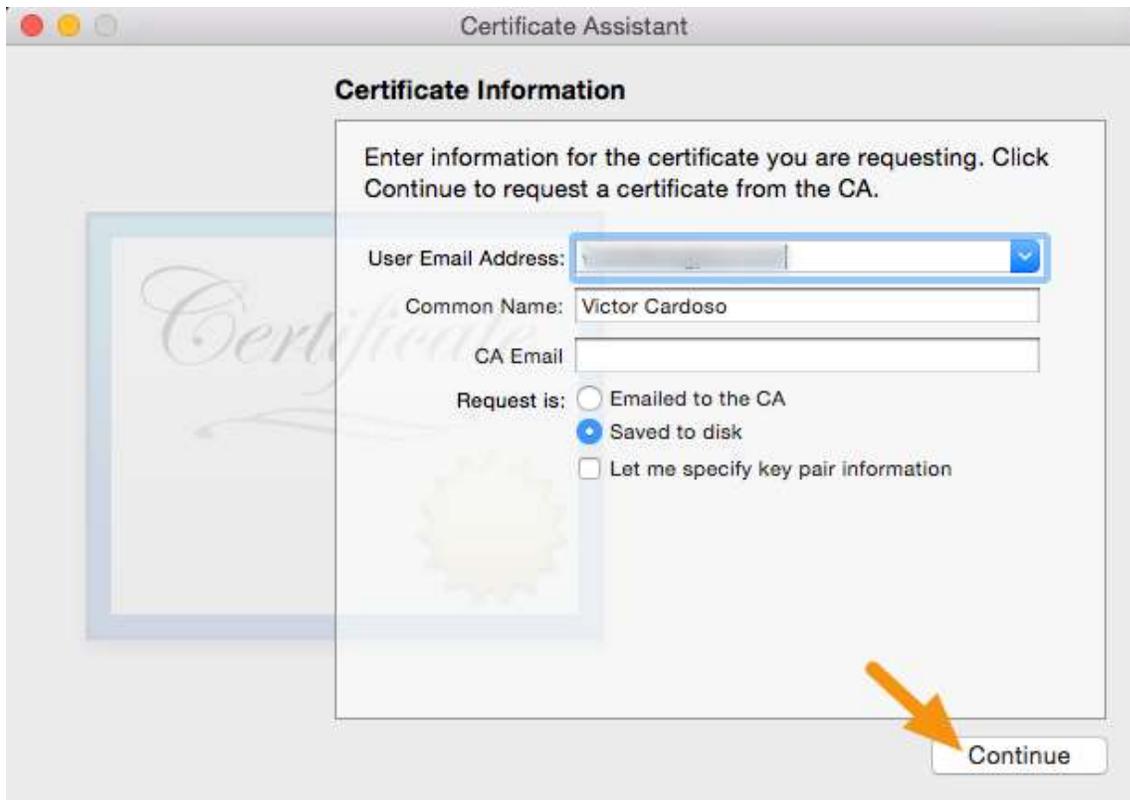
Keychain Access.app

2. Go to Keychain Access > Certificate Assistant > Request a Certificate from a Certificate Authority.



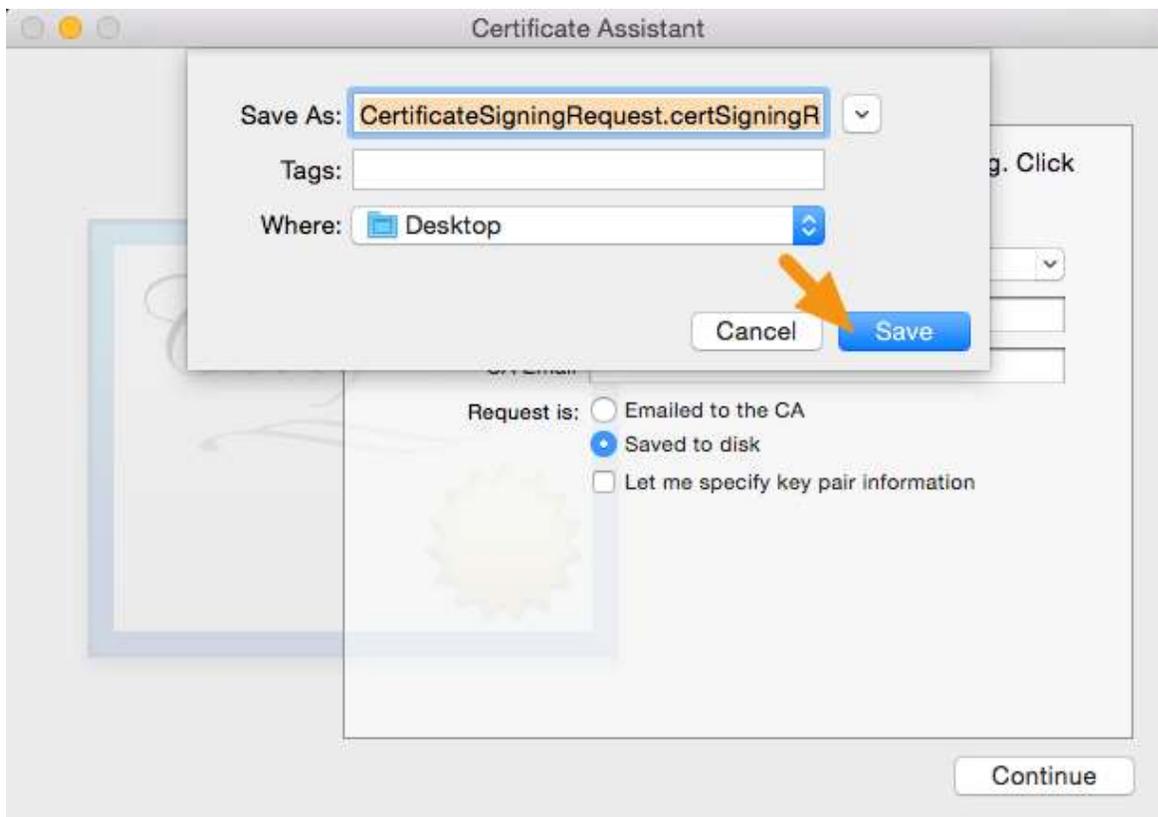
3. Fill out the information in the Certificate Information window as specified below and click "Continue.".

- In the User Email Address field, enter the email address to identify with this certificate
- In the Common Name field, enter your name
- In the Request group, click the "Saved to disk" option



4. Save the file to your hard drive.

The assistant creates a Certificate Signing Request (CSR) file which contains a public/private key pair.



B.) Certificate

To obtain a certificate, go to your [Developer account](#) -> Certificates, Identifiers & Profiles -> Certificates -> + -> iOS Distribution (App Store and Ad Hoc)

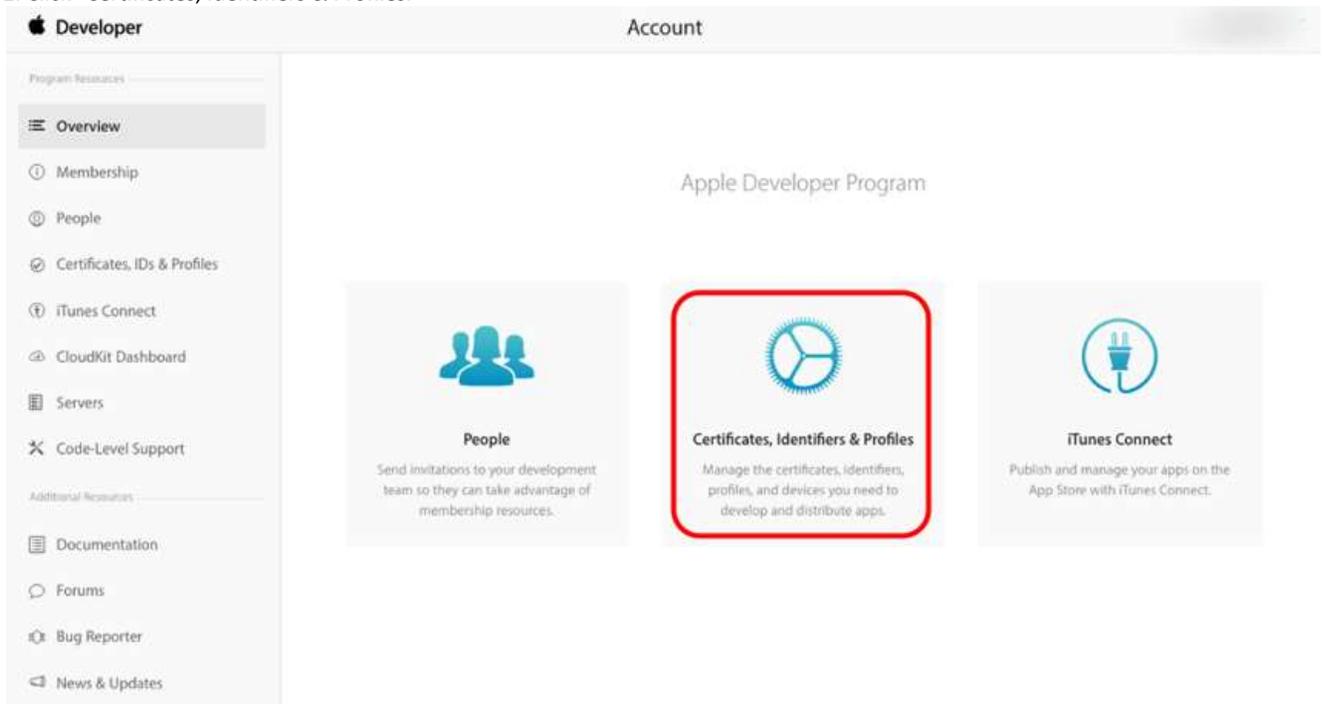
Below you can find step-by-step instructions, but these have been captured in 2020 and the apple website changed meantime (current website is similar with some modifications).

Follow these steps to create a Distribution certificate (deprecated):

1. Login into the iOS Developer Console goto <<https://developer.apple.com>> and click Account at the top



2. Click "Certificates, Identifiers & Profiles."



3. Click on the "Certificates" link under "iOS Apps."



4. Go to Certificates > Production and click on the "+" button to add a new certificate.

Certificates, Identifiers & Profiles

iOS Apps

Certificates

- All
- Pending
- Development
- Production**



Identifiers

- App IDs
- Pass Type IDs
- Website Push IDs
- iCloud Containers
- App Groups
- Merchant IDs

Devices

- All

Provisioning Profiles

- All
- Development
- Distribution

iOS Certificates (Production)

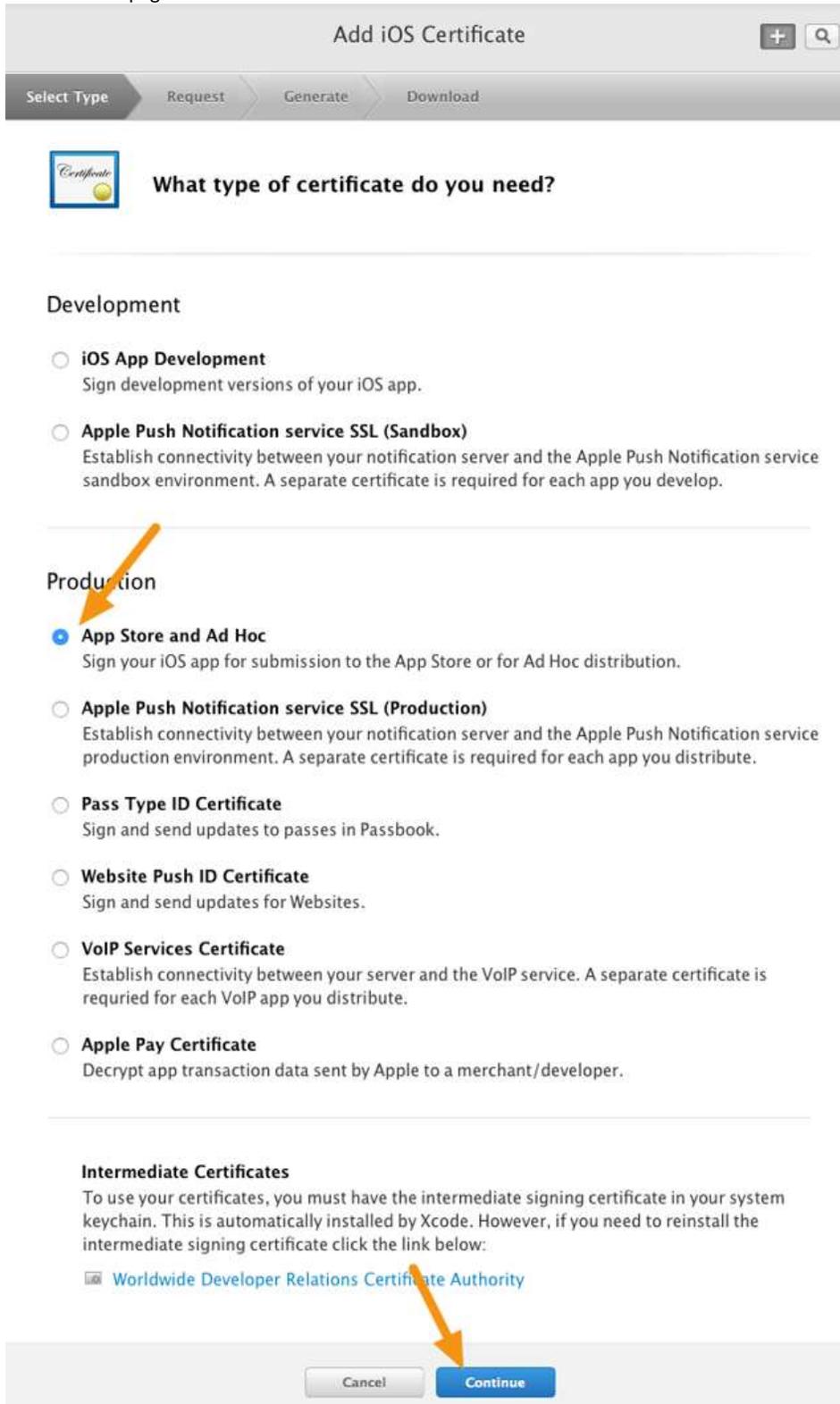


3 Certificates Total

Name	Type	Expires
[Redacted]	APNs Production iOS	Feb 06, 2015
Victor Cardoso	iOS Distribution	Sep 18, 2015
Victor Cardoso	iOS Distribution	Feb 06, 2015

5. Select "App Store and Ad Hoc" from the Production options and click "Continue."

Note: To use your certificates, you must have the intermediate signing certificate in your OS X system keychain. This is automatically installed by Xcode. However, if you need to reinstall the intermediate signing certificate click the link at the bottom of the page.



Add iOS Certificate

Select Type | Request | Generate | Download

What type of certificate do you need?

Development

- iOS App Development**
Sign development versions of your iOS app.
- Apple Push Notification service SSL (Sandbox)**
Establish connectivity between your notification server and the Apple Push Notification service sandbox environment. A separate certificate is required for each app you develop.

Production

- App Store and Ad Hoc**
Sign your iOS app for submission to the App Store or for Ad Hoc distribution.
- 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 Passbook.
- Website Push ID Certificate**
Sign and send updates for Websites.
- VoIP Services Certificate**
Establish connectivity between your server and the VoIP service. A separate certificate is required for each VoIP app you distribute.
- Apple Pay Certificate**
Decrypt app transaction data sent by Apple to a merchant/developer.

Intermediate Certificates

To use your certificates, you must have the intermediate signing certificate in your system keychain. This is automatically installed by Xcode. However, if you need to reinstall the intermediate signing certificate click the link below:

[Worldwide Developer Relations Certificate Authority](#)

Cancel | Continue

6. Click on "Continue" again.

You created your CSR file in the previous steps, so there's no need to create another one.

+ Q

Select Type Request Generate Approval



About Creating a Certificate Signing Request (CSR)

To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Mac. To create a CSR file, follow the instructions below to create one using Keychain Access.

Create a CSR file.

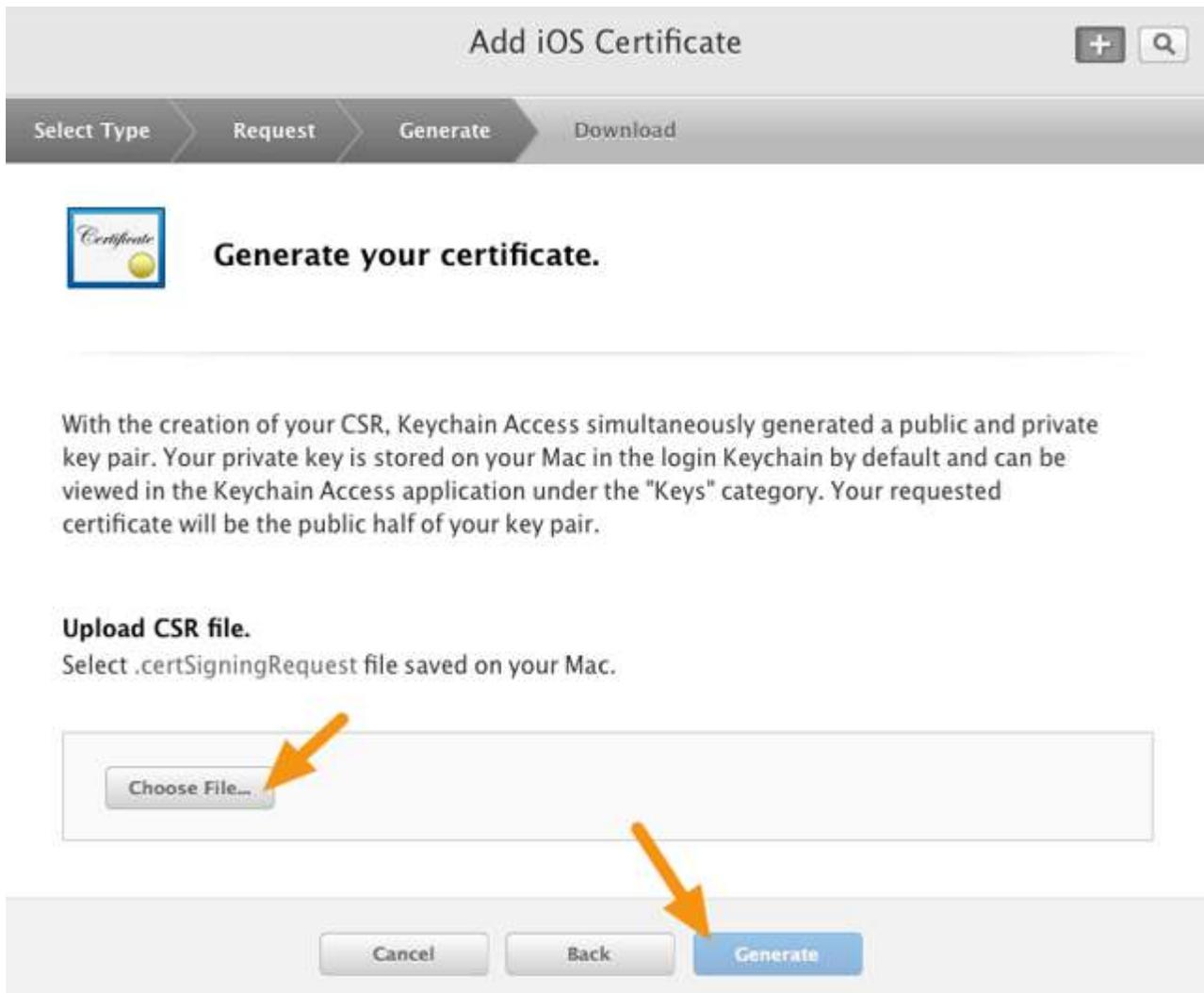
In the Applications folder on your Mac, open the Utilities folder and launch Keychain Access.

Within the Keychain Access drop down menu, select Keychain Access > Certificate Assistant > Request a Certificate from a Certificate Authority.

- In the Certificate Information window, enter the following information:
 - In the User Email Address field, enter your email address.
 - In the Common Name field, create a name for your private key (e.g., John Doe Dev Key).
 - The CA Email Address field should be left empty.
 - In the "Request is" group, select the "Saved to disk" option.
- Click Continue within Keychain Access to complete the CSR generating process.

Cancel Back Continue

7. Click on "Choose File" and select the CSR file you created previously and then click on "Generate."



8. Click on the "Download" button to download a .cer file to your machine and then click "Done."

Add iOS Certificate

+ Q

Select Type Request Generate Download



Your certificate is ready.

Download, Install and Backup

Download your certificate to your Mac, then double click the .cer file to install in Keychain Access. Make sure to save a backup copy of your private and public keys somewhere secure.



Name:	IOS Distribution:	
Type:	IOS Distribution	
Expires:	Nov 25, 2015	

Download

Documentation

For more information on using and managing your certificates read:

[App Distribution Guide](#)

Add Another Done

Set the file at the VoIP app builder website on the iOS page -> Certificate file (as-is or in a zip archive)
This will become the iosdistributioncertificatename parameter.

3. App Profile

Create a Distribution Provisioning Profile for the application.

This consists of two steps:

A.) App ID: Create an App ID

Identifiers -> + -> App IDs -> Explicit App ID -> enter the app bundle ID

B.) Profile: Create a Distribution provisioning profile

Certificates, IDs & Profiles -> Profiles -> + -> App Store -> chose App ID -> distribution certificate -> enter profile name -> download (iosprovisioningprofilename)

A.) App ID

Create an App ID (Identifier) which mainly consists of the name of the app and a Bundle ID.

To obtain your App ID, go to your [Developer account](#) -> Certificates, Identifiers & Profiles -> Identifiers -> + -> App IDs -> Explicit App ID -> enter the app bundle ID

Below you can find step-by-step instructions, but these have been captured in 2020 and the apple website changed meantime (current website is similar with some modifications).

“App ID” has been recently renamed to “Identifiers” by Apple.

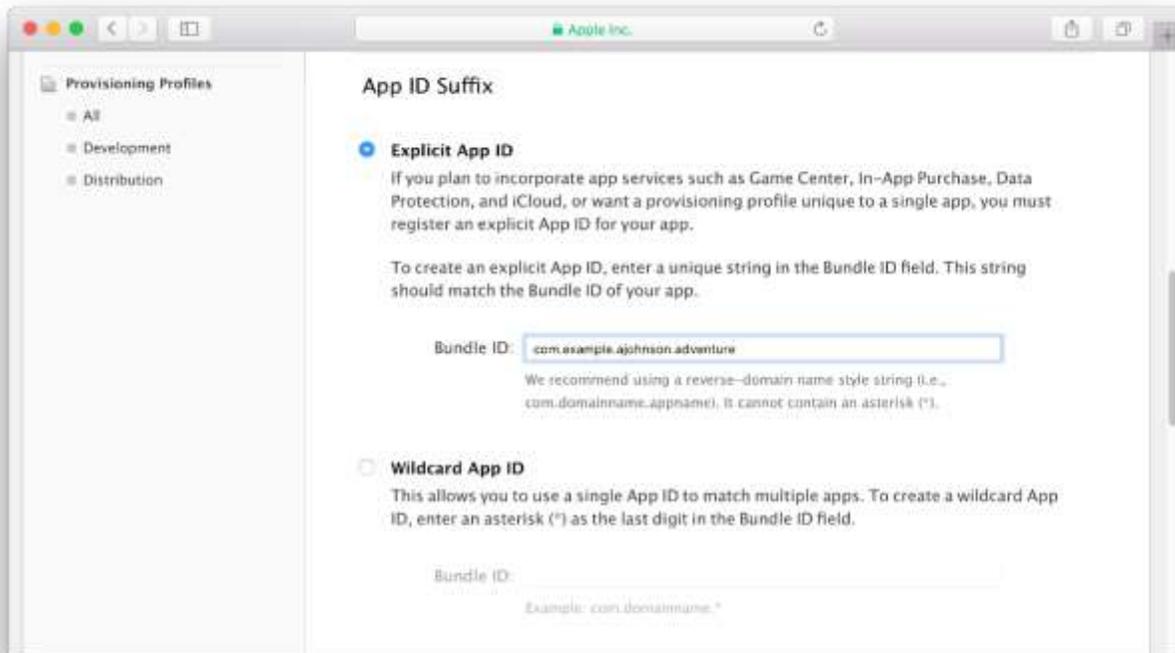
Follow these steps to create your App ID (deprecated):

Now that we have our certificate ready, let's create an App ID which mainly consists of the name of the app and a Bundle ID.

1. Sign in to developer.apple.com/account, and click Certificates, IDs & Profiles.
2. Go to Identifiers
3. Click the Add button (+).
4. Select “App IDs”
5. Enter a name or description for the App ID in the Description field.



6. Create an explicit App ID, select Explicit App ID and enter the app's bundle ID in the Bundle ID field. The format of the bundle ID is a reversed domain, for example: com.yourcompanyname.brandname or com.webdomain.appname.



7. Under "Capabilities" enable "Push Notifications".
8. Click Continue.
9. Review the registration information, and click Register.
10. Click on the App ID that you have just created and scroll down to Capabilities section.
11. Next to the "Push notifications" capability a "Configure" button is displayed.
12. Click "Configure" to create the necessary certificate for Push notifications.
13. Create a Production SSL Certificate:
 - a. Click Create button
 - b. Select platform "iOS" and upload your CSR file
 - c. Once ready, download it.
14. Set the same App ID on the VoIP app builder website (iOS page -> App ID)
This will become the iospackagename parameter

B.) Profile

To obtain the Distribution Provisioning Profile, go to your [Developer account](#) -> Certificates, IDs & Profiles -> Profiles -> + -> App Store -> chose App ID -> distribution certificate -> enter profile name -> download.

*Note: For Ad-Hoc distribution (demo/test) you will need an **Ad-Hoc** provisioning profile instead!*

Below you can find step-by-step instructions, but these have been captured in 2020 and the apple website changed meantime (current website is similar with some modifications).

Follow these steps to create a Distribution Provisioning Profile:

1. Sign in to developer.apple.com/account, and click Certificates, IDs & Profiles.
2. Go to Profiles and click the Add button (+).
3. Select "App Store" and click Continue.
4. Choose the App ID You created earlier and click Continue.

5. Select your distribution certificate, and click Continue.
App store provisioning profile contains a single distribution certificate.
6. Enter a profile name, and click Continue.
Wait while your developer account generates the provisioning profile.
7. Download the Distribution Provisioning Profile
8. At the bottom of the page, Click Done.
9. Set the file at the VoIP app builder website on the iOS page -> Profile file (as-is or in a zip archive)
This will become the iosprovisioningprofilename parameter

4. PushKit

Create a certificate for VoIP Push Notifications (optional).

A PushKit certificate is required if you wish to enable VoIP Push notification capable to launch or wake-up your device/app on incoming calls even if your device is sleeping or the app is closed.

This option will be applied only if you select App Store distribution (not for ad-hoc demo/test).

To obtain a PushKit certificate, go to your [Developer account](#) -> Certificates, Identifiers & Profiles -> Certificates -> VoIP Services -> download as voip_services.cer -> open in Keychain Access -> Export

Step by step instructions:

- Login to the Apple Developer portal.
- Go to “Certificates, Identifiers & Profiles” -> “Certificates” and generate a “VoIP Services” certificate.
- Go to “Certificates, Identifiers & Profiles” -> “Identifiers” -> click on your “App ID” -> “Capabilities” and enable “Push Notifications”. Make sure that there are no conflicts with Push Notifications certificates here.
- Download certificate from Apple developer page -> voip_services.cer.
- Export to a key-store:
After you download the certificate, open it up, and this should open the Keychain Access application on your macOS, and you should see the certificate under the My Certificates section.
Expand the new certificate in Keychain Access, you should see the private key with either your name or your company name.
Select both items by using the “Select” key on your keyboard, right click (or cmd-click if you use a single button mouse), choose “Export 2 items”.
Then save the p12 file with name “pushkit.p12” to your Desktop – now you will be prompted to enter a password to protect it, you can either click Enter to skip the password or enter a password you desire.
- Upload the keystore file to the VoIP app builder website (iOS page -> Push file)

PushKit can work even if your VoIP server doesn’t have push notification support. In this case the app can use the push proxy service. Otherwise, you might integrate the push notification support directly into your VoIP service as described [here](#).

Here is the same process again with more details (with screenshots):

You will need a VoIP push certificate which will be used by the server to be able to connect to apple push cloud service (APNS).

Follow the below steps to generate and set the certificate:

Login to your with your developer account and go to [Certificates, Identifiers & Profiles](#).

To generate a VoIP push certificate you first need to click on the All button in the Certificates section on the left-hand side. Then, click the + button:



On the next page you need to select the VoIP Services Certificate:

- 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 Certificate**
Decrypt app transaction data sent by Apple to a merchant/developer.
- Apple Pay Merchant Identity**
A client TLS certificate that is used to authenticate you to Apple Pay Servers

Then select the App ID for which you're creating this VoIP certificate:



Next, you'll be presented with instructions on how to create a so-called CSR (Certificate Signing Request) file:



About Creating a Certificate Signing Request (CSR)

To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Mac. To create a CSR file, follow the instructions below to create one using Keychain Access.

Create a CSR file.

In the Applications folder on your Mac, open the Utilities folder and launch Keychain Access.

Within the Keychain Access drop down menu, select Keychain Access > Certificate Assistant > Request a Certificate from a Certificate Authority.

- In the Certificate Information window, enter the following information:
 - In the User Email Address field, enter your email address.
 - In the Common Name field, create a name for your private key (e.g., John Doe Dev Key).
 - The CA Email Address field should be left empty.
 - In the "Request is" group, select the "Saved to disk" option.
- Click Continue within Keychain Access to complete the CSR generating process.

Once you create that file, you'll select it for upload on the next screen. If everything goes well you'll be given the certificate which you **have to** download:



Your certificate is ready.

Download, Install and Backup

Download your certificate to your Mac, then double click the .cer file to install in Keychain Access. Make sure to save a backup copy of your private and public keys somewhere secure.



Name:	Apple VoIP Services:com.nikola-breznjak.voipTestNikola
Type:	VoIP Services
Expires:	Sep 18, 2017

[Download](#)

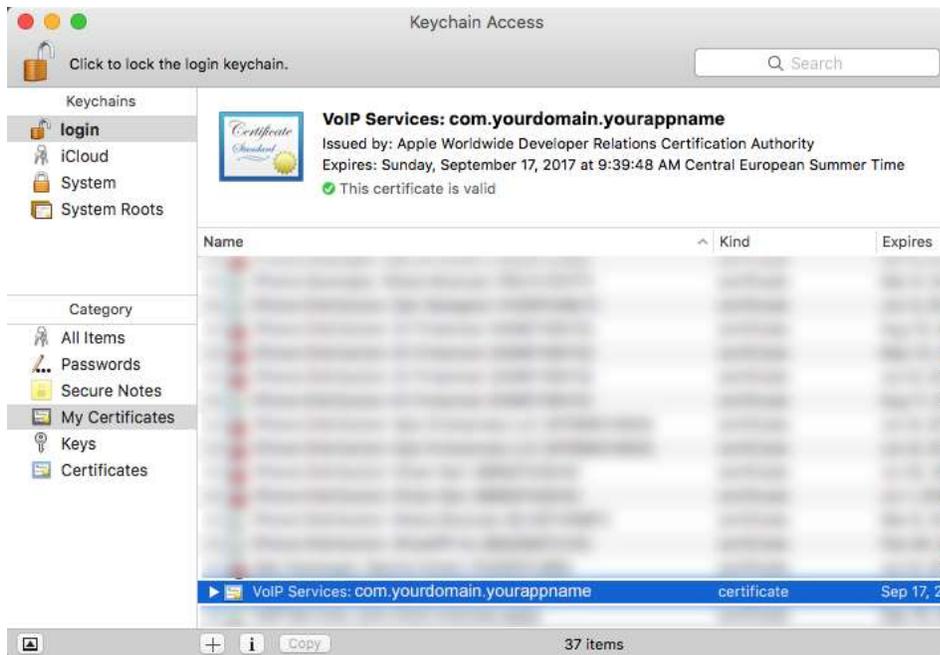
Documentation

For more information on using and managing your certificates read:

 [App Distribution Guide](#)

Download certificate from Apple developer page -> voip_services.cer.

After you download the certificate, open it up, and this should open the Keychain Access application, and you should see the certificate under the My Certificates section:



Expand the new certificate in Keychain Access, you should see the private key with either your name or your company name. Select both items by using the “Select” key on your keyboard, right click (or cmd-click if you use a single button mouse), choose “Export 2 items”.

Then save the p12 file with name “pushkit.p12” to your Desktop – now you will be prompted to enter a password to protect it, you can either click Enter to skip the password or enter a password you desire.

Upload this pushkit.p12 keystore file to the VoIP app builder website (iOS page -> Push file).

5. VoIP App Builder

Go to our [VoIP app builder website](#) and configure your softphone by supplying all the settings required for iOS.

General settings:

There are a few mandatory settings required by the builder service such as your app name, company name and SIP server address.

Design related:

- For the launcher icon we recommend to supply a high resolution image file of 1024 x 1024 pixels preferably in PNG format (if missing, a default icon will be set)
- The logo image should be minimum width of 250 pixels rectangle PNG (the logo is optional)
- *Optionally a launch screen image might be also possible to be configured in PNG format with exact size: 1242x2208 pixels. Logo or icon will be used if not set. This doesn't have much importance as the app starts quickly so any launch screen could be barely seen by the users.*

iOS specific details:

Upload/set the required iOS related details (obtained by following the above described steps) on the VoIP app builder website, under the iOS link:

- Team ID which consists of 10 characters and can be found here:

- login to <https://developer.apple.com>
- select Account in top menu bar
- select Membership in the left menu and you will see the Team ID displayed.
-this will become the developerteamid parameter
- App ID *-this will become the iospackagename parameter*
- Distribution certificate *-this will become the iosdistributioncertificatename parameter*
- Distribution Provisioning Profile *-this will become the iosprovisioningprofilename parameter*
- PushKit certificate (optional) *-voip_services.cer -> pushkit.p12*

Other settings:

You might configure any other settings allowed by the VoIP app builder website.

For the initial build we recommend to go with conservative settings and change/add only really required advanced settings.

Please note that not all the advanced settings can be applied for all platform.

The VoIP app builder cloud service will generate your softphone and will send it to you by email or you will be able to download it from the website.

6. Publish

Upload the received iOS softphone to App Store Connect and publish it.

Now that you have created your app profile on [App Store Connect](#) and you have received the softphone you can upload it by following these steps:

A.) Create your app entry on [App Store Connect](#) page and fill in all the data required by Apple so when you receive the softphone you can upload it immediately.

B.) After you have received the iOS softphone, upload it to App Store Connect and submit it for review by the Apple review team. This can take anywhere from a few days to a couple of weeks.

Once accepted by Apple, you can immediately publish it on the App Store.

To upload the iOS softphone to App Store Connect, you need the following:

- Mac OS X machine
- Transporter application

Download your softphone.zip archive file and unpack it. You should find an .ipa file in the archive.

Download Transporter application from here:

<https://apps.apple.com/app/transporter/id1450874784?mt=12>

Install it on your Mac and start it. Follow the below steps to upload the iOS softphone to App Store Connect:

1. Launch Transporter and login with your Apple Developer account.
2. Click on "ADD APP" button, and select your .ipa file that you have unpacked.
3. Click "DELIVER" button.

4. A progress bar will show the progress of your upload. Once complete, your build will be listed under your app in App Store Connect.

Once the upload finished, login to <https://appstoreconnect.apple.com> and select My Apps, select the app that you have created for the softphone (usually the same brand name as the softphone) and in the left menu, under "iOS APP", click on the current version.

Submission guidelines:

Make sure to properly fill out your developer profile and create an unique app description, otherwise Apple might reject your submission.

Apple might reject your app for various reasons, including bad app name, your company history, reputation, poor quality or inaccurate app description, duplicate content, not working SIP account, broken/bad links, incorrect privacy policy and more importantly, if it doesn't comply with the [App Store submission guidelines](#) or their various other policies. More details can be found [here](#).

MizuTech is not responsible for rejecting your app from the App Store!

Once published/available in the store, you should receive a notification about it by email from Apple.

Test account:

You should supply one or two working SIP account for Apple so their test team can verify your app. Make sure that the account(s) will work correctly (your VoIP server is online and test calls will succeed), otherwise your app might be rejected.

Payments:

You should either implement in-app purchases or hide any mentions for payment or payment links from your app description and user interface.

Apple requires that all kind of payments made from an application or for a service offered by an application must go through Apple.

If this is not fulfilled, your app might be rejected for "Guideline 5.0 - Legal" reasons.

App description:

We recommend to write a good unique description for your app, preferable with a video: Provide a video of your iOS softphone demonstrating the features (don't mention anything about payment/recharge).

China:

CallKit integration is not allowed for China. We recommend to not promote your app for the China App Store or disable CallKit in your softphone.

More help can be found [here](#).

7. Ad-Hoc

In case if you select the Ad-Hoc option on the app builder website then you receive an iOS softphone to be used on your own phone only (bind to UDID) and can't be published on the App Store.

You can find your UDID by connecting your iOS device to your computer -> start iTunes -> select your device -> then Summary tab on the left.

In the first "box" you will see a field called "ECID:" or "Serial Number", click it a few times, until you see your UDID, then right click and copy.

More help can be found [here](#).

Once you have the UDID you will need to add it to your account at the iOS Provisioning Portal (the same way as it would be a development device). Go to the Devices section and click on the 'Add Device' button to add your device.

You can find a "Provisioning profile" in the ZIP archive.

You can install this softphone using iTunes. Just follow the instructions below:

In iTunes you can go to the view menu, and "Show Sidebar", this will give you the sidebar, that you can drag 'n drop to.

You'll drag 'n drop to the open area that will be near the bottom of the sidebar (I'm typically doing this with both an IPA and a provisioning profile). After you do that, there will be an apps menu that appears in the sidebar with your app in it. Click on that, and you'll see your application in the main view. You can then drag your application from there to your device

More help: <http://stackoverflow.com/questions/26720764/install-ipa-with-itunes-12>

Demo video: <https://www.youtube.com/watch?v=0ACq4CRpEJ8&feature=youtu.be>

Technical details: <https://useyourloaf.com/blog/creating-iphone-ad-hoc-distribution-builds/>

Copyright © Mizutech SRL

www.mizu-voip.com