

Intégration SIM SDK iOS Version 2 dans un projet XCode

I. Package des Frameworks

Dans le package livré il y a des framework suivant à intégrer dans le projet XCode:

SIM SDK framework

- *SimSDK.framework*

Facebook SDK framework

- *FBSDKCoreKit.framework*
- *FBSDKLoginKit.framework*

Google SDK framework

- *GoogleOpenSource.framework*
- *GooglePlus.framework*
- *GooglePlus.bundle*

Twitter SDK framework

- *Fabric.framework*
- *TwitterCore.framework*
- *TwitterKit.framework*
- *TwitterKitResources.bundle*

II. Les frameworks de système

Intégrer les frameworks suivante dans le projet XCode:

- *Accelerate.framework*
- *Accounts.framework*
- *AddressBook.framework*
- *AssetsLibrary.framework*
- *AVFoundation.framework*
- *AudioToolbox.framework*
- *Foundation.framework*
- *CoreData.framework*
- *CoreLocation.framework*
- *CoreMedia.framework*
- *CoreMotion.framework*
- *CoreGraphics.framework*

- *CoreText.framework*
- *MediaPlayer.framework*
- *MessageUI.framework*
- *MobileCoreServices.framework*
- *Security.framework*
- *SystemConfiguration.framework*
- *UIKit.framework*

III. Configuration du projet XCode

Sélectionner le target du projet dans *Targets*, puis *Build Settings* -> *Other Linker Flags* puis saisir **-lc++ -ObjC**

Pour Facebook

S'il y a besoin d'utilisation Facebook copier les codes ci-dessous dans le fichier info plist du projet:

```
<key>CFBundleURLTypes</key>
<array>
  <dict>
    <key>CFBundleURLName</key>
    <string></string>
    <key>CFBundleURLSchemes</key>
    <array>
      <string>fb501620563319874</string>
    </array>
  </dict>
  ...
</array>
```

Attention:

1. vérifier bien si le `<key>CFBundleURLTypes</key>` existe
2. remplacer le bonne Facebook application Id

Pour Google

S'il y a besoin d'utilisation Google copier les codes ci-dessous dans le fichier info plist du projet:

```
<key>CFBundleURLTypes</key>
<array>
  <dict>
    <key>CFBundleTypeRole</key>
    <string>Editor</string>
    <key>CFBundleURLName</key>
    <string>application bundle ID</string>
```

```

        <key>CFBundleURLSchemes</key>
        <array>
            <string>application bundle ID</string>
        </array>
    </dict>
    ...
</array>

```

Attention:

1. vérifier bien si le <key>CFBundleURLTypes</key> existe
2. remplacer l'application bundle id par votre application bundle Id

Pour Twitter

S'il y a besoin d'utilisation Twitter copier les codes ci-dessous dans le fichier info plist du projet:

```

<key>Fabric</key>
<dict>
    <key>APIKey</key>
    <string>api-key</string>
    <key>Kits</key>
    <array>
        <dict>
            <key>KitInfo</key>
            <dict/>
            <key>KitName</key>
            <string>Crashlytics</string>
        </dict>
    </array>
</dict>

```

IV. Utilisation du SIM SDK dans le projet XCode

l'utilisation du SIM SDK est très simple, vous pouvez voir le projet d'exemple livré avec le SIM SDK

1. Pour utiliser le SIM SDK dans une classe importer le SDK SDK dans la classe:

```
#import <SimSDK/SimSDK.h>
```

2. Initialisation du SIM SDK dans la classe Delegate du application

```

- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {

```

```

// Init callback block
R5InitFinishedCallback callBack = ^(R5Exception * error)
{
    if (error)
    {
        // Error handle
    }
    else
    {
        // SIM SDK Init with success handle!";
    }
};

//Init SIM SDK
[R5Client initNativeSDKWithDomain:@"reach5.og4.me"
    apiKey:@"BvnynWPdIUOGwC3yERXi"
    providers:nil
    onInitFinish:callBack];

//Dispatch l'événement
[R5Client application:application didFinishLaunchingWithOptions:launchOptions];

...
return YES;
}

- (void)applicationDidBecomeActive:(UIApplication *)application {
    //Dispatch l'événement
    [R5Client didBecomeActive];
}

- (BOOL)application:(UIApplication *)application openURL:(NSURL *)url
sourceApplication:(NSString *)sourceApplication annotation:(id)annotation {
    //Dispatch l'événement
    return [R5Client application:application
        openURL:url
        sourceApplication:sourceApplication
        annotation:annotation];
}

```

3. pour appeler le login d'un provider

voir le projet d'exemple pour plus détail.

```

//Login callback block
R5LoginViewControllCallback callBack = ^(R5LoginResponse *response, R5Exception *error)
{
    if (response)

```

```

{
    // Login sucessfull handle
}
else
{
    // Login FAIL handle
}
};

```

//Login with a provider (R5Facebook|R5Google|R5Twitter|R5Paypal)
[R5Client nLoginWithProvider:<provider> origin:<origin> onLoginResponse:callback];

4. pour appeler la liste des logins

voir le projet d'exemple pour plus détail.

```

//Login callback block
R5LoginViewControlleCallback callback = ^(R5LoginResponse *response, R5Exception *error)
{
    if (response)
    {
        // Login sucessfull handle
    }
    else
    {
        // Login FAIL handle
    }
};

//list of login buttons
[R5Client nDisplayLoginButtonsWithOrigin:<origin> onView:<viewController>
onLoginResponse:callback];

```

V. Modifier les labels

pour modifier des label dans le SIM SDK (le Close et les Labels des logins buttons) il faut définir des clés suivant dans la fichier *Localizable.strings* :

"Close"="Close";

"Login with Facebook"	= "Login with Facebook";
"Login with Google Plus"	= "Login with Google Plus";
"Login with Twitter"	= "Login with Twitter";
"Login with Paypal"	= "Login with Paypal";