

Android SDK

Last Updated: February 14, 2013

Introduction

Placed Nearby provides mobile application developers with a simple SDK that allows easy integration with Placed Location Analytics.

It exposes simple public API calls that can be used to register strategic events that occur while an application is being used.

The SDK also collects location and sensor data that is relayed along with the application event data, enabling strategic location-based application usage analytics.

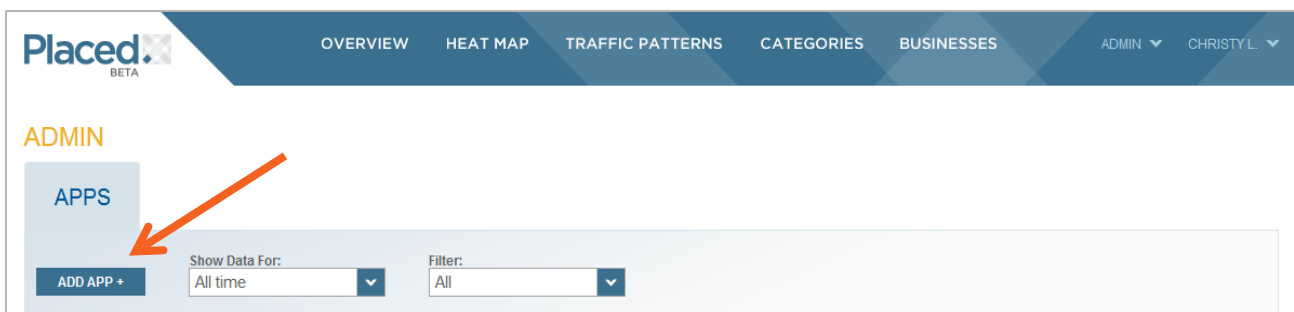
The SDK has been designed for easy set up and integration with both new and existing mobile applications.

Steps to integrate your Application with the Placed SDK

Registration

(If you've already registered an account, skip to the Set Up section of the document)

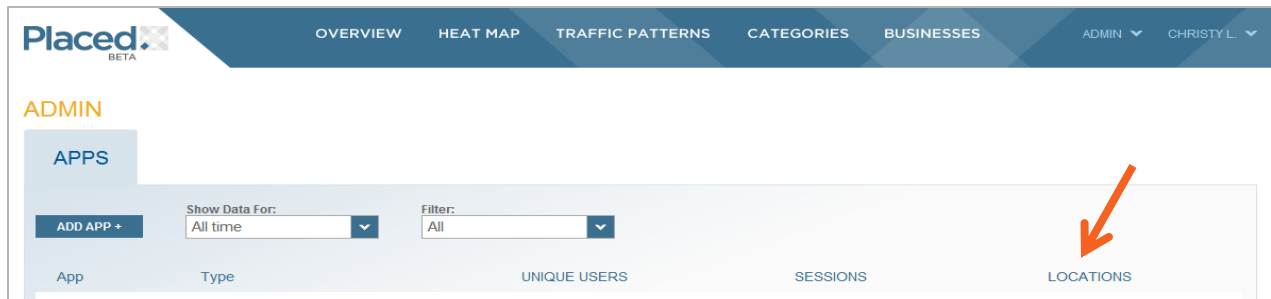
- Create a developer account with Placed
 - Visit the *Placed Nearby App Dashboard* to set up your developer account.
- Add your Application
 - Once signed into the *Placed Nearby App Dashboard*, you can add an application
 - Click on "Add App," enter a name for your application and select Android as platform



- Upon adding your application you will be provided a unique **SDK Application Key**
- Download the App API library
 - Upon adding your application you can download the **placed-sdk-android.zip**
 - This zip file contains these instructions and the SDK library that you need to reference in your application to enable Placed Location Analytics

Set Up

- Add the **SDK jar** to your Java classpath
- In regular Java projects, add the **SDK jar** to your project's *libs* folder.
- In Eclipse, add the SDK jar to your Build Path using the *Add External JAR* wizard.
- Confirm proper implementation by logging into Placed.com and viewing the **Location** count for your application. Note that it takes at least one day for activity to be reflected in the dashboard.



Configuration

- Add the following permissions to the application's **AndroidManifest.xml**
 - These permissions are required to collect and send event and location data back to the Placed servers
- **Required Permissions**
 - android.permission.INTERNET
 - android.permission.ACCESS_COARSE_LOCATION
 - android.permission.ACCESS_FINE_LOCATION

Integration

- **Initialize**
 - Instantiate the PlacedAgent at the creation of your application.
 - We advise placing this in the *onCreate()*
 - `PlacedAgent placedAgent = PlacedAgent.getInstance(Context, Application Key)`
 - Provide a reference to the Android Activity or Service Context and your Application Key
- **Start**
 - Invoke the StartSession call to indicate the beginning of a session for which you wish to associate subsequent events
 - We advise placing this in the *onStart()*

- *placedAgent.logStartSession()*
 - If there is an active session, and the application ends the active session and starts a new session within the 'Resume Interval':
 - We will resume the previous session
 - If there is an active session and the application starts a new session without ending the current active session:
 - We will always terminate the previous session and start a new session
 - If there is no active session and the application attempts to invoke any API event call:
 - We will automatically start a new session and start data collection
- **Page View**
 - Invoke the PageView call to indicate landing on a new screen/page within the application
 - *placedAgent.logPageView(Page Title)*
 - Provide the name of the screen/page you wish to track
- **Events**
 - Once you have registered the session start and a landing page, you can proceed to invoke various Custom Events
 - *placedAgent.logCustomEvent(Event Title, Attributes)*
 - Provide a meaningful event title that you wish to track and report on
 - Optionally you can provide a set of string attributes associated with the Event that will be used for more granular analytics
 - Note that you can register up to ten unique Event and Attributes
 - All events are associated with the most recent PageView invoked
- **Custom Variables**
 - You may choose to register custom variables during the operation of your Application
 - *placedAgent.logCustomVariable(Scope, Value)*
 - You can specify a free form value and scope it to the User, Session or Page
- **Demographics**
 - You may choose to provide demographics including a unique user identifier, age or gender.
 - *placedAgent.logUniqueld(Unique Identifier)*
 - You can use this to log your application specific user/device unique identifier. We will use this information internally for aggregated reporting only.

- *placedAgent.logAge(Age)*
 - You can use this to log your user's age. Valid values range from 1 to 115
- *placedAgent.logGender(Gender)*
 - You can use this to log your user's gender. Valid values include MALE, FEMALE and UNSPECIFIED
- **End**
 - Invoke the EndSession call to indicate the completion of a session
 - We advise placing this in the *onDestroy()*
 - *placedAgent.logEndSession()*
 - This closes the current active sessions and stops Placed location tracking

Advanced Concepts

- **Logging**
 - You can use this call to enable logging for your debugging
 - *placedAgent.setEnableLogging(True or False)*
 - You may choose to disable logging when releasing your application to the market
- **Data Sync**
 - Data is collected only during an active session, and is synchronized with the Placed servers periodically. This does not block your application's activities.
 - This sync cycle is not configurable.
- **Data Collection Mode**
 - You can configure the mode in which the Placed SDK collects data. We recommend that you keep the setting at DEFAULT. Any changes to a level below DEFAULT will result in lower coverage and accuracy in the analytics.
 - *placedAgent.setDataCollectionMode(DataCollectionMode)*
 - The modes to select from are as follows
 - DataCollectionMode.PRECISE
 - DataCollectionMode.DEFAULT
 - DataCollectionMode.CONSERVATIVE
 - DataCollectionMode.PASSIVE

Support

For further guidance contact support@placed.com