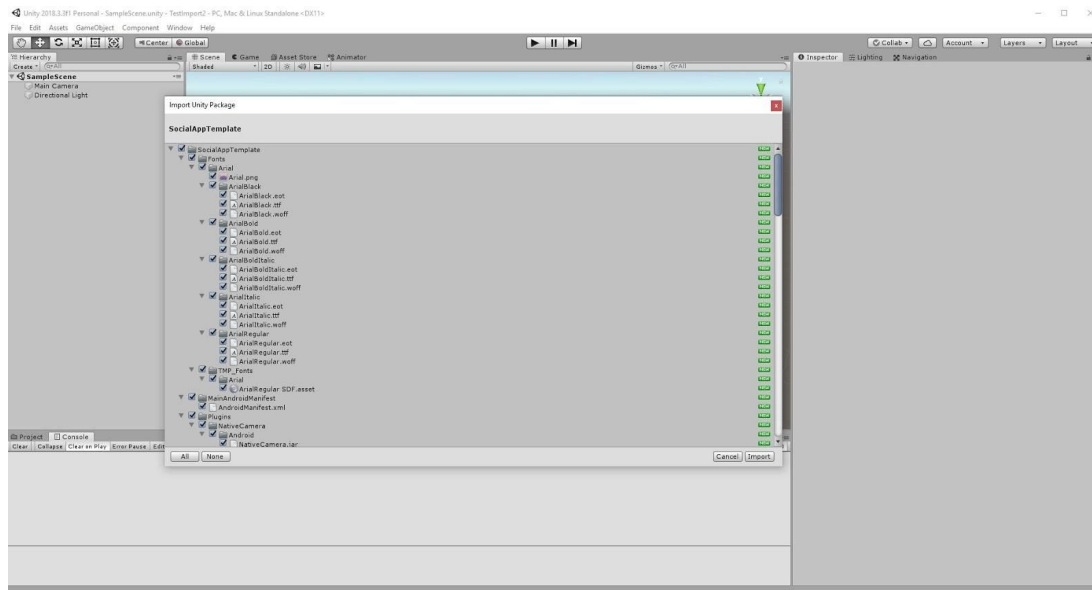
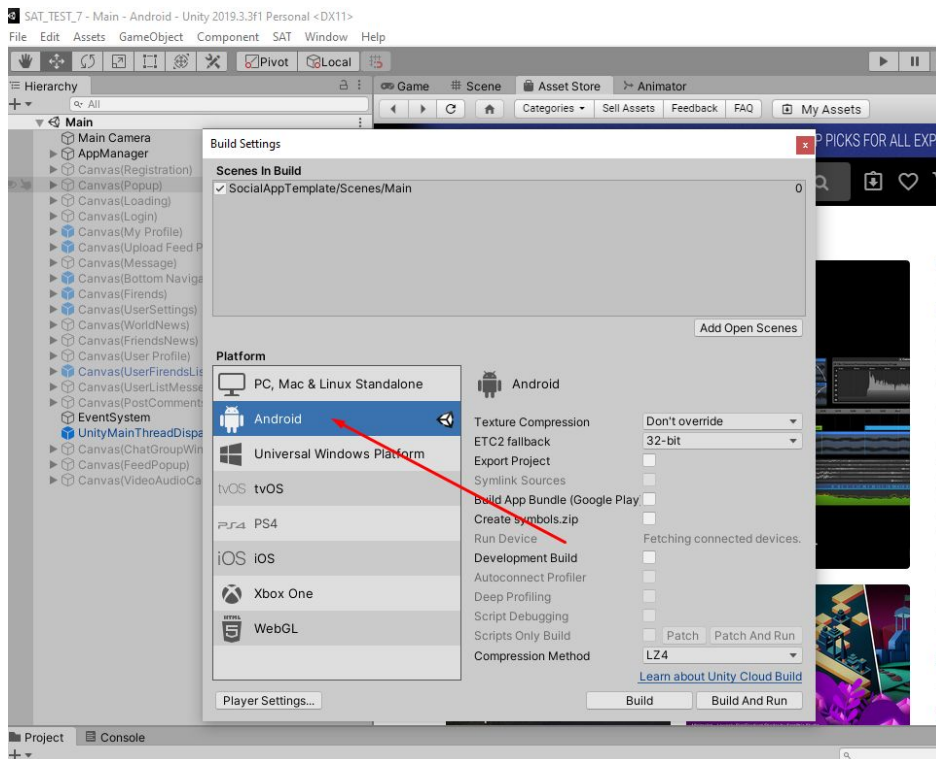


## Stage 1. Import package

### 1. Download and import package.

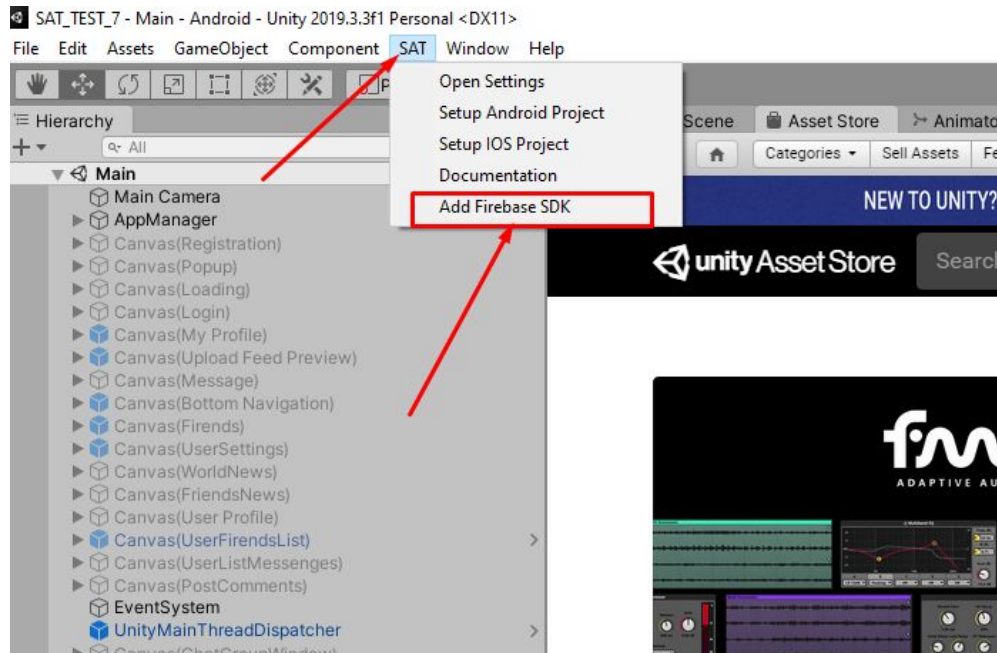


### 2. Switch to android platform

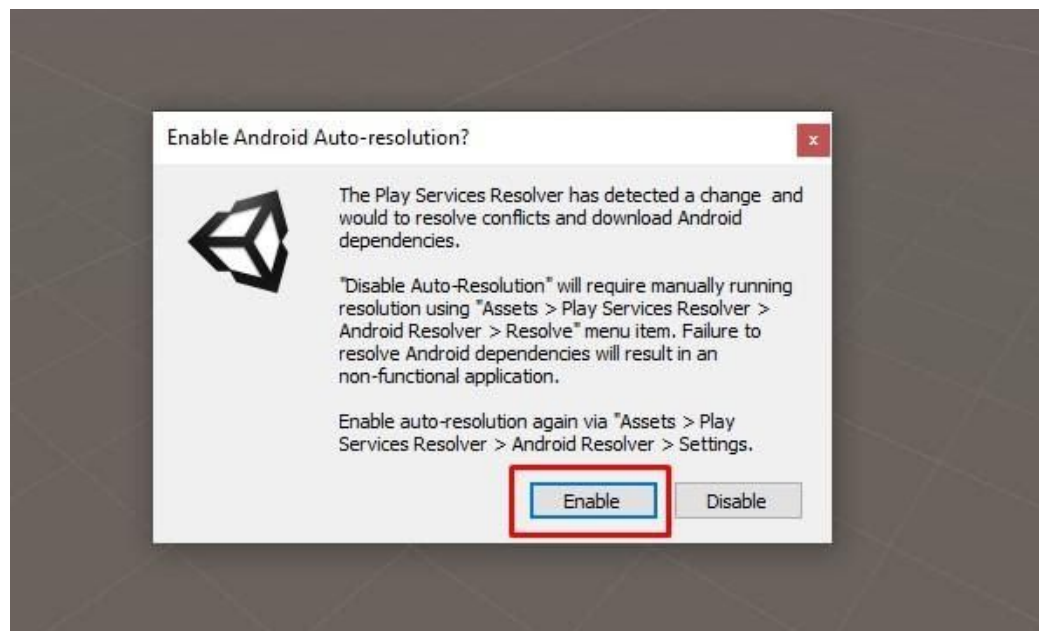


**IMPORTANT! If you don't have a context menu "SAT" - just restart the Unity.**

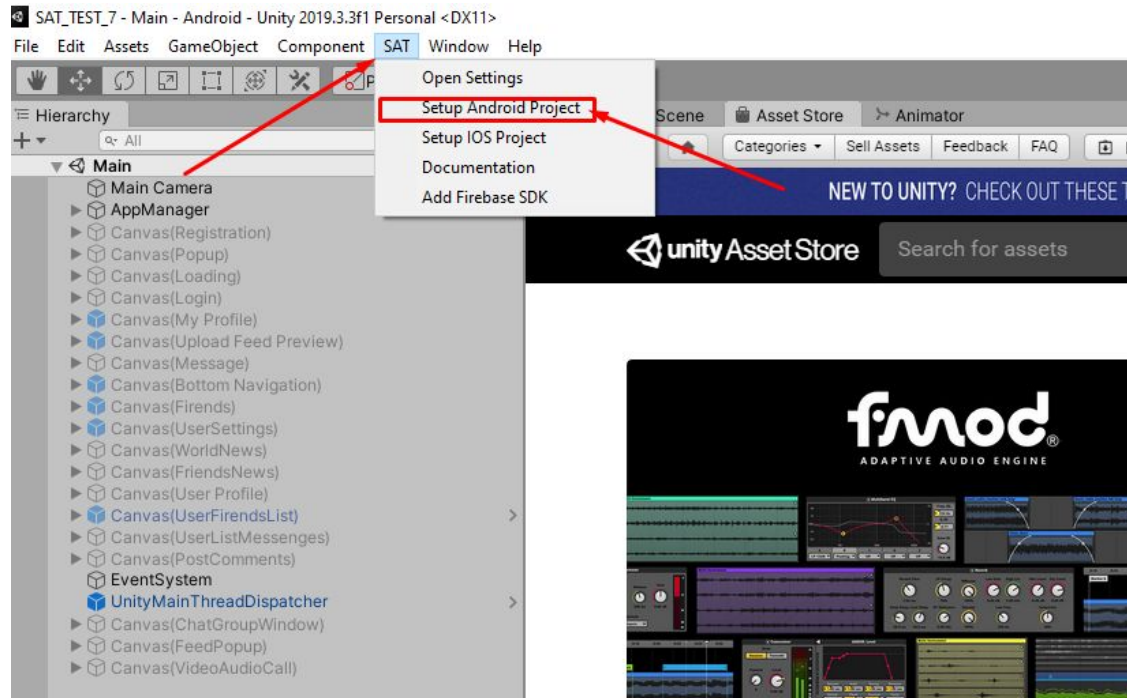
3. Add firebase SDK. Execute SAT => Import Firebase SDK



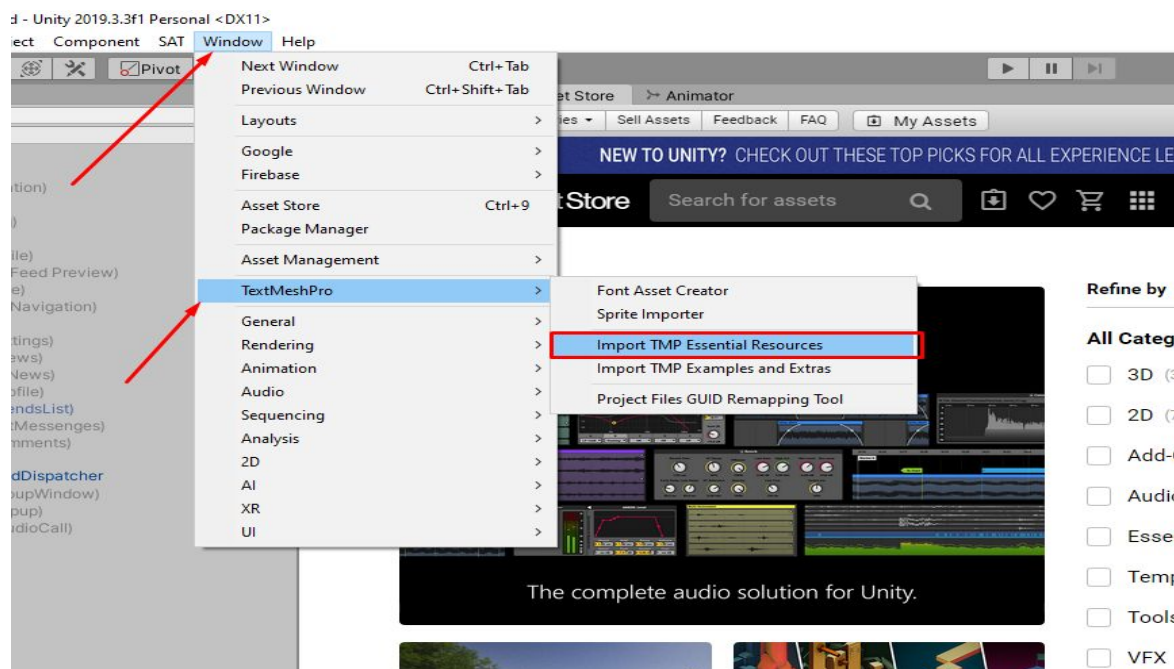
4. Enable Android Auto-resolution



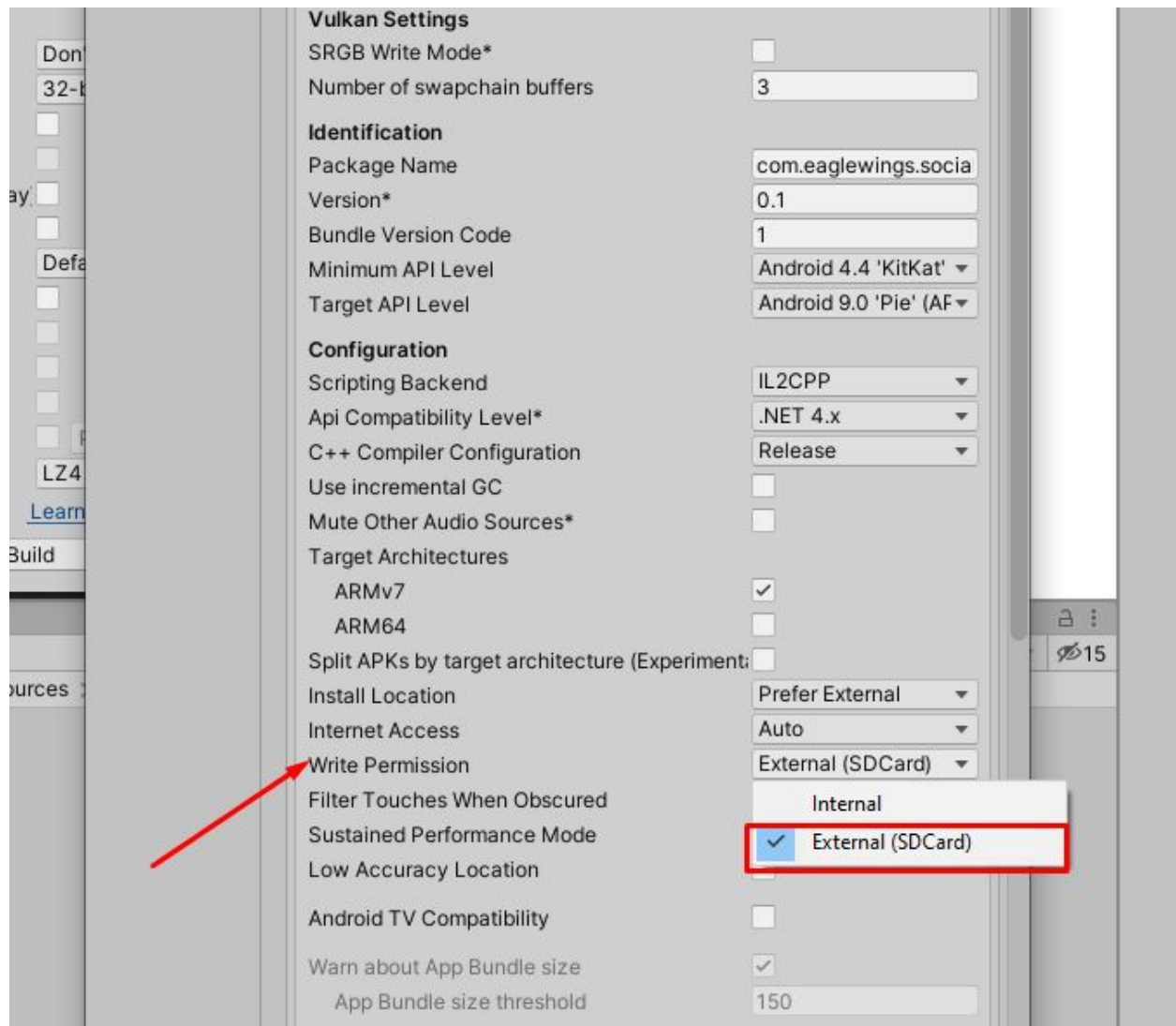
5. Setup android project. Execute SAT => Import Firebase SDK



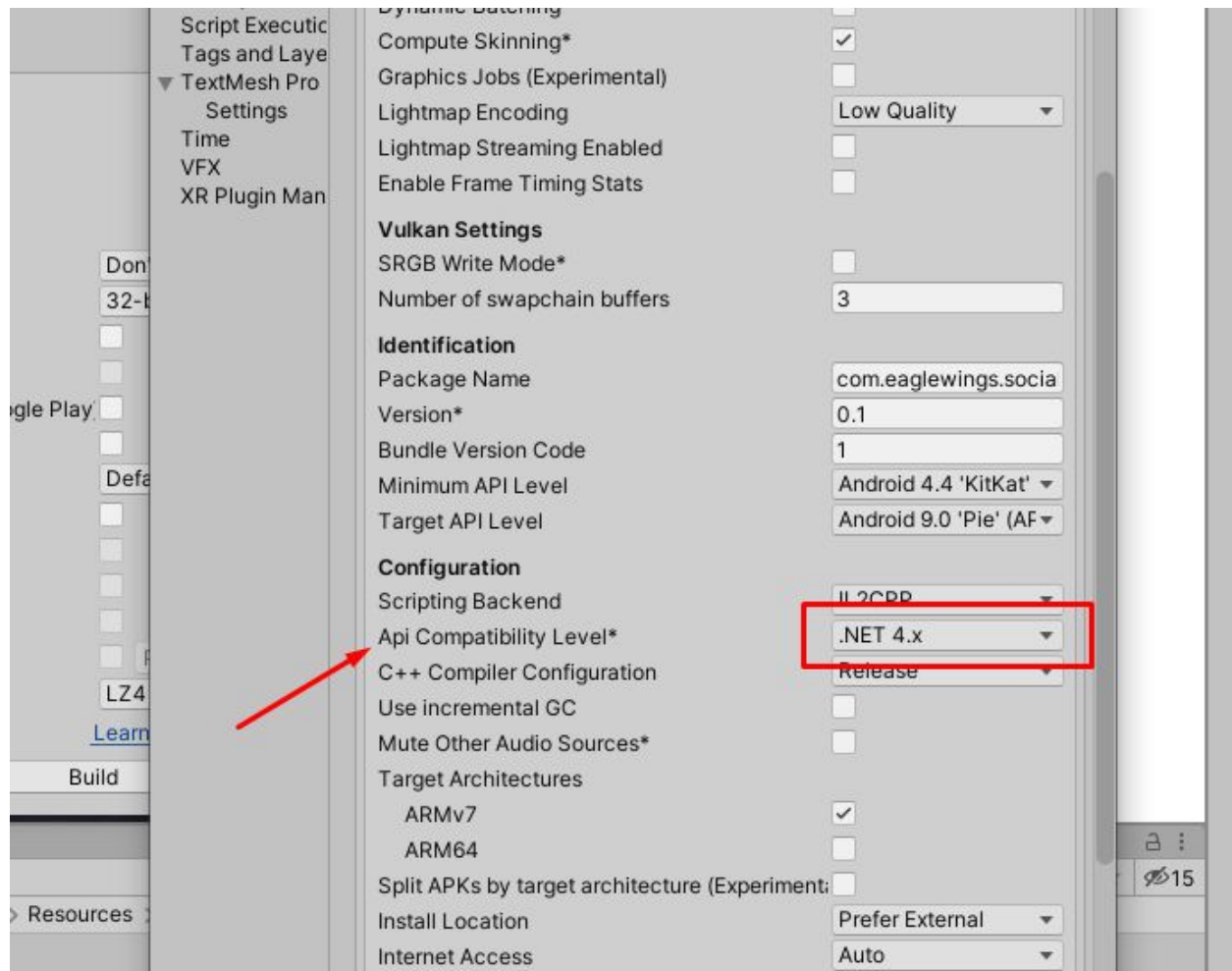
6. Import TMP. Window->TextMeshPro->Import TMP Essential Resources



7. Navigate to PlayerSetting and change **Write Permission** to **External (SDCard)**



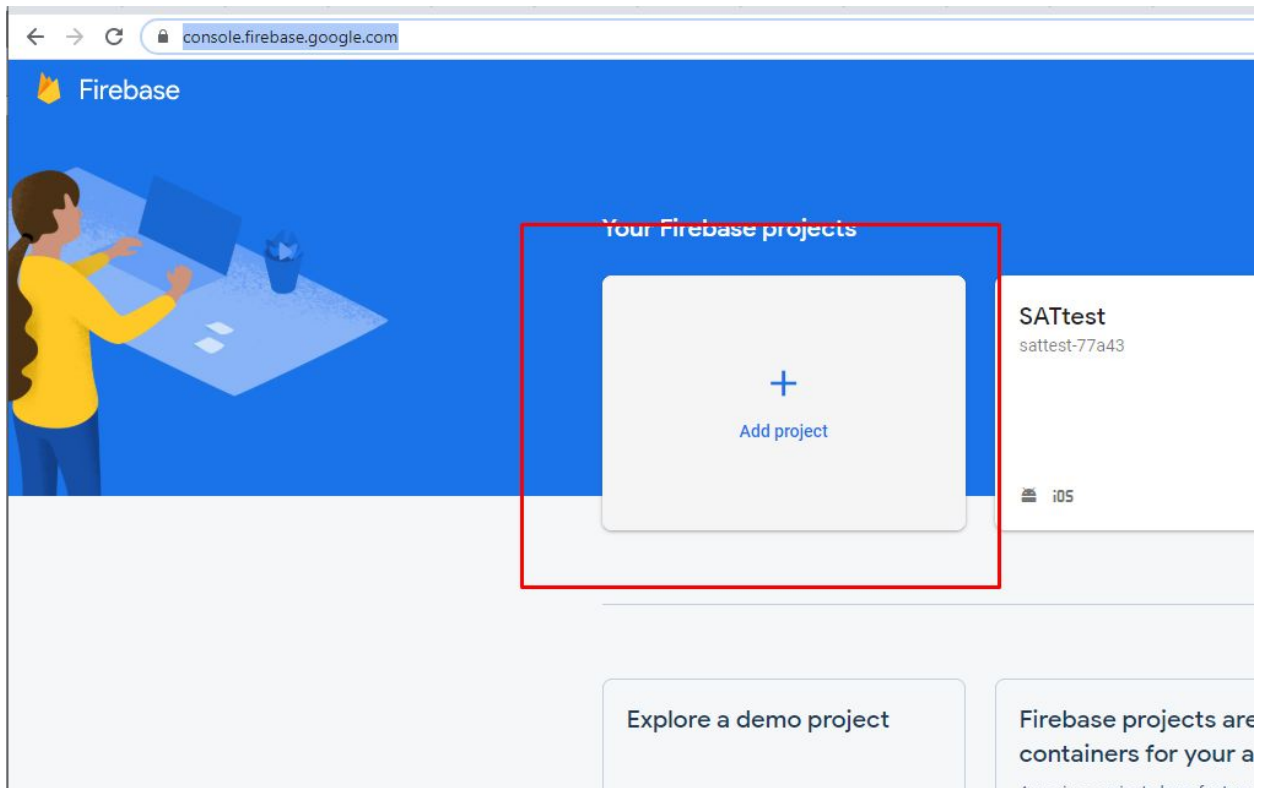
8. Navigate to PlayerSetting and change **Api Compatibility Level** to **.NET 4.x**



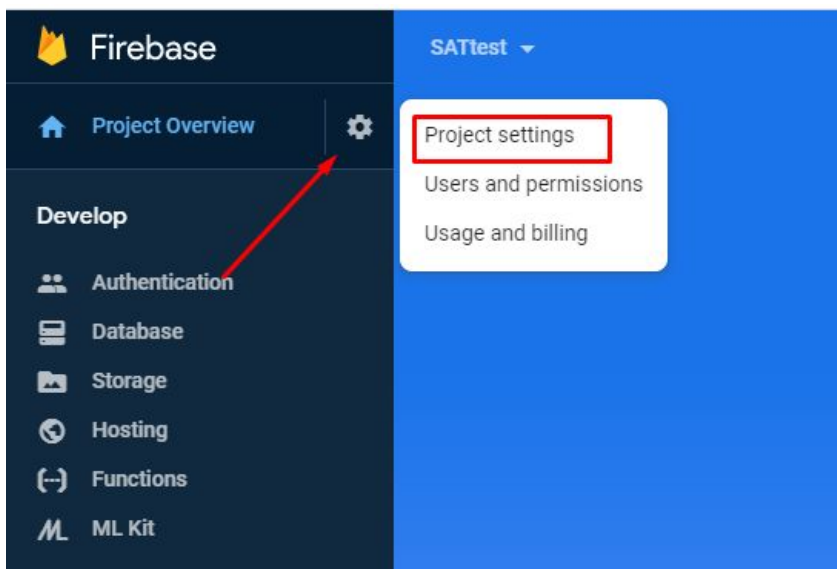
## Stage 2. Setup firebase console

1. Go to Firebase console and create new project

<https://console.firebase.google.com/>

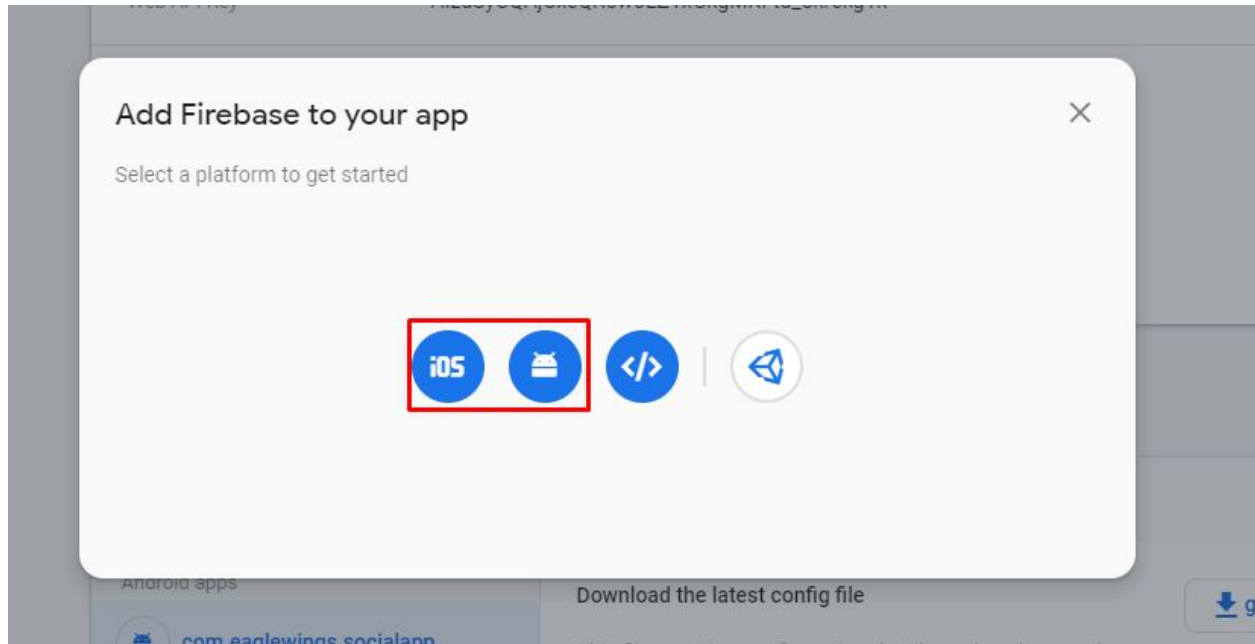


2. Navigate to Setting->Project Setting

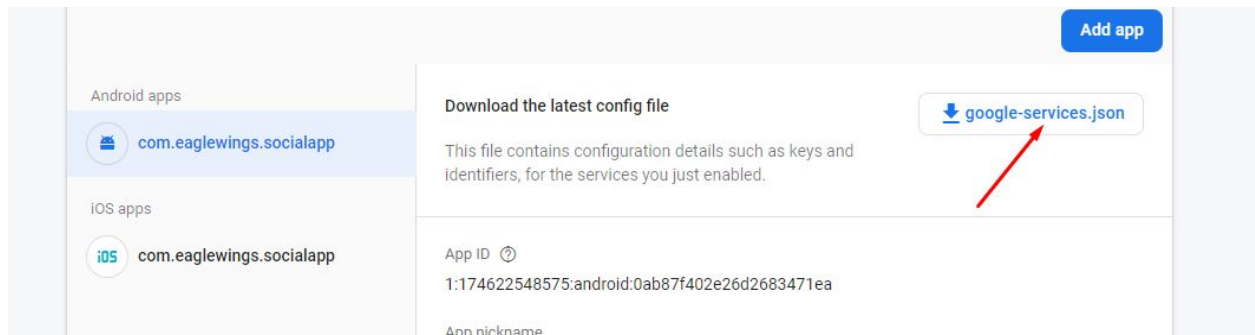


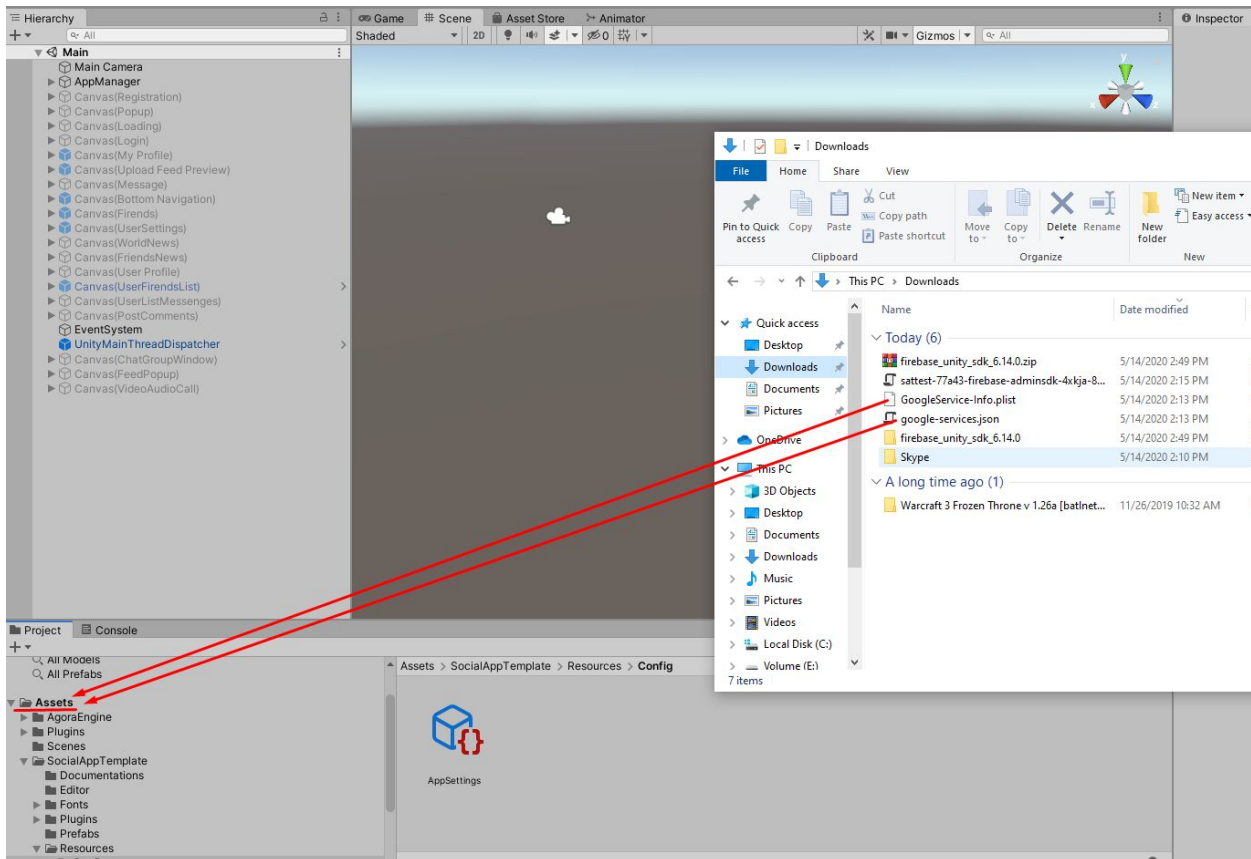
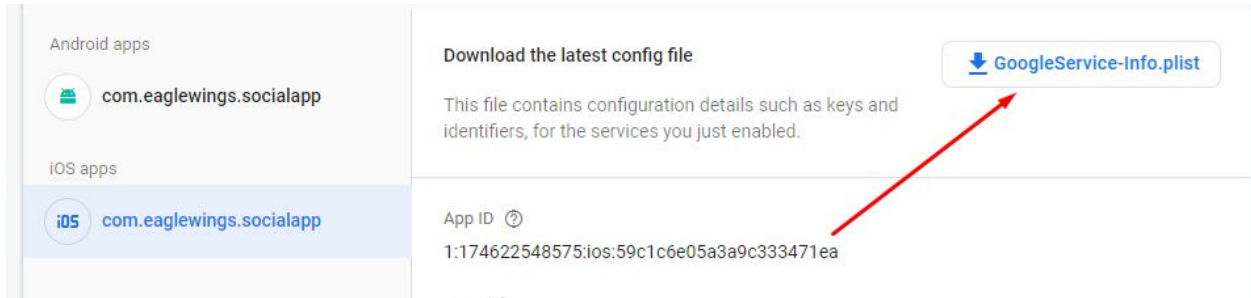
And setup iOS and Android app





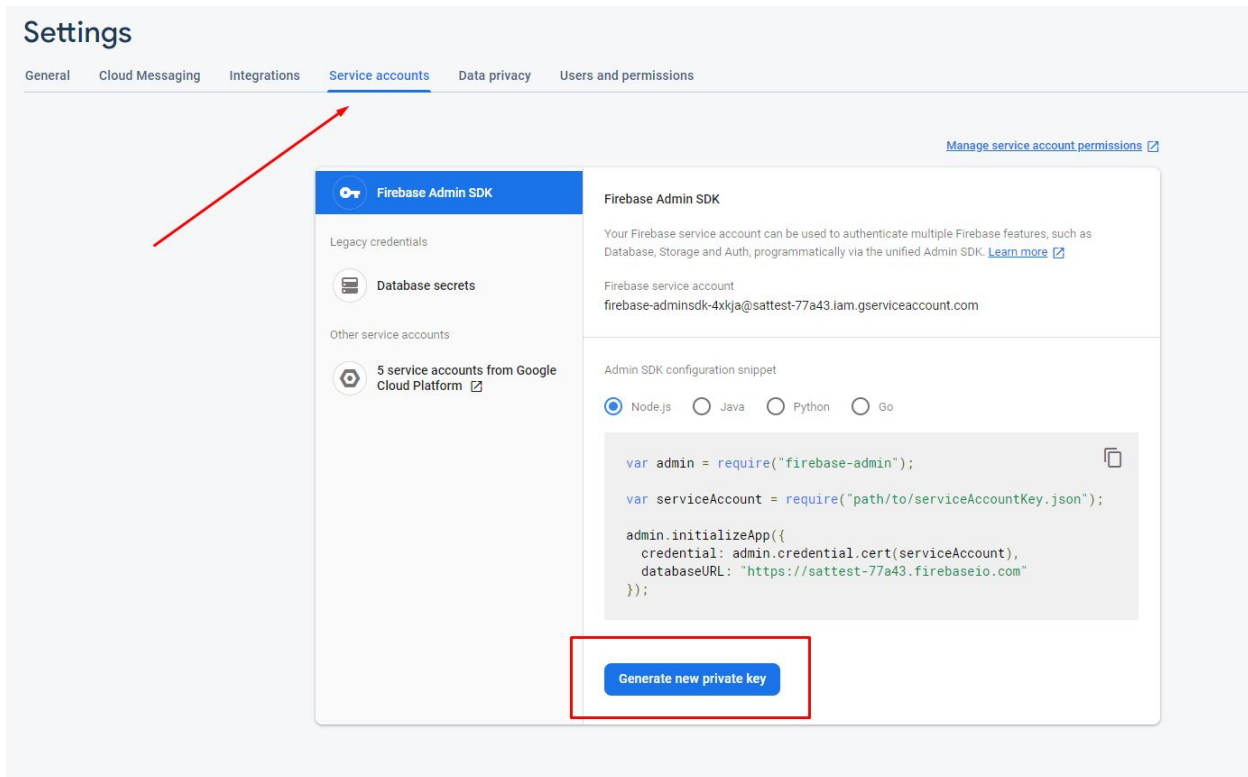
Download **google-services.json** for android and **GoogleService-Info.plist** for iOS. Import files to Assets folder. Before importing, make sure that the files are named that way and not for example **google-services (1).json**



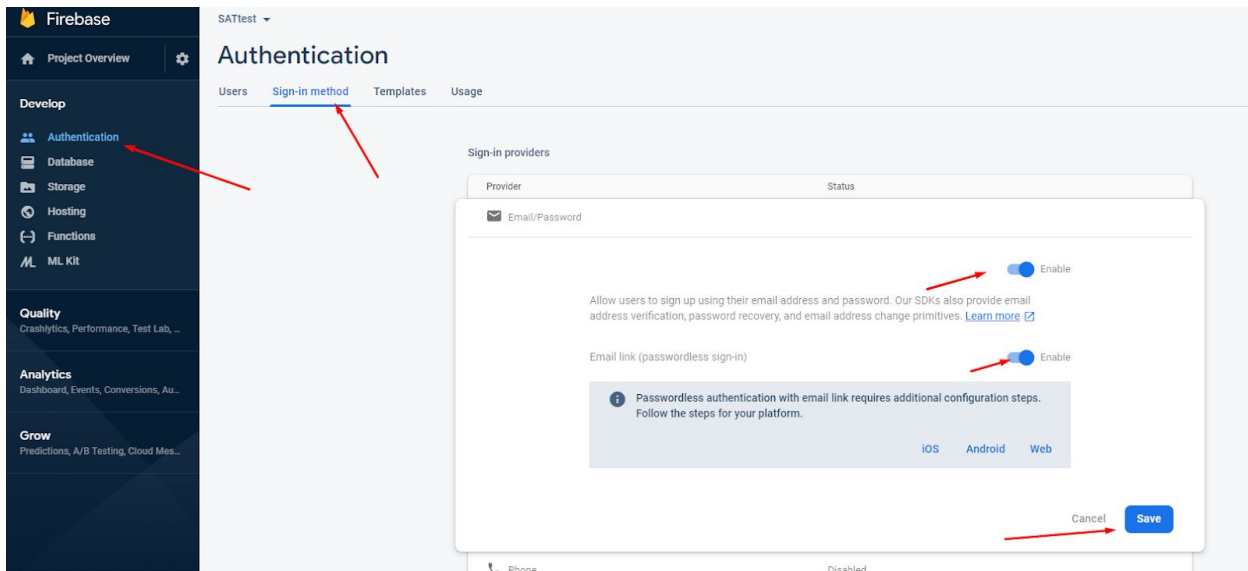


3. Navigate to **Service accounts** and generate new private key

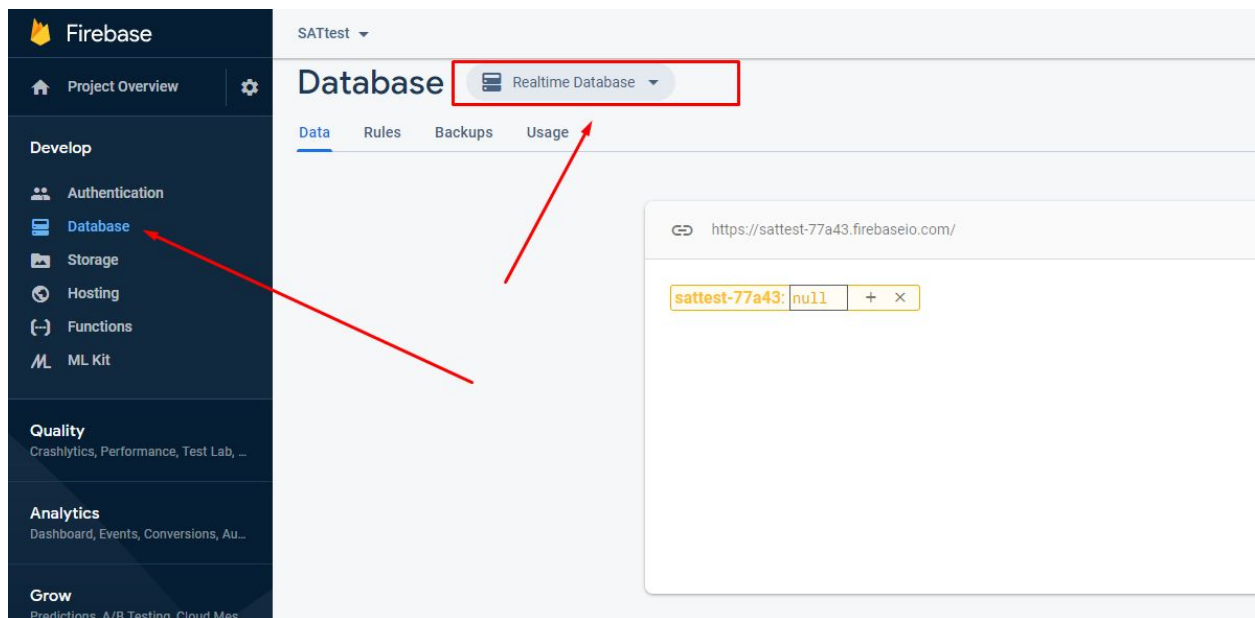
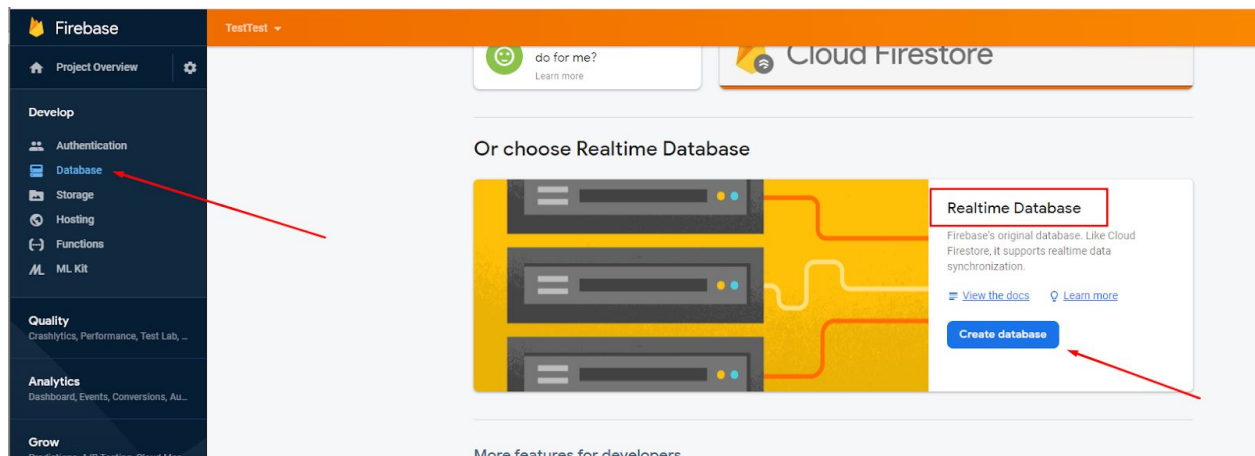




4. Navigate to **Authentication->Sing-in method** and enable **Email/Password provider**

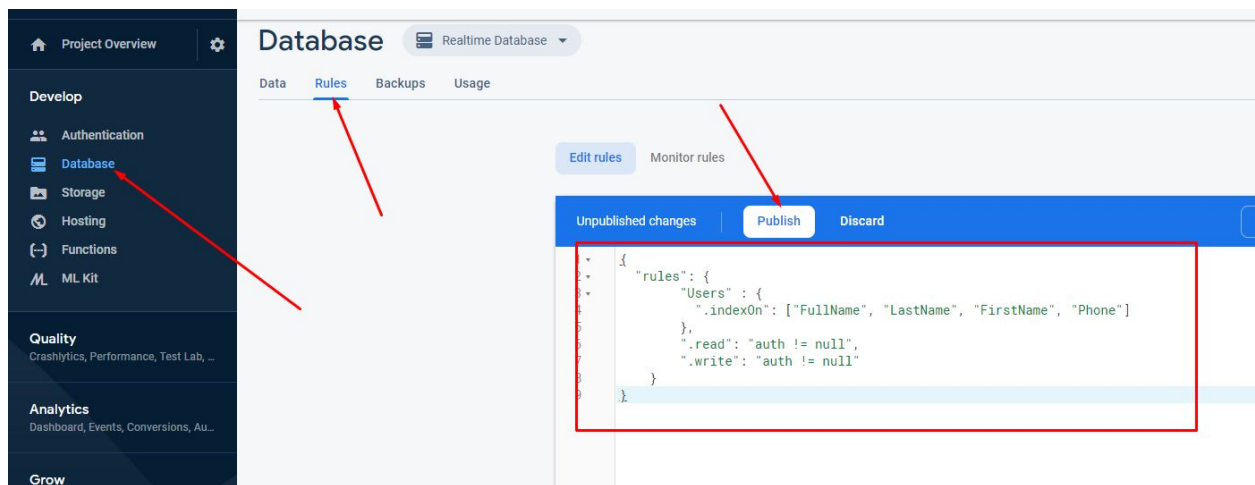


5. Navigate to Database create/select **RealtimeDatabase**



6. Navigate to **Database->Rules** and copy/paste this code

```
{  
  
  "rules": {  
  
    "Users" : {  
  
      ".indexOn": ["FullName", "LastName", "FirstName", "Phone"]  
  
    },  
  
    ".read": "auth != null",  
  
    ".write": "auth != null"  
  
  }  
  
}
```



7. Navigate to **Storage->Rules** and copy/paste this code

```
{  
  
  "rules": {  
  
    "Users" : {  
  
      ".indexOn": ["FullName", "LastName", "FirstName", "Phone"]  
  
    },  
  
    ".read": "auth != null",  
  
    ".write": "auth != null"  
  
  }  
  
}
```

The screenshot shows the Firebase console interface. On the left sidebar, under the 'Develop' section, 'Storage' is highlighted with a red arrow. The main area shows the 'Rules' tab selected, with a red arrow pointing to the 'Publish' button in the top right corner. The rules editor displays a sample rule for allowing read and write access if the user is authenticated.

```
1 service firebase.storage {  
2   match /b/{bucket}/o {  
3     match /{allPaths=**} {  
4       allow read, write: if request.auth != null  
5     }  
6   }  
7 }  
8  
9
```

### Stage 3. Upload firebase cloud function

1. Install Node.js <https://nodejs.org/en/> and unzip FirebaseFunctions.rar located at Assets/SocialAppTemplate/FirebaseCloudFunctions/FirebaseFunctions.rar
2. Run Command Prompt
3. cd unzipped folder (Example "cd C:\Users\luavrf\OneDrive\Desktop\FirebaseFunctions")
- 4 npm install npm@latest -g
5. npm install -g firebase-tools
6. firebase login
7. firebase init functions
8. Are you ready to proceed? **Yes**
9. What language would you like to use to write Cloud Functions? **JavaScript**
10. Do you want to use ESLint to catch probable bugs and enforce style? **Yes**
11. File functions/package.json already exists. Overwrite? **No**
12. File functions/.eslintrc.json already exists. Overwrite? **No**
13. File functions/tsconfig.json already exists. Overwrite? **No**
14. File functions/index.js already exists. Overwrite? **No**

15. File functions/.gitignore already exists. Overwrite? **No**
16. Do you want to install dependencies with npm now? **Yes**
17. firebase use --add
18. "Select you project from list"
19. What alias do you want to use for this project? (e.g. staging) **staging**
20. cd functions
21. npm install tslint typescript -g
22. npm install firebase-admin@latest
23. npm install firebase-functions@latest
24. npm install
25. npm audit fix
26. firebase deploy
27. Finally you will have this window on firebase console



Develop

Authentication

Database

Storage

Hosting

Functions

ML Kit

Quality

Crashlytics, Performance, Test La...

Analytics

Dashboard

Events

Conversions

Audiences

Funnels

User Properties

Latest Release

Retention

StreamView

Extensions

Spark

Upgrade

Dashboard

Health

Logs

Usage

Function	Trigger	Region	Runtime	Memory	Timeout
GetServerTimeStamp	HTTP Request https://us-central1-social-app-template-4c3fa.cloudfunctions.net/GetServerTimeStamp	us-central1	Node.js 8	2 GB	540s
SendFCM	HTTP Request https://us-central1-social-app-template-4c3fa.cloudfunctions.net/SendFCM	us-central1	Node.js 8	2 GB	540s
SharePostWithFriends	HTTP Request https://us-central1-social-app-template-4c3fa.cloudfunctions.net/SharePostWithFriends	us-central1	Node.js 8	2 GB	540s
UploadAndCompressVideo	HTTP Request https://us-central1-social-app-template-4c3fa.cloudfunctions.net/UploadAndCompressVideo	us-central1	Node.js 8	2 GB	540s
countallunreadfriendsc...	ref.write UnreadMessages(userId)/List(friendId)	us-central1	Node.js 8	256 MB	60s
countallunreadmessagetc...	ref.write UnreadMessages(userId)/List(friendId)	us-central1	Node.js 8	256 MB	60s
countfriendchange	ref.write UserFriends(userId)/List(friendId)	us-central1	Node.js 8	256 MB	60s
countfriendspostchange	ref.write FriendPosts(userId)/List(postId)	us-central1	Node.js 8	256 MB	60s
countpendingfriendchan...	ref.write UserPendingFriends(userId)/List(friendId)	us-central1	Node.js 8	256 MB	60s
countpostcommentschange	ref.write PostComments(postId)/List(userId)	us-central1	Node.js 8	256 MB	60s
countpostlikeschange	ref.write PostLikes(postId)/List(userId)	us-central1	Node.js 8	256 MB	60s
countrequestedchange	ref.write UserRequestedFriends(userId)/List(friendId)	us-central1	Node.js 8	256 MB	60s
countunreadmessagessan...	ref.write	us-central1	Node.js 8	256 MB	60s

if you have any questions, please contact us [assets@uni-bit.com](mailto:assets@uni-bit.com)