

# NEW TAXI APP INSTALLATION GUIDE

New Taxi is a Complete Solution for Taxi Service. In this app User can easily Book Taxi and travel between different locations. User can call and chat the Cab driver for taxi status. In this App, all drivers must be verified by Admin after submitting valid documents.

We have Payment Gateways and Multi Currency Option, if you want to start own Taxi Service App. This is a Complete Solution.

## Features of NewTaxi

- ✓ Latest and high-end development technologies.
- ✓ Easy Sign Up / Sign In via Apple, Facebook & Google.
- ✓ Secure payment methods through PayPal, Stripe, Cash and Braintree.
- ✓ Easy Search and filter Options.
- ✓ SMS notifications and verifications.
- ✓ Instant Notification.
- ✓ User/Driver Wallet & Promo codes.
- ✓ User/Driver Referrals.
- ✓ In app Chat option.
- ✓ In app calls (Sinch calls).
- ✓ Ride Scheduling.
- ✓ Manual booking.
- ✓ Past/Current/Upcoming Trips History.
- ✓ Live Tracking.
- ✓ Driver/User Rating & Reviews.
- ✓ Optional Tips to Driver.
- ✓ Multiple Currency options.
- ✓ Fare Break Down Options.
- ✓ Multiple languages translation.

- ✔ Ride Cancellation option for trip.
- ✔ Payment Flow Updates.
- ✔ Covid Safety Features.

## Server Requirements

- ✔ Operating System – Linux
- ✔ Web Server – Apache
- ✔ Database – MySQL
- ✔ PHP – 7.4
- ✔ Ability to Setup Cron Jobs

## Installation Details

### Admin Backend and Frontend installation instructions.

#### Step 1

Download File source code from [codecanyon.net](https://www.codecanyon.net) and extract it.

Select Admin Panel Zip file.

Upload it to your Server Directory.

Upload the file via the File Manager

You can upload your files directly through cPanel using the File Manager. File Manager is a web interface that allows you to manage all files associated with your account.

cPanel's File Manager may look slightly different depending on which version of cPanel your server is running, however, all the functionalities are available in all.

To upload a file through cPanel's File Manager, please do the following:

- ✔ Login to your cPanel account.
- ✔ Under Files, click File Manager.
- ✔ Select Web Root and Show Hidden Files, then click Go.

- ✔ Click to open the folder to which you'd like to upload files.
- ✔ Click Upload from the top toolbar.
- ✔ Click Browse (some web browsers will have Choose File instead).
- ✔ Select the file on your computer that you download Admin Panel Zip file to upload.
- ✔ Once the file has been selected, it will automatically upload. You may select additional files while uploads are in progress.
- ✔ After uploading files to your account, Extract it your current Directory

## Database Configuration

First, Create MySQL Database and User

- ✔ Open Cpanel find MySQL Database
- ✔ Create MySQL Database
- ✔ Create MySQL User
- ✔ Select All User Privileges

Goto Cpanel Find phpMyAdmin

- ✔ Open phpMyAdmin
- ✔ Select Your Database
- ✔ Click Import
- ✔ Browse SQL File
- ✔ Select database.sql File (db.sql under database folder)

Second step, you need to type your database details in .env file.

**DB\_HOST=localhost**

**DB\_DATABASE= yourdatabasename**

**DB\_USERNAME= yourdbusername**

**DB\_PASSWORD=yourdbpassword**

**Admin Access**

Go to <http://yoursite.com/admin> after successfully configuration

We created an admin account for you by default

Username: admin

Pass: 1234567890

### System Requirements

- ✓ PHP version 7.4
- ✓ MySQL.
- ✓ The [OpenSSL](#) PHP module.
- ✓ The [PDO](#) PHP module.
- ✓ The [mbstring](#) PHP module.
- ✓ The [Tokenizer](#) PHP module.

The bcmath, mod\_rewrite, mod\_gzip, mod\_filter and mod\_expires modules.

### API Settings

You can enter your API credentials in Admin panel in API Credentials page.

Firebase - for push notification. [Instructions](#)

Twilio - for send sms notification. [Instructions](#)

Google API Keys (Browser and Server Key) - for Google Map and Places.

Please enable, "Google Places API Web Service", "Google Maps Geocoding

API" in Google Developer Console. [Server Key Instructions](#) [Browser Key Instructions](#)

Facebook APP ID [Login with facebook](#) then follow the [Instruction](#)

### Final Notes

If you have any suggestions, ideas, or comments, or if you find any disturbing bugs, please send email to <mailto:modhiamboisatieno@gmail.com>

## Creating Google Map API Server Key

Google Map Server Key is for getting Google Places detail in JSON format.

Follow the below steps to create Google Server Key,

- ✓ Go to Google API Console Page, <https://console.developers.google.com>
- ✓ Create or Select Project.
- ✓ Enable Google Maps Geocoding API in Library page.
- ✓ On the Credentials page, get an API key (and set the API key restrictions). Note: If you have an existing unrestricted API key, or a key with browser restrictions, you may use that key.

Google Map JavaScript Key is for Google Autocomplete (Suggestion box) and Map.

Follow the below steps to create Google JavaScript Key,

- ✓ Go to Google API Console Page, <https://console.developers.google.com> ✓ Create or Select Project.
- ✓ Enable Google Maps JavaScript API in Library page.
- ✓ On the Credentials page, get an API key (and set the API key restrictions). Note: If you have an existing unrestricted API key, or a key with browser restrictions, you may use that key.

## Firestore

Firestore service is used in the app for sending & receiving notifications from the backend/chat with user/driver.

Create a Firestore project

You need to create a firestore project in order to connect the flutter app to firestore.

- ✓ Go to <https://firebase.google.com/>.

- ✓ Click on Get started.
- ✓ Create a new project by tapping on the Add project box.
- ✓ Enter the Project name & select Analytics location. Analytics location represents the country/region of your organization and sets the currency for revenue reporting.
- ✓ Tap on Create project. Following this, Firebase will set up a new project for you. This will take just a minute.
- ✓ After the project is created successfully, Firebase will show a prompt saying Your new project is ready. Click on Continue to complete the flow.

**IMPORTANT:** Ensure that you use a new copy of google\_service.json after every firebase feature related configurations.

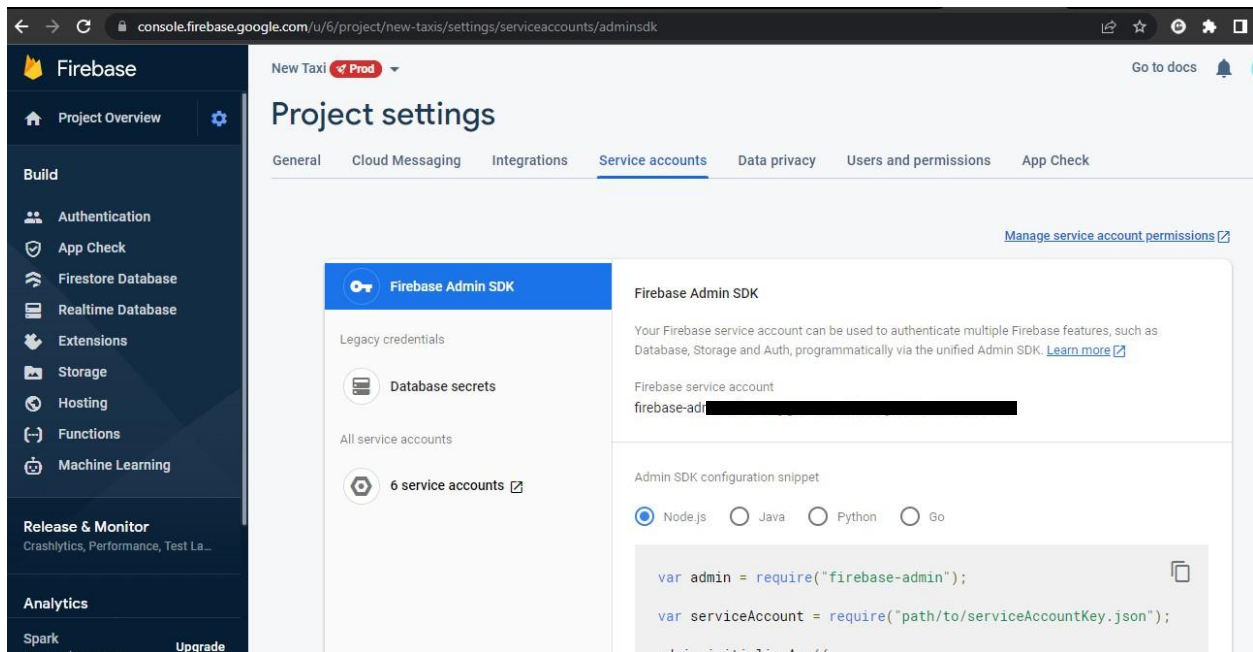
## Google Service Account

In order to access a Firebase project using a server SDK, you must authenticate your requests to Firebase with [Service Account credentials](#).

To authenticate a service account and authorize it to access Firebase services, you must generate a private key file in JSON format.

To generate a private key file for your service account:

1. Open [https://console.firebase.google.com/project/\\_/settings/serviceaccounts/adminsdk](https://console.firebase.google.com/project/_/settings/serviceaccounts/adminsdk) and select the project you want to generate a private key file for.



2. Click **Generate New Private Key**, then confirm by clicking **Generate Key**

3. Securely store the JSON file containing the key.

After downloading the service account key, you need to upload this to the backend.

On the backend go to **Admin > Api Credentials**. Then upload the just downloaded **Firebase Service Account File (JSON)** \*

New Taxi

- Dashboard
- Admins Management <
- Manage Company
- Manage Drivers
- Manage Riders
- Manage Documents
- Send Messages
- Manage Emails <
- Manage Manual Booking <
- Vehicles Management <
- Additional Reasons

Apple Service Id \*

#

Apple Team Id \*

#

Apple Key Id \*

#

Apple Key File \*

Choose File No file chosen

Firebase Database URL \*

https://new-taxi-default-rtdb.firebaseio.com

Firebase Service Account File (JSON) \*

Choose File No file chosen

Submit

Reset

## **Setting Up Apps in Android Studio**

Download the latest android studio from given link and install it using standard installation process. <https://developer.android.com/studio>.

- a. Setup project in Android Studio
- b. Open your zip file and locate in your drive and extract the android app file.
- c. Open Android Studio
- d. Click on Open Project

## **Updating App Icon**

- a. Click File → New → Image Asset
- b. Set Icon type as Launcher Icons (Adaptive and Legacy)
- c. Set your image name as ic\_launcher.
- d. Foreground Layer → Set your Layer Name as ic\_launcher\_foreground.
- e. Set Asset Type to Image
- f. Foreground Layer → Path and select your icon
- g. Resize your icon (if needed)
- h. Background Layer → Set your Layer Name as ic\_launcher\_background.
- i. Background Layer → Source Assets → Asset Type → Color
- j. Set your background color for the icon
- k. Click Next and at the next page click finish

## **Updating App Logos**

### **Steps To Update App Logo**

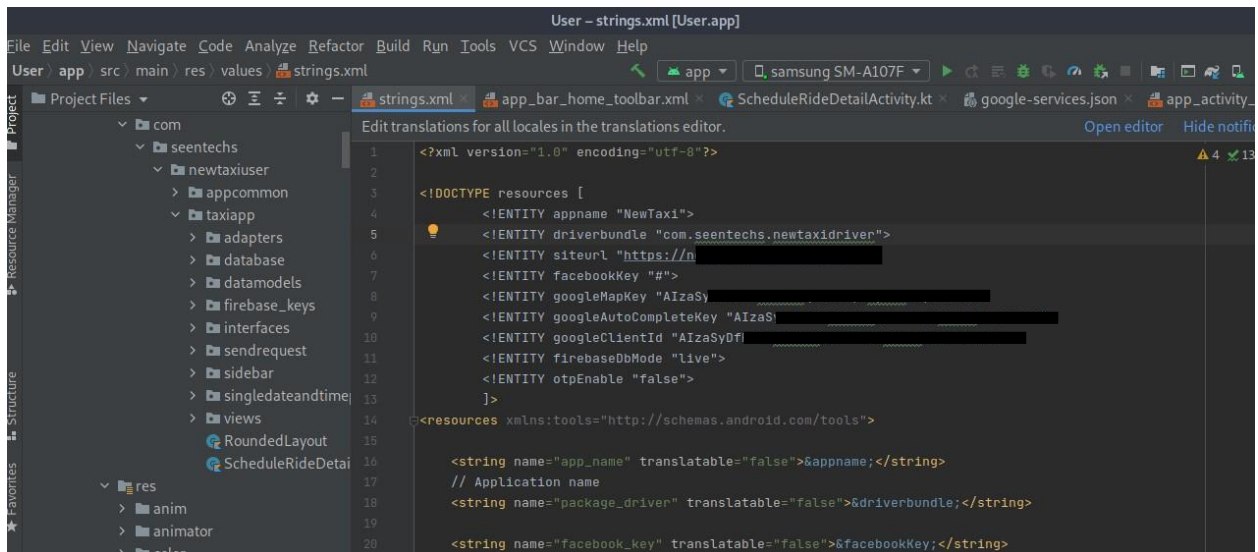
To change the app logo in selected folder (image name is logo. In Source folder -> app-> src-> main-> res -> drawable to replace image which name: -



- app\_a\_app\_logo.png (User & Driver)
- app\_a\_app\_splash\_icon.png (User & Driver)
- app\_a\_notification\_icon.png (User & Driver)
- applogo.png (Driver)
- app12logo.png (Driver)

## Updating Base URL, Facebook, and Google API Key, Drivers' app

You Can Create Facebook App Id and Google login for either rider or driver, and use the same Google Map Key for both rider and driver (change in Android Studio ->app/res/values/string.xml)



- ✔ Get Facebook APP ID used to authenticate into different types of Web Application, Mobile Application.
- ✔ Follow the link to create Facebook APP ID <https://developers.facebook.com/>
- ✔ Get Google APP ID used to authenticate into different types of Web Application, Mobile Application.
- ✔ Follow the link to create Google API Key <https://console.developers.google.com/>
- ✔ Change the Driver Package name to open the driver app from rider app if driver app is installed otherwise open the play store link to download.

- ✔ Update Service ID in admin panel api\_credentials

#### Enable libraries in google API

- ✔ Google+ API
- ✔ Google Maps SDK for Android
- ✔ Google Places API
- ✔ Google Directions API
- ✔ Google Distance Matrix API

#### Steps To Update Base URL, Facebook, Apple and Google API Key

- ✔ Open the CommonKeys java file in app/java/com/seentechs/user/utls(Android Studio)
- ✔ Update baseUrl
- ✔ Open the app folder in android studio
- ✔ And then open the res folder
- ✔ And then open the values folder

String.xml file contains

- ✔ Application name
- ✔ Facebook\_Key ✔ Facebook provider
- ✔ google maps key -Replace Your API Key here ✔ Google client id ✔ driver package name here you can change it.
- ✔ App name should be updated for all Languages in all the string files ✔ Change live

and demo in real\_time\_db for Firebase real time db Updating FCM Key (Firebase Cloud Messaging)

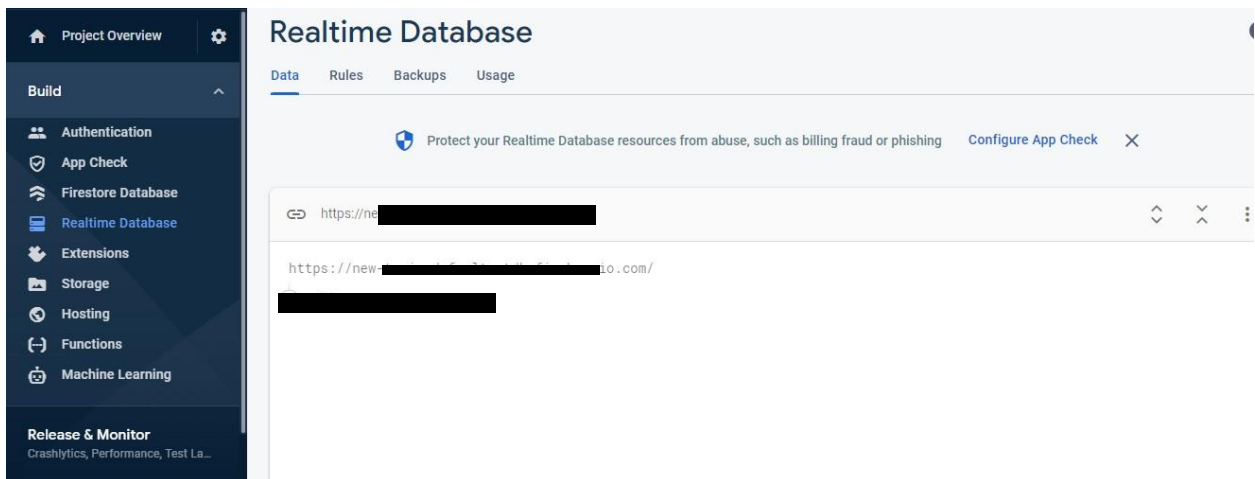
- ✔ Get FCM Key used to push notification into different types of Web Application, Mobile Application.
- ✔ Follow the link to create FCM <https://console.firebase.google.com>

- ✓ Create the project and create both driver and rider app
- ✓ Click setting in selected app
- ✓ Download the google-services.json from firebase console **You should use**

**the FCM Key for admin panel too.**

## Steps To create Firebase Database

- a. On left side panel of console click database
- b. Choose real-time database Click create database
- c. Choose start in locked mode and enable
- d. Update the Database rules (You can begin with setting permissions to demo mode).



## Steps To update FCM Key

- a. Open the Source folder
- b. And then open project folder (New Taxi driver / rider)
- c. And then open the app folder
- d. Now to place / replace the google-services.json file

## Twilio SMS Intergration

Generate SID, Token, Number from below url h <https://www.twilio.com/login> and update in admin panel as shown in the below screenshot.



Twilio Token *	bc887b0e7159ab5cb0945c3fc59b345a
Twilio From Number *	+19898282822
Twilio Service SID *	ACf64f47c56b592b6dec3919aed6b2a55e

## 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 .