OutSystems における Push 通知の実装方法

tdi 情報技術開発株式会社

スマートフォンやタブレット端末などのモバイル端末で利用されている Push 通知は、Push 通知サービスを提供しているサイトと連携することで OutSystems でも実現することができます。本書では、Push 通知サービスを OutSystems でどのように実装して実現するか、その方法を解説します。

内容

Push 通知とは

Push 通知とは、ユーザのモバイル端末に直接メッセージを送信し、ロック画面やモバイル端末の上部に表示させることができる機能です。Push 通知の代表例として、SNS メッセージの受信通知、ニュースサイトからの速報通知、カレンダー登録した予定が近づいたときに表示されるメッセージなどがあります。

OutSystems でもこの Push 通知を実装することができます。本書では OneSignal*1 を利用した Push 通知の実装について解説します。

*1：OneSignal：本書で紹介しているような Push 通知や Web メッセージ配信などのメッセンジャーサービスを SaaS で提供されるサービス。

OutSystems における Push 通知アプリの実装

OutSystems で Push 通知アプリを作成する手順は以下となります。

1．OutSystems のアプリを作成する

2．ネイティブアプリを作成するために証明書を発行する

3．OneSignal の AppID と RestAPIKey を取得する

4．Push 通知受信用のネイティブアプリを生成する

では、この手順に沿ってアプリを作成します。

1.　OutSystems のアプリ作成

最初に OutSystems でアプリケーションを作成します。ここでは Push 通知に必要な OneSignal の設定を中心に解説しますので、OutSystems の細かい設定の解説は省略します。本書では、一番簡単な Push 通知の方法について解説しますので、特定のユーザやデバイスにではなく、通知可能な端末すべてに対して Push 通知を行う方法となっています。

１．１．アプリ作成の準備

OutSystems の実装作業を行う前に Forge*2 から OneSignalPlugin をインストールします。

図 1 Forge の OneSignalPlugin

*2：Forge：OutSystems のコミュニティメンバーが開発した OutSystems 用のソフトウェアを他のコミュニティメンバーに簡単に共有できるオンラインリポジトリです。

１．２．メッセージ送信機能の作成

Push 通知のメッセージを送信する機能を作成します。メッセージは Web アプリの画面から送信することにしますので、Push 通知のメッセージを送信するアプリは「Reactive Web App」を選択します。新しいアプリの画面が開いたら、Management Dependencies で OneSignalAPI ＞ Server Action ＞ SendPushNotificationToAll を選択します。

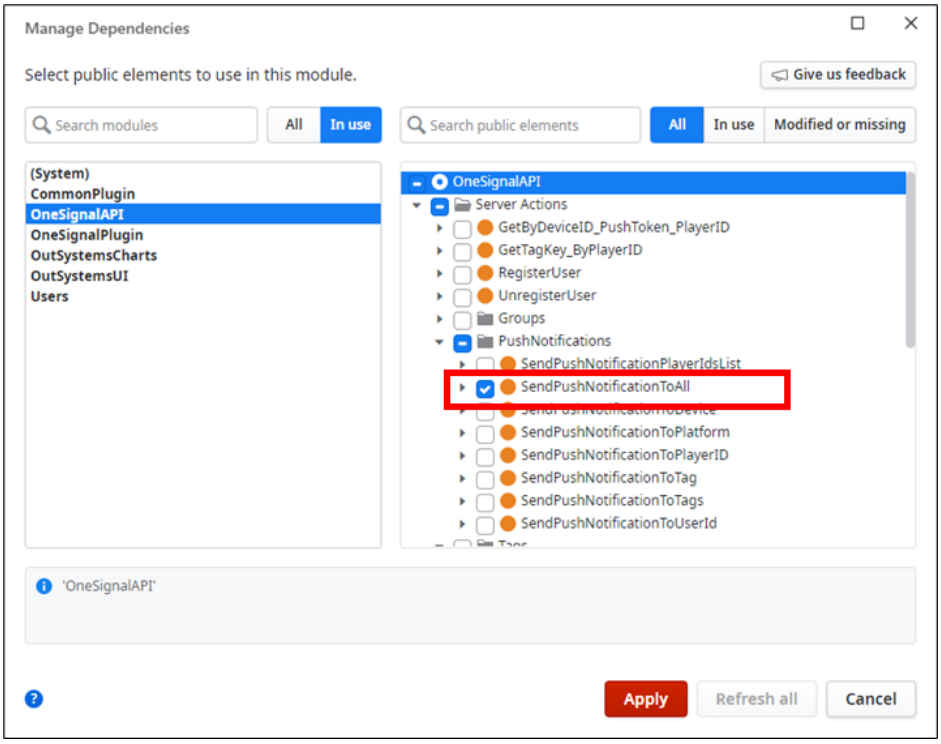

図 2 Management Dependencies で SendPushNotificationToAll を選択した画面

次にメッセージを入力するための画面を作成します。画面については、Push 通知のメッセージを送信する場合でも特別な実装は行いません。メッセージを入力する Input ウィジェットとメッセージを送信するたに Action を実行するためのボタンがあれば OK です。図 3 が画面イメージになります。

図 3 メッセージ送信画面イメージ

続いてロジックを作成します。ロジックはボタンから呼び出される Screen Action と Screen Action から呼び出される Server Action の 2 つを作成します。

まずは Screen Action です。Screen Action は、メッセージの入力チェックとメッセージを送信するための Server Action の呼び出し、そしてメッセージ送信後にエラーが返されていないか、をチェックしています。ロジックは図 4 のとおりです。図では確認できませんが、Server Action を呼び出すときに画面に入力されたメッセージを引数として渡しています。

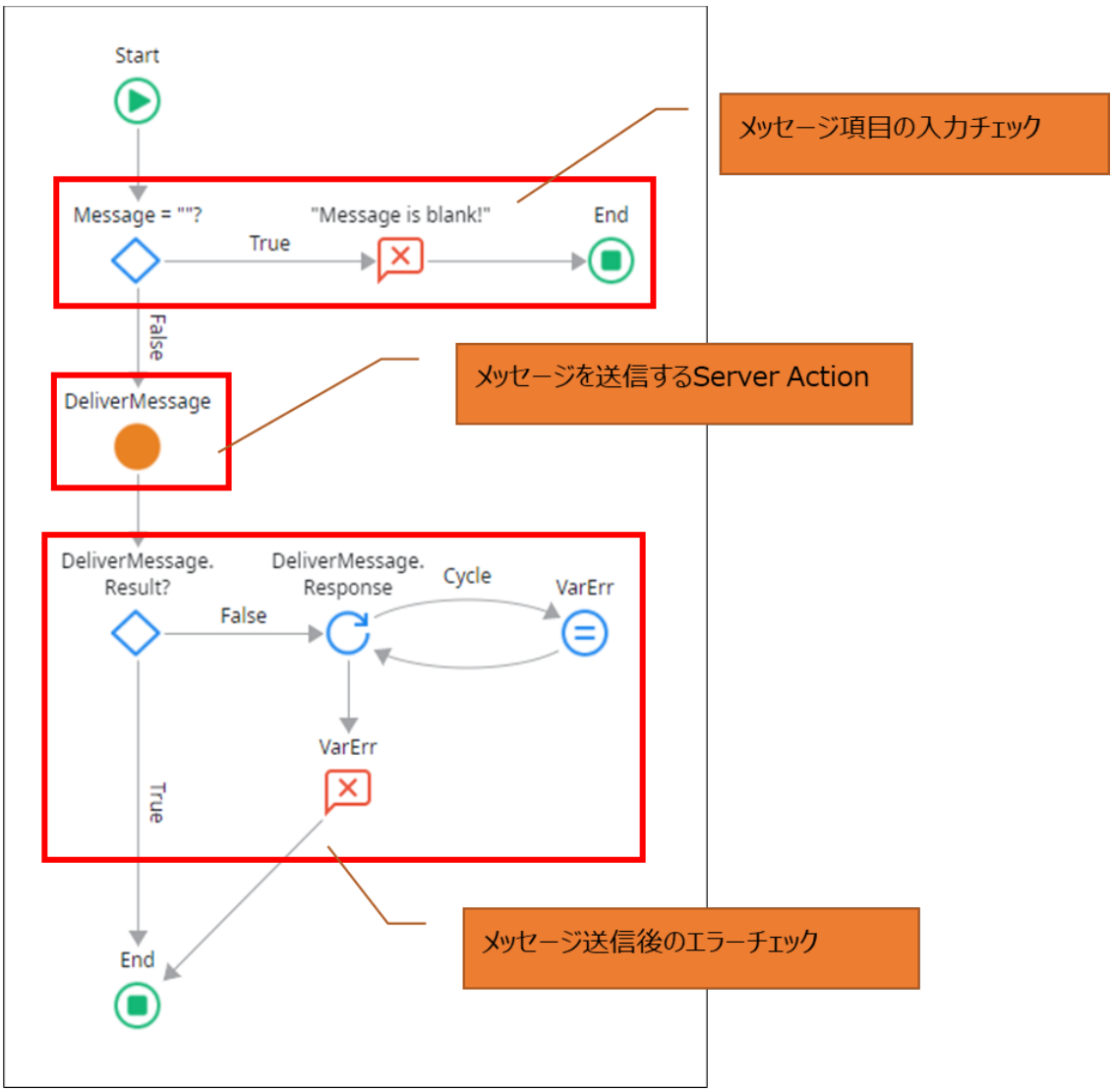

図 4 メッセージ送信の Screen Action

Server Action のロジックは図 5 のようになります。Server Action では実際に Push 通知を行うために利用する OneSignal へ必要な項目をセットしサービスの呼び出しを行っています。

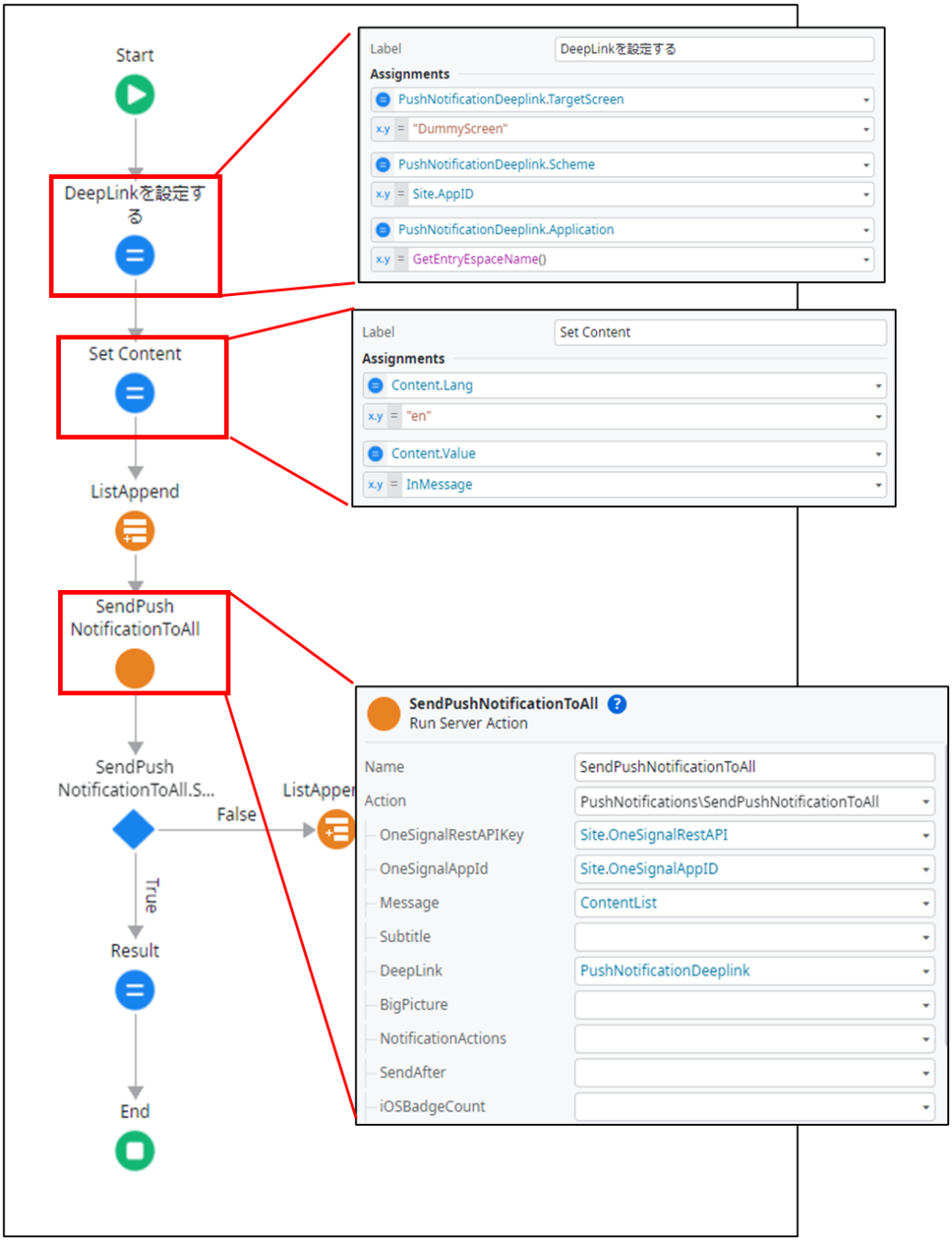

図 5 メッセージ送信の Server Action

Push 通知のメッセージを送信する処理は以上となります。

※このフローでは DeepLink*3 を設定してメッセージがタップされたら”DummyScreen”という名称のページに遷移するようにしていますが、DeepLink は設定しなくても問題はありません。

※OneSignal の RestAPIKey と AppID は、値を変更しても Publish しなくて済むようにサイトプロパティに定義しているので図 5 のようなコードになります。

*3：DeepLink：アプリを特定の画面に遷移させるリンクのこと。Push 通知のメッセージをタップした場合、特定の画面に誘導したい場合に設定します。

１． ３． メッセージ受信アプリの作成

続けてメッセージを受信するアプリを作成します。メッセージ受信アプリはモバイル端末にインストールして利用しますので、ネイティブアプリが生成できるように作成する必要があります。そのためアプリを作成する際は、「Phone App」、または「Tablet App」を選択してください。続いてモジュールを作成し、Management Dependencies を開いてインストールした Forge：OneSignalPlugin を参照し、Push 通知に必要な部品を参照します。

まず、OneSignalPlugin の UI Flow から OneSignal、Client Action から Register を選択します。

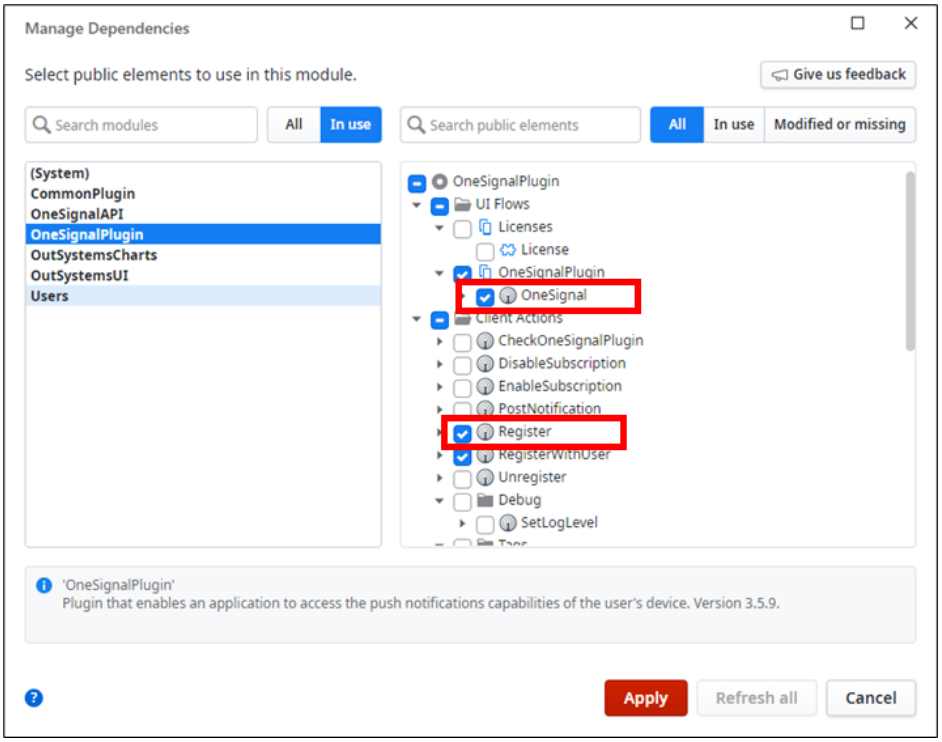

図 6 Management Dependeicies で OneSignal と Register を選択したところ

OneSignal の参照設定が完了したら、アプリケーション起動時に OneSignal をロードするための設定を行います。モジュールを作成すると自動的に作成される Web Block の Layout を開き、そこに OneSignalPlugin の UI Flow の OneSignal を配置します。その際、コンテナを作成し、 OneSignal Wrapper と名前を付け、そこに配置すると構造がわかりやすくなります。画面デザインは図 7 のようになります。

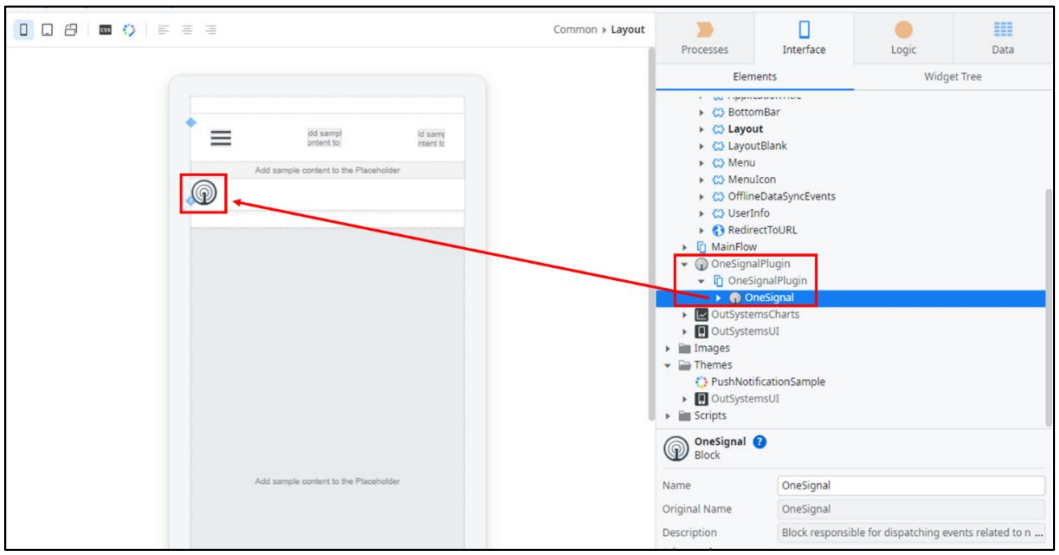

図 7 UI Flow の OneSignal の配置例

Widget Tree では図 8 のような配置になります。

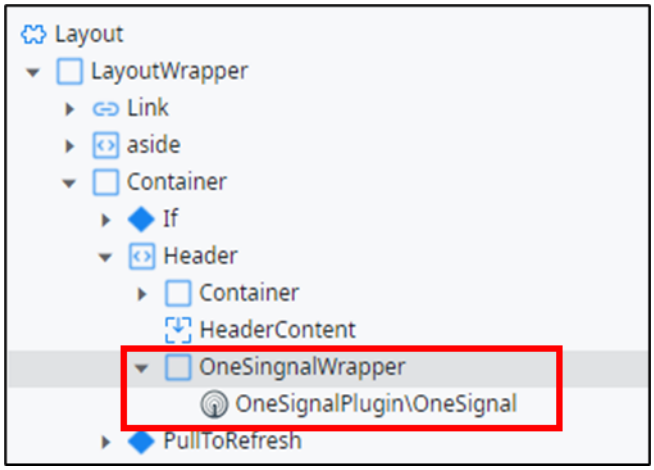

図 8 Widget Tree での実装例

メッセージ受信アプリを起動したときに OneSignal 側に App ユーザを登録する OneSignalPlugin の Client Action：Register を配置します。Register は Client Action の OnApplicationReady で実行するようにします。OnApplicationReady は通常デフォルトでは定義されないので、Logic タブの Client Action で右クリック→Add System Event を選択することで定義することができます。

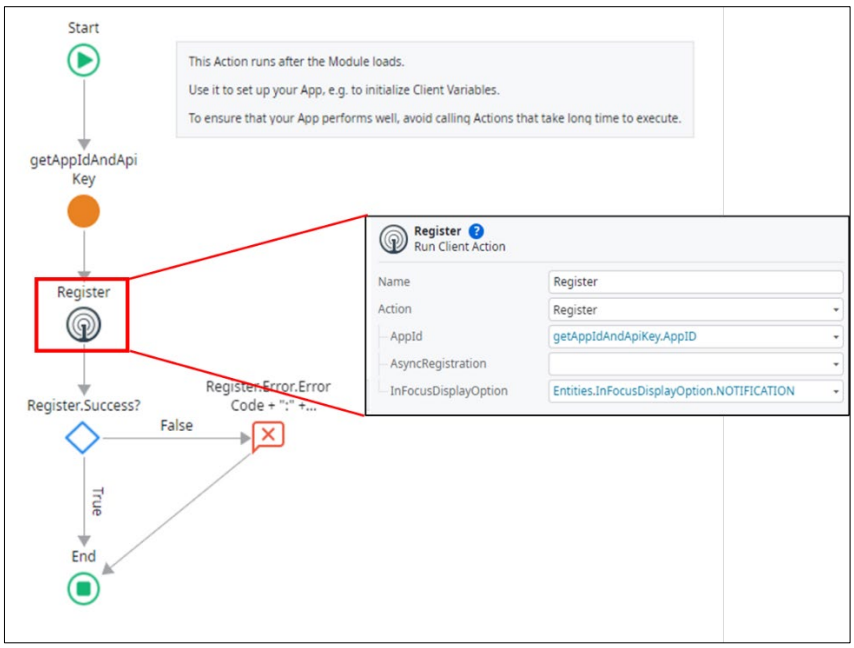

図 9 OnApplicationReady の Action フロー

これで OutSystems のロジックの実装は完了です。この後は、実際に Push 通知のサービスを利用するために必要な証明書の発行、OneSignal の Push 通知サービス利用のための設定を行います。OneSignal の設定を行うことで、RestAPI、AppID を取得できます。RestAPI、AppID 値の取得方法は、「3．OneSignal の AppID と RestAPIKey の取得」の章で説明します。

2. ネイティブアプリ作成のための証明書発行

OutSystems でネイティブアプリの作成を行うためには、いくつか証明書が必要になります。なお、Android では通常証明書がなくてもネイティブアプリは作成できますが、今回のように外部サービスとのやりとりを行う Push 通知では必要になります。

2．1．Android の場合

本書でご紹介している Push 通知サービスである OneSignal を利用する場合には Android でも証明書が必要になります。

Android の証明書（Keystore）の発行は、Java の keytool コマンドを使って発行できます。証明書の発行方法は、OutSystems のサイトに記載がありますので、そちらを参照ください。

【Keystore を発行する参考 URL】

https://success.outsystems.com/ja-jp/documentation/11/delivering_mobile_apps/generate_and_distribute_your_mobile_app/more_information_on_generating_and_distributing_mobile_apps/

2．2．iOS の場合

Apple Development Program（以下 ADP）で証明書を発行します。OutSystems で作成するネイティブアプリに必要なものは.p12 ファイルとプロビジョニングプロファイルの 2 つになります。作成の仕方については、本書では省略しますが、作成にあたって参考になるウェブサイトをご紹介します。

【AppStore 用証明書の作成の参考 URL】

https://zenn.dev/iwatos/articles/b131cf60d20131513bcb

3. OneSignal の AppID と RestAPIKey の取得

OutSystems で OneSignal のサービスを利用するためには、OneSignal の AppID と RestAPIKey が必要です。同サービスを利用するにあたって必要な情報は OS ごとに異なります。本章では Android、iOS で必要な情報について記載します。なお、OneSignal のサイトはすべて英語表記ですが、操作自体は難しいものではありません。

まず、OneSignal のアプリケーション名と組織を入力します。入力値は、図 10 にあるようにアプリケーション名は任意、組織はデフォルト値で問題ありません。次に Push 通知を利用する端末の OS を選択します。ここでは Android、iOS のいずれかを選択する必要があります。OneSignal のアプリ登録は複数の OS に対応していますので、Android→iOS、iOS→Android、のようにどちらから先に設定しても問題ありません。

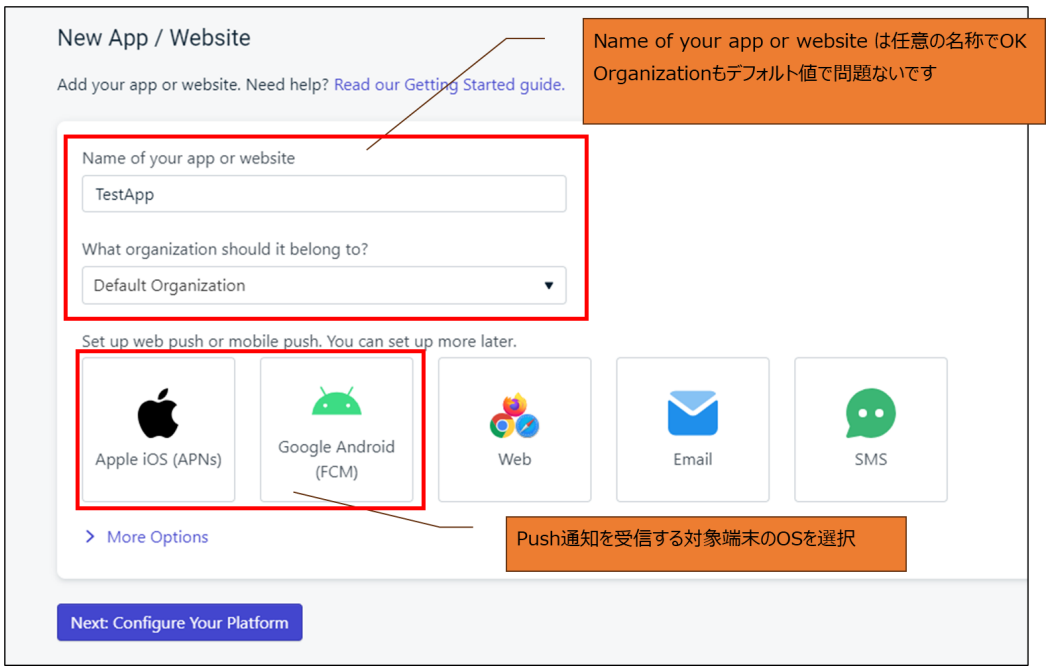

図 10 OneSignal の App 登録画面

3．1．Android の設定

Android 用のコンフィギュレーション画面です。

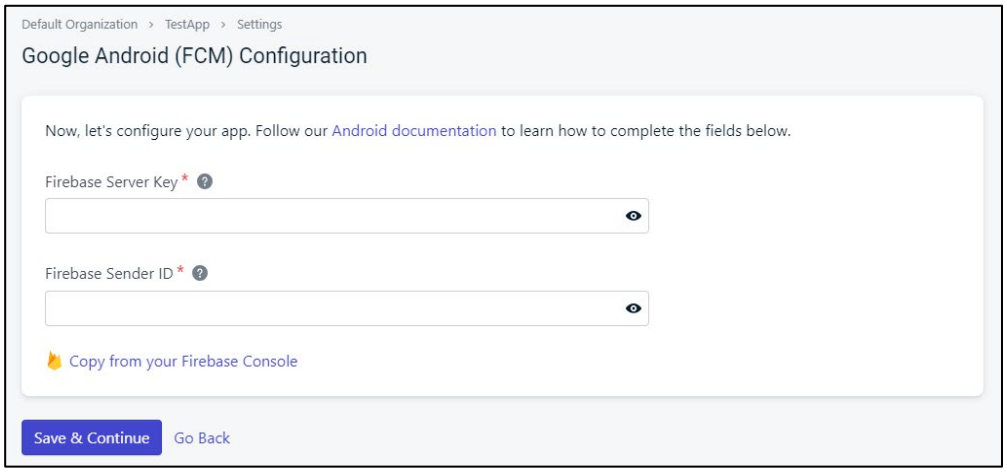

図 11 Android 用のコンフィギュレーション画面

図 11 の通り、Android では Firebase*4 の Server Key と Sender ID の 2 つです。Firebase にて Cloud Messaging サービスを作成すると取得することができます。

Firebase における Server Key と Sender ID は、図 12 の通りです。

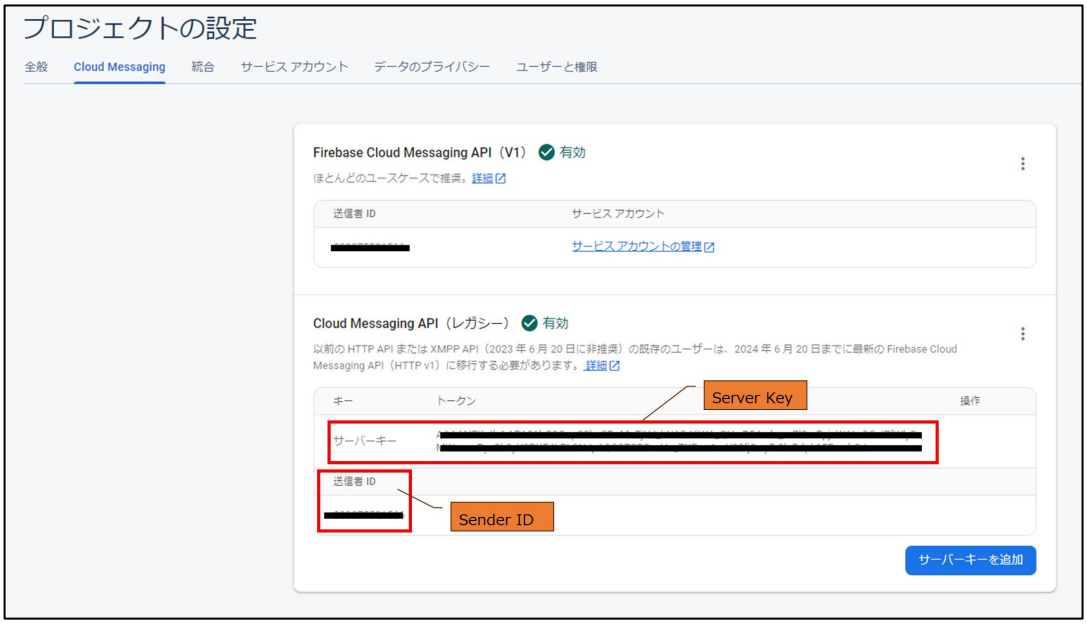

図 12 Firebase の Cloud Messaging 画面

*4：Firebase：Google 社が提供するモバイル・Web アプリを開発するためのプラットフォームで、MBaaS（Mobile Backend as a Service）/BaaS（Backend as a Service）に分類されるクラウドサービスです。

3．2．iOS の設定

iOS 用のコンフィグレーション画面とその設定値です。

図 13 iOS 用コンフィグレーション画面

OneSignal の設定で必要な、ADP 画面で確認できる ID は、図 14 の通りです。

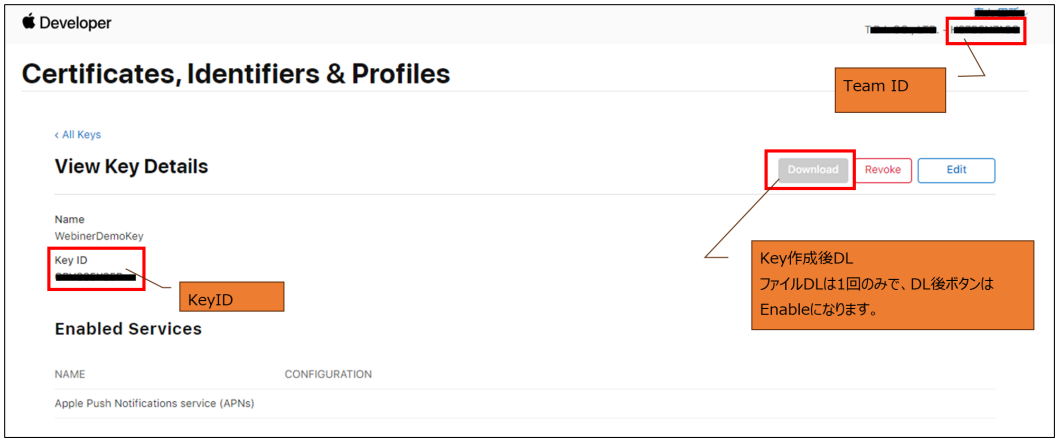

図 14 ADP の Key 詳細画面

OneSignal での認証形式は、以前は.p12 形式のみ（OneSignal のサイトで発行する.p12 ファイルは ADP で発行する.p12 ファイルとは別物になります）でしたが、最近は.p8 形式に対応し、また図 13 でも確認できるように.p8 形式が Recommend されています。そのため今回は、.p8 による認証を利用しています。.p8 形式のファイルは、図 14 にもあるように ADP の Certificates, Identifiers & Profiles ＞ Keys にて作成できますが、ファイルのダウンロードは Key 作成後 1 回のみとなるので、注意が必要です。

App Bundle ID は、OutSystems の Service Studio のネイティブアプリを作成する画面で App Identifier 欄で確認できますので、そこから取得すると便利です（４．Push 通知受信用のネイティブアプリを生成する、を参照）。

３．３．OneSignal での AppID と RestAPIKey の取得

OneSignal のアプリ設定が完了すると AppID と RestAPIKey が発行され画面で取得することができ、この内容はいつでも確認することができます。

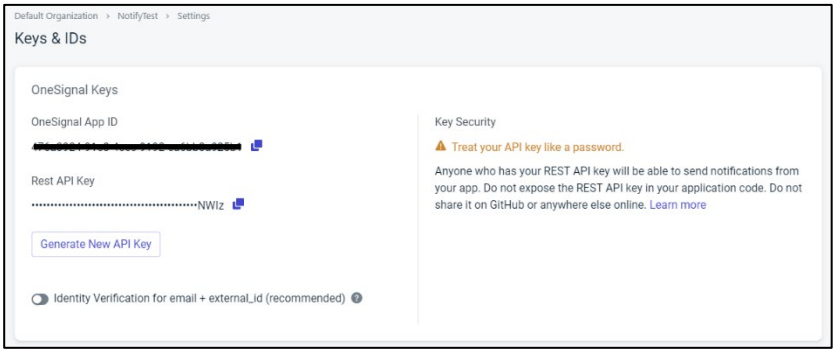

図 15 OneSignal の AppID と RestAPIKey 確認画面

OutSystems の設定を修正します。OneSignal で取得した AppID と RestAPIKey を OutSystems のサイトプロパティにセットします（サイトプロパティは Service Center で行うとアプリを再 Publish する必要がありません）。これで実装は完了です。

4. Push 通知受信用のネイティブアプリの生成

OutSystems の Service Studio に戻り、Push 通知受信用するためのネイティブアプリを生成します。Service Studio ではアプリケーションのトップ画面の Distribute タブでネイティブアプリの生成を行います。

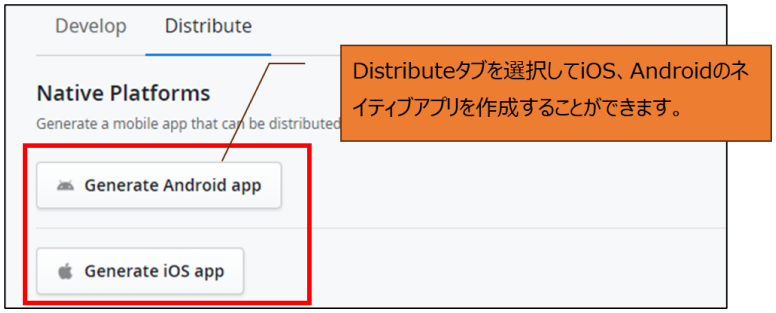

図 16 OutSystems・Service Studio の Distribute タブ

4．1．Android のネイティブアプリの生成

Generate Android app ボタンを押し、図 17 の画面を表示して、必要な情報を入力し、Generate Android app ボタンを押してネイティブアプリを生成します。Android の場合は、Keystore 情報が不要な場合があります。

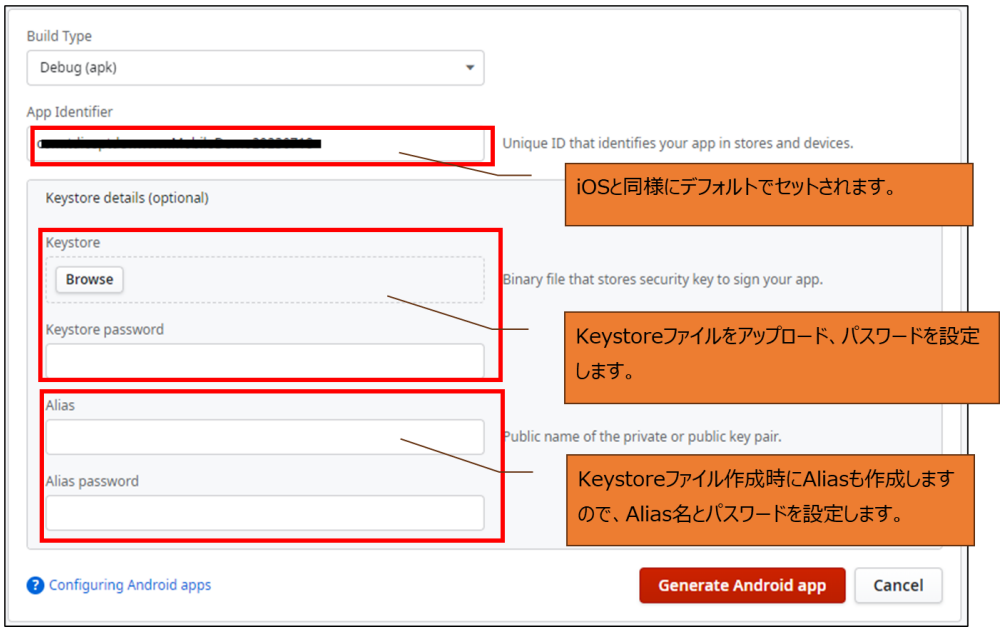

図 17 Generate Android app 画面

4．2．iOS のネイティブアプリの生成

Generate iOS App ボタンを押し、以下の図 18 の画面を表示し、必要な情報を入力し Generate iOS app ボタンを押しネイティブアプリを生成します。iOS の場合は、すべての項目に対して入力が必須となっています。

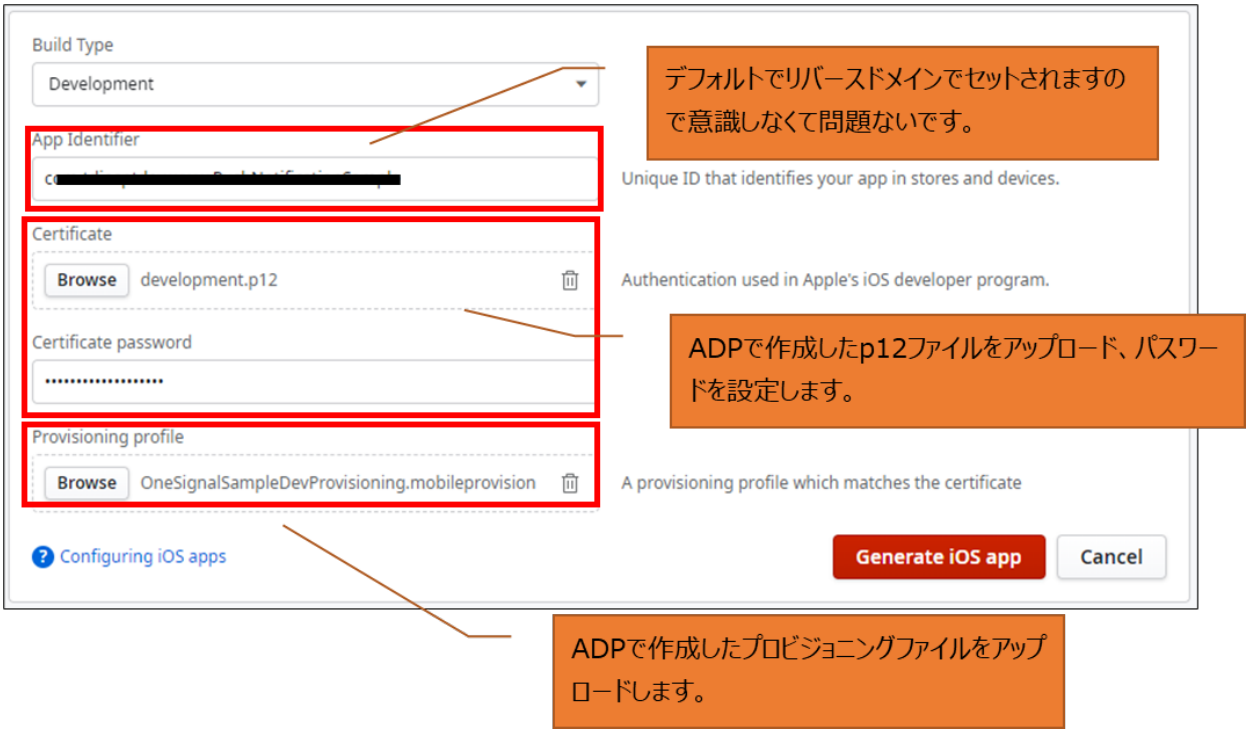

図 18 Generate iOS App 画面

5.　Push 通知アプリの実行

では実際に 4．で生成したネイティブアプリをデバイスにインストールして、実行してみます。画面からメッセージを入力し、メッセージ送信ボタンを押します。Push 通知のメッセージは iOS で受信した結果です。

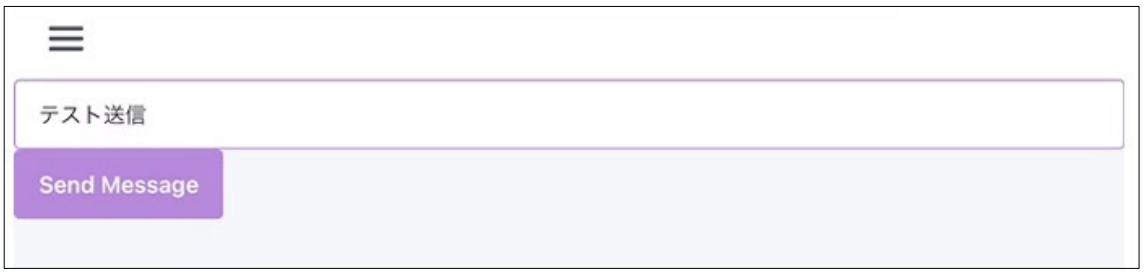

図 19 メッセージ送信画面でメッセージを入力

図 20 受信した Push 通知のメッセージ

終わりに

OutSystems における Push 通知の実装方法及びその注意点について解説いたしました。OutSystems を利用して Push 通知が実装できることがお判りいただけたと思います。

tdi では、 OutSystems の機能や技術について十分な知識を持った多くの技術者を有しており、資格保有者数は国内トップクラスです。ローコード開発が一般的に注目される以前（2016 年）から重ねた OutSystems 開発の実績をもとに、IT 戦略コンサルティングや OutSystems 導入から運用までをトータルサポートします。また、お客様に合わせた人財育成や内製化もご支援いたします。本件を含め、何かお困りごとがございましたらどうぞお気軽にお問合せください。

【ローコード開発基盤「OutSystems」】

https://www.tdi.co.jp/outsystems/

【お問い合わせ】

https://tdi.smktg.jp/public/application/add/1095

情報技術開発株式会社
営業本部

東京:〒163-1332 東京都新宿区西新宿六丁目 5 番 1 号　新宿アイランドタワー32 階

TEL：03-5325-4811 (代表)　FAX 03-5325-4812

中部:〒451-6027 愛知県名古屋市西区牛島町 6 番 1 号 名古屋ルーセントタワー27 階

TEL 052-571-6871（代表）　FAX 052-571-3856

関西:〒530-0005 大阪府大阪市北区中之島二丁目 2 番 7 号 中之島セントラルタワー20 階

TEL.06-6201-7739(代表) FAX.06-6201-7740

九州:〒812-0013 福岡県福岡市博多区博多駅東二丁目 10 番 1 号 福岡ビル S 館 7 階

TEL.092-451-8218(代表) FAX.092-474-7379