



20CSI605 - Mobile Application Development

1



Crashlytics



Crashlytics

2

- ❑ **Firebase Crashlytics** is a lightweight, realtime crash reporter that helps you track, prioritize, and fix stability issues that erode your app quality.

Crashlytics helps using 3 main aspects :

- ❑ **Logs:** Each events in app is logged so that context can be provided alongwith crash reports on the event of app crash
- ❑ **Crash reports:** Crash reports is made up each time a crash occurs and sent up on the application being run the next time .
- ❑ **Stack traces:** If the app recovers from an error , Dart Stack trace help in still Reporting the error.



Firebase Crashlytics : Key Capabilities

3

- **Curated Crash Reports** — Firebase Crashlytics amalgamate an avalanche of crashes Into a well curated & identifiable list of issues providing contextual information highlighting the seriousness & pervasiveness of crashes so that you can easily pinpoint the root crash cause .
- **Cures for Common Crash** — Crashlytics provides Crash Insights highlighting the common stability problems . Also providing great resources that make them easier to troubleshoot — **Prioritize & Resolve** .



Firebase Crashlytics : Key Capabilities

4

- **Analytics Integration** — Crashlytics alongwith Analytics provide Audience Insights with Crash Analytics report for users & Simplify debugging by giving you access a list of other events leading up to each crash .
- **Realtime alerts** — Crashlytics will help you get realtime alerts for new issues, regressed issues, and growing issues that might require immediate attention.



Implementation Pattern : Firebase Crashlytics

5

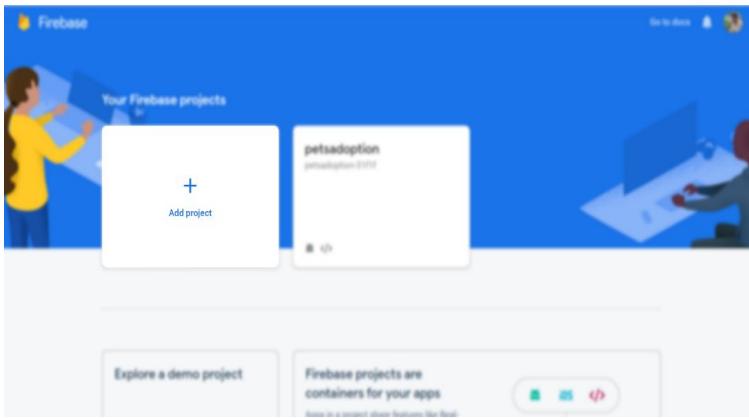
- Firebase Crashlytics saves you troubleshooting time by Intelligently Grouping Crashes and Highlighting the circumstances that lead up to them.
- **Connect Your App** — Add On Firebase To Your Apps.
- **Integrate The SDK** — Add the Crashlytics SDK
- **Check Firebase Console** — Use the Firebase console to track, prioritize, and fix issues in your app.



Firebase Crashlytics — Setup & Initial Configuration

6

- In order to Integrate Crashlytics, We need to create a firebase project for the Firebase Crashlytics from firebase.google.com by logging in with your Google account. This brings us to the following screen:



Create a project(Step 1 of 3)

Let's start with a name for your project®

Project name: **Firebase Crashlytics**

You're 2 projects away from the project limit. Consider adding Firebase to an existing project or request an increased limit.

Request an increase

Continue

Create a project(Step 2 of 3)

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, Predictions and Cloud Functions.

Google Analytics enables:

- A/B testing
- User segmentation and targeting across Firebase products
- Predicting user behaviour
- Crash-free users
- Event-based Cloud Functions triggers
- Free unlimited reporting

Enable Google Analytics for this project Recommended

Previous Continue

- Click on Add Project Button (+) initiating firebase project creation process
- You can either select the firebase analytics and then clicking on Continue Project.



Firebase Crashlytics — Setup & Initial Configuration

7

Screenshots

The image displays three screenshots of the Firebase Crashlytics interface:

- Project Setup:** Shows the initial project setup screen with a green checkmark indicating "Your new project is ready". A "Continue" button is visible.
- Welcome Screen:** Shows the main Firebase dashboard with the Crashlytics service highlighted. It features a "Get started by adding Firebase to your app" section, developer icons (iOS, Android, Java, Swift), and a call to action "Add an app to get started".
- Crashlytics Dashboard:** Shows detailed crash statistics for the "Flutter Demo App". The "Crash-free statistics" chart shows 100% crash-free users on Sep 30, 2020. The "Event trends" section shows 0 crashes and 0 users affected. The "Issues" section lists open issues with a search bar and filter for "Issue state = 'Open'".



Adding Crashlytics plugin In Flutter App :

8

- Add Package Dependency in **pubspec.yaml** and then run **Flutter pub get** :

```
firebase_crashlytics: "^\u2022.0.0"
```

- Add the following code in **build.gradle** file :

```
dependencies {  
    classpath 'com.google.firebase:firebase-crashlytics-gradle:2.2.0'}
```
- Apply the following Plugin to the end of your **android/app/build.gradle** file :

```
apply plugin: 'com.google.firebase.crashlytics'
```

- Add the following code in **build.gradle** file of our **android/app** folder :

```
dependencies {  
    // Adding Firebase Crashlytics SDK  
    implementation 'com.google.firebase:firebase-crashlytics:17.0.0-beta01'  
}
```



Adding Crashlytics plugin In Flutter App :

```
void main() {  
  // Set `enableInDevMode` to true to see reports while in debug mode  
  // This is only to be used for confirming that reports are being  
  // submitted as expected. It is not intended to be used for  
  // everyday development.  
  
  Crashlytics.instance.enableInDevMode = true;  
  
  // Pass all uncaught errors to Crashlytics.  
  FlutterError.onError = Crashlytics.instance.recordFlutterError;  
  WidgetsFlutterBinding.ensureInitialized();  
  
  SystemChrome.setEnabledSystemUIOverlays([SystemUiOverlay.bottom]).then((  
    _)  
  {  
    SharedPreferences.getInstance().then((prefs) {  
      var darkModeOn = prefs.getBool('darkMode') ?? true;  
      runZoned(() {  
        runApp(ChangeNotifierProvider<ThemeNotifier>(  
          create: (_) => ThemeNotifier(darkModeOn ? darkTheme :  
lightTheme),  
          child: MyApp(),  
        );  
      }, onError: Crashlytics.instance.recordError);  
    });  
  });  
}
```

Rebuild your app:

\$ flutter run



Adding Crashlytics plugin In Flutter App :

Firebase

Project Overview

Develop

- Authentication
- Cloud Firestore
- Realtime Database
- Storage
- Hosting
- Functions
- Machine Learning

Quality

- Crashlytics
- Performance
- Test Lab

Extensions

Spark Free \$0/month

Upgrade

Flutter Demo App

Crashlytics

Flutter Demo App

Go to docs

M

?

Last 7 days
24 Sep – 30 Sep

Filter Event type = "Crashes" X

Event trends

Crashes Users affected

0 0

Latest Release

No data
Try selecting another time range

Crash-free statistics

Crash-free users

100% 50% 0%

Sep 24 Sep 27 Sep 30

Sep 30, 2020

Crash-free users 100.00%

Issues

Search by user ID

Filter issues Issue state = "Open" X

Search issue title or subtitle

Issues Details Versions Events Users



THANK YOU