



GoferDeliveryAll User App Android Configuration

Trioangle Technologies Pvt Ltd

Madurai, Tamil Nadu, India.

www.trioangle.com

support@trioangle.com

© trioangle

Table of Contents

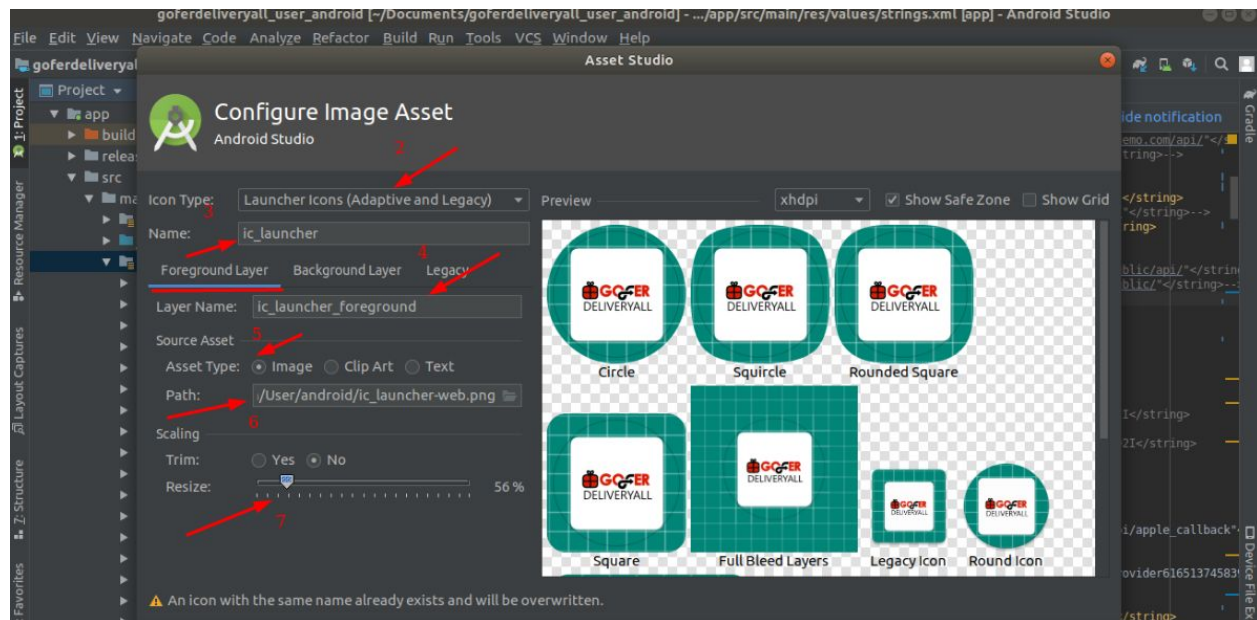
| | |
|--|-----------|
| Table of Contents | 2 |
| Updating APP Icon | 3 |
| Steps To Update App Icon | 3 |
| Reference Below | 3 |
| Updating App Logo | 5 |
| Steps To Update App Logo | 5 |
| Reference Below | 5 |
| Updating App Name, Base URL and Google API Key | 5 |
| Reference Below | 6 |
| Updating FCM Key (Firebase Cloud Messaging) | 7 |
| Steps To create Firebase Database | 8 |
| Reference Below | 8 |
| Update below firebase realtime db rules | 10 |
| Steps To update google-service.json | 13 |
| Reference Below | 13 |
| Set up Crashlytics in the Firebase console | 14 |
| Click Crashlytics in the left-hand nav panel of the Firebase console. | 14 |
| If your Firebase project has multiple apps registered in it, select the app you just added from the dropdown next to Crashlytics in the top bar of the console. | 14 |
| Click Set up Crashlytics. | 14 |
| Select This app is new to Crashlytics | 14 |
| API Config document | 14 |

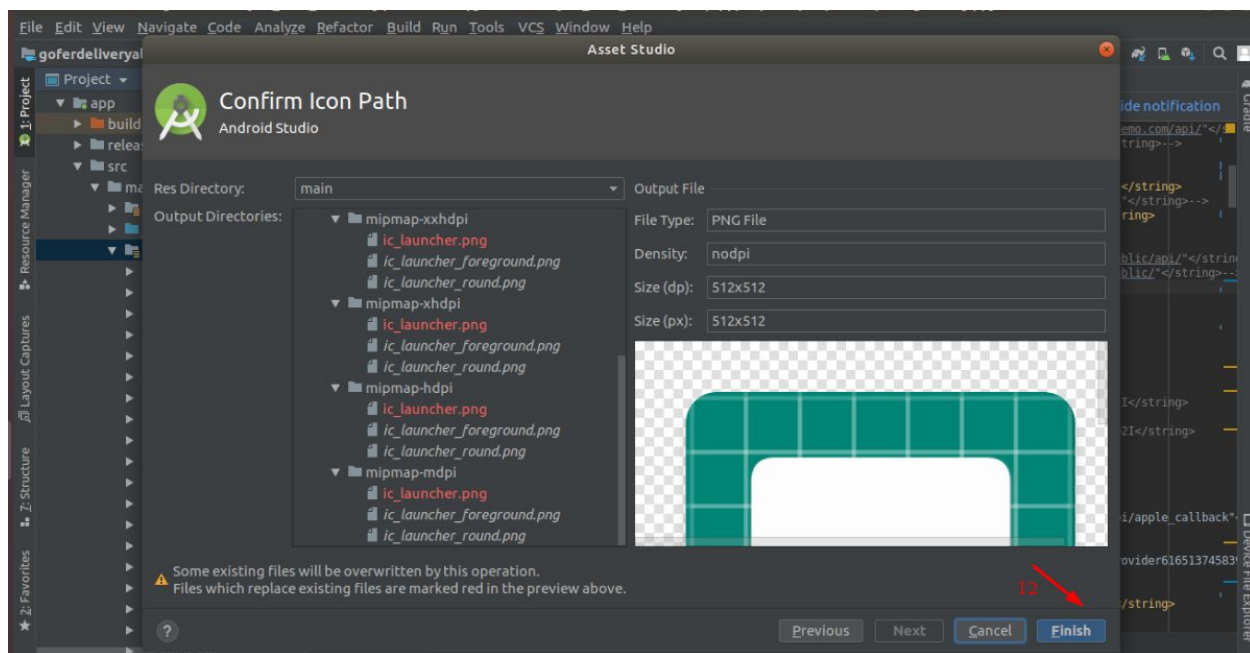
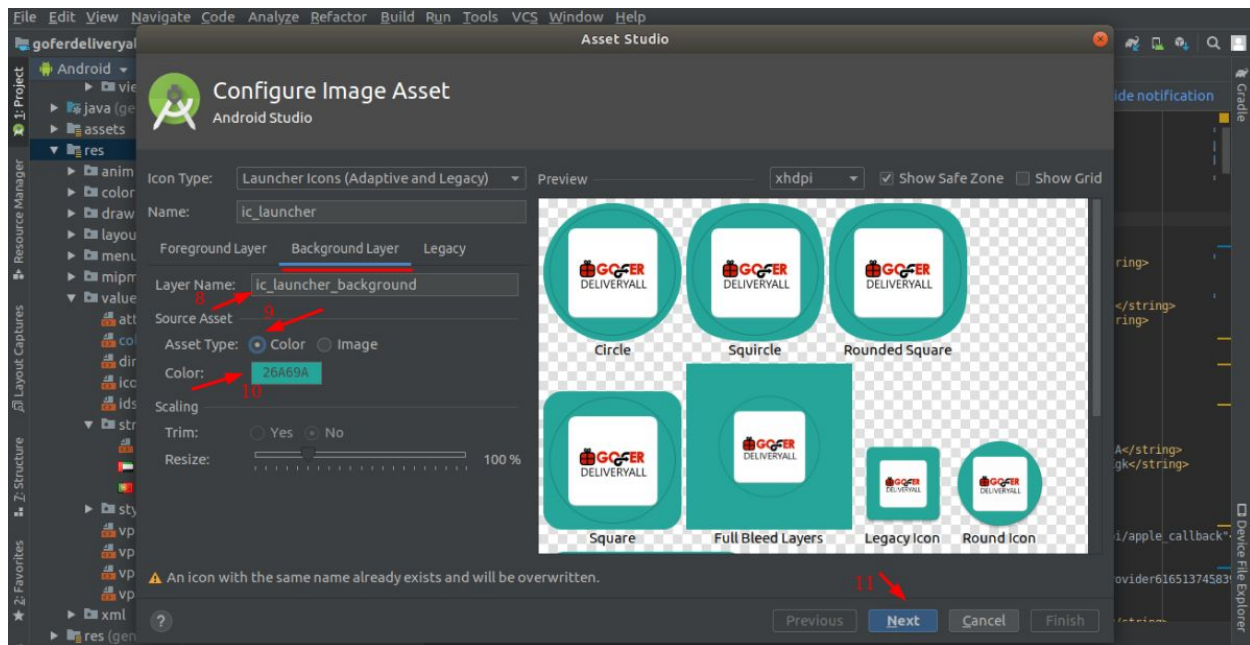
Updating APP Icon

Steps To Update App Icon

1. Click File → New → Image Asset
2. Set Icon type as Launcher Icons (Adaptive and Legacy)
3. Set your image name as ic_launcher.
4. Foreground Layer → Set your Layer Name as ic_launcher_foreground.
5. Set Asset Type to Image
6. Foreground Layer → Path and select your icon
7. Resize your icon (if needed)
8. Background Layer → Set your Layer Name as ic_launcher_background.
9. Background Layer → Source Assets → Asset Type → Color
10. Set your background color for the icon
11. Click Next and at the next page click finish

Reference Below





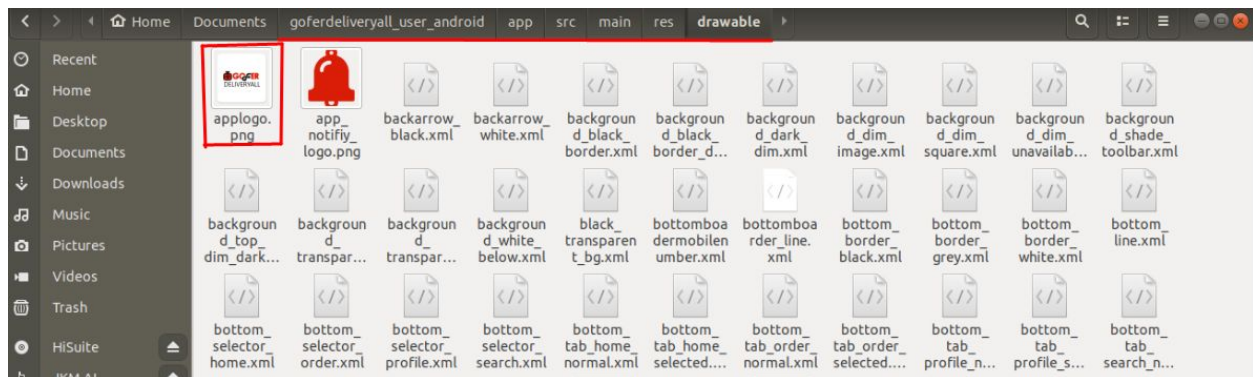
Updating App Logo

- Updating app logo

Steps To Update App Logo

1. To change the app logo image follow the steps
 - a. Set your image name as corresponding to selected image name like (*applogo.png*)
 - b. You can design your own image and change the below images.
 - c. In Source folder → app → src → main → res → drawable → replace corresponding image

Reference Below

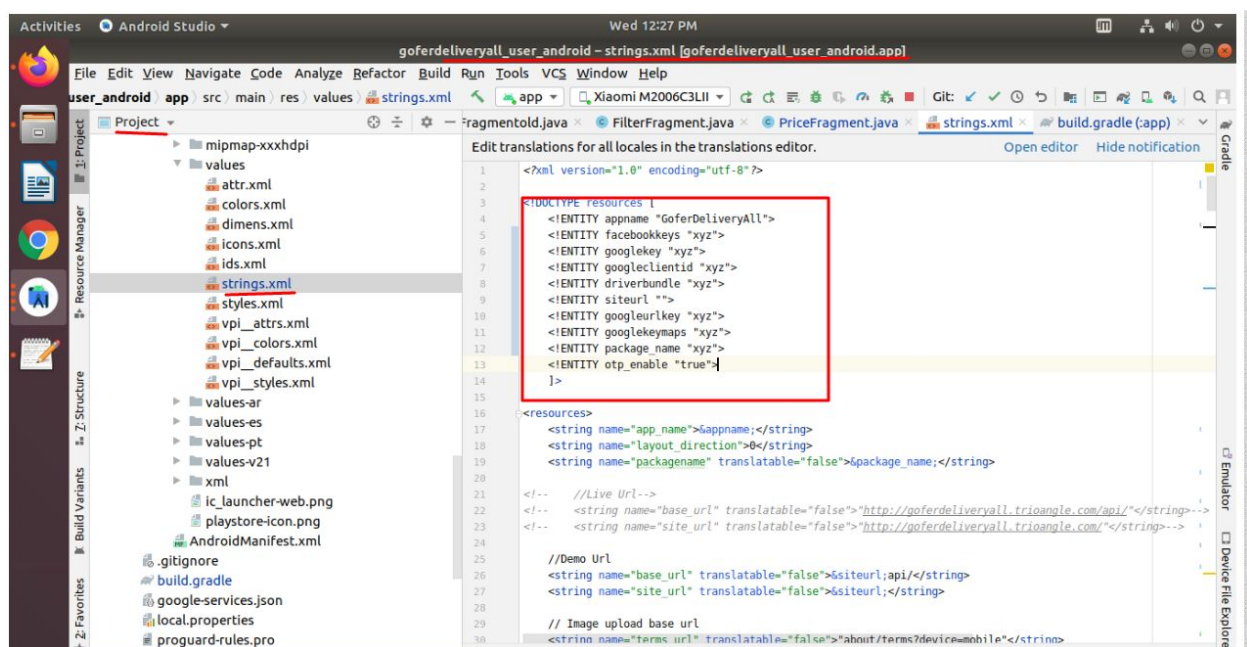


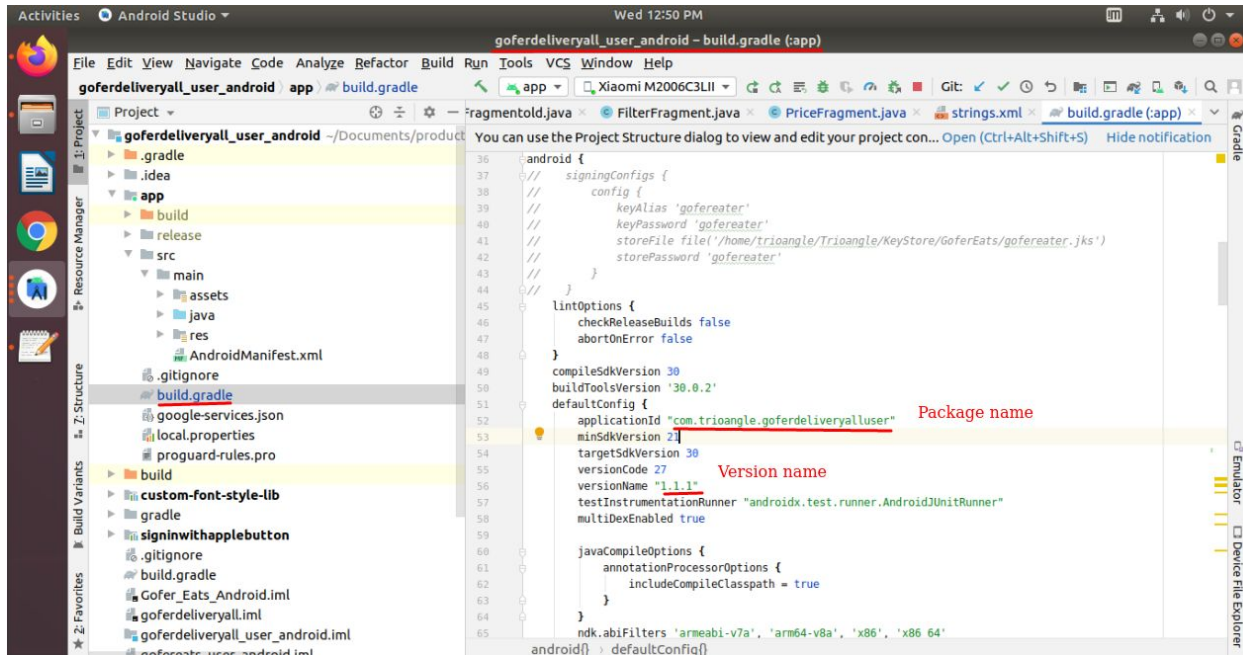
Updating App Name, Base URL and Google API Key

- Get **Google APP Key** used to authenticate into different types of Web **Application**, **Mobile Application**.

- Follow the link to create Google API Key <https://console.developers.google.com/> and follow the steps <http://www.devexchanges.info/2015/07/integrating-with-google-plus-google.html>
- To generate signed APK follow these steps <http://www.androiddocs.com/tools/publishing/app-signing.html>
- To release updated version change version name and version code and generate signed APK

Reference Below





Note:

- Use the same **Google API key** on the web and enable lib in google API.
- Change the **app name**.

Updating FCM Key (Firebase Cloud Messaging)

- Get **FCM Key** used to push notification into different types of Web Application, Mobile Application.
- Follow the link to create FCM <https://console.firebase.google.com>
- Create the project and then create app for all driver, eater and restaurant
- Change or get the **package name** to follow the steps
 - In Android studio → Open the project → After build success → Gradle scripts → build.gradle (Module: app) → Find applicationId (Get or change)
- Follow the links steps
 - <https://webkul.com/blog/generate-api-key-fcm-sender-id/>
 - <https://developers.intercom.com/docs/android-fcm-push-notifications>
- Select the app and Download the **google-services.json** from firebase console
- Get Server id and Legacy server key update in admin panel → site settings → API Credential → change **Fcm Sender Id** and **Fcm Sender key**

Note:

Use same FCM key in web and iOS also.

Steps To create Firebase Database

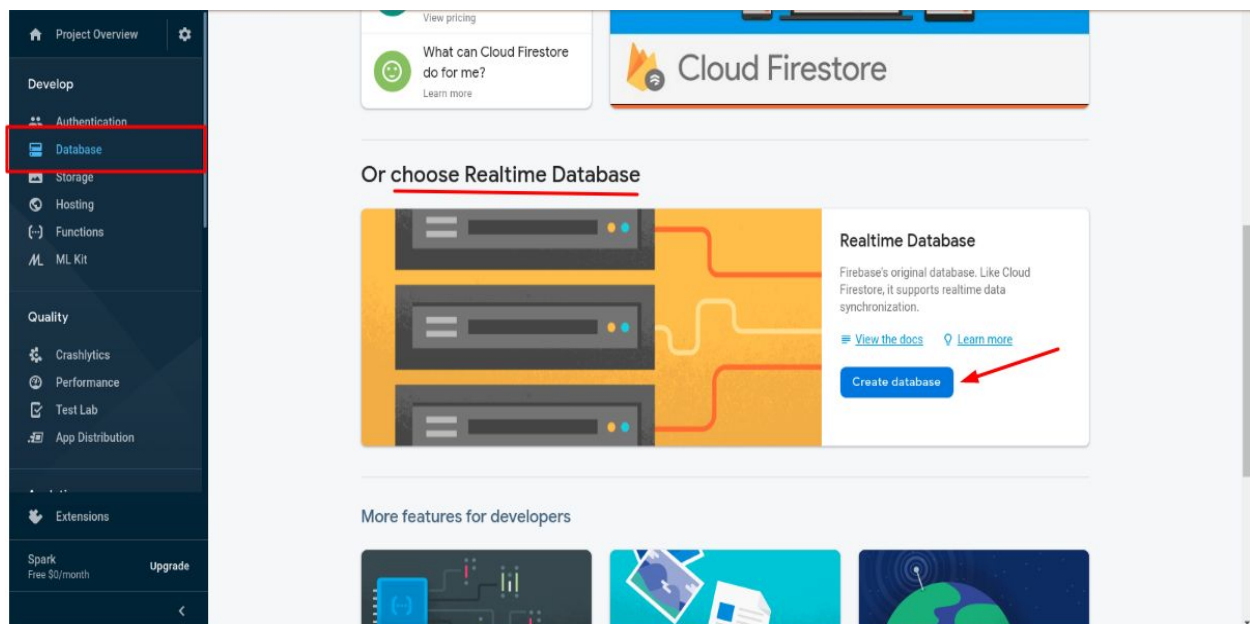
- On left side panel of console click **database**
- Choose **realtime database** Click create database
- Choose **start in test mode** and enable

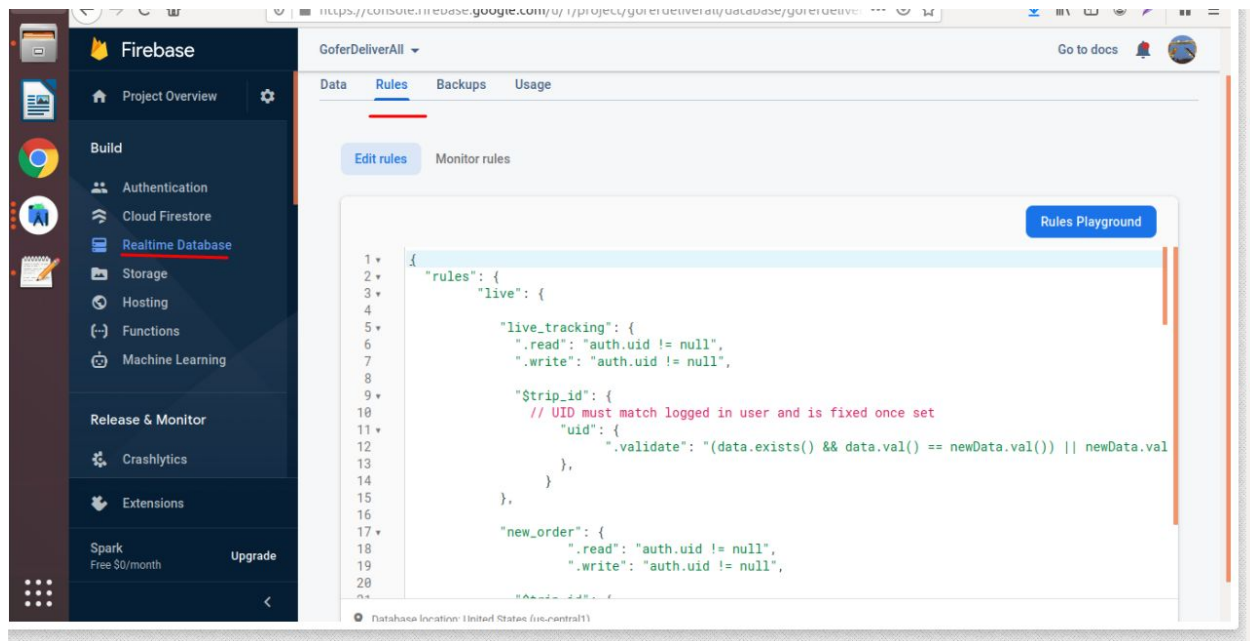
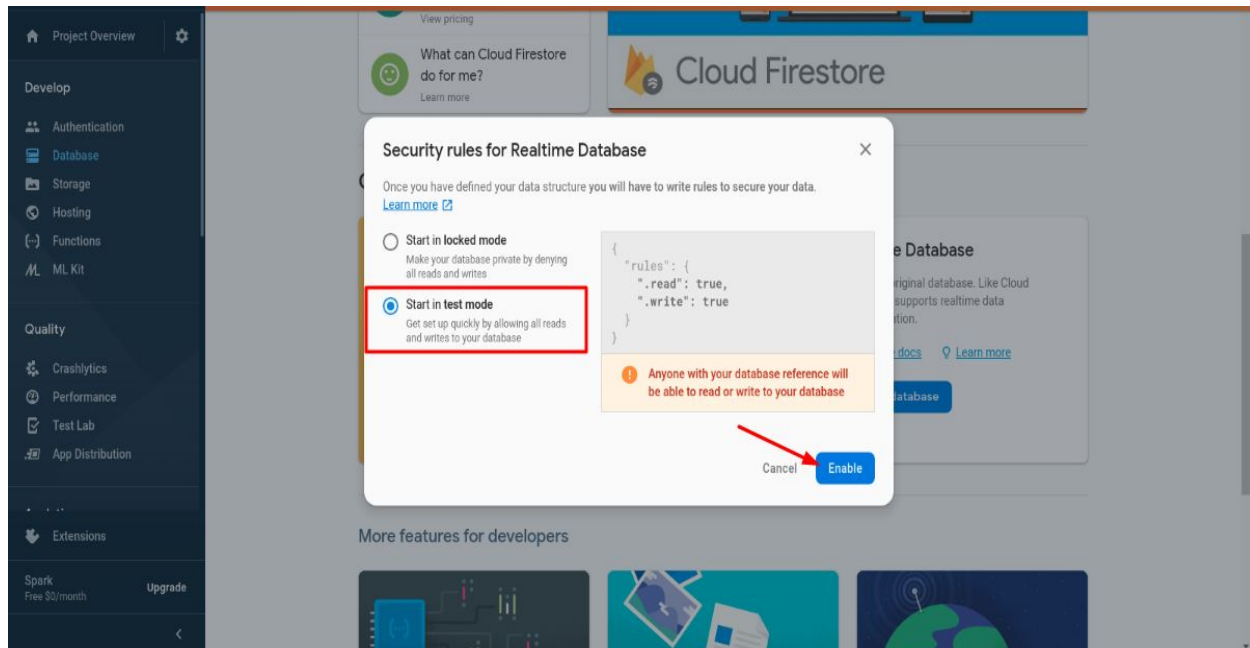
Note :

The Rules of the database must be **true** for **read** and **write**

- Get Server id and Legacy server key update in admin panel → site settings → API Credential → change **Fcm Sender Id** and **Fcm Sender key**

Reference Below





Update below firebase realtime db rules

```
{
  "rules": {
    "live": {

      "live_tracking": {
        ".read": "auth.uid != null",
        ".write": "auth.uid != null",

        "$trip_id": {
          // UID must match logged in user and is fixed once
          "uid": {
            ".validate": "(data.exists() && data.val()
set
== newData.val()) || newData.val() == auth.uid"
          },
        }
      },

      "new_order": {
        ".read": "auth.uid != null",
        ".write": "auth.uid != null",

        "$trip_id": {
          "uid": {
            ".validate": "(data.exists() && data.val()
== newData.val()) || newData.val() == auth.uid"
          },
        }
      },

      "trip": {
        ".read": "auth.uid != null",
        ".write": "auth.uid != null",

        "$trip_id": {
          "uid": {
            ".validate": "(data.exists() && data.val()
```

```

== newData.val()) || newData.val() == auth.uid"
        },
    },
    "trip_request": {
        ".read": "auth.uid != null",
        ".write": "auth.uid != null",

        "$trip_id": {
            "uid": {
                ".validate": "(data.exists() && data.val()
== newData.val()) || newData.val() == auth.uid"
            },
        },
    },
    "GeoFire": {
        ".read": "auth.uid != null",
        ".write": "auth.uid != null",
        ".indexOn": "g",
        "$trip_id": {
            "uid": {
                ".validate": "(data.exists() && data.val()
== newData.val()) || newData.val() == auth.uid"
            },
        },
    },
    "demo": {
        "live_tracking": {
            ".read": "auth.uid != null",
            ".write": "auth.uid != null",

            "$trip_id": {
                "uid": {
                    ".validate": "(data.exists() && data.val()
== newData.val()) || newData.val() == auth.uid"
                },
            },
        },
    },
}

```

```

    },
    "new_order": {
        ".read": "auth.uid != null",
        ".write": "auth.uid != null",

        "$trip_id": {
            "uid": {
                ".validate": "(data.exists() && data.val()
== newData.val()) || newData.val() == auth.uid"
            },
        }
    },

    "trip": {
        ".read": "auth.uid != null",
        ".write": "auth.uid != null",

        "$trip_id": {
            "uid": {
                ".validate": "(data.exists() && data.val()
== newData.val()) || newData.val() == auth.uid"
            },
        }
    },

    "trip_request": {
        ".read": "auth.uid != null",
        ".write": "auth.uid != null",

        "$trip_id": {
            "uid": {
                ".validate": "(data.exists() && data.val()
== newData.val()) || newData.val() == auth.uid"
            },
        }
    },

    "GeoFire": {
        ".read": "auth.uid != null",

```


Set up Crashlytics in the Firebase console

- Click **Crashlytics** in the left-hand nav panel of the [Firebase console](#).
- If your Firebase project has multiple apps registered in it, select the app you just added from the dropdown next to Crashlytics in the top bar of the console.
- Click Set up Crashlytics.
- Select This app is new to Crashlytics

Note : The SDK's are already added

API Config document

- Follow the link to get API list and details.
- <https://documenter.getpostman.com/view/682036/RW8CHTWm>