

# Section 1 - Getting started

iOS iOS 5.2.x and earlier iOS 5.3.x iOS 5.4.x iOS 5.5.x iOS 5.6.x iOS 6.0.x iOS 6.1.x iOS - 6.3.x Latest version - 6.4.x

In this section, we will help you get started by going through the preliminary steps for a smooth integration:

- Configure Push Notifications in your app and generate a certificate on the Apple developer portal
- Set the Accengage dashboard correctly
- Create URL Schemes for deep linking

During these steps, you will need to use Xcode, your Apple Developer account, as well as the Accengage Dashboard.

## 1.1 Requirements and important information on the iOS SDK

iOS iOS 5.2.x and earlier iOS 5.3.x iOS 5.4.x iOS 5.5.x iOS 5.6.x iOS 6.0.x iOS 6.1.x iOS - 6.3.x Latest version - 6.4.x  
The Accengage iOS SDK is provided as a dynamic framework and is compatible with:

- Xcode 9
- iOS 8 and higher.

In terms of size, the SDK will increase your app download size by ~608 Kb.

In order to integrate it, you will need:

- A partner id and a private key for each application and each environment (Development and production), available for registered customers on our platform. If you do not have a partner id and private key yet, please contact our support (<http://ticket.accengage.com>).

See below to set these correctly with your other credentials :

- To generate a Push Notification certificate (see below)
- The Bundle Id of your application which should be entered in the application form in the Accengage dashboard (Settings > Manage application > Edit).

## 1.2 Activate Push Notifications

iOS iOS 5.2.x and earlier iOS 5.3.x iOS 5.4.x iOS 5.5.x iOS 5.6.x iOS 6.0.x iOS 6.1.x iOS - 6.3.x Latest version - 6.4.x

### Enable Push Notifications in Xcode

1. In the project editor, choose a target and click **Capabilities**.
2. In the Push Notifications section, click the switch to turn it from OFF to ON.

## Generate and export an APNS client TLS certificate from the Apple developer portal

### Generate an APNS client TLS certificate

You need to generate a separate client TLS certificate for each app you distribute that uses push notifications including Development and Production versions.

It is important that you generate a Development (Sandbox) Push Notification certificate to use with a Development Provisioning Profile, and/or a Production Push Notification certificate to use with an Ad Hoc or Distribution Provisioning Profile. They will be used in 2 separate applications within Accengage Dashboard (see below) and it will be important that you respect this differentiation.

Generating the certificate fully enables push notifications for the associated App ID. In your [developer account](#), you will notice that the Push Notifications service for the App ID changes automatically from Configurable to Enabled.

1. In your developer account, go to [Certificates, Identifiers & Profiles](#) and if necessary, choose the operating system from the pop-up menu on the left (for macOS apps, choose OS X).
2. Under Certificates, select All.

3. Click the Add button + in the upper-right corner.
4. Under Production, select the “Apple Push Notification service SSL (Sandbox & Production)” checkbox, then click Continue.
5. Choose an App ID from the App ID pop-up menu, and click Continue.
6. Choose the explicit App ID that matches your bundle ID.
7. Follow the instructions to create a certificate signing request on your Mac, and click Continue.
8. Click Choose File.
9. In the dialog that appears, select the certificate request file (a file with a `.certSigningRequest` file extension), and click Choose.
10. Click Continue.
11. Click Download and double-click the downloaded file (a file with a `.cer` file extension) to add the certificate to your keychain.
12. Click Done.

## Create a certificate signing request

1. Launch Keychain Access located in `/Applications/Utilities`.
2. Choose Keychain Access > Certificate Assistant > Request a Certificate from a Certificate Authority.
3. In the Certificate Assistant dialog, enter an email address in the User Email Address field.
4. In the Common Name field, enter a name for the key (for example, Gita Kumar Dev Key).
5. Leave the CA Email Address field empty.
6. Choose “Saved to disk”, and click Continue.

## Export the client TLS identity from your Mac

Export the identity from the keychain on the Mac where you created it, and copy it to the appropriate place on the server that runs the provider code and connects with the development or production version of APNs.

1. Launch Keychain Access.
2. In the Category section, select My Certificates.
3. Find the certificate you want to export and disclose its contents.
4. You'll see both a certificate and a private key.
5. Select both the certificate and the key, and choose File > Export Items.
6. From the File Format pop-up menu, choose a file format that your server accepts.
7. Enter a filename in the Save As field, and click Save.
8. The certificate and key are saved to the location you specified as a text file in the Personal Information Exchange format (a file with a `.p12` file extension).

## 1.3 Configure the Accengage dashboard for Push Notifications

iOS iOS 5.2.x and earlier iOS 5.3.x iOS 5.4.x iOS 5.5.x iOS 5.6.x iOS 6.0.x iOS 6.1.x iOS - 6.3.x Latest version - 6.4.x

Once you have exported your Push Notification certificate, you need to upload it in the Accengage Dashboard. Please go to Settings > Manage Application and click on the pencil to edit the configurations. You should have 2 separate applications, 1 for the Development environment, and 1 for Production. In the relevant application, you should:

- upload the push notification certificate
- enter its password
- tick the right type of push notification certificate
- enter the right Bundle Id

It is very important that you respect the following configuration:

- Use your Development Partner Id and private key, with a Development Provisioning profile within your application, a Development Push Notification Certificate, as well as set the Push Notification certificate as “Development” in the Accengage Dashboard (Settings > Manage application > Edit > Development certificate)
- Use your Production Partner Id and private key, with a Distribution or Ad Hoc Provisioning profile within your application, a Production Push Notification Certificate, as well as set the Push Notification certificate as “Production” in the Accengage Dashboard (Settings > Manage application > Edit > Development certificate)
- Also make sure that you enter the right Bundle Id for the right application within the Accengage dashboard.

If you do not respect this consistency, then you will end up with Development tokens within your Production environment or the opposite and this will create disconnections when you try and send Push Notifications, which means you might not receive some Push Notifications!

This is a common mistake which causes time consuming investigations. Please make sure that you have followed the previous instruction with attention to avoid those mistakes.

Once you have set this information up, you are now ready to integrate the SDK!

## 1.4 Configure URL Schemes

iOS iOS 5.2.x and earlier iOS 5.3.x iOS 5.4.x iOS 5.5.x iOS 5.6.x iOS 6.0.x iOS 6.1.x iOS - 6.3.x Latest version - 6.4.x

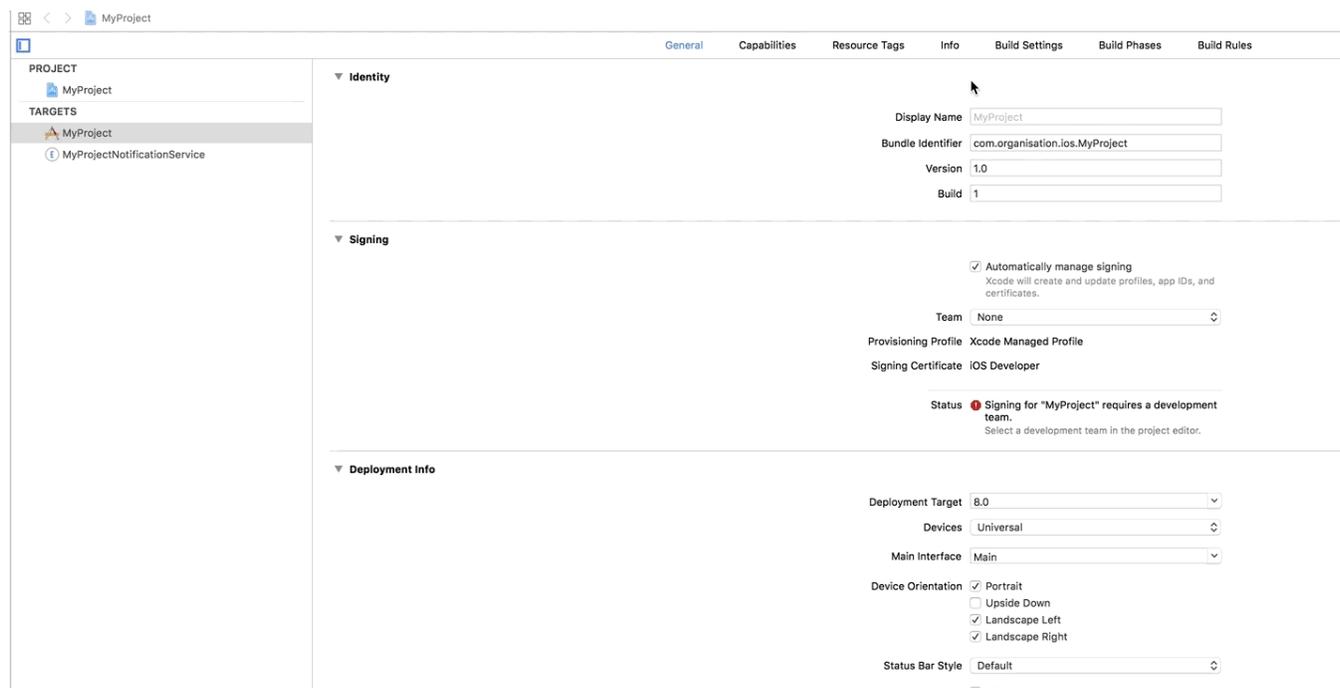
### Creating URL schemes

Deep Linking is becoming very important, it's a great way to enhance the effectiveness of your app's marketing campaigns. It will allow you to open a specific view of your app from another app, from a push message, from an in-app message or even from a website to your app.

To take advantage of it, you must create a custom URL Scheme associated with your application. A URL scheme lets you communicate with other apps through a protocol that you define. To find out more, check out [Apple documentation](#).

To register your URL Scheme:

1. Select your app target from the sidebar, then select the Info tab.
2. Open the URL Types expander and click +.
3. In the URL Schemes field, enter `your_app_url_scheme`.
4. Assign a unique identifier to the scheme to ensure uniqueness and avoid name collisions, so we'll use `com.myapp.url` for the example.



That's it! To verify that everything works:

1. Run your app
2. Open Safari on the same device
3. Type `your_app_url_scheme://` in the address bar, then press `Go`
4. You should arrive on the expected page

### Whitelisting URL Schemes

iOS 9 introduces changes that impact URL scheme management. For more details, you can consult [Apple's documentation](#).

Make sure that your application Info.plist includes the `LSApplicationQueriesSchemes` set with your app's URL Scheme (see previous section to find out how to create your app's URL scheme) and every scheme your app will query.

```
<key>LSApplicationQueriesSchemes</key>
<array>
  <string>your_app_url_scheme</string>
  <string>other_app</string>
</array>
```