

Customize your Crashlytics crash reports

You can click into an issue and get a detailed event report in the **DevOps & Engagement > Crashlytics dashboard** (https://console.firebase.google.com/project/_/crashlytics) of the Firebase console. You can customize those reports to help you better understand what's happening in your app and the circumstances around events reported to Crashlytics.

- [Log custom keys](#) (#add-keys)
- [Log custom log messages](#) (#add-logs)
- [Log user identifiers](#) (#set-user-ids)
- [Report exceptions](#) (#log-excepts)
- [Get breadcrumb logs](#) (#get-breadcrumb-logs)
- [Enable opt-in reporting](#) (#enable-reporting)
- [Include GWP-ASan reports](#) (#gwp-asan)

Log custom keys

Custom keys help you get the specific state of your app leading up to a crash. You can associate arbitrary key-value pairs with your crash reports, then use the custom keys to search and filter crash reports in the Firebase console.

View instructions for logging custom keys: [iOS](#)

([/docs/crashlytics/ios/customize-crash-reports#add-keys](#)) [Android](#)

([/docs/crashlytics/android/customize-crash-reports#add-keys](#)) [Flutter](#)

([/docs/crashlytics/flutter/customize-crash-reports#add-keys](#)) [Unity](#)

([/docs/crashlytics/unity/customize-crash-reports#add-keys](#))

Log custom log messages

To give yourself more context for the events leading up to a crash, you can add custom Crashlytics logs to your app. Crashlytics associates the logs with your crash data and displays them in the Crashlytics dashboard.

View instructions for logging custom log messages: [iOS+](#)

([/docs/crashlytics/ios/customize-crash-reports#add-logs](#)) [Android](#)

([/docs/crashlytics/android/customize-crash-reports#add-logs](#)) [Flutter](#)

([/docs/crashlytics/flutter/customize-crash-reports#add-logs](#)) [Unity](#)

([/docs/crashlytics/unity/customize-crash-reports#add-logs](#))

Log user identifiers

To diagnose an issue, it's often helpful to know which of your users experienced a given crash. Crashlytics includes a way to anonymously identify users in your crash reports.

View instructions for logging user identifiers: [iOS+](#)

([/docs/crashlytics/ios/customize-crash-reports#set-user-ids](#)) [Android](#)

([/docs/crashlytics/android/customize-crash-reports#set-user-ids](#)) [Flutter](#)

([/docs/crashlytics/flutter/customize-crash-reports#set-user-ids](#)) [Unity](#)

([/docs/crashlytics/unity/customize-crash-reports#set-user-ids](#))

Report exceptions

In addition to automatically reporting your app's *fatal* events (like crashes), the Crashlytics SDK can report non-fatal exceptions as *non-fatal* events. For Futter and Unity, if you have exceptions that are expected, you can have the SDK report them as non-fatal events, as well.

These non-fatal events are logged on-device and then sent along with the next fatal event report or when the end-user restarts the app.

View instructions for reporting exceptions: [iOS+](#)

([/docs/crashlytics/ios/customize-crash-reports#log-excepts](#)) [Android](#)

(/docs/crashlytics/android/customize-crash-reports#log-excepts) [Flutter](#)

(/docs/crashlytics/flutter/customize-crash-reports#log-excepts) [Unity](#)

(/docs/crashlytics/unity/customize-crash-reports#log-excepts)

Get breadcrumb logs

Breadcrumb logs give you a better understanding of the interactions that a user had with your app leading up to a crash, non-fatal, or ANR event. These logs can be helpful when trying to reproduce and debug an issue.

View instructions for getting breadcrumb logs: [iOS+](#)

(/docs/crashlytics/ios/customize-crash-reports#get-breadcrumb-logs) [Android](#)

(/docs/crashlytics/android/customize-crash-reports#get-breadcrumb-logs) [Flutter](#)

(/docs/crashlytics/flutter/customize-crash-reports#get-breadcrumb-logs) [Unity](#)

(/docs/crashlytics/unity/customize-crash-reports#get-breadcrumb-logs)

Enable opt-in reporting

By default, Crashlytics automatically collects crash reports for all your app's users. To give users more control over the data they send, you can enable opt-in reporting by disabling automatic reporting and only sending data to Crashlytics when you choose to in your code.

View instructions for enabling opt-in reporting: [iOS+](#)

(/docs/crashlytics/ios/customize-crash-reports#enable-reporting) [Android](#)

(/docs/crashlytics/android/customize-crash-reports#enable-reporting) [Flutter](#)

(/docs/crashlytics/flutter/customize-crash-reports#enable-reporting) [Unity](#)

(/docs/crashlytics/unity/customize-crash-reports#enable-reporting)

Include GWP-ASan reports to debug memory corruption issues

Crashlytics can help you debug crashes caused by native memory errors by collecting GWP-ASan reports. These memory-related errors can be associated with memory corruption within your app, which is the leading cause of app security vulnerabilities.

View instructions for including GWP-ASan reports: [Android NDK](#)

(/docs/crashlytics/android/customize-crash-reports#gwp-asan) [Unity](#)

(/docs/crashlytics/unity/customize-crash-reports#gwp-asan-unity)

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](#) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](#) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](#) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2026-08-06 UTC.