



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

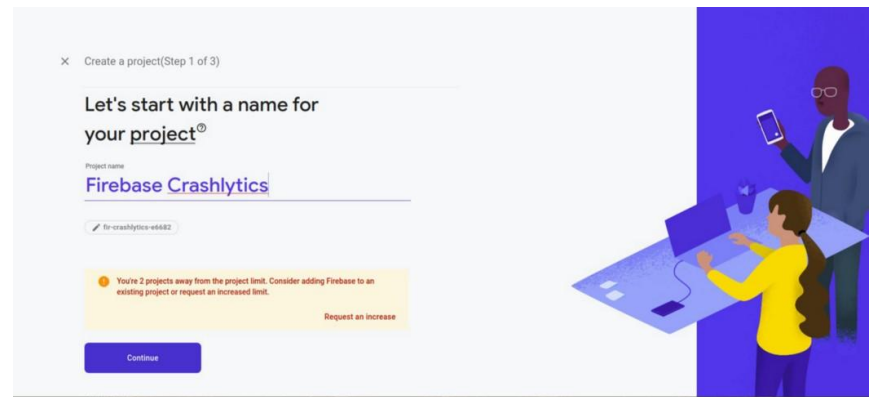
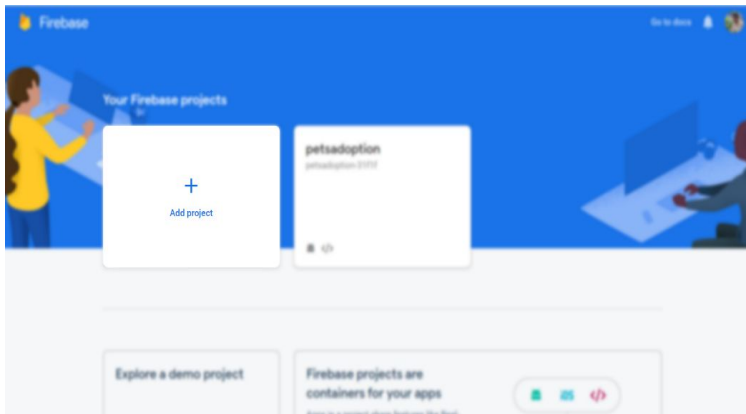
- Firebase Crashlytics saves you troubleshooting time by Intelligently Grouping Crashes and Highlighting the circumstances that lead up to them.
- **Connect Your App** — Add On Firebase To Your Apps.
- **Integrate The SDK** — Add the Crashlytics SDK
- **Check Firebase Console** — Use the **Firebase console**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



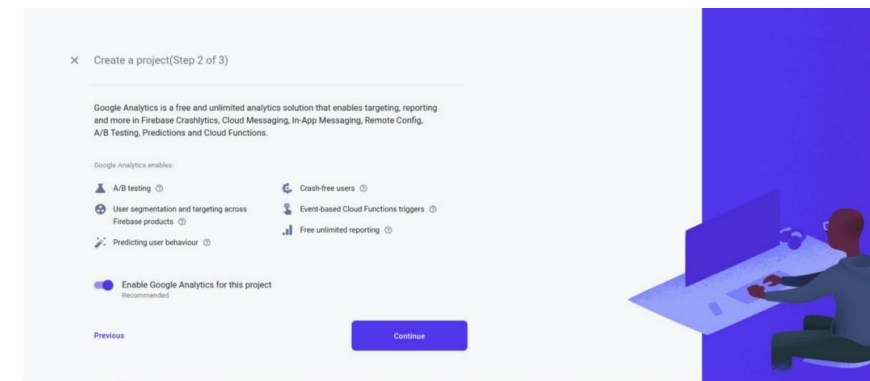
Firestore Crashlytics — Setup & Initial Configuration

6

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



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

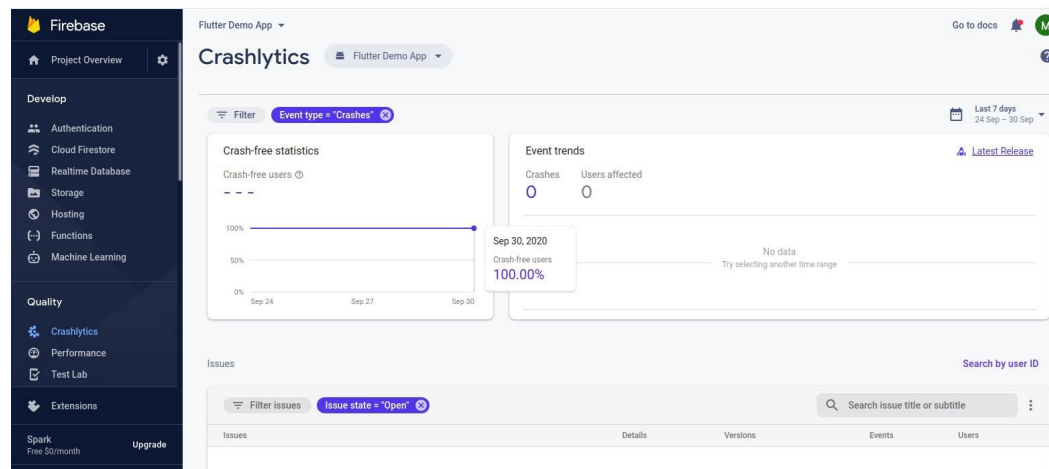
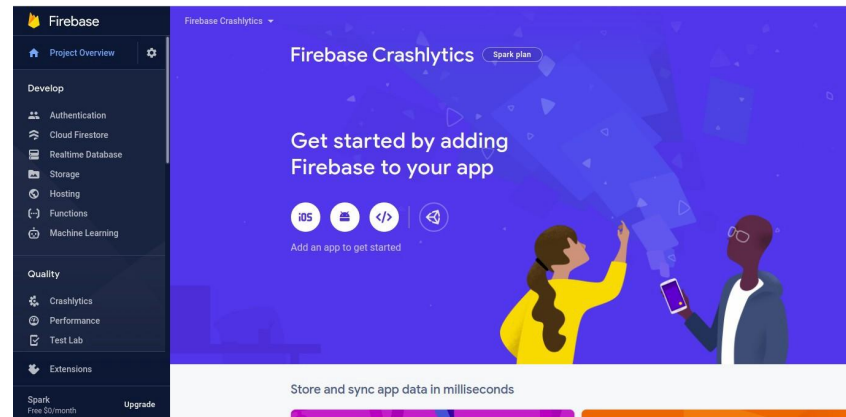
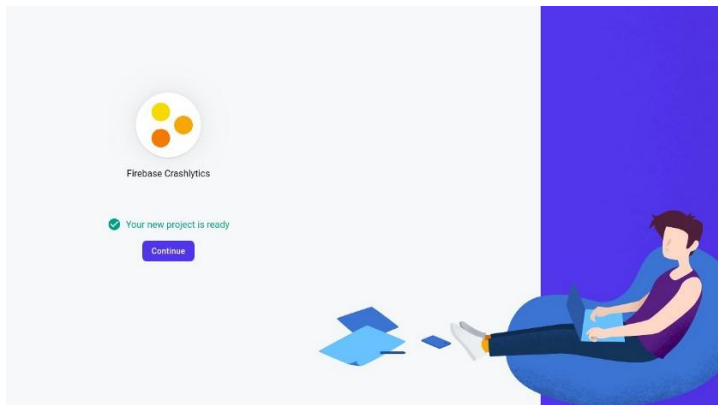




Firebase Crashlytics — Setup & Initial Configuration

7

□ Screenshots





Adding Crashlytics plugin In Flutter App :

8

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

```
firebase_crashlytics: "^0.2.0"
```

- *Add the following dependencies to your **android** build file:*

```
dependencies {  
    classpath 'com.google.firebase:firebase-crashlytics-gradle:2.2.0'}
```

- *Apply the following Plugin to the end of your **android** build file:*

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

- *Add the following dependencies to your **android** build file:*

```
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 :

The screenshot displays the Firebase Crashlytics interface for the 'Flutter Demo App'. The left sidebar contains the Firebase navigation menu with sections for Develop, Quality, and Spark. The main content area shows the Crashlytics dashboard with a filter for 'Event type = Crashes'. The 'Crash-free statistics' chart shows 100% crash-free users for the period Sep 24 to Sep 30, 2020. The 'Event trends' section shows 0 crashes and 0 users affected. The 'Issues' section at the bottom shows a filter for 'Issue state = Open' and a search bar for issue titles or subtitles.

Crashlytics Dashboard Data:

Section	Filter	Value
Crash-free statistics	Event type = "Crashes"	100.00% (Sep 30, 2020)
Event trends	Crashes	0
Event trends	Users affected	0
Issues	Issue state = "Open"	0



THANK YOU