Embed iOS SDK in to a workforce marketplace app

Objective:

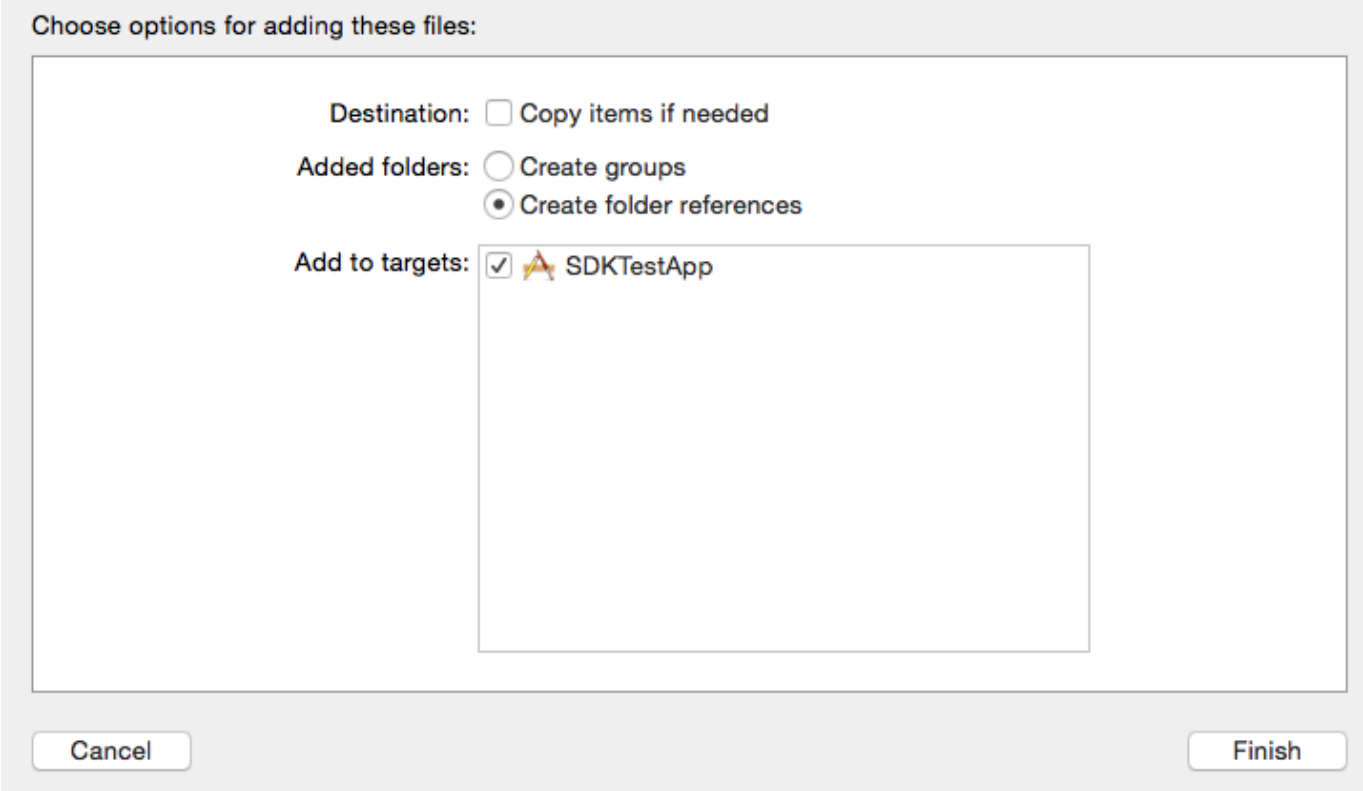
To embed smart messaging in to an existing iOS app for workforce marketplace

Implementation:

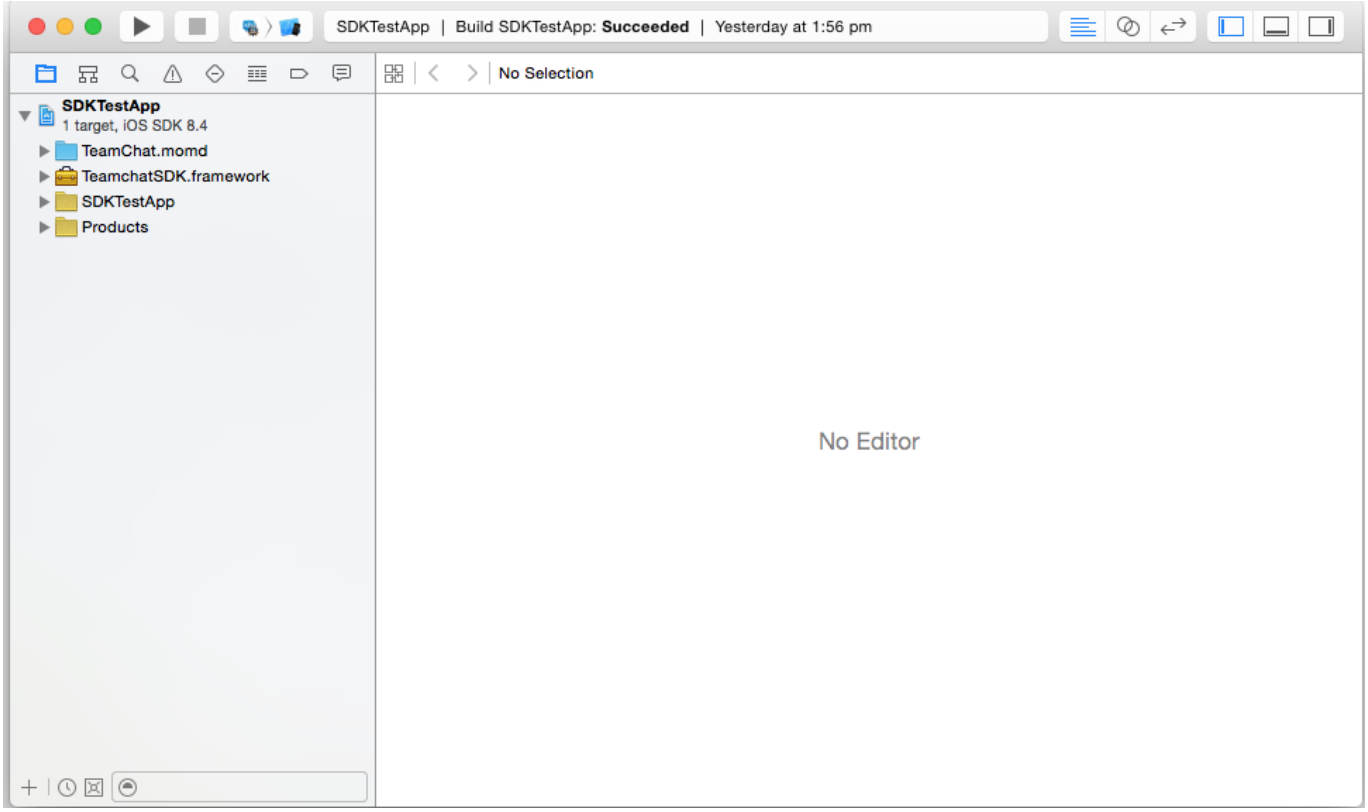
1. Add TeamchatSDK.framework to your project.

2. Add TeamchatSDK.framework to 'Embedded Binaries' field under 'General' tab of your project target.

3. Add TeamchatSDK.framework/TeamChat.momd folder to your project using the "Create folder references" option as shown below.



Now project navigator will look like:



4. To support some HTTP requests in iOS 9.0 and above, add the key 'NSAppTransportSecurity' as a dictionary in Info.plist of your project and within the dictionary add the key 'NSAllowsArbitraryLoads' with BOOL value 'YES'.

Now you need to initialize the Teamchat object with appID in the AppDelegate method *application:didFinishLaunchingWithOptions:.*

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

        [Teamchat initializeWithAppID:<App ID> withCompletionHanlder:^(BOOL success,
NSError *error) {
                if (!success)
                {
                        [[[UIAlertView alloc] initWithTitle:@"Failed"
                        message:[error localizedDescription] delegate:nil
                        cancelButtonTitle:@"OK" otherButtonTitles:nil, nil] show];
                }
        }];

        return YES;
}
```

Enable remote notifications by setting deviceID in the AppDelegate method *application:didRegisterForRemoteNotificationsWithDeviceToken:*

```
- (void)application:(UIApplication *)application
didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken {
```

```
        [Teamchat setRemoteNotificationsDeviceToken:deviceTokenString];
}
```

You can login to Teamchat in two ways:

1. Set Authentication code, emailID and userID.

```
[Teamchat setAuthenticationCode:@"<Auth-code>"];
[Teamchat setUserEmailID:@"<Email-ID>"];
[Teamchat setUserID:@"<User-ID>"];
```

2. Use Teamchat login view controller.

```
UIViewController *loginViewController = [Teamchat
teamchatLoginViewController:^(BOOL success, NSError *error, NSString *errMsg) {
    if (success)
    {
        // Successful login
    }
    else
    {
        // Login error
    }
}];

if (loginViewController)
{
    [self presentViewController:loginViewController animated:YES completion:nil];
}
```

If you want to configure UI of the Teamchat screens, you can set them as follows:

```
[Teamchat setNavigationTitle:@"Chats"];
[Teamchat setNavigationTitleColor:[UIColor greenColor]];
[Teamchat setTableViewSeparatorColor:[UIColor redColor]];
```

To get the chat groups list controller, you can use the following API:

```
[Teamchat initWithCompletionHandler:^(BOOL success, NSError *error, NSString
*errMsg) {
    if (success)
    {
        UIViewController *teamchatGroupsViewController = [Teamchat

TeamchatGroupsViewController];
        if (teamchatGroupsViewController)
        {
            [self presentViewController:teamchatGroupsViewController animated:YES
completion:nil];
```

```
        }
        else
        {
            // Error
        }
    }
    else
    {
        // Error
    }
}];
```

+(void)initWithCompletionHandler:(^(BOOL, NSError*,NSString*))handler method should be called before calling groups list API.

To get the chat window controller, you can use the following API:

```
[Teamchat initWithCompletionHandler:^(BOOL success, NSError *error, NSString
*errMsg) {
    if (success)
    {
        NSError *error;
        TeamchatGroup *group = [Teamchat teamchatGroupWithID:@"<GroupID>"
error:&error];

        UIViewController *groupViewController = [Teamchat

createTeamchatWindowViewControllerWithGroup:group];
        if (groupViewController)
        {
            [self presentViewController:groupViewController animated:YES
completion:nil];
        }
        else
        {
            // Error
        }
    }
    else
    {
        // Error
    }
}];
```

+(void)initWithCompletionHandler:(^(BOOL, NSError*,NSString*))handler method should be called before calling chat window controller API.

To get user profile view controller, you can use the following API:

```
[Teamchat initWithCompletionHandler:^(BOOL success, NSError *error, NSString
*errMsg) {
    if (success)
```

```
    {
        UIViewController *userProfileViewController = [Teamchat
userProfileViewController];

        if (userProfileViewController)
        {
            [self presentViewController:userProfileViewController animated:YES
completion:nil];
        }
        else
        {
            // Error
        }
    }
    else
    {
        // Error
    }
}];
```

+(void)initWithCompletionHandler:(^(BOOL, NSError*,NSString*))handler method should be called before calling chat window controller API.

To get the public groups view controller, you can use the following API:

```
[Teamchat initWithCompletionHandler:^(BOOL success, NSError *error, NSString
*errMsg) {
    if (success)
    {
        UIViewController *publicGroupsViewController = [Teamchat
publicGroupsViewController:^(TeamchatGroup *selectedGroup) {
            if (selectedGroup)
            {
                // Push the selected group's chat window.
                UIViewController *chatWindowViewController = [Teamchat

createTeamchatWindowViewControllerWithGroup:selectedGroup];
                if (chatWindowViewController)
                {
                    [self.navigationController
pushViewController:chatWindowViewController animated:YES];
                }
                else
                {
                    // Error
                }
            }
        }];

        if (publicGroupsViewController)
        {
            [self presentViewController:publicGroupsViewController animated:YES
completion:nil];
        }
    }
```

```
    else
    {
        // Error
    }
}];
```

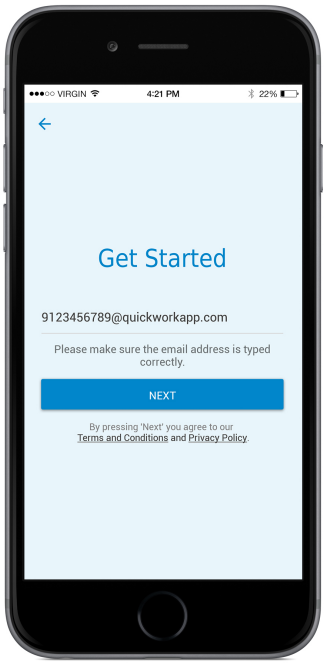
+(void)initWithCompletionHandler:(^(BOOL, NSError*,NSString*))handler method should be called before calling public group API.

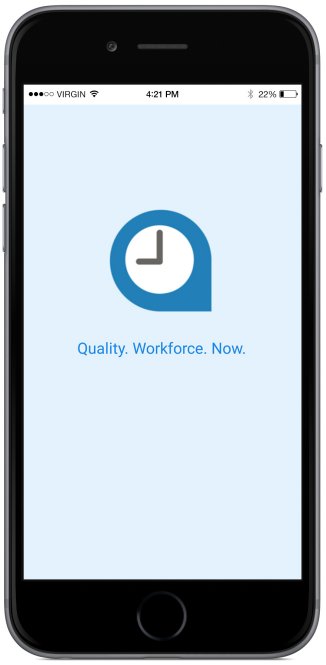
To get Teamchat settings view controller, you can use the following API:

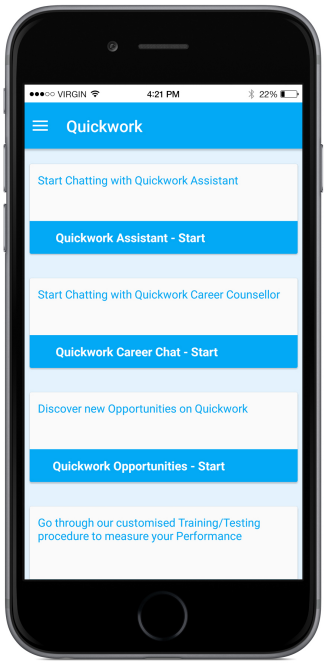
```
[Teamchat initWithCompletionHandler:^(BOOL success, NSError *error, NSString
*errMsg) {
    if (success)
    {
        UIViewController *settingsViewController = [Teamchat
settingsViewController];

        if (settingsViewController)
        {
            [self presentViewController:settingsViewController animated:YES
completion:nil];
        }
        else
        {
            // Error
        }
    }
    else
    {
        // Error
    }
}];
```

+(void)initWithCompletionHandler:(^(BOOL, NSError*,NSString*))handler method should be called before calling public group API.


●●●○○ VIRGIN
4:21 PM
22%
Get Started
9123456789@quickworkapp.com
Please make sure the email address is typed correctly.
NEXT
By pressing 'Next' you agree to our
Terms and Conditions and Privacy Policy.


●●●○○ VIRGIN
4:21 PM
22%
Quality. Workforce. Now.


●●●○○ VIRGIN
4:21 PM
22%
Quickwork
Start Chatting with Quickwork Assistant
Quickwork Assistant - Start
Start Chatting with Quickwork Career Counsellor
Quickwork Career Chat - Start
Discover new Opportunities on Quickwork
Quickwork Opportunities - Start
Go through our customised Training/Testing procedure to measure your Performance