

アンドロイド用 **BeaconPlus SDK** ドキュメンテーション

バージョン **1.1**

2014 年 11 月

PinMicro 株式会社 機密・専有

配布制限：このドキュメントは PinMicro 株式会社の財産である。PinMicro 株式会社の書面による事前の承諾なしに、文書の全部あるいは一部を使用すること、複製すること、複写すること、あるいは第三者にいかなる方法によっても開示することは禁じられている。

質問がある場合には、当社 (support@pinmicro.com) までご連絡ください。

改訂履歴

日時	バージョン	内容	担当者
2014年8月13日	0.4	初版	Jithin
2014年8月22日	1.0	レビューポイント反映	Jithin
2014年11月3日	1.1	エラーコード更新	Jithin

レビュー詳細

日時	バージョン	担当者	レビュー担当者
2014年8月13日	0.4	Jithin	Nidheesh
2014年8月22日	1.0	Jithin	Nidheesh
2014年11月3日	1.1	Jithin	Nidheesh

目次

1	はじめに	4
1.1	本文書の目的	4
1.2	本文書のスコープ	4
2	アクリニム	4
3	BeaconPlus SDK 概要.....	4
4	SDK のセットアップ	5
4.1	要件	5
4.2	BeaconPlus SDK を Eclipse プロジェクトに統合する方法.....	5
5	SDK の使用方法.....	9
6	アクセストークンによる認証.....	11
6.1	アクセストークンの入手.....	11
7	コールフロー・ダイアグラム.....	13
8	エラーコード	14
9	文献	15

1 はじめに

1.1 本文書の目的

本 BeaconPlus SDK ドキュメントは、BeaconPlus SDK およびこれを Android のアプリに統合する方法に関して、その概要を説明するものです。詳細な API の仕様およびその他の実装方法の詳細については、[BeaconPlus API spec \(http://pinmicro.com/docs/android-sdk/index.html\)](http://pinmicro.com/docs/android-sdk/index.html) をご参照ください。

1.2 本文書のスコープ

Android 用コンテキスト・アウェアネス（文脈認識）アプリケーションの開発者は、まず最初に本文書をお読みください。

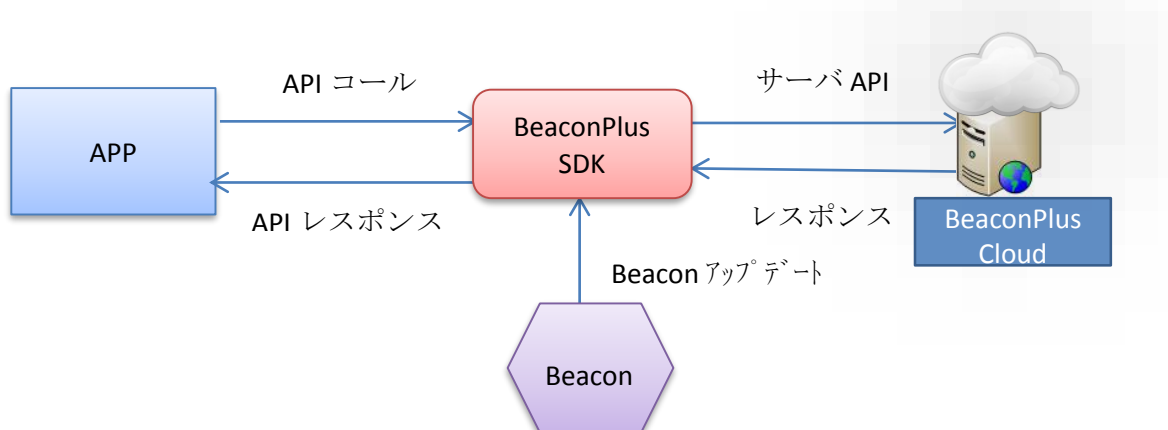
2 アクロニム

本文書では以下のアクリニムを使用します。

アクリニム	詳細
SDK	ソフトウェア開発キット（Software Development Kit）
API	アプリケーションプログラムインターフェース（Application Programming Interface）
BLE	Bluetooth Low Energy

3 BeaconPlus SDK 概要

BeaconPlus SDK により、幅広い自由度をもってコンテキスト・アウェアネスのアプリケーションを開発することができます。BeaconPlus SDK は、クライアントアプリと BeaconPlus クラウドサービス間の橋渡しとしての役割を持ちます。



SDK は定期的クラウドと同期しており、プロジェクトに対して適切なタイミングでの正確な情報を保持しています。またすべての内部ロジックを処理してデバイスを検出し、特定の領域で「エントリー」と「エクシット」の事象が発生するたびにアプリに通知します。

4 SDK のセットアップ

本節では、Android のアプリを利用するための SDK セットアップ方法を説明します。

4.1 要件

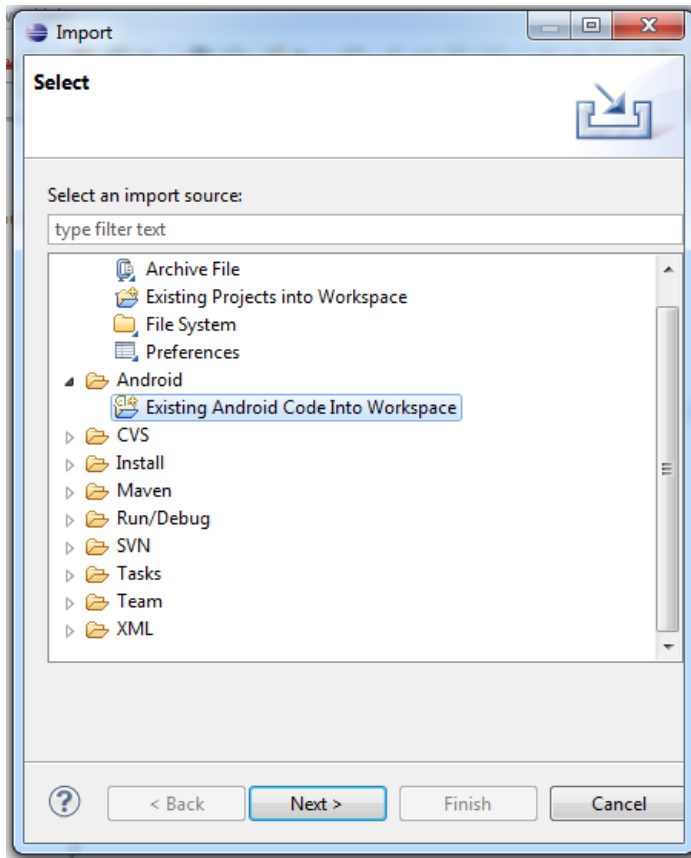
BeaconPlus SDK は BLE プロトコルを利用してビーコンを特定するので、最低限 Android のバージョン 4.3 が必要です。ハードウェアの仕様としては、SDK が以下のデバイス上だけで動作していることが必要です。

- ブルートゥース 4.0 による Android 電話
- ブルートゥース 4.0 による Android タブレット

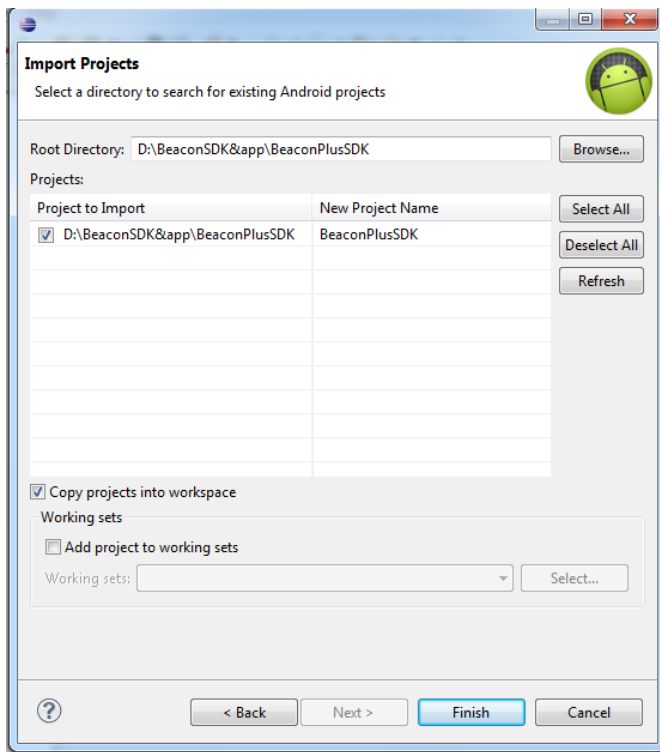
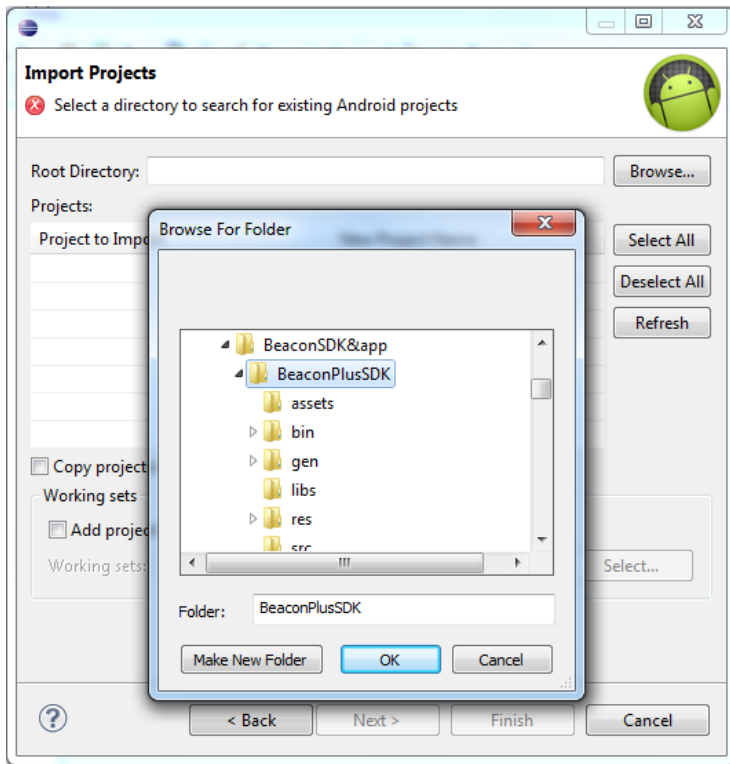
4.2 BeaconPlus SDK を Eclipse プロジェクトに統合する方法

お手元の Android アプリケーションに BeaconPlus SDK の機能を統合するためには、Eclipse の Android プロジェクトを設定する必要があります。Android 用の BeaconPlus SDK は、Android ライブラリープロジェクトとして提供されています。ユーザは BeaconPlus SDK をインポートして参照し、JAVA ビルドパスとライブラリを設定する必要があります。

- a. Eclipse を立ち上げる。
- b. Eclipse において新規に Android アプリ・プロジェクトを作成し、たとえば「GetMeShoes」などのように命名する。
- c. BeaconPlus SDK > Import > Android > Existing Android Code Into Workspace を選択して Next をクリック

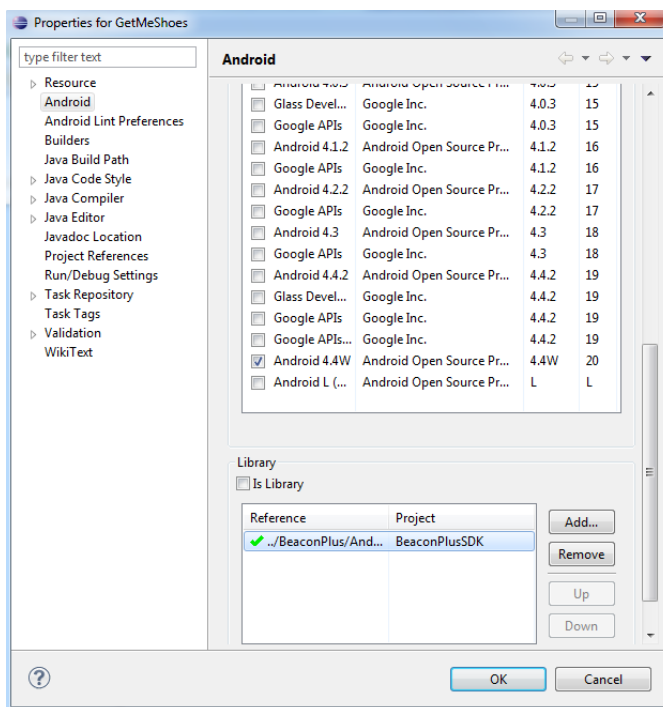
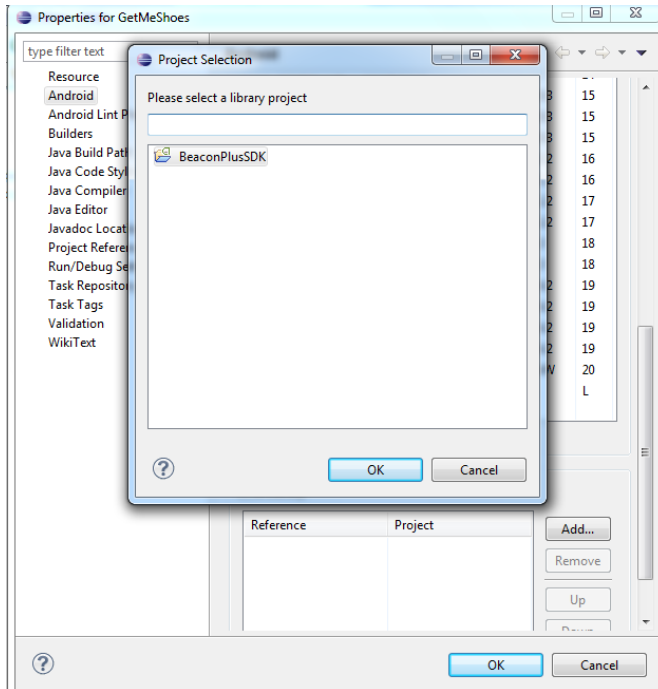


- d. Browseを選択... BeaconPlus SDKの場所を確認し、copy project into workspace option を選択して、Finishをクリックする。



- e. プロジェクトのプロパティを更新する。
 - e.1. Projectを右クリックしてPropertiesを選択。project propertiesのダイアログが表示される。

- e.2. Androidを選択し、ライブラリセクションでAddをクリック。BeaconPlus SDK projectを選択してOKをクリックする。



- f. オプション : Project > Clean とクリックして、プロジェクトが意図通りに設定されているかどうか確認する。

5 SDK の使用方法

a. パーミッションを宣言する

SDKはAndroidのいくつかの特色ある機能を全面的に活用しているので、以下に示すパーミッションブロックがユーザのアプリであるAndroidManifest.xml に含まれているかどうかを確認してください。

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

b. サービスとレシーバを宣言する

BeaconPlus SDKが正常に動作するためには、ユーザのアプリであるAndroidManifest.xml 中に、以下のサービスとレシーバを追加しておく必要があります。

同期プロセス、ネットワーク、ブルートゥース状態の特定のために、以下のレシーバを利用します。

```
<receiver android:name="com.innovature.beaconplussdk.BeaconPlusSDK"
  android:enabled="true" >
  <intent-filter>
    <action android:name="com.innovature.beaconplussdk.ALARM_SYNC_SERVICE" />
    <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
    <action android:name="android.bluetooth.adapter.action.STATE_CHANGED" />
  </intent-filter>
</receiver>
```

ビーコン検出プロセスには以下のサービスを利用します。

```
<service android:name="com.innovaturelabs.ibeacon.service.IBeaconService"
  android:enabled="true"
  android:isolatedProcess="false"
  android:label="iBeacon">
</service>
<service android:name="com.innovaturelabs.ibeacon.IBeaconIntentProcessor"
  android:enabled="true">
</service>
```

最後に、ユーザ・アプリ中に次のメタデータを入れておいてください。

```
<meta-data android:name="com.google.android.gms.version"
  android:value="4242000"/>
```

c. SDKの初期化

BeaconPlusSDKクラスは、アプリとBeaconPlus SDKのインタフェースのことです。

- a.1. ユーザがSDKに関連したアクションを実装する予定のクラスに、BeaconPlusSDKクラスをインポートする。

```
import com.innovature.beaconplussdk.BeaconPlusSDK;
```

- a.2. 以下のいずれかの方法を利用してSDKを初期化する。

```
BeaconPlusSDK sdkManager = BeaconPlusSDK.initialize(<CONTEXT>,
<SERVERURL>, <ACCESSTOKEN>, <ONINITIALIZATIONLISTNER>);
```

```
BeaconPlusSDK sdkManager = BeaconPlusSDK.initialize(<CONTEXT>,
<SERVERURL>, <ACCESSTOKEN>, <LOCALE>, <ONINITIALIZATIONLISTNER>);
```

```
BeaconPlusSDK sdkManager = BeaconPlusSDK.initialize(<CONTEXT>,
<SERVERURL>, <ACCESSTOKEN>, <LOCALE>, <ONINITIALIZATIONLISTNER>, <CACHEPLACE>)
;
```

```
BeaconPlusSDK sdkManager = BeaconPlusSDK.initialize(<CONTEXT>,
<SERVERURL><ACCESSTOKEN>, <LOCALE>, <ONINITIALIZATIONLISTNER>, <CACHEPLACE>,
<MAXCACHELIMIT>);
```

- b. OnInitializationListnerを実行する。

OnInitializationListnerを実行して、SDKからの初期化コールバックを入手してください。OnInitialization コールバックを実行することにより、SDKメソッドのいくつかを上書きする必要があります。

```
@Override
    public void onInitializationError(int arg0, String arg1) {
        // TODO Auto-generated method stub
    }
    @Override
    public void onInitializationSuccess() {
        // TODO Auto-generated method stub
```

```
    }
```

- c. スキャンを開始する。コールによりSDKが初期化された後、ユーザはビーコンのスキャンを開始することができます。

```
boolean scanStatus = sdkManage.startScanning(<SPOTNOTIFIER>);
```

- d. **SpotNotifier**を実行する。
SDKからのスキャンコールバックを入手するために**SpotNotifier**を実行してください。
SpotNotifier コールバックを実行することにより、SDKメソッドのいくつかを上書きする必要があります。

```
@Override
public void onSpotExit(Spot spot) {
    // TODO Auto-generated method stub
}

@Override
public void onSpotEntry(Spot spot) {
    // TODO Auto-generated method stub
}

@Override
public void onScanError(int errorCode, String message) {
    // TODO Auto-generated method stub
}
```

6 アクセストークンによる認証

BeaconPlus SDK は、BeaconPlus クラウド中に自分のアカウントを作成した後、各組織に提供されているアクセストークンにより各ユーザを特定します。その後 SDK は、クラウドからのアプリ用のマッチング情報を引き出して、タスクを実行します。

主要な特徴

- アクセストークンは開発者を特定する「鍵」の役割を果たし、クラウドからどのデータにアクセスすべきかをシステムに知らせます。そのため開発者はログインや認証のためだけに特別なロジックを実装する必要はありません。
- 同一組織内では、複数のアプリに対して同じアクセストークンを使用することができます。アプリの利用可能情報は同じであっても無くてもよく、アプリ開発者は自らの要件に合うようにアプリのロジックを実装することができます。

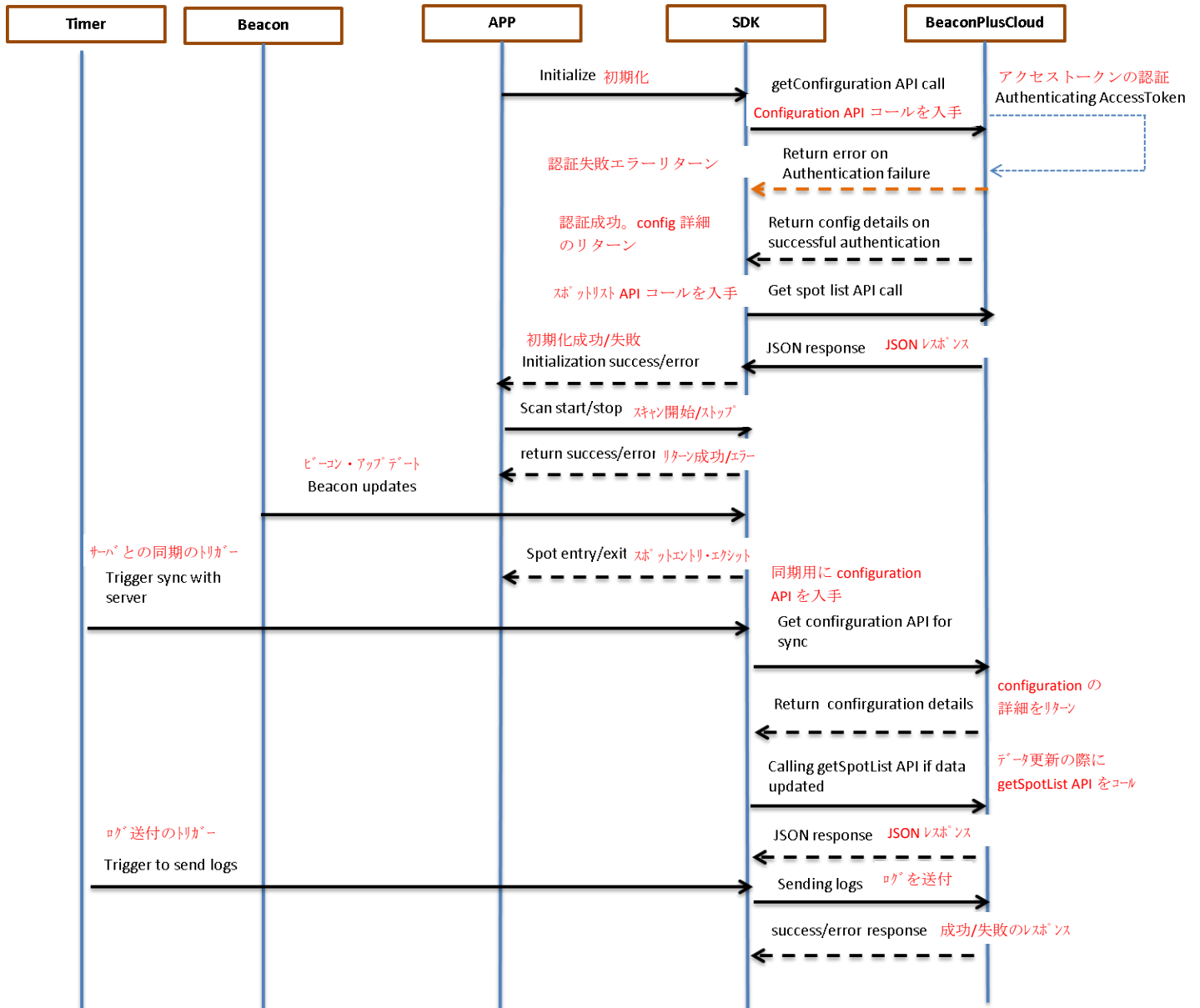
6.1 アクセストークンの入手

Innovature Technologies にお問い合わせの際には、下記の事項も併せてお知らせください。

- 組織名

- 住所
- 連絡先の番号
- 電子メール

7 コールフロー・ダイアグラム



8 エラーコード

以下に SDK からアプリに返すエラーコードを示します。

エラーコード	エラー理由
738	ネットワークエラー
723	アプリがブロックされた
724	組織がブロックされた
733	アプリの有効期限切れ
740	組織の有効期限切れ
702	SDK が初期化されていない
732	無効なアクセストークン
731	無効な時間
768	デバイスがサポートされていない
744	ファイルダウンロードに失敗した
701	コンテキストは空白でないこと
703	OnInitializationListner は空白でないこと
764	リージョンが追加されていない
765	デバイスが追加されていない
734	サーバへの接続不能
721	サーバ詳細が設定されていない
725	アクセストークンが見つからない
726	サーバの url が見つからない
743	キャッシュの最小基準を満たしていない
739	一時的なリンク発生に失敗した
781	無効なパラメータ
765	デバイスが未登録
761	SpotNotifier は空白でないこと
736	アプリにスポットが未追加
766	スキャン時間超過
767	スキャン時間が不十分
769	ブルートゥースが有効になっていない
768	Bluetooth low energy はサポートされていない

9 文献

初めての Android 開発用に

<https://developer.android.com/training/basics/firstapp/index.html>

ライブラリ・プロジェクトの参考として

<http://developer.android.com/tools/projects/projects-eclipse.html>

BLE の参考ガイド

<https://developer.android.com/guide/topics/connectivity/bluetooth-le.html>