

BeaconPlus SDK for Android Documentation

V1.3

April 2015

PinMicro K.K. Confidential and Proprietary

Restricted Distribution: This documentation is the property of PinMicro K.K. Not to be used, copied, reproduced in whole or in part, nor its contents revealed in any manner to others without the prior written permission of PinMicro K.K.

In case of any queries please contact us on support@pinmicro.com

Revision History

Date	Version	Description	Author
2014-08-13	0.4	Initial release	Jithin
2014-08-22	1.0	Review points incorporated	Jithin
2014-11-03	1.1	Error code updated	Jithin
2015-03-17	1.2	Error code updated	Jithin
2015-04-17	1.3	Document updated	Jithin

Review Details

Date	Version	Author	Reviewer
2014-08-13	0.4	Jithin	Nidheesh
2014-08-22	1.0	Jithin	Nidheesh
2014-11-03	1.1	Jithin	Nidheesh
2015-03-17	1.2	Jithin	Nidheesh
2015-04-17	1.3	Jithin	Nidheesh

Table of Contents

1	Introduction	4
1.1	Purpose of the document	4
1.2	Scope of the document.....	4
2	Acronyms	4
3	BeaconPlus SDK Overview	4
4	Setting up the SDK	5
4.1	SDK download details	5
4.2	Requirements.....	5
4.3	Registering you organization	5
4.4	Getting an Access Secret.....	6
4.5	Setting up BeaconPlus cloud.....	7
4.6	Integrating BeaconPlus SDK in your Eclipse project	7
5	Using the SDK.....	11
6	Authentication via Access Token	13
7	How to troubleshoot.....	13
8	Call flow diagram	15
9	Error Codes	16
10	Bibliography	17

1 Introduction

1.1 Purpose of the document

The BeaconPlus SDK documentation provides insight into the BeaconPlus SDK and how it could be integrated to Android Apps. For detailed API spec and other implementation details, please refer [BeaconPlus API spec](http://pinmicro.com/docs/android-sdk/index.html) (<http://pinmicro.com/docs/android-sdk/index.html>).

1.2 Scope of the document

Any developer who would like to develop contextual awareness applications for Android could use this documentation as the starting point for the same.

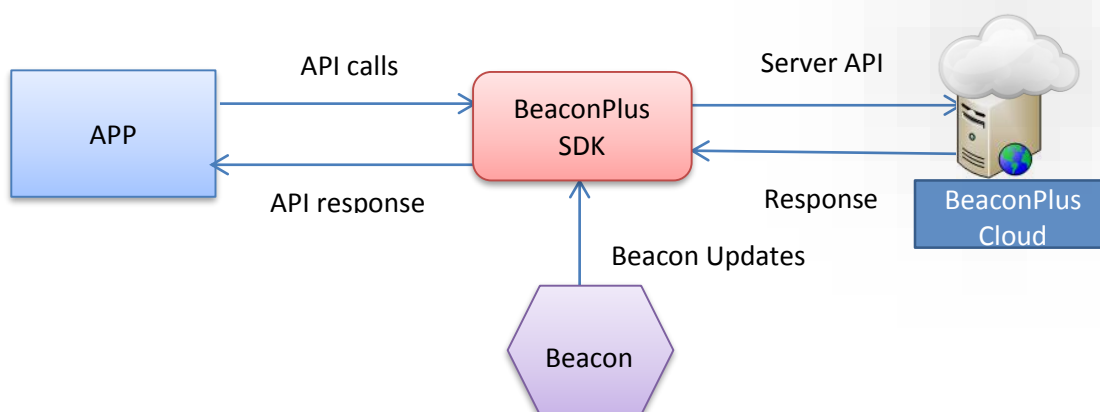
2 Acronyms

The following acronyms are used in this documentation

Acronym	Details
SDK	Software Development Kit
API	Application Programming Interface
BLE	Bluetooth Low Energy

3 BeaconPlus SDK Overview

BeaconPlus SDK is offering developers the freedom to develop contextual awareness applications. It acts as a bridge between the Client App and the BeaconPlus cloud services.



The SDK synchronizes with the cloud periodically and thereby holds the right information to project at the right time. It also handles all the internal logic for device detection and notifies the app on the 'Entry' and 'Exit' events for a specific region.

4 Setting up the SDK

This section describes how to setup the SDK for using in an Android application.

4.1 SDK download details

BeaconPlus SDK can be downloaded from the following url :

<https://pinmicro.com/blog/category/beacon-sdk/#downloads>

The SDK bundle zip file consists of two parts

- BeaconPlus SDK
- Sample Application source code (GetMeShoes)

4.2 Requirements

The BeaconPlus SDK uses BLE protocol to identify beacons, hence requires a minimum Android version 4.3. The hardware specifications demands that the SDK runs only on the following devices:

- Android phone with Bluetooth 4.0
- Android tablet with Bluetooth 4.0

4.3 Registering your organization

The first and foremost step to do before using BeaconPlus SDK is to register your organization. To register, click on the following link.

<https://pinmicro.com/register/>

The link will redirect the user to the registration page as shown in the below screenshot

PinMicro

English 日本語

Call +81-3-6439-1205

Home Products SDK Cloud About Us Contact Us

Register Your Organization

Play with BeaconPlus Admin Console for 30 days free trial. You can enjoy all web features and API's (API service will be available only till the expiry date, other services will continue to work as before)

Organization Name

Address

Contact Number

Administrator Name

Administrator Email

Administrator Password

Confirm Password

Fill up all the necessary details in the fields given, as shown in the above screenshot and once it is done, a confirmation mail will be send to the mail id with which user has registered.

4.5 Setting up BeaconPlus cloud

Before starting the actual development, make sure all the necessary components have been configured correctly in BeaconPlus cloud. For eg .

1. Deployment is created
2. The application under development is added to this deployment
3. Beacons and its contents are added to deployment

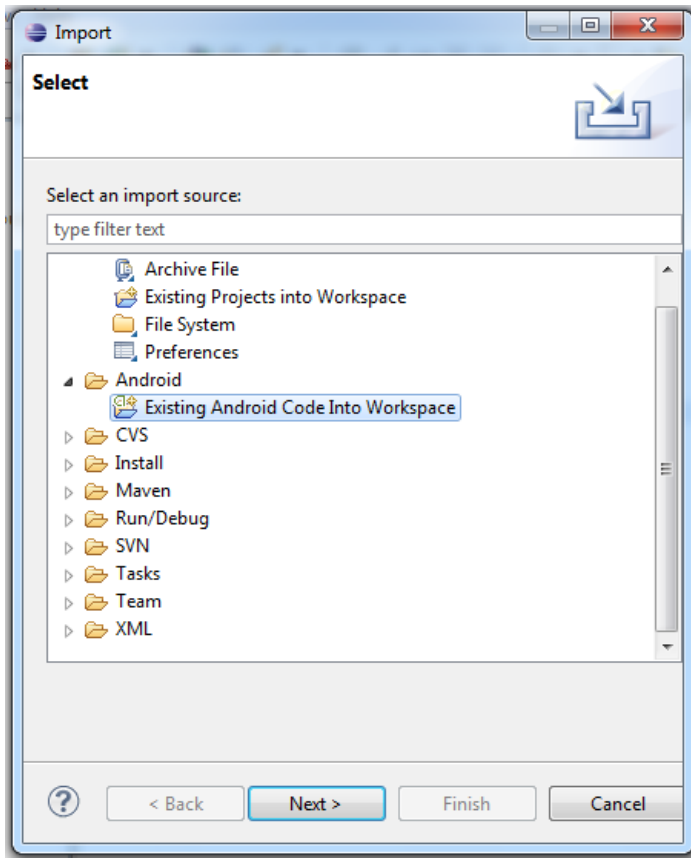
For more details please refer the doc from the link below:

<https://pinmicro.com/docs/BeaconPlus-Admin-UserManual.pdf>

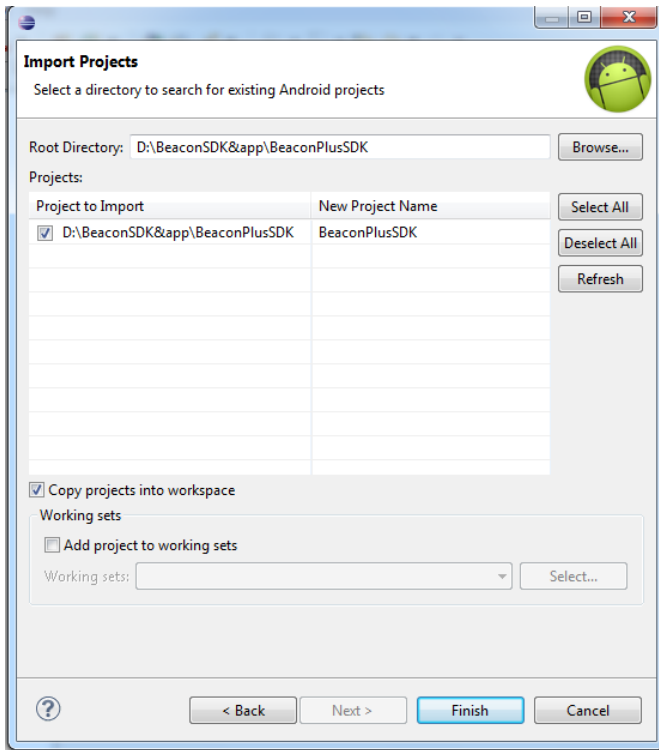
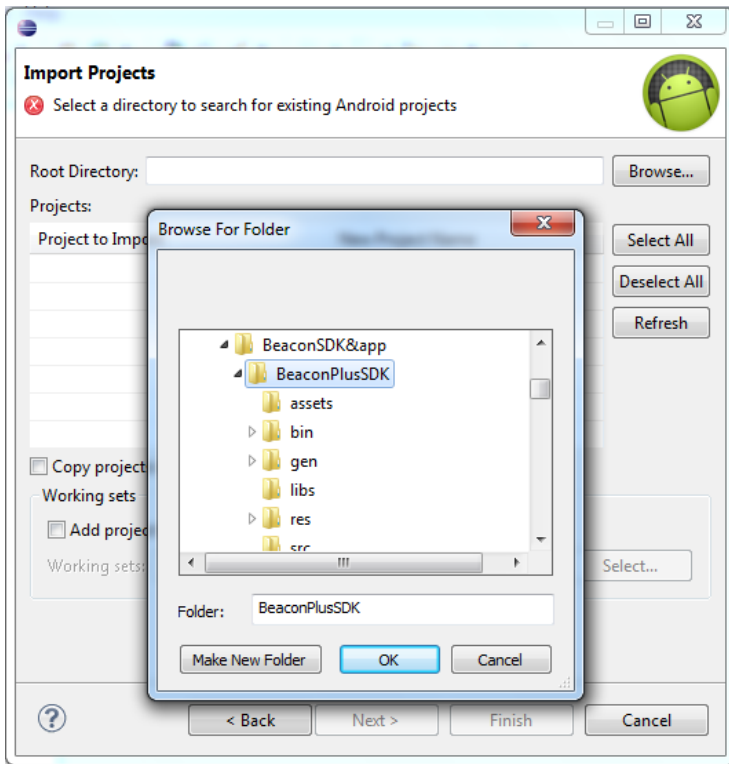
4.6 Integrating BeaconPlus SDK in your Eclipse project

Integrating BeaconPlus SDK feature in your Android application requires configuring your Android project in Eclipse. BeaconPlus SDK for Android is delivered as an Android Library Project. You must import and reference the BeaconPlus SDK and set up the Java build path and libraries.

- a. Launch Eclipse.
- b. Create a new Android Application project in Eclipse, name it, say, 'GetMeShoes'
- c. Select BeaconPlus SDK > Import > Android > Existing Android Code Into Workspace and click Next.

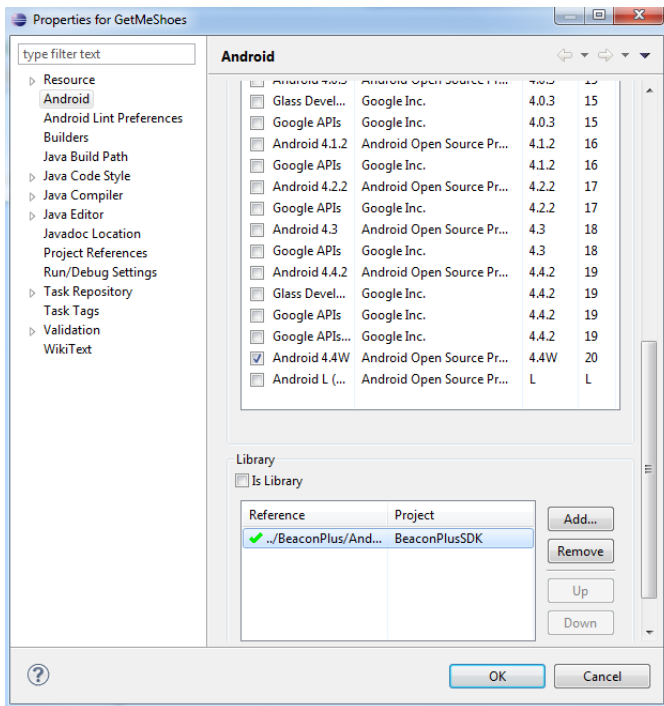
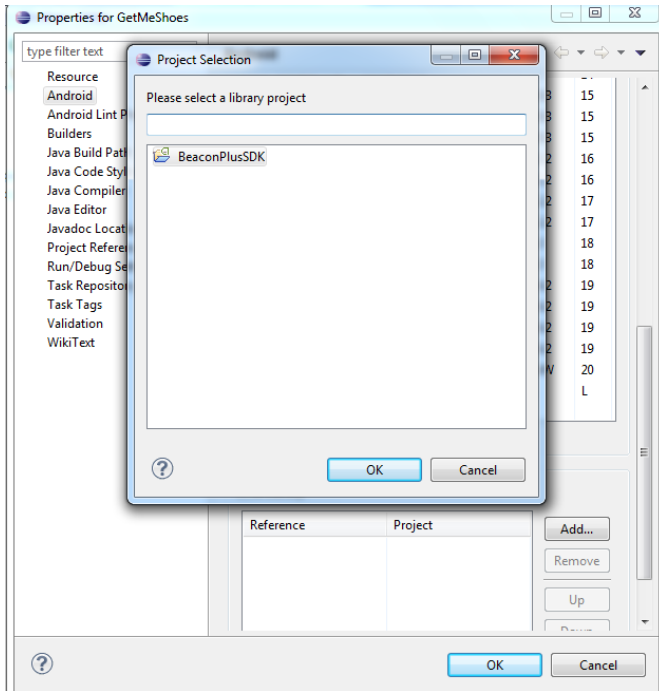


- d. Select Browse... and identify the location of BeaconPlus SDK , select copy project into workspace option and click Finish



- e. Update your project's properties
 - e.1. Right click Project > Properties. The project properties dialog displays.

e.2. Select Android and in the Library section, click Add. Choose the BeaconPlus SDK project and click OK



- f. Optional: Click Project > Clean to ensure your project picks up the settings

5 Using the SDK

a. Declare permissions

The SDK completely relies on some unique Android features, so please make sure that following block of permissions are included in your application's AndroidManifest.xml.

```
<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. Declare services and receiver.

In your application's AndroidManifest.xml you must add the following receiver and services for proper working of the BeaconPlus SDK.

Following receiver is used for sync process, network and Bluetooth status identification.

```
<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>
```

Following services are used for beacon detection process.

```
<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>
```

Finally, include this meta-data inside your application.

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

c. Initializing the SDK

BeaconPlusSDK class is an interface between application and BeaconPlus SDK.

- a.1. Import BeaconPlusSDK class to the class in which you would like to implement the SDK related actions.

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

- a.2. Initialize SDK by using one of the following methods.

```
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. Implement the OnInitializationListner.

Implement OnInitializationListner for getting initialization call back from the SDK. By implementing OnInitialization callback you must override some SDK methods.

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

c. Start scanning operation.

User can start beacon scanning operation after SDK has initialized by calling

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

d. Implement the SpotNotifier.

Implement SpotNotifier for getting scanning call back from the SDK. By implementing SpotNotifier callback you must override some SDK methods.

```
@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 Authentication via Access Token

The BeaconPlus SDK identifies each user via the Access Token provided to each Organization after they have created their own account in BeaconPlus cloud. The SDK then retrieves matching information for the application from the cloud and performs further tasks.

Key features:

- Access token serves as a “key” that identifies a developer and lets the system know what data to access from the cloud. Hence the developer doesn’t have to implement any separate logic for login or authentication.
- An organization may use the same access token for multiple applications. The deployment information of the applications may or may not be the same and the app developer could implement the app logic to suit his requirements.

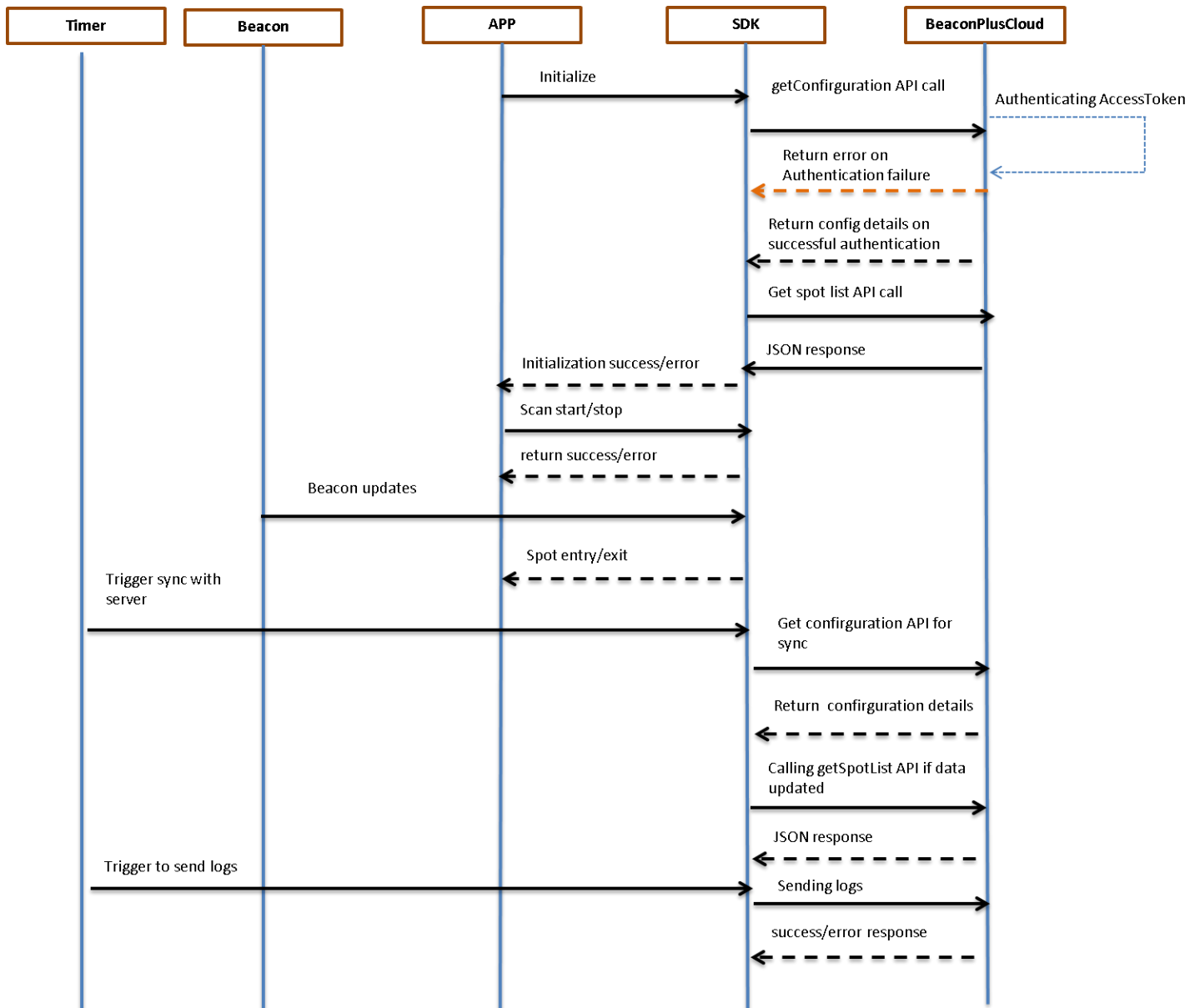
7 How to troubleshoot

Please go through the steps mentioned in the below document:

https://pinmicro.com/docs/sdk_help.pdf

Please feel free to contact us at support@pinmicro.com for further assistance.

8 Call flow diagram



9 Error Codes

Following are the error codes that would be returned to the app from the SDK.

Error Code	Reason
738	Network error
723	Application blocked
724	Organization blocked
733	Application validity expired
740	Organization validity expired
702	SDK uninitialized
732	Invalid access token
731	Invalid time
768	Device unsupported
744	File download failed
701	Context should not be null
703	OnInitializationListner should not be null
764	Regions not added
765	Devices not added
734	Unable to connect to the server
721	Server details not set before
725	Access token not found
726	Server url not found
743	Minimum cache limit not met
739	Error in generating temporary link
781	Invalid parameter
765	No devices registered
761	SpotNotifier should not be null
736	No spots added for the application yet
766	Scan time period exceeded
767	Scan time period is low
769	Bluetooth is not enabled
768	Bluetooth low energy is not supported
704	Same language as that set now

10 Bibliography

Starting point for Android development:

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

Referencing a library project

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

BLE reference guide

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