Contents

[Mobile Application Configuration 1](#_Toc11598725)

[1. Pre-requisites for configuring Mobile application 1](#_Toc11598726)

[2. Android Platform 3](#_Toc11598727)

[3. IOS Platform: 8](#_Toc11598728)

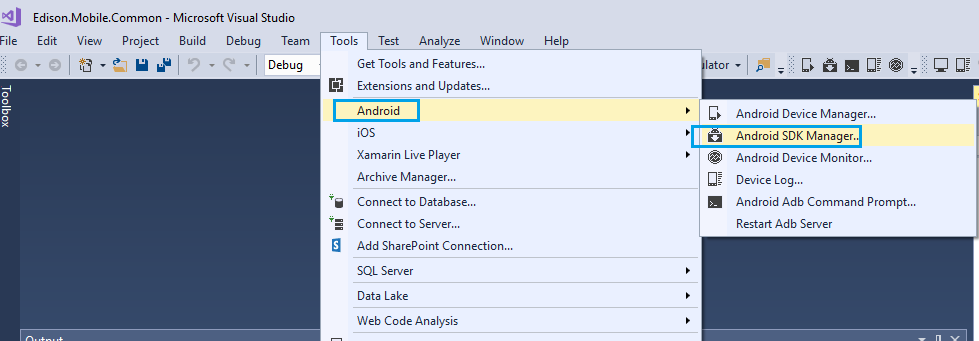
# Mobile Application Configuration

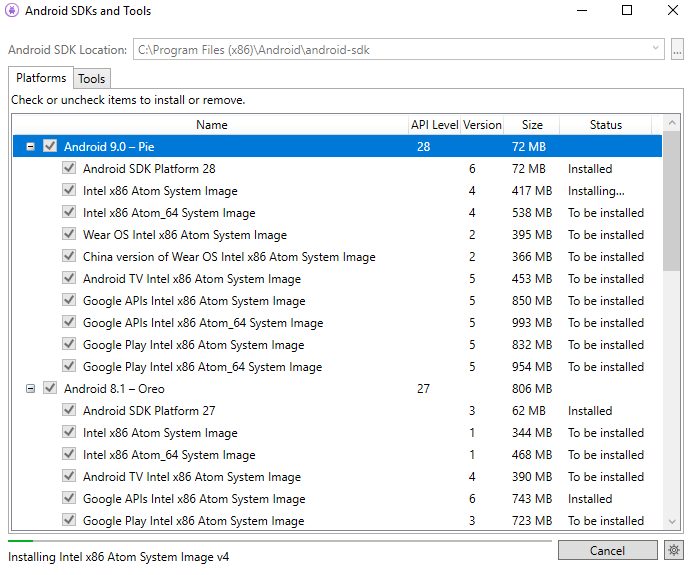
## Pre-requisites for configuring Mobile application

1. Install Visual Studio 2017 with the following packages:

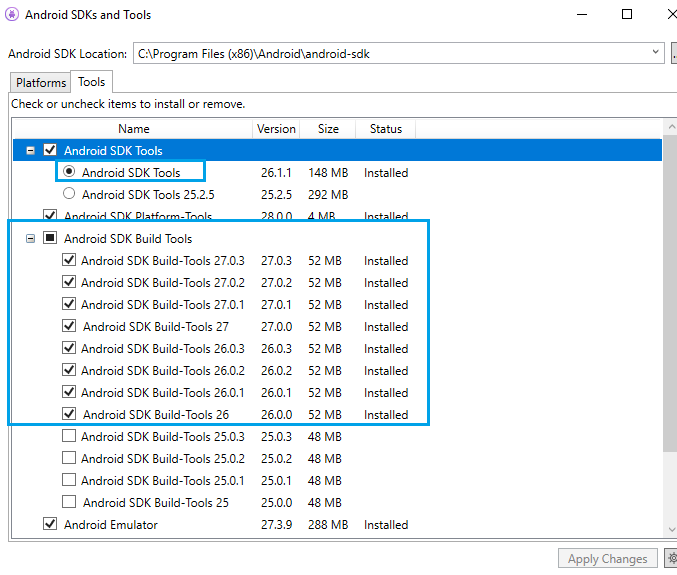
* Mobile Development with .NET
* Mobile development with JavaScript
* NET Framework 4.7.1 SDK and targeting pack
* Connected Service for Azure IoT Hub
* NET Core cross-platform development
* Azure development
* Data storage and processing
* Android SDK 8.0, 8.1 and 9.0

2. Install below **Android SDKs under platforms and Tools**.





3. Install below **Android SDKs under Tools as shown below**.



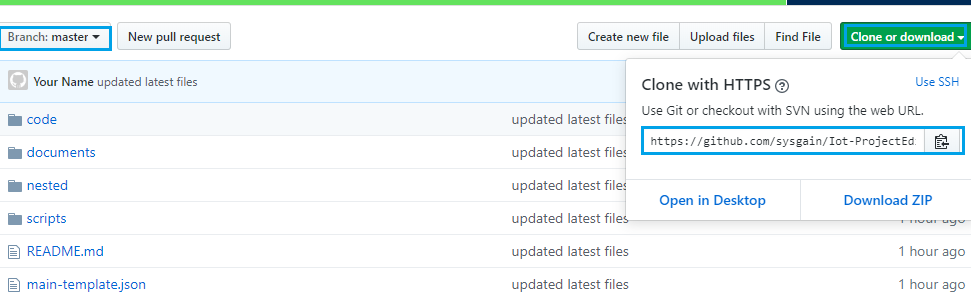
## Android Platform

Download the script using below command.

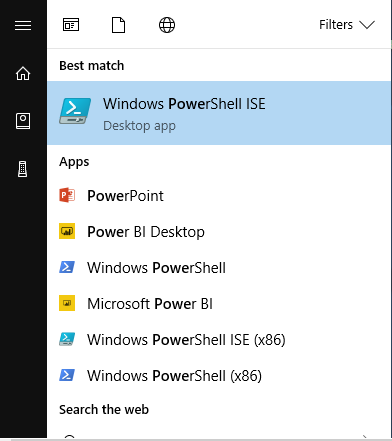
**git clone <git url>**

**Ex:** git clone https://github.com/xxxx/xxxxx.git

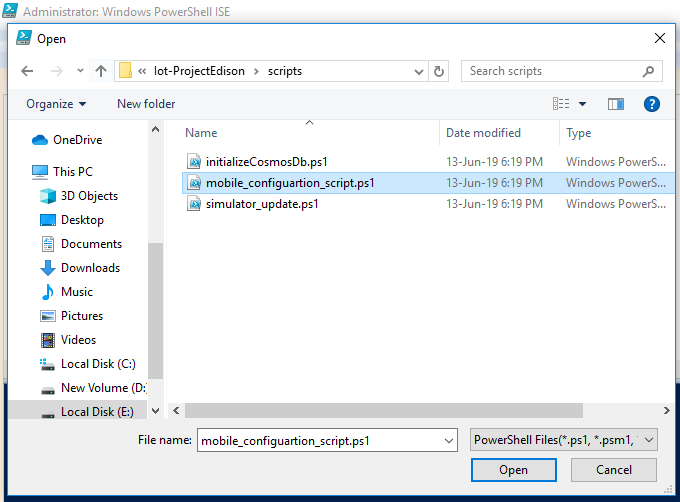
Take the URL as shown like below.

****

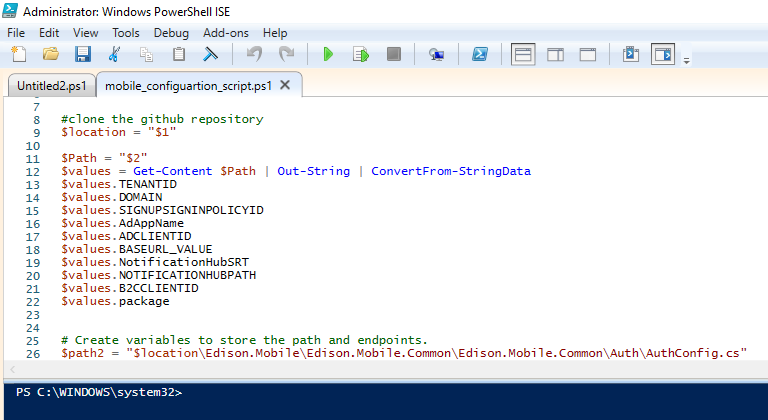
Open PowerShell ISE using Administration permission from start menu.



Open the script from below downloaded project path Iot-ProjectEdison/scripts/ mobile\_configuartion\_script.ps1.



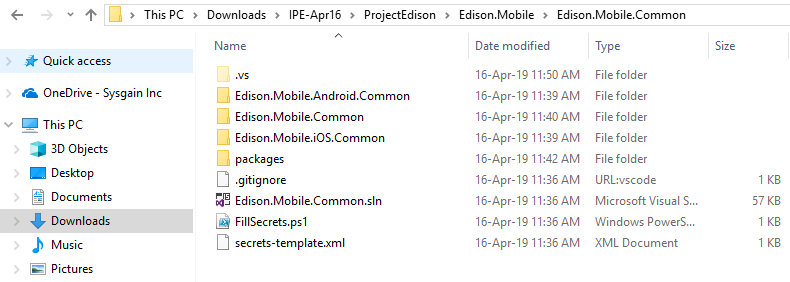
Click on **Run** script (icon).



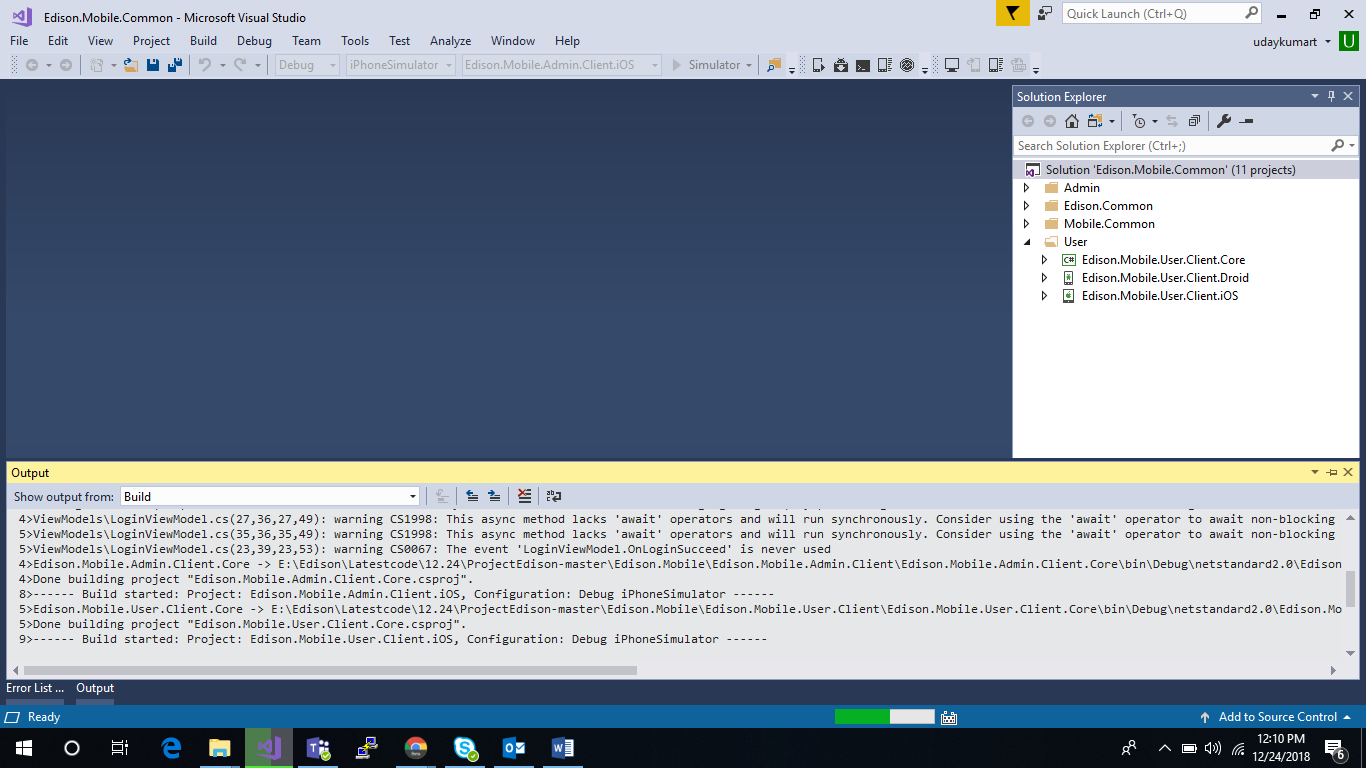
The above script updates the Mobile code with required values.

Configure and Build the User Mobile application using **Edison.Mobile.Common.sln**

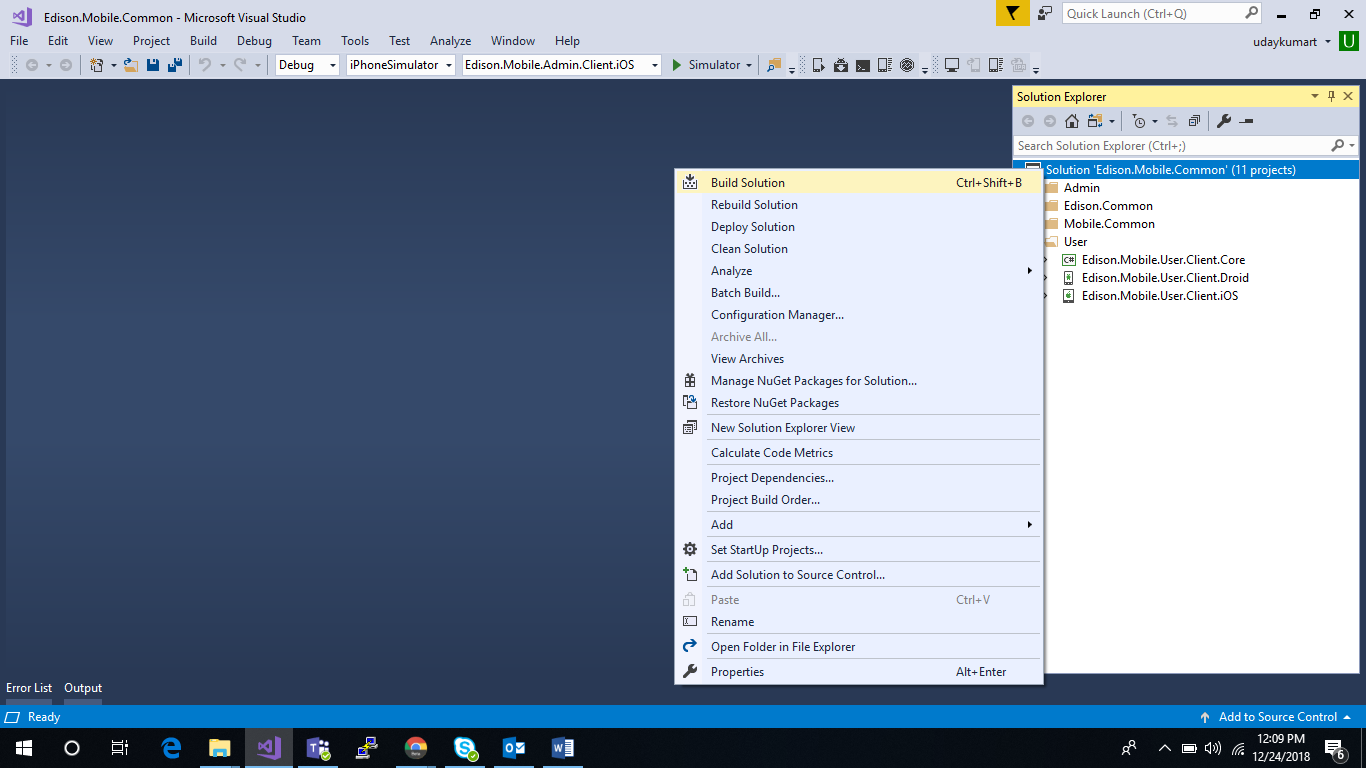
Open **Edison.Mobile.Common.sln** by double clicking on it.



1. Wait for some time visual Studio will restore the dependencies automatically, until you see **Ready** on left side bottom of the screen of the output section.

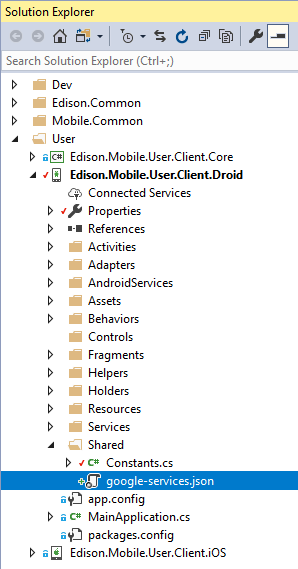


1. Right click on the solution and click on **Build Solution.**



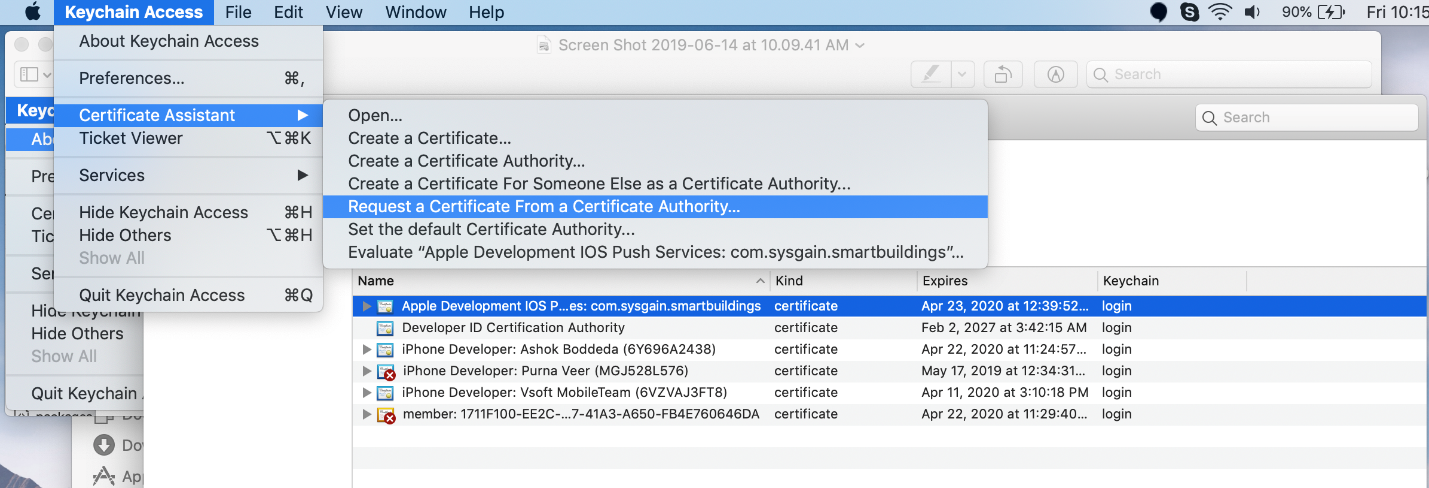
1. Once the build is succeeded. Add the downloaded **google-services.json** file in shared folder of **Edison.Mobile.User.Client.Droid** Project as shown in below screen shot.

**Note:** Please refer User Guide documentation 5th section 3rd point for downloading **google-service.json**.

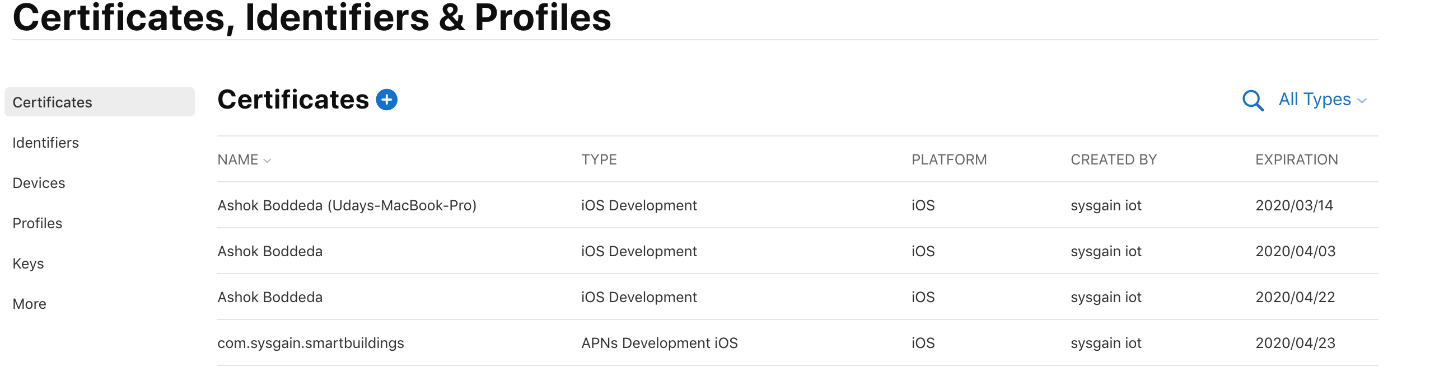


## IOS Platform:

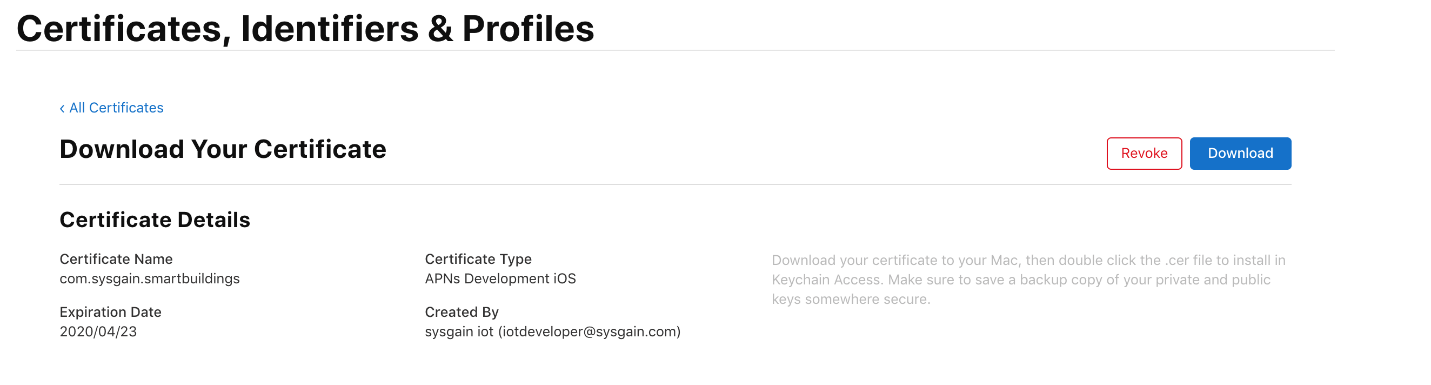
1. Request a certificate from certificate authority.



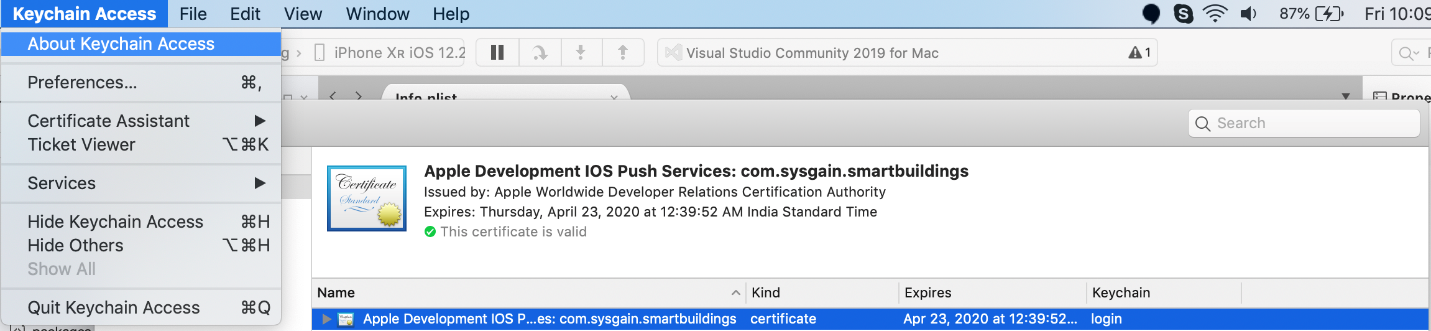
1. Create a cer file in apple developer portal by choosing APNS Option.



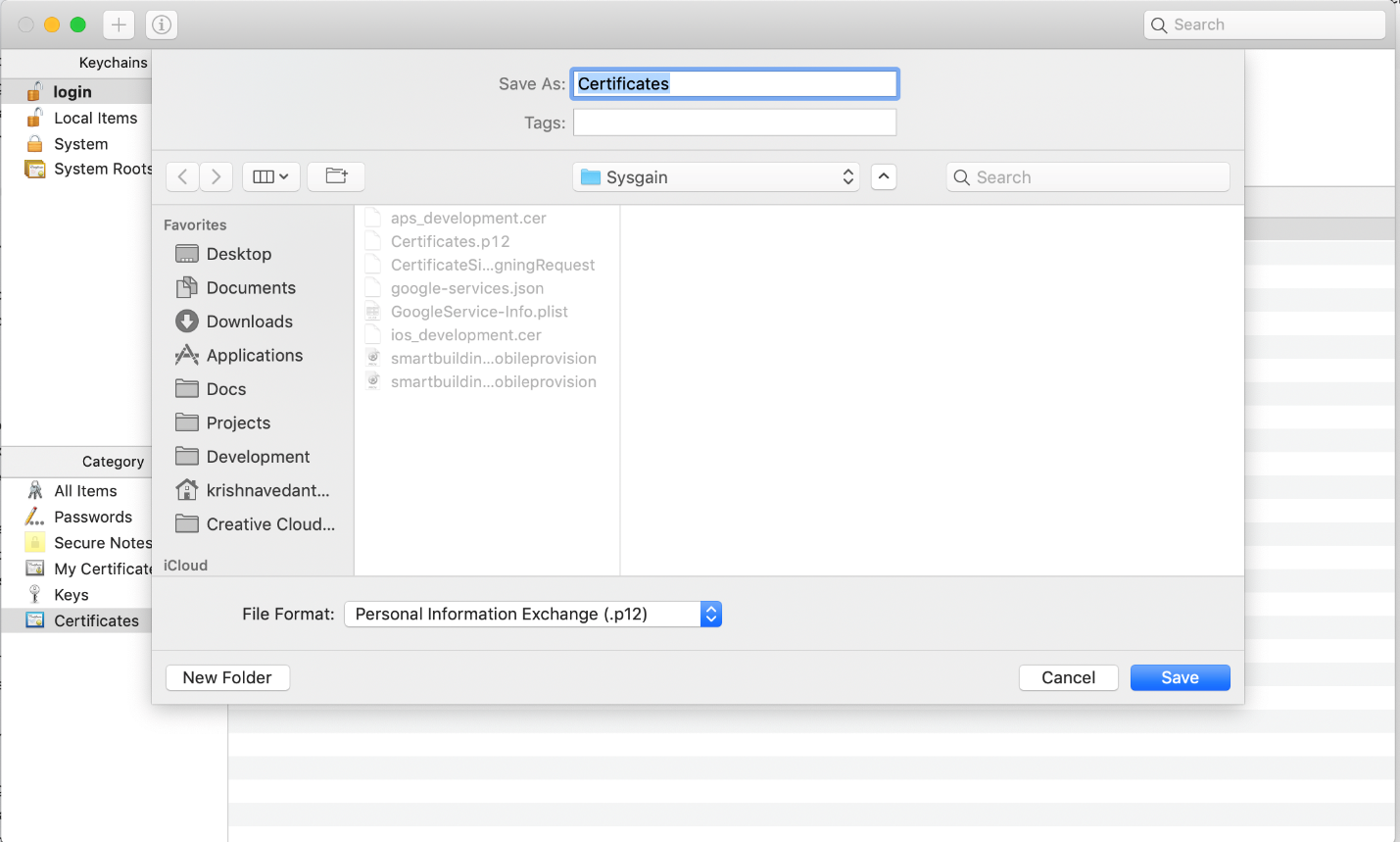
1. Download the cer file and install into development system.



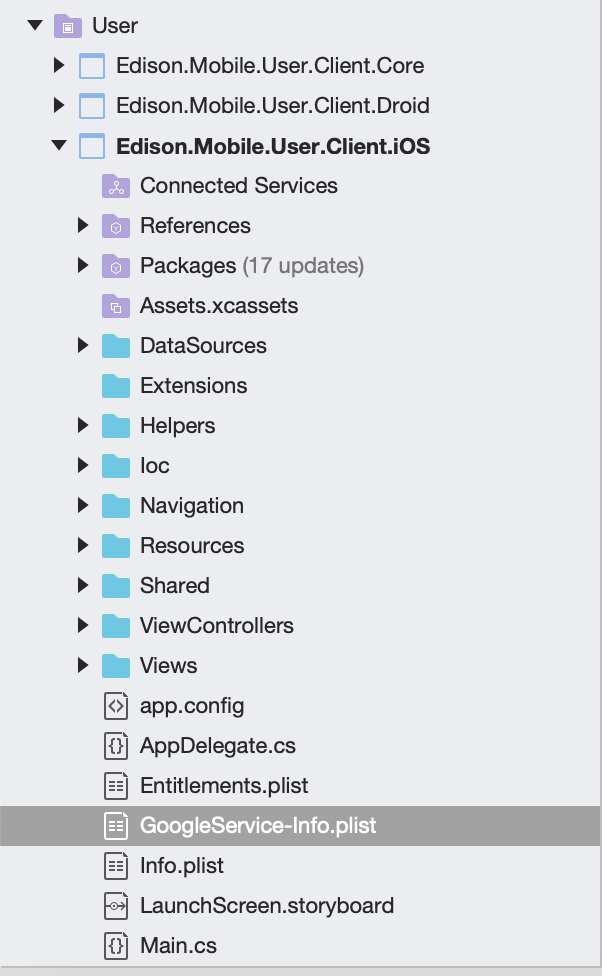
1. Export the installed file to generate a P12 file with a password protection.



1. Create an application into Firebase for iOS with a unique bundle identifier.



1. Upload the exported p12 file into Azure and Firebase.



1. Download the **GoogleServices-Info.plist** into platform specific project and change to Build Action settings as Bundle Resource.

