# Legacy - Android Quick Start Guide

This document helps you quickly start using Alchemer Mobile (Formerly Apptentive) in your Android app. For complete documentation, as well as system requirements, see the Android Integration guide.

## 1. Add Alchemer Mobile

In your `build.gradle`, add a dependency to Alchemer Mobile, replacing [[SDK_version]] with most recent SDK

```
repositories {
  jcenter()
}

dependencies {
  implementation 'com.apptentive:apptentive-android:[[SDK_version]]'
}
```

When prompted by Android Studio, click **Sync Now**

## 2. Register Alchemer Mobile

When your app starts, it will need to register Alchemer Mobile (Apptentive). This step must happen in your `Application` class's `onCreate()` method.

```
public class YourApplication extends Application {
    @Override
    public void onCreate() {
      super.onCreate();
      ApptentiveConfiguration configuration = new ApptentiveConfiguration("YOUR_APPTENTIVE_APP_KEY", "YOUR_APPTENTIVE_APP_SIGNATURE");
      // Set the log level to debug Apptentive
      configuration.logLevel = ApptentiveLog.Level.VERBOSE
      Apptentive.register(this, configuration);
    }
}
```

Make sure you use the Alchemer Mobile (Apptentive) App Key and Alchemer Mobile (Apptentive) App Signature for your Android app on the API & Development page. Sharing keys in two apps, or using keys from the wrong platform is not supported, and will lead to incorrect behavior.

## 3. Styling Alchemer Mobile

Alchemer Mobile will inherit your app's styles by default. If you are using a Light/Dark AppCompat theme, Alchemer Mobile will look like your app by default, and you can skip this step. But if you are using another theme, or if you want to force Alchemer Mobile to adopt different styles than your app, please follow instructions in Android Interface Customization.

# 4. Add Events

Events record user interaction. You can use them to determine if and when an Interaction will be shown to your customer. At a minimum, you should include 20 – 50 Events in your app to start taking advantage of Alchemer Mobile, but for now, let's just create one. To trigger an Event, call the `engage()` method. This will record the Event, and then check to see if any Interactions targeted to that Event are allowed to be displayed, based on the logic you set up in the Alchemer Mobile Dashboard.

In this example, trigger an Event when your Main Activity resumes.

```
@Override
protected void onResume() {
  super.onResume();
  Apptentive.engage(this, "main_activity_resumed");
}
```

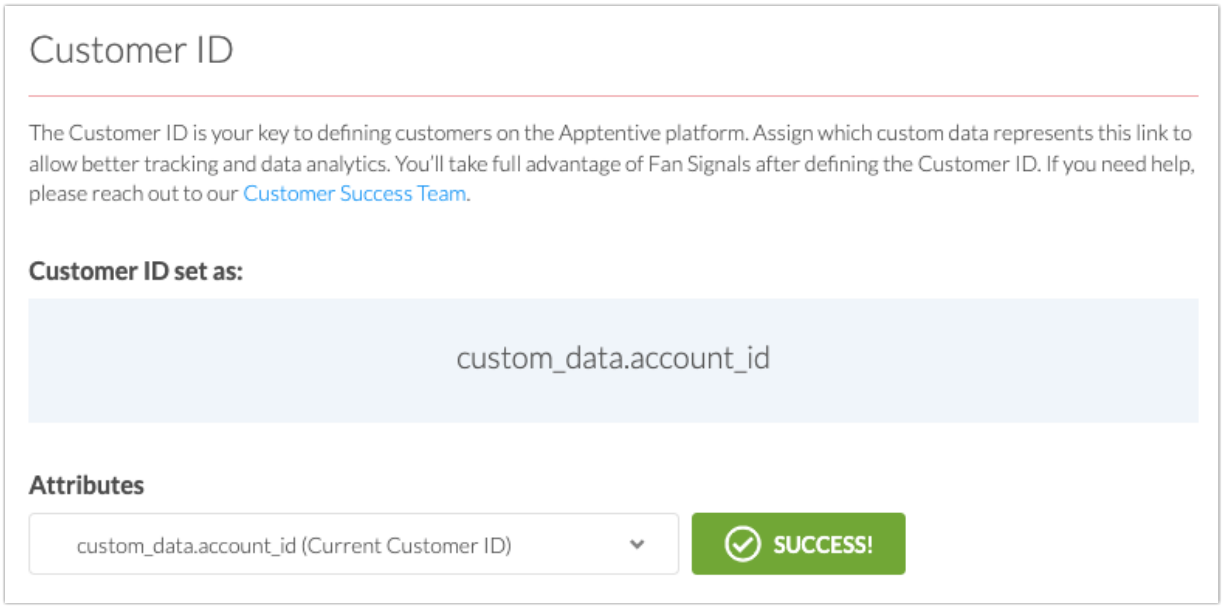
# 5. Add Customer ID

You can send Custom Data associated with a person's profile that is using the app, or the device. In particular, this is useful for sending a Customer ID and other information that helps you understand and support your users better. Custom Data can also be used for configuring when Interactions will run. You can add custom data of type `String`, `Number`, and `Boolean`.

Below is an example of a Customer ID value being passed in, along with whether that customer is on a premium account of the app or not.

```
 Apptentive.addCustomPersonData("customer_id", 1234567890); Apptentive.addCustomPersonData("is_prem
ium", true);
```

After setting your Customer ID and other custom data, you can choose which field is your Customer ID in the Alchemer Mobile Platform.




If you are interested in learning more about the Customer ID feature, please review this article.

# 6. Show a Survey

Now that you've created an Event, you can create a Survey and display it when the Event is triggered.

1. Go to the Surveys page.
2. Click "New Survey".
3. Give the Survey a name, title, introduction, add a question, choose whether to end with a *Thank You* message, and click **Save & Continue**.
4. Choose *Publish survey as an independent Interaction*.
5. Under the **Where** section, chose the Event `main_activity_created` (or whatever Event name you used). If your app hasn't connected to the server after triggering that Event, you will need to add it manually at this point, by clicking **Create new Event** on the Events page.
6. Near the bottom, check *Allow multiple responses from the same person* so you can display this survey more than once.
7. Click **Save & Continue.**
8. Click **Launch Survey.**
9. Finally, **uninstall then reinstall** the app to ensure you have downloaded that newly launched Survey from our servers.

Now, you will see this survey when you trigger the `main_activity_created` Event.

# 7. Add Message Center

Find a place in your app for a button that will launch Message Center. This will allow customers to contact you with feedback, or questions if they are having trouble using your app, as well as allow them to see your responses.

If Message Center is available, show a `Button` that will launch it when clicked. This example assumes you have an `Activity` called `SettingsActivity` that you has a `Button` you would like to open Message Center with.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  setContentView(R.layout.settings_layout);

  final Button button = (Button) findViewById(R.id.message_center_button);
  Apptentive.canShowMessageCenter(new Apptentive.BooleanCallback() {
    @Override
    public void onFinish(boolean canShowMessageCenter) {
      if (canShowMessageCenter) {
        button.setVisibility(View.VISIBLE);
        button.setOnClickListener(new View.OnClickListener() {
          @Override public void onClick(View v) {
          Apptentive.showMessageCenter(YourActivity.this);
        }
      });
    } else {
      button.setVisibility(View.GONE);
    }
  }
}
```

Related Articles