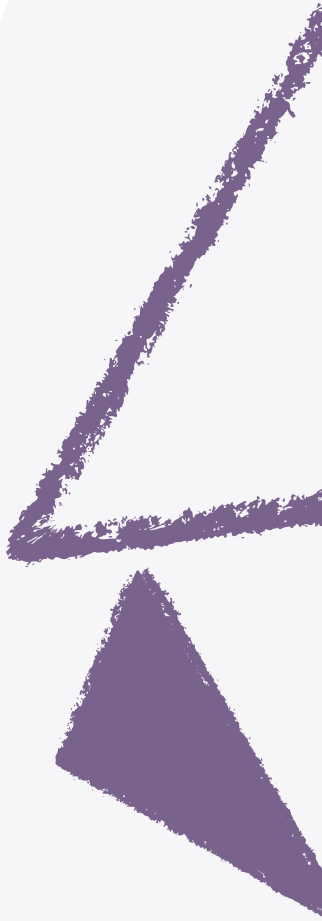
SENTRY

Sentry vs. Crashlytics:

The Mobile Developer's Decision-Making Guide

Contents



Introduction

Many mobile developers weigh the differences between Firebase Crashlytics and Sentry when they are trying to choose the right crash, error, and performance monitoring solution for their apps. This guide, which includes a feature comparison, tools to assess ROI, customer stories, and more, will help you make an informed decision.

At-a-glance overview

Firebase Crashlytics is a lightweight crash reporting tool. Part of the Firebase app development platform from Google, it offers support for Android, iOS, Flutter, and Unity. Firebase also offers Performance Monitoring, which collects and analyzes performance characteristics of Apple, Android, and web apps.

Sentry is a complete mobile monitoring solution, including crash reporting and error and performance monitoring. Part of a platform that also provides monitoring for web apps and back end projects, Sentry mobile monitoring offers out-of-the-box support for Android, iOS, Flutter, Unity, React Native, and more.

In this guide, **we will focus on the crash reporting and error monitoring capabilities of Crashlytics and Sentry.** Both Crashlytics and Sentry synthesize crashes and errors into manageable lists of issues, offering contextual information to help you find and fix root causes. However, the two offerings vary in their levels of customization, product depth, and flexibility.

In general, Firebase Crashlytics is a fine solution for hobbyists, solopreneurs, and small teams. Sentry is a better solution for growing teams that need deeper context and insights into solving crashes and errors, customizable workflow tools, and product investments to enable innovation and scale.

"As we grow and move upmarket, we have tighter requirements around reliability and issue management. Reporting through Crashlytics wasn't working well. We wanted to up our game for mobile. We wanted to capture non-fatal errors and triage them so we could respond to them faster."

Staff Engineer,
Health Benefits Platform

Here's a high-level overview of the differences between Firebase Crashlytics and Sentry.

	Sentry	Firebase Crashlytics
Context and Insights	Offers metadata, stack traces, breadcrumbs, suspect commits, screenshots, view hierarchy	Offers metadata, stack traces, keys, logs
	Source Map support for React Native	No Source Map support for React Native
	Search and filter by custom tags; view trends and analyze data with Dashboards and Discover	Limited search and filter capabilities; export and analyze data in BigQuery
Workflow	Fully customizable alerts	Pre-set alerts
	Automatic issue assignment and notifications	No issue assignment or targeted notifications
	Two-way data flow between other tools	One-way data flow to other tools
Innovation and Scale	Product investments focused on enabling developers to adopt new technologies	Product investments focused on deeper integration with Firebase Platform
	Vendor agnostic	Reliant on Firebase Platform
	EU Data residency	Not GDPR compliant

Features comparison

Let’s take a deeper dive into these 3 categories.

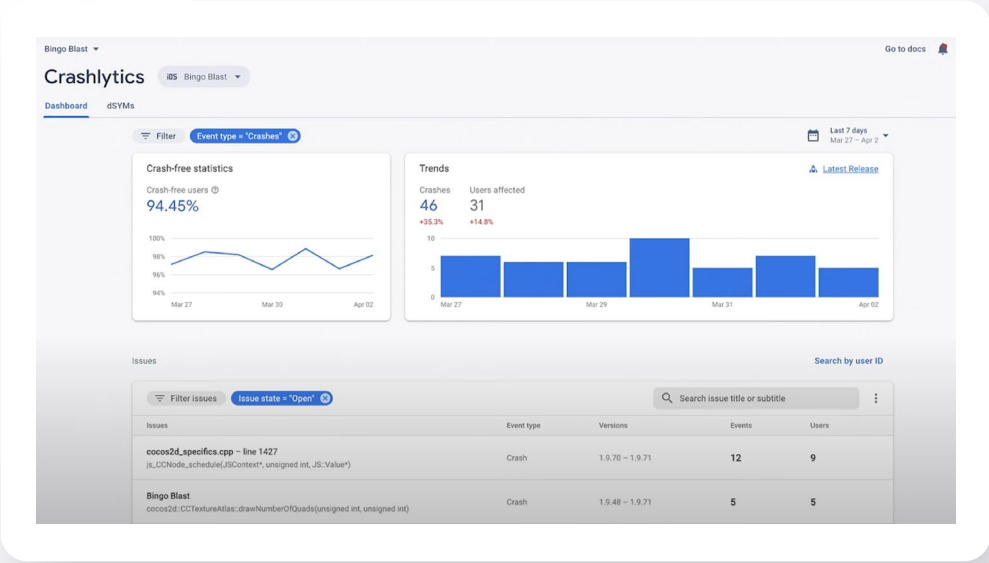
- Context and Insights
- Workflow
- Innovation and scale

Context and Insights

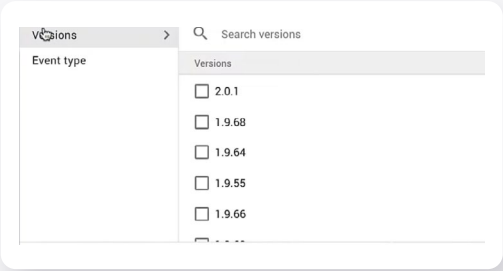
Prioritizing crashes and errors and identifying root causes in order to solve them fast is the primary use case for crash reporting tools. So let's start there.

Here's a look at the **Crashlytics dashboard**, where you can view crash-free rate statistics, event trends, and issues for one project at a time.

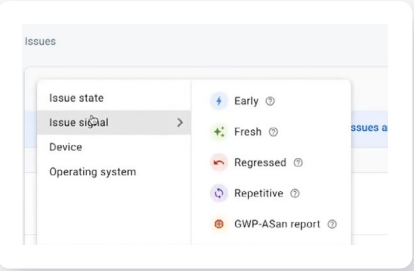
To get more granular details, you can filter crash-free stats and event trends by release version or event type. You can also drill down into specific issues and filter by issue state, device, operating system, or custom keys. Additionally, you can search issues by title or custom keys.



Filter crash-free rates:



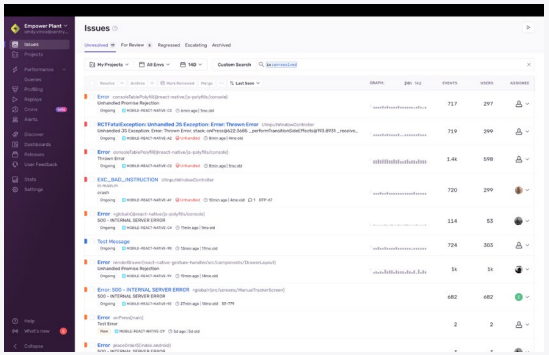
Filter issues:

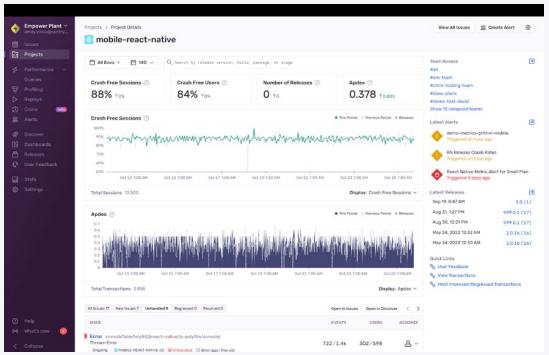


Let’s compare Sentry, starting with the Issues page. Here, you can search and filter issues across projects and perform advanced searches with default and custom tokens.

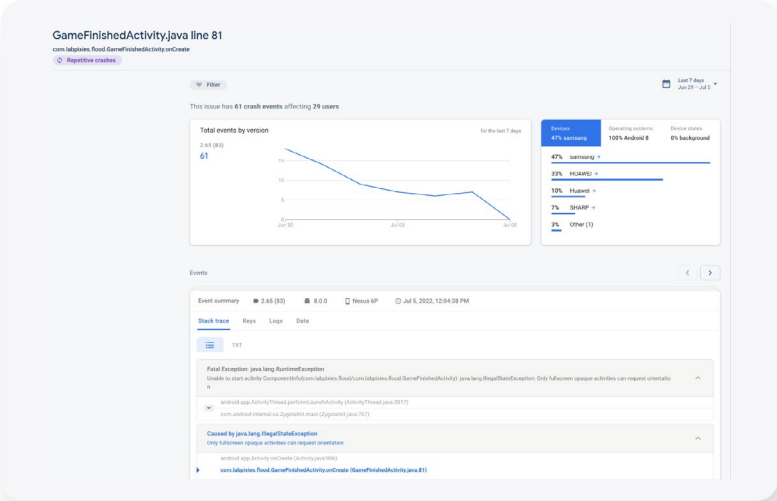
On the Project Details page, you can see current crash free sessions and Apdex scores for specific release versions, build, package, or stage of usage. You can also visualize crash free sessions over time and see any recent alerts.

We have only just started the guide, but you likely already have a sense for the deeper functionality available with Sentry. Let’s move on to context.





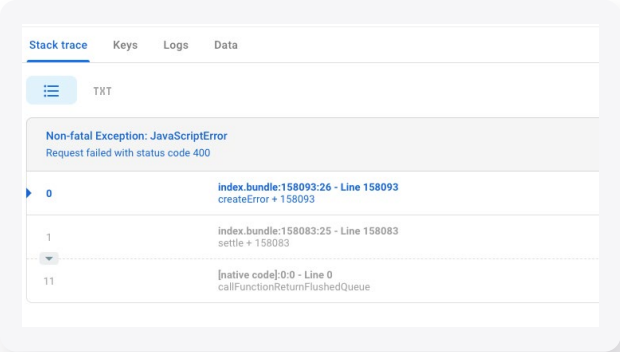
Once you have identified a crash or non-fatal error to solve, Crashlytics offers stack traces, keys, logs, and data. We hear from a lot of users that the content isn't usually enough to solve most bugs.



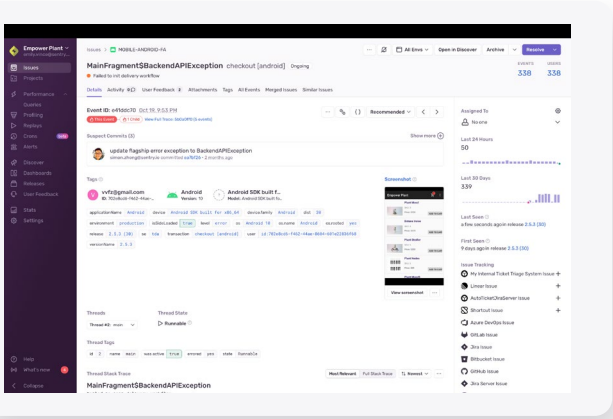
"There's not a lot of depth to the data that Crashlytics collects. There's not a lot of breadcrumbing, there's not a lot of ability to dig in deeper other than one line of stack trace causing an issue"

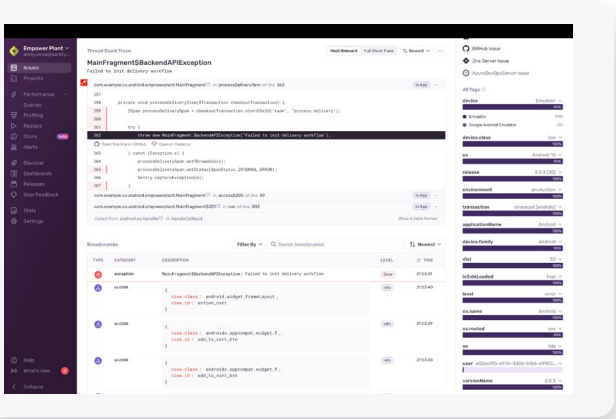
Head of Engineering,
Music Streaming Service

Debugging with Crashlytics is especially difficult if you build with React Native. Crashlytics does not provide source map support for React native out-of-the-box, which results in minified stack traces that are difficult to read.

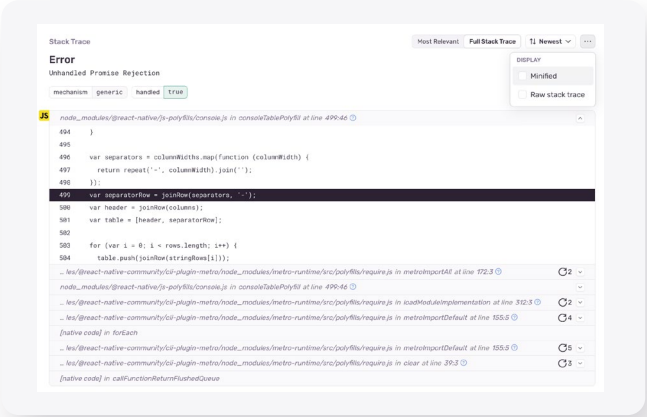


Sentry, on the other hand, provides deep actionable context to help you solve issues faster.



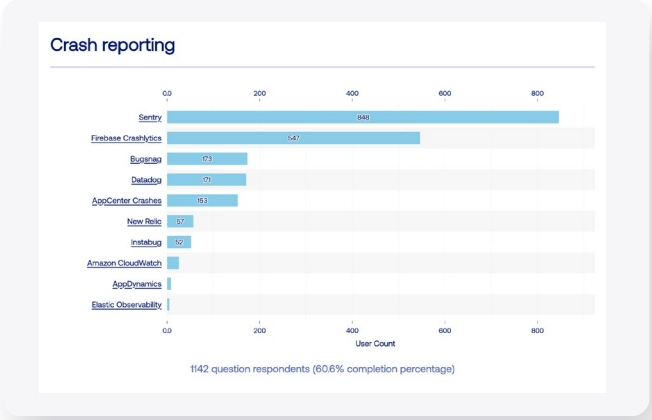


- Suspect Commits to uncover the commits that introduced the error
- Tags to index and search key information like device type or OS
- Screenshots to see what your users saw when the app crashed
- Gain insight into the sequence of events that led to the error, as well as the line of code that contains the bug, with Stack Traces
- If the cause of an error isn't obvious, trace your user's steps with Breadcrumbs



In fact, in the recent [State of React Native survey](), Sentry was identified as the most popular crash reporting solution among survey respondents.

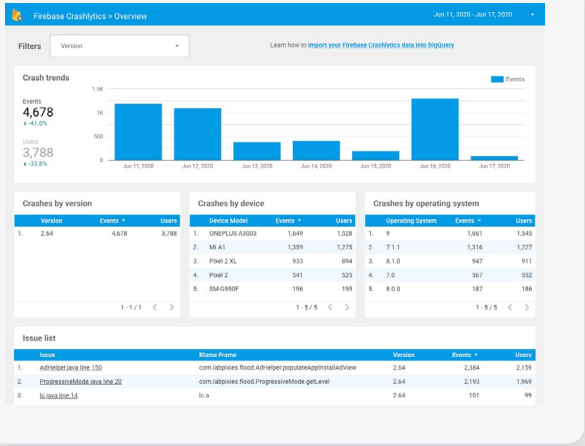
For React Native users, Sentry provides unminified stack traces.



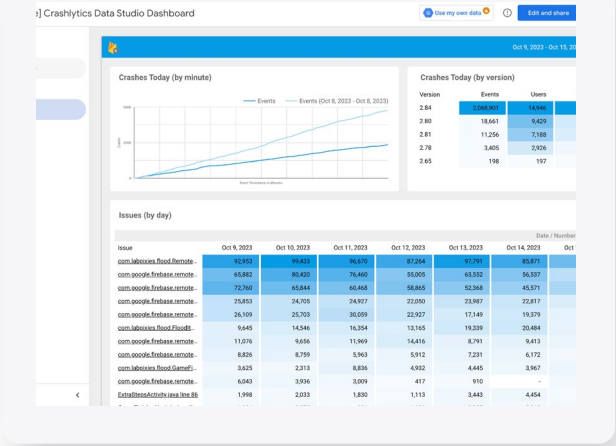
"We had been using Firebase Crashlytics but it does not support source maps from React Native, so it was always difficult trying to understand exactly where a crash originated from and what change might've introduced it. Sentry makes it easy to identify new crashes and where they are coming from."

Frikkie Snyman,
Mobile Platform Engineer, Relive

Lastly, let's take a look at the comparative insights and analytics offerings from Crashlytics and Sentry.



BigQuery gives you access to your raw Crashlytics data through an integration so you can get more in-depth crash reporting and run custom queries. You can set up ways to automate your release process.



Crashlytics provides a Data Studio template, which is powered by exported crash data.

If you are using BigQuery streaming, which adds additional costs, data is shown in real-time if you refresh the page.

Sentry provides two products that offer analytics tools built right into the platform: **Discovery and Dashboards**

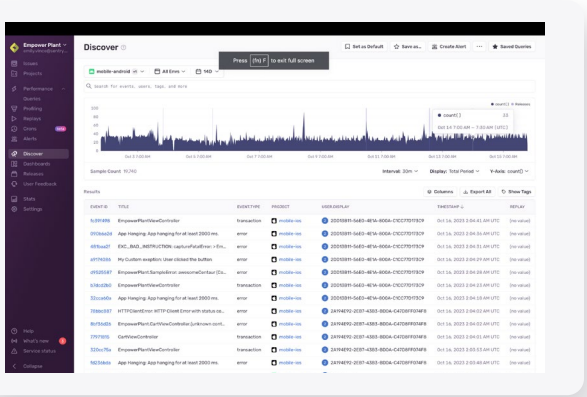
Discover provides visibility into your data across environments by building upon and enriching your error data. You can query and unlock insights into the health of your entire system and get answers to critical business questions — all in one place. Use Discover to view comprehensive information sent to Sentry.

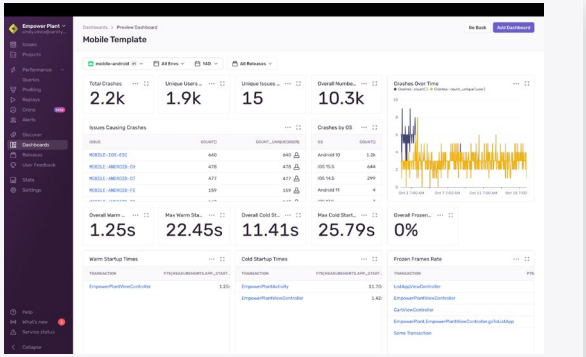
Sentry's Dashboards provide you with a broad overview of your application's health by allowing you to navigate through crash and error across multiple projects. Dashboards are made up of one or more widgets, and each widget visualizes one or more Discover queries.

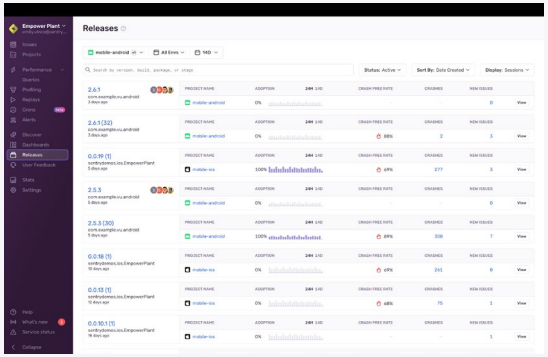
You can build custom queries in Discover and custom Dashboards on Sentry's Business plan.

The Releases page provides a visualization of your releases. It presents adoption of releases from the past 24 hours and provides a high-level view of:

- Each release version
- The associated project
- The adoption stage of each release
- The authors of each commit
- The percentage of crash-free users
- The percentage of crash-free sessions





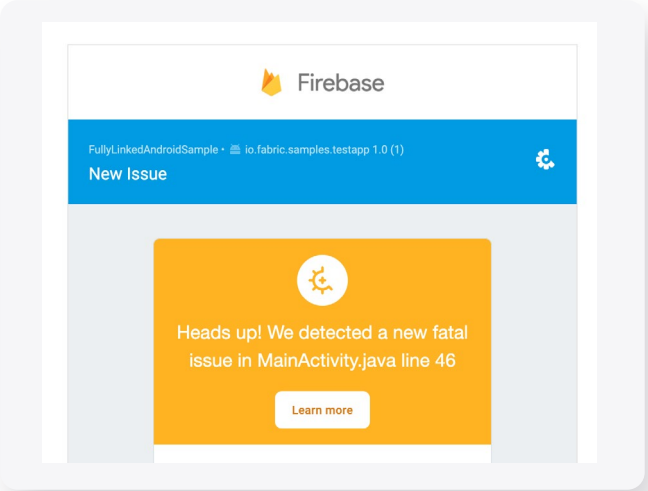


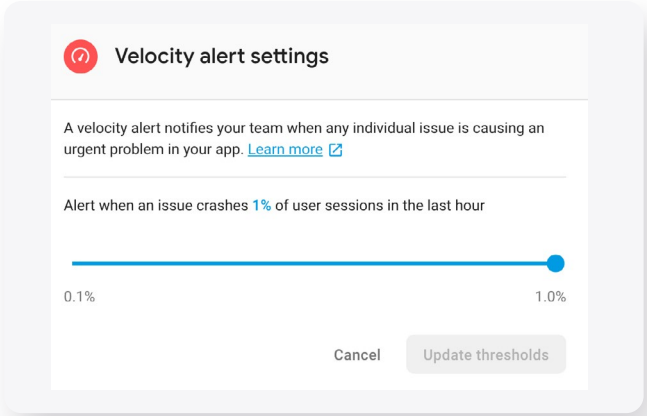
Workflow

Workflows are an area where you will see clear advantages to Sentry over Crashlytics, especially for growing teams that rely on collaboration to ship and improve their apps. Most workflows to solve crashes start by finding out that there is a problem. So let's take a look at alerts and notifications.

Crashlytics offers the following pre-set alerts:

- New issue alerts
- Regressing detection alerts
- Trending issue digests
- Velocity alerts



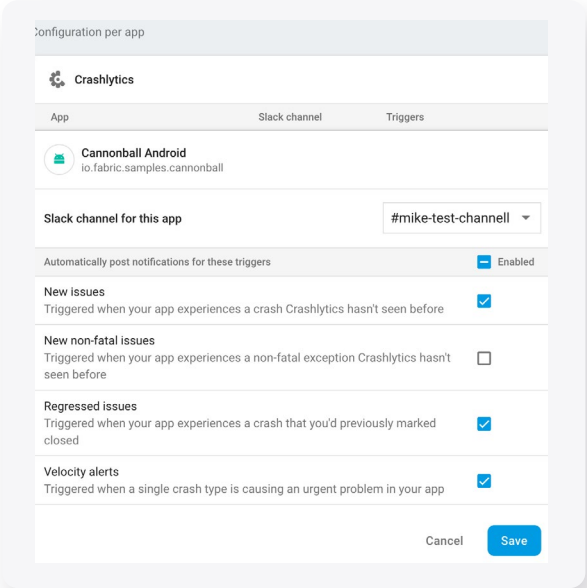


You can adjust the Velocity alert settings.

By default, Firebase will send Crashlytics alerts via email to every project member.

For velocity alerts and regressions, Firebase will show these alerts in the Firebase console. Firebase can also send one-way summaries to Slack, Jira, or PagerDuty.

On the paid Blaze plan for Firebase, you can use Cloud Functions to send customized notifications for velocity alerts to additional third-party systems.



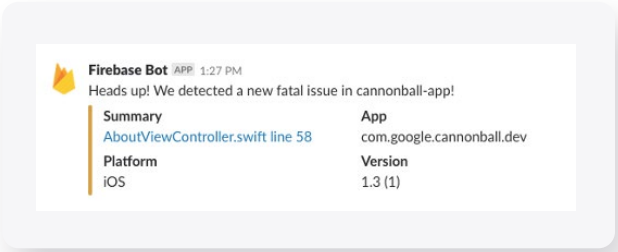
With Crashlytics, you cannot assign issues to a specific team or developer. As such, you cannot send notifications only to an assigned person or team. One developer's bug becomes everyone's problem.

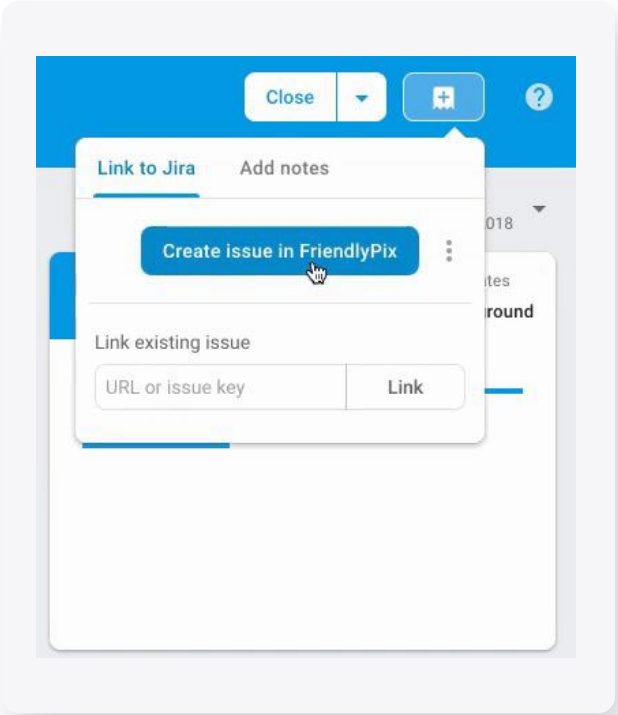
"There's a lot of noise [with Crashlytics] and it's not useful noise. We'll get alerts all hours of the night. I'll get alerts that something's going wrong and it ends up being one user causing a lot of noise. And so the alerting I find to not be super helpful. Alerts should be intelligent. And when you're saying that there's a critical issue and it impacts five users... not worth waking the team up for."

Head of Engineering, Music Streaming Service

Crashlytics notifications to Slack and Jira are “one-way”, meaning that updates don’t sync between tools. If you receive a notification in Slack, you then must log back into Crashlytics to take action.

Similarly, you can create tickets in Jira with Crashlytics, but any updates made in Jira will not be automatically reflected in Crashlytics.



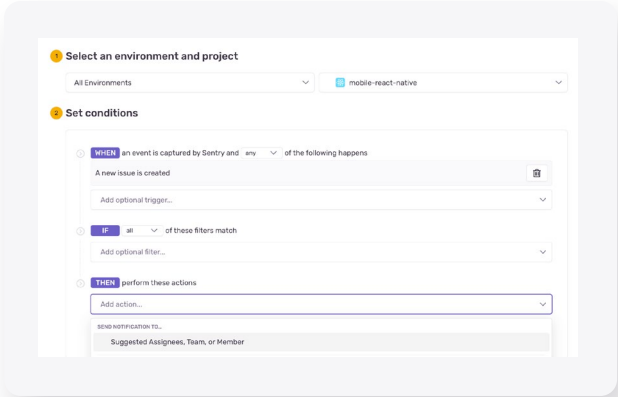


Let's take a look now at the ability to customize your workflow with Sentry, which can help developers stay focused only on crashes and bugs that need their attention.

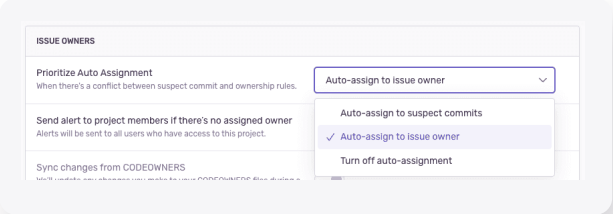
Sentry offers fully customizable alerts that allow you to fine-tune notifications and reduce noise.

You can create two types of alerts:

- Issue alerts that trigger when an issue matches a specific criteria
- Metric alerts when macro-level metrics cross specific thresholds

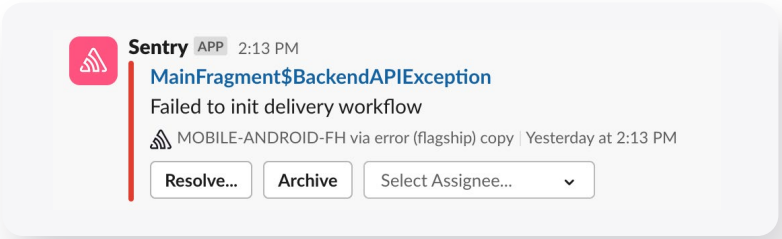


With Ownership Rules, you can automatically assign issues and send alerts to the specific team or developer that is best positioned to solve the issue fast. With Code Owners, you can import GitHub or GitLab CODEOWNERS and automatically assign issues according to those file paths.

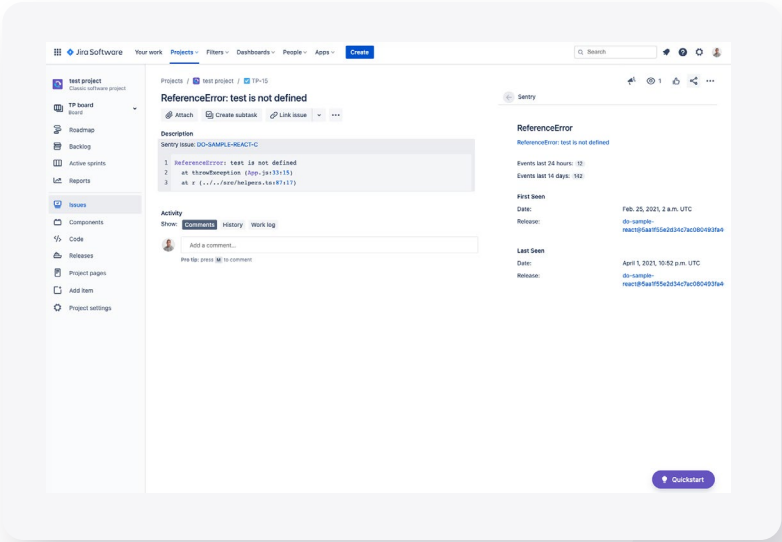


You can choose to notify specific teams or developers via email, Slack, multiple supported integrations (Opsgenie, PagerDuty, Discord, Microsoft Teams, Jira, and more), and custom integrations through webhooks.

Sentry offers two-way integrations with many communication and issues tracking tools. For example, you can triage, resolve, and archive Sentry issues directly in Slack, without switching back to Sentry.



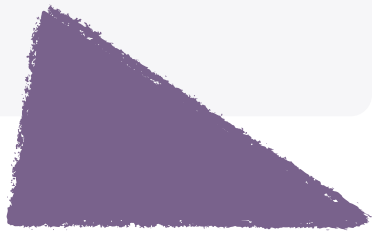
With our Jira integration, Sentry will automatically create Jira tickets. Any updates made to the Jira ticket, like delegating the issue to an assignee or updating a status on Jira, will also populate in Sentry.



If you are building a mobile app solo or with a very small team, you may not notice minor snags in your workflow. But, as your app grows and multiple developers are collaborating across teams, inefficiencies in your workflow can really slow you down.

Sentry's workflow tools can streamline processes for your team, helping them identify and solve crashes faster, allowing them to deliver more stable apps and focus more time on building more features for their customers.

> ***"Our workflow has changed from being reactive to proactive, with the ability for each team to monitor issues over time, customize alerts based on what's important for their product area."***
>
> **Senior Head of Engineering, Global Food Delivery Service**



Innovation and Scale

Let’s move on to the final part of our product Crashlytics vs Sentry comparison: innovation and scale. Here we will start with Sentry.

Since launching our first mobile SDKs for Android and iOS in 2016, Sentry has continued to invest in our mobile offerings. We have especially doubled down in the last three years as mobile adoption has continued to accelerate after the pandemic.

Here is a look at Sentry's recent crash reporting investments, which are focused on helping developers solve crashes and non-fatal errors faster and adopt new technologies.

2020

- **Release Health**
- **Semantic Versioning**
- **Android Symbol Collection**
- Andrioid SDK v3.0
- Cocoa SDK v6.0
- React Native SDK v2.0
- Flutter SDK launched

2021

- **Mobile Screenshots**
- Android SDK
  - Auto activity instrumentation
- Cocoa SDK v7.0
- React Native SDK v3.0
- Flutter SDK v6.0
- Unity SDK launched

2022

- **Dexguard Support**
- Android SDK v6.0
  - Jetpack compose support
- Cocoa SDK
  - iOS App Hangs
- React Native Attachment Support
- Flutter SDK
  - Mobile Vitals

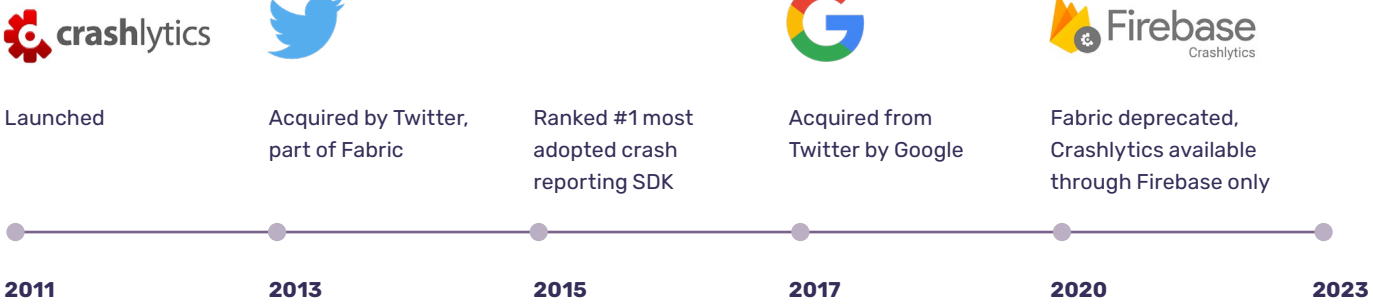
2023

- **View Hierarchy**
- **Device Classification**
- Android SDK v7.0
  ANR improvements
- Cocoa SDK v8.0
  - SwiftUI support
- React Native SDK 5.0
  New architecture support
- Flutter SDK v7.0
  - Screenshots
- Unity SDK v1.0

“Sentry’s team is often working on something interesting, at a super fast cadence. Other tools I’ve used before just can’t compare in that regard. We’re keeping an eye on their Kotlin Multiplatform SDK and we might adopt it eventually. ”

Fred Porciuncula, Mobile Engineer, Planet Wild

Let's take a look at the Crashlytics product investment areas. A quick reminder of their history helps inform their investment approach.



Crashlytics was first launched in 2011. Two years later, in 2013, it was acquired by Twitter, and integrated with the Fabric developer platform. In 2017, Google acquired Crashlytics and then focused development over the next three years on integrating Crashlytics with the Firebase Platform, eventually deprecating Fabric and the previous version of Crashlytics.

Many of the major updates to Crashlytics are around a deeper integration with the Firebase ecosystem, rather than investments in new technologies in mobile app development (Jetpack Compose support, for example).

2020

Firebase ecosystem

- Firebase Crashlytics SDK (removing Fabric dependencies)

2021

- Crashes correctly attributed to .kt files
- Bulk issues management
- Crashlytics Signals

2022

Firebase ecosystem

- Firebase Crashlytics insights in Android Studio

Other updates

- Custom alerting with Cloud Functions for Firebase
- Updated Flutter plugin to track fatal errors on demand
- Filter crashes by Google Play track
- dSYM file “Drag and Drop”
- Moved event type into new metadata line

2023

Firebase ecosystem

Navigate from stack trace to code in Android Studio

Other updates

- Filter Android app events by device category
- GWP-ASan reporting
- Crash grouping with variants

On a related note – it's important to consider the pros and cons of relying on a crash reporting tool that is so deeply embedded within the Firebase Platform. Although the Firebase Platform can simplify many aspects of mobile application development, it has its downsides and limitations, including:

CRASHLYTICS	SENTRY
• **Vendor lock-in:** Migrating off platform may be challenging.	Conversely, Sentry offers the following benefits: • **Vendor agnostic:** Sentry is vendor-agnostic and supports various platform integrations as a first-class citizen. If you decide to use non-Google products (e.g. Postgres or Mongo instead of BigQuery for database querying), Sentry can integrate easily while Firebase cannot.
• **Cost:** While Firebase offers a free tier, it can become expensive as your app scales. You may find yourself incurring costs for services like Cloud Firestore, Realtime Database, or Firebase Hosting.	• **Cost:** Sentry offers flexible and transparent user-based pricing.
• **Limited investment in iOS and React Native:** While advertised as a cross-platform solution, Firebase focuses more on Android and Flutter (both part of the Google ecosystem) instead of iOS and React Native.	• **Equal investment across multiple mobile SDKs** including Android, iOS, React Native, Flutter, Unity, and others.
• **Data ownership and privacy:** Storing sensitive or critical data in Firebase means it's hosted on Google's servers. Depending on your app's use case, this could raise concerns about data ownership and privacy.	• **Data ownership and privacy:** As part of GDPR requirement for data residency, you can opt into having Sentry host your data in the EU.

Assessing ROI

We understand that for many users Firebase Crashlytics may seem like the more economical choice, especially if you already use many parts of the Firebase Platform and Crashlytics is included at no cost. When assessing the true cost of a solution, it's important to assess the financial impact of not solving crashes fast. You can try Sentry free and see for yourself.

There are a few variables to consider:

Customer Acquisition Costs

Given the competitive landscape, especially for consumer apps, it is expensive to differentiate and market a mobile app. You need to develop and pay for ad placement to target your primary audience segments.

On average, it costs $75 to acquire a *paying* mobile user.

Consider that 62% of users will uninstall an app if they experience crashes, freezes, or other major errors.

If you and your team are unable to solve a crash fast, **that crash could cost up to $4,650 for every 100 users of your app**, just in terms of acquisition costs

Customer churn costs

You also need to take into account losses in terms of customer value.

Given the competitive landscape, especially for consumer apps, it is expensive to differentiate and market a mobile app. You need to develop and pay for ad placement to target your primary audience segments.

On average, it costs $75 to acquire a *paying* mobile user.

Consider that 62% of users will uninstall an app if they experience crashes, freezes, or other major errors.

If you and your team are unable to solve a crash fast, **that crash could cost up to $4,650 for every 100 users of your app**, just in terms of acquisition costs

Developer productivity costs

One last lens to consider when evaluating the ROI of your crash reporting tool is the cost to developer productivity.

Although developer productivity cannot depend on a single metric, one important metric to consider when it comes to crash reporting is MTTR or Mean Time to Repair. It measures the time from when the incident is first identified to the time it is fixed. You can calculate MTTR by adding up the total time spent on repairs during any given period and then dividing that time by the number of repairs.

Without proper alerting, assignment, and debugging tools, MTTR can be very high. If your team needs to solve one crash every week, and they spend one hour longer solving that crash with Crashlytics than they would with Sentry, that is **more than a full week of developer time spent on fixing crashes alone.** That is time that could be spent focused on further developing the feature set of your app.

Getting started with Sentry

If you would like to see for yourself how Sentry can help your team build crash free apps with less developer time, it's easy to get started – for most mobile SDKs, you just need to run one line of code.

Get Started for free,

START TRIAL

Android:	`brew install getsentry/tools/sentry-wizard && sentry-wizard -i android`
iOS:	`brew install getsentry/tools/sentry-wizard && sentry-wizard -i ios`

React Native:	Add the @sentry/react-native dependency:
	`npx @sentry/wizard@latest -s -i reactNative`
	
	Wrap your root component
	`export default Sentry.wrap(App);`

Flutter	Get the SDK from `pub.dev` by adding the following to your `pubspec.yaml`
	`dependencies:`
	`  sentry _ flutter: ^7.10.1`
	
	Import `Sentry` and initialize it:
	`  import 'package:flutter/widgets.dart';`
	`  import 'package:sentry _ flutter/sentry _ flutter.dart';`
	
	`Future<void> main() async {`
	`  await SentryFlutter.init(`
	`  (options) => options.dsn = 'https://<key>@sentry.io/<project>',`
	`  appRunner: () => runApp(MyApp()),`
	`  );`
	`  // or define SENTRY _ DSN via Dart environment variable (--dart-define)`
	`  }`

In conclusion

Now that you have a firm understanding of the differences between Sentry and Crashlytics, you can make an informed decision. If you are building a prototype app solo or with a small team and are deeply invested in the Firebase Platform, then Crashlytics may be a fine choice.

If you and your teammates are committed to building crash-free apps in less time, then Sentry is an easy choice. With Sentry, you can find and fix errors fast with deeper context and insights into solving crashes and errors, customizable workflow tools, and product investments to enable innovation and scale.

To learn more:

Join our Discord

Check out GitHub

Read our docs

Still have questions about how Sentry can uplevel your mobile crash reporting? Talk to a Sentry expert to request a demo.

REQUEST A DEMO