# Introduction

This document will instruct to integrate the iOS SDK step by step

# Installation

## **Configuration**

Before using the DocSDK, place the configuration file into the DocServices-Info.plist project folder and one copy into the app store.

If you do not have configuration file, contact the development team or follow [this process](https://my-doc.atlassian.net/wiki/spaces/MYD/pages/459702668/MyDoc+SDK+-+Client+Onboarding+Checklist)

## **Download & Install SDK**

**NOTES:** Once the configuration file is placed into your project folder, set-up your Podfile to include the Doc SDKs. As our SDK is modularized by multiple SDKs, you can install the SDK based on your requirement.

**Setting up the Podfile to use the Beta version:**

1. Click [this process](https://my-doc.atlassian.net/wiki/spaces/MYD/pages/459702668/MyDoc+SDK+-+Client+Onboarding+Checklist) to get access to your Github account. After accessing github, you can use Doc SDK Hub (hosted by Github), containing SDK beta builds
2. Include the source 'git@github.com-doc:doc/ios-sdk-hub.git' to your Podfile so that CocoaPods can download and install the beta builds of SDK.
3. Click [here](https://help.github.com/articles/generating-an-ssh-key/) to add the SSH public key to your Github account.
4. Add config in your ~/.ssh/config file. The IdentityFile should point to the SSH private key which you have added in Step 3.

For the official release, since the SDKs are all public, you don’t need to source our private Hub.

You should only install the low app size SDK that can serve your flow. For example, if you want to have the Authentication flow, you need to include pod DocAuth to your Podfile so that CocoaPods can download and install the DocAuth.

## **Initialization**

Config the *MDCore* before you access the APIs from Doc SDK. By doing this, *MDCore* initialize the configuration session and inject those configuration sessions to other SDKs

# Usage

## **DocAuth**

You can register or login your account with Doc system as the authentication APIs is in DocAuthSDK

Configure the *DocAuth* to retrieve the configuration session which is initialized by *Doccore*.

To configure DocAuth, click [here](https://my-doc.atlassian.net/wiki/spaces/MYD/pages/459605414/MyDoc+SDK+-+iOS+SDK+Integration#Initialization) and specify the MDAuth.self as an instance of the library

After configuring the DocAuth, you can then use APIs exposed by it for your flow.

* Each API returns the response structure as data
* Each API also returns the Error as info

After logging or registering into your Doc account, your token is stored in the SDK. You can use DocSDK to identify any unauthorized user.

* Calling the MDAuth.Configurator().configureToken() so that DocSDK can fetch and initialize the persisted token.

This *method* calling is mandatory for each app session so that the Doc token can be distributed to other SDKs. For example, DocConsultationSDK requires the user to log-in to the system before they can access the Consultation service,

You can use MDAuth method to check the log-in and log-out of the user account.

## **DocConsultation**

Currently, to access the consultation service of Doc, the user should have a valid token. This token can be retrieved after logging or registering into the Doc service.

Therefore, to use this SDK, you need to install the DocAuthSDK and then handle the authentication flow before accessing the APIs exposed by DocConsultationSDK.

Configure the DocConsultation to retrieve the configuration session which is initialized by DocCore.

In the process of configure, click [here](https://my-doc.atlassian.net/wiki/spaces/MYD/pages/459605414/MyDoc+SDK+-+iOS+SDK+Integration#Initialization) and specify the MDConsultation.self as an instance of the library.

After configuring the DocConsultation, you can use exposed APIs.

## **DocProfile**

The configuration of DocProfile should follow the same process as other SDK

## **DocChat**

The configuration of DocChat should follow the same process as other SDK