

Legacy - Troubleshooting the Alchemer Mobile Android SDK

Here are a few of the more common issues that customers have run into when integrating the Alchemer Mobile Android SDK in their app.

Poor Text Contrast

If the text isn't contrasting with the background of part or all of the Alchemer Mobile UI, you may have forgotten to tell Alchemer Mobile which type of AppCompatActivity theme your app is using. For more information, please visit our [Android Interface Customization](#) guide.

Interactions Aren't Displaying

If you are confident that the criteria you have configured for displaying an Interaction when you engage a specific event has been met, check your logs. It could be that the criteria hasn't really been met, or it has, but there is a problem displaying the Interaction.

Criteria Hasn't Been Met

Sometimes you think that the criteria has been met, and that an Interaction you've targeted through the "Where" section of the Interaction configuration should display when you call `Apptentive.engage()` with that Event name. If the Interaction isn't displaying, check the logs in Verbose mode.

To enable Verbose logging for the Alchemer Mobile SDK, set the log level on your Alchemer Mobile Configuration before registering it:

```
configuration.logLevel = ApptentiveLog.Level.VERBOSE
```

We print out the results of the criteria evaluation to help debug which piece is causing the evaluation to fail. Here is an example:

```

V/Apptentive: + Parsing Interaction Criteria.
V/Apptentive: + LogicalClause of type "$and"
V/Apptentive: + ConditionalClause for query: "code_point/com.apptentive#app#launch/invokes/version_name"
V/Apptentive: + ConditionalTest: $gte: BigDecimal(1)
V/Apptentive: + ConditionalClause for query: "interactions/00f51c33b553b54409000000/invokes/version_name"
V/Apptentive: + ConditionalTest: $seq: BigDecimal(0)
V/Apptentive: + Finished parsing Interaction Criteria.
I/Apptentive: Evaluating Criteria
V/Apptentive: - <$and>
V/Apptentive: - code_point/com.apptentive#app#launch/invokes/version_name
V/Apptentive: - BigDecimal(1) $gte BigDecimal(1)?
V/Apptentive: - => true
V/Apptentive: - interactions/00f51c33b553b545c9000056/invokes/version_name
V/Apptentive: - BigDecimal(0) $seq BigDecimal(5)?
V/Apptentive: - => false
V/Apptentive: - <$and>
I/Apptentive: - => false

```

Each clause in the criteria corresponds to configuration you've specified in the Interaction configuration, so you may be able to spot the problem. If not, we are here to help.

Interaction Is Covered

Sometimes, an Interaction is displayed, but then immediately covered by another Activity launched by your app. Since our UI is composed of Activities, you need to make sure you don't launch another Activity right after you engage an Event if you plan on targeting an Interaction to that Event.

If you see something like this in the logs, but you don't see an Interaction display, check to make sure you aren't calling `startActivity()` right after you call `Apptentive.engage()`.

```
I/Apptentive: Launching interaction: EnjoymentDialog
```

This log means that the Love Dialog launched. You just need to make sure you aren't covering it up, and you should be good to go.

Google Play Rating Dialog Not Shown

The Google Play Rating Dialog is a specific type of Interaction, and in addition to following the above guidelines about meeting criteria and ensuring that the UI is not covered, there are several more reasons that this dialog may not appear.

- You must be using [SDK version 5.6.0 or higher](#) (or corresponding versions for other integrations, listed [here](#)). Your customers must also be on Android OS 5 or higher.
- You must be testing on a version of the app downloaded from Google Play. If testing in a simulator or similar, the dialog cannot be presented.
- The account that's attempting to see the dialog has hit its quota. This limit is imposed by Google directly, though they do not say exactly what that limit is.
- The user has already reviewed the app. If testing and you wish to leave another, you can delete your original review directly from Google Play.
- If using a Google Business account.

- The customer already has a review within a test track.
- There may also be other silent, undocumented failures by Google.

Whenever that the Google Play Rating Dialog is called but is *not* presented, the following messages would be seen in the logs:

```
E/Apptentive: [IN_APP_REVIEW] In-app review flow completed too fast (549 ms) and we have good reasons to believe it just failed silently.  
I/Apptentive: [Apptentive Queue] [INTERACTIONS] Engage event: 'com.apptentive#InAppRatingDialog#not_shown'
```

This means that Alchemer Mobile attempted to present the dialog but Google ultimately prevented it from being displayed at that particular time.

The Rating Dialog Shown Instead of Google Play Rating Dialog

Are you on an eligible version of the SDK and OS, but seeing the *Alchemer Mobile* Rating Dialog rather than the *Google Play* Rating Dialog? If so, you may be experiencing an issue with the Where Event that you used for targeting. To be certain of this, you would see the following message in the logs:

```
E/Apptentive: [IN_APP_REVIEW] Unable to create in-app review manager  
java.lang.ClassCastException: com.your_app.app_name.your.app cannot be cast to android.app.Activity
```

This means the Where Event that you're using for launching the Rating Dialog was likely passed an Application Context instead of an Activity Context. To display the Google Rating Dialog, pass an Activity Context as the parameter to `Apptentive.engage()`.

App Crashes When Google Play Rating Dialog is Launched

Please note that, if your app implements specific Play Core or Play Services, the Google In-App Review requires at least Play Core version 1.8.0+ and Play Services Base 17.4.0+.

If you do not use those versions, you may see a crash when trying to show the dialog.

App Behaves or Looks Weird After Integrating Alchemer Mobile

Sometimes after integrating Alchemer Mobile, you will notice an EditText is missing, a password field icon is duplicated, or some other unexpected change happens to a part of your app that is seemingly unrelated to your Alchemer Mobile integration.

Most of the time, this is caused by a conflicting support library dependency. Check the [Alchemer Mobile Integration Reference: Dependencies](#). It will list the version of the support libraries that the latest release of Alchemer Mobile depends on. If your app was using an older version of the support libraries, and Alchemer Mobile is using a newer version, then when you add Alchemer Mobile to your app gradle will use the version Alchemer Mobile depends on instead. This can lead to unexpected behavior, as newer versions of the support libraries often change behavior and break any workarounds you may be using.

You can also check for conflict resolution by running the following gradle command and looking at the output for a specific build variant.

```
./gradlew :app:dependencies
```

ProGuard/R8/Obfuscation issues

You may have an obfuscation problem if you're saying/thinking:

- "Works in Debug but not Release"
- "When we try updating from the previous version..."
- "We recently updated Android Studio and..."

Or see any of the following in your codes/logs:

- `java.lang.NoClassDefFoundError`
- `java.lang.NoSuchMethodError`
- `java.lang.ClassCastException`

When you build your project using Android Gradle plugin 3.4.0 or higher, the plugin no longer uses ProGuard to perform compile-time code optimization. Instead, the plugin works with the R8 compiler to handle the following compile-time tasks.

If you have any issues with R8 optimization – you can disable it by adding the following line in your `gradle.properties` file:

```
android.enableR8=false
```

If the issue is with R8 removing a class, try adding the class as a dependency in your app or add a rule to your R8 configuration to keep the class in question.

Related Articles