PREREQUISITES

To set the android app up, we need to download and install a few tools that is used to generate the Thinkin service android application.

the Development software available at https://developer.android.com/

INSTALLATION

Thinkin service User Application:

1. Extract the android package.

2. Open the Thinkin service package in Android Studio and wait till the Gradle build successfully.

3. On the project explorer window of Android studio, find and edit the following files

Gradle Scripts/build.gradle (Module: app)

a.   applicationId "com.Thinkinservice.app" -> change this to your app's Bundle ID

b.   change values and the credentials in the screenshot down below

4. Configure your Application on the Google Firebase console, enable authentication and
Databasebase for the chat and download the googleservices.json, and replace them in app/ folder.

5. Change the splash screen and icon for your own brand.

Open the project folder from the PC and follow the below-mentioned path: Thinkin service->app->src->main->res

You will find 'mipmap' folders in the different resolution names, you must save the icons and splash screens in these folders according to the device size.

Also, You will find 'drawable' folders in the different resolution names, you must save the icons and splash screens in these folders too according to the device size.

Now hit on the 'run' button the application will run successfully in the virtual device.

If you are planning to upload the app in the Play Store, you must generate the build. To do so, click on 'Build' in Android Studio, choose 'Generate Signed APK' from the drop-down. Now, click on create new and hit 'Next' to proceed.

Choose the path to the location where you want to save the key, give a folder name for the key and click 'OK'. Enter the information in the fields and click on 'Finish'.

The APK will be generated and stored in the folder named by you.

Thinkin service Provider/Driver Application:

1. Extract the android package.
2. Open the Thinkin service package in Android Studio and wait till the Gradle build successfully.
3. On the project explorer window of Android studio, find and edit the following files
   a. Gradle Scripts/build.gradle (Module: app)
      i. applicationId "com.Thinkin service.app" -> change this to your app's Bundle ID
      ii.

4. Configure your Application on the same project in Google Firebase console, and download the googleservices.json, and replace them in app/ folder.
5. 5. Change the splash screen and icon for your own brand.

Open the project folder from the PC and follow the below-mentioned path: Thinkin service->app->src->main->res

You will find 'mipmap' folders in the different resolution names, you must save the icons and splash screens in these folders according to the device size.

Also, You will find 'drawable' folders in the different resolution names, you must save the icons and splash screens in these folders too according to the device size.

Now hit on the 'run' button the application will run successfully in the virtual device.

If you are planning to upload the app in the Play Store, you must generate the build. To do so, click on 'Build' in Android Studio, choose 'Generate Signed APK' from the drop-down. Now, click on create new and hit 'Next' to proceed.
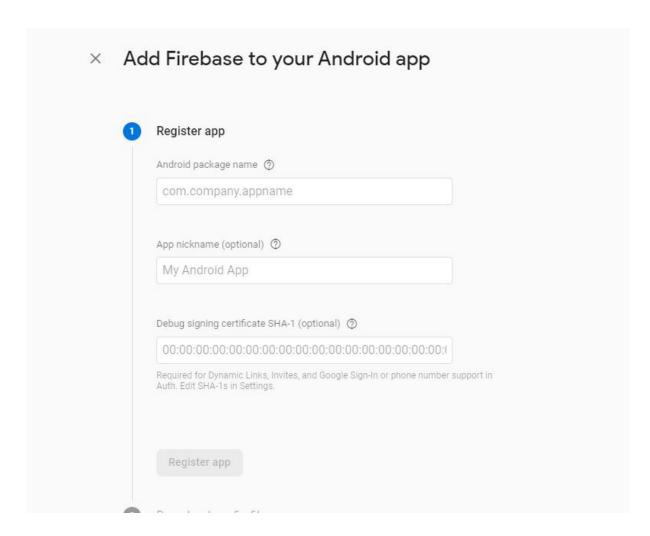
Choose the path to the location where you want to save the key, give a folder name for the key and click 'OK'. Enter the information in the fields and click on 'Finish'.

The APK will be generated and stored in the folder named by you.

Firebase

To get the FCM function in app for push and instant message between user and provider

1. Set the firebase project adding both app and get the package name
2. Download google-services.json file tomorrowm

Setting the live chat DB make sure you set the rule to true for both read and write

Security rules for Realtime Database

×

Once you have defined your data structure **you will have to write rules to secure your data.**
Learn more ↗

○ Start in **locked mode**
Make your database private by denying all reads and writes

◉ Start in **test mode**
Get set up quickly by allowing all reads and writes to your database

```
{
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

⚠ **Anyone with your database reference will be able to read or write to your database**

Cancel   **Enable**