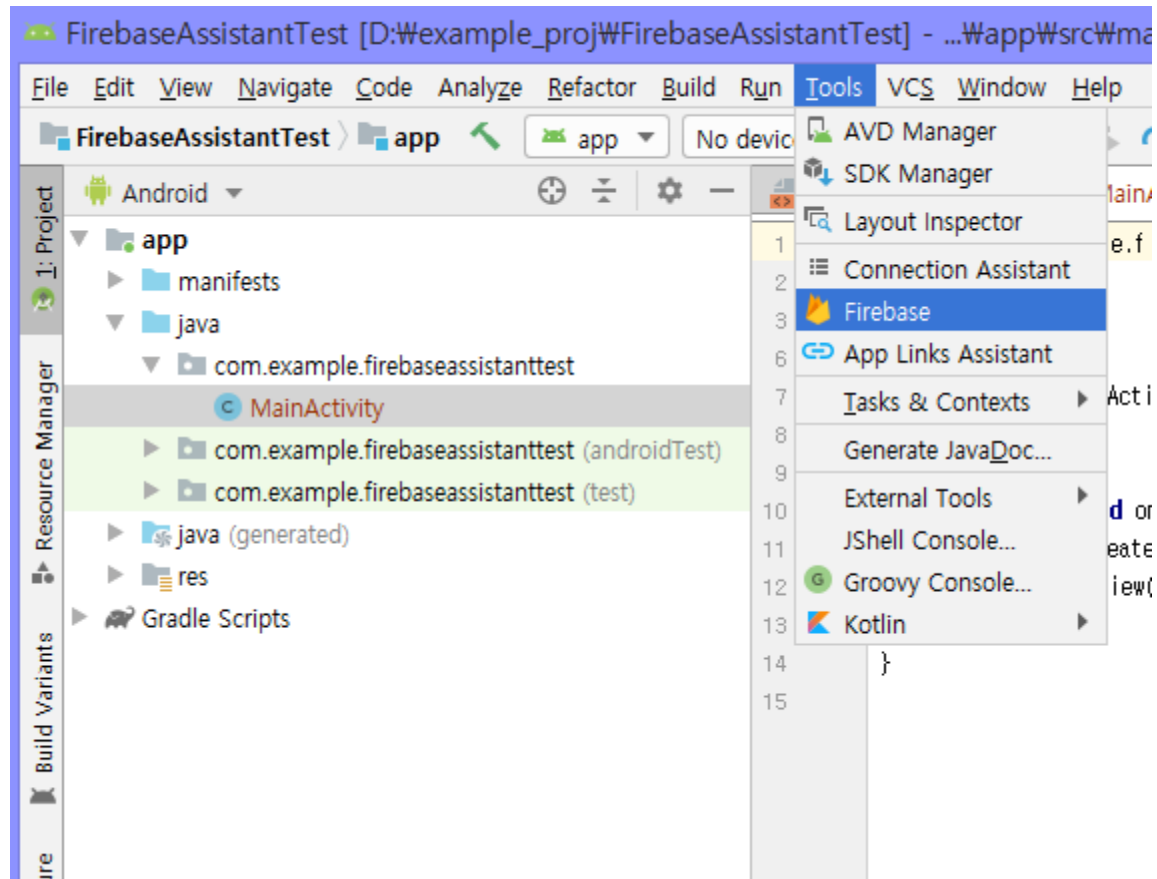


Firestore Push App


Innova Lee(이상훈)
gcccompil3r@gmail.com







The screenshot shows the Android Studio IDE. At the top is a toolbar with various icons for development tasks. Below the toolbar is a tab bar with 'Assistant' and 'Firebase' tabs. The 'Firebase' tab is active, displaying the Firebase logo and a brief description: 'Firebase gives you the tools and infrastructure from Google to help you develop, grow and earn money from your app.' with a 'Learn more' link. Below this, there is a list of Firebase services, each with an icon, a title, a description, and a 'More info' link. The services listed are Analytics, Cloud Messaging, Authentication, and Realtime Database. A vertical sidebar on the right contains icons for 'Gradle' and 'Assistant'.

Assistant **Firebase** ⚙️ —

||

 **Firebase**

Firebase gives you the tools and infrastructure from Google to help you develop, grow and earn money from your app. [Learn more](#)

- ▶  **Analytics**
Measure user activity and engagement with free, easy, and unlimited analytics. [More info](#)
- ▶  **Cloud Messaging**
Deliver and receive messages and notifications reliably across cloud and device. [More info](#)
- ▶  **Authentication**
Sign in and manage users with ease, accepting emails, Google Sign-In, Facebook and other login providers. [More info](#)
- ▶  **Realtime Database**

Gradle Assistant

The screenshot shows the Android Studio IDE. At the top, there's a toolbar with various icons. Below it, the 'Assistant' tab is active, showing the 'Firebase' section. The breadcrumb navigation shows '← Firebase > Cloud Messaging'. The main content area is titled 'Set up Firebase Cloud Messa...' and contains a description of Firebase Cloud Messaging. Below the description is a link 'Launch in browser'. The first step, '1 Connect your app to Firebase', is highlighted with a yellow background and includes a 'Connect to Firebase' button. The second step, '2 Add FCM to your app', includes an 'Add FCM to your app' button. The third step, '3 Handle messages', includes a paragraph of text about creating a new Service.

Assistant Firebase

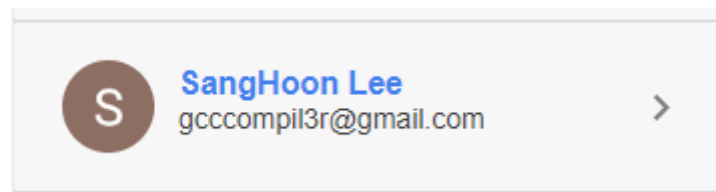
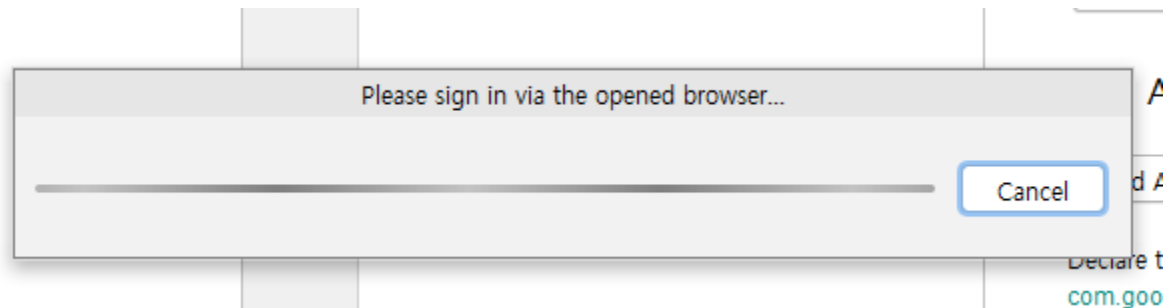
← Firebase > Cloud Messaging

Set up Firebase Cloud Messa...


Firebase Cloud Messaging lets you receive and send messages from your app to the server and other clients. This tutorial explains how to set up FCM and enable your app to receive notifications.

[Launch in browser](#)



- 1 Connect your app to Firebase
Connect to Firebase
- 2 Add FCM to your app
Add FCM to your app
- 3 Handle messages
If you wish to do any message handling beyond receiving notifications on apps in the background, create a new Service (**File > New > Service > Service**) that extends `FirebaseMessagingService` .



Android Studio이(가) 내 Google 계정에 액세스하려고 합니다

 gcccompil3r@gmail.com

이렇게 하면 **Android Studio**에서 다음 작업을 할 수 있습니다.

- Google 클라우드 플랫폼 서비스 전반에서 데이터 조회 및 관리 ⓘ
- Google App Engine에 배포된 애플리케이션 조회 및 관리 ⓘ
-  Google에서 작업 보기 및 관리 ⓘ
-  모든 Firebase 데이터 및 설정 확인 및 관리 ⓘ

r Challenge is back! [Submit your idea before December 2.](#)

Login to Android Studio

Use your Google Account to login to Android Studio and unlock the following services for your Android app:



Firebase



Google Cloud Platform




Twitter


Twitter에서 @AndroidDev 팔로
우



YouTube

YouTube에서 Android 개발자 확
인

Connect to Firebase×

 **Firebase**

☒ Create new Firebase project [What's this?](#)

Signed in as null [Sign out](#)

☐ Choose an existing Firebase or Google project

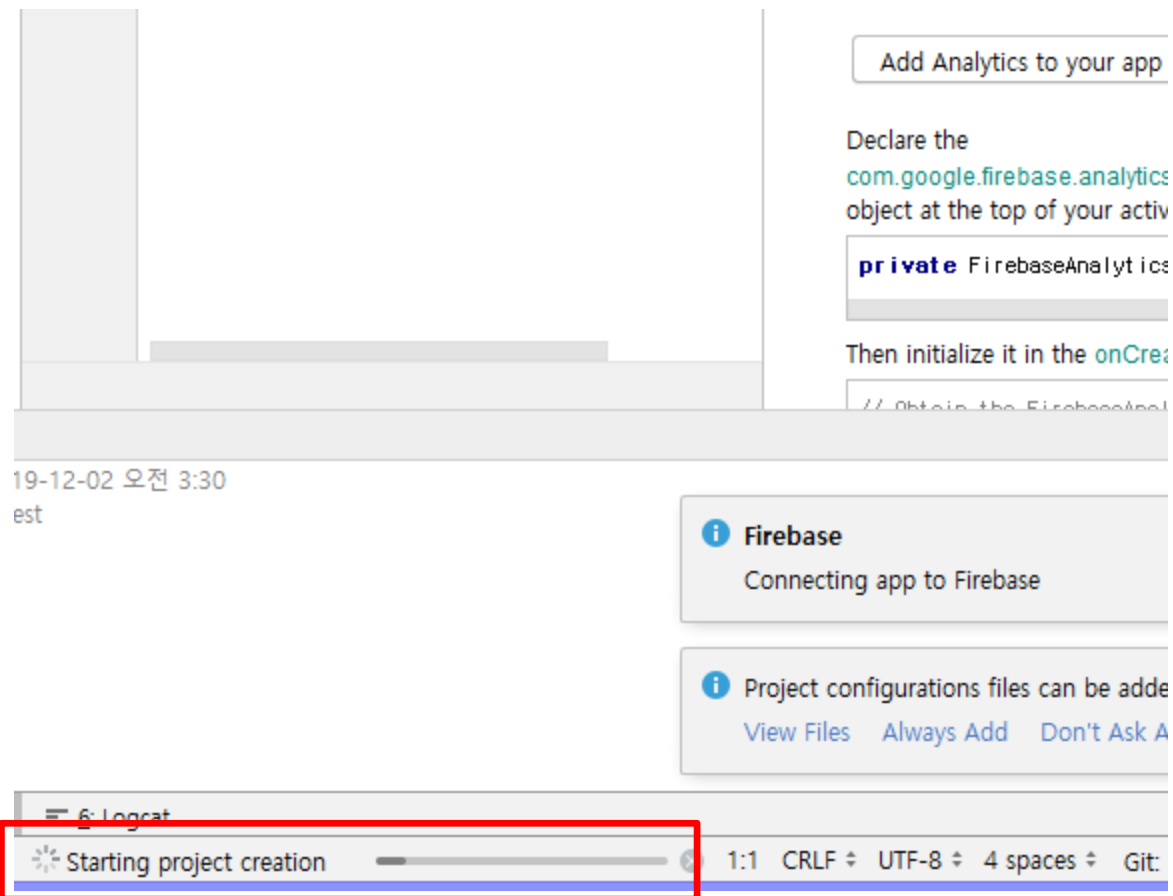
FCMTest	1 Android app(s) connected
Firebase Demo Project	1 Android app(s) connected, 1 iOS app(s) connected
FirebaseAssistantTest	1 Android app(s) connected
FirebaseAuthTest	1 Android app(s) connected
FireBasePushTest	1 Android app(s) connected

Country/region [What's this?](#)

By default, your Firebase Analytics data will enhance other Firebase features and Google products. You can control how your Firebase Analytics data is shared in your settings at anytime. [Learn more](#)

Connect to Firebase

Cancel



The screenshot shows the Android Studio IDE with the 'Assistant' tab selected. The 'Firebase' section is active, displaying the 'Cloud Messaging' setup wizard. The title is 'Set up Firebase Cloud Messa...'. The description states: 'Firebase Cloud Messaging lets you receive and send messages from your app to the server and other clients. This tutorial explains how to set up FCM and enable your app to receive notifications.' A link 'Launch in browser' is present. The first step, '1 Connect your app to Firebase', is highlighted with a yellow bar and shows a green checkmark and the text 'Connected'. The second step, '2 Add FCM to your app', includes a button labeled 'Add FCM to your app'. The third step, '3 Handle messages', includes the text: 'If you wish to do any message handling beyond receiving notifications on apps in the background'.

Assistant **Firebase**

← **Firebase** > Cloud Messaging

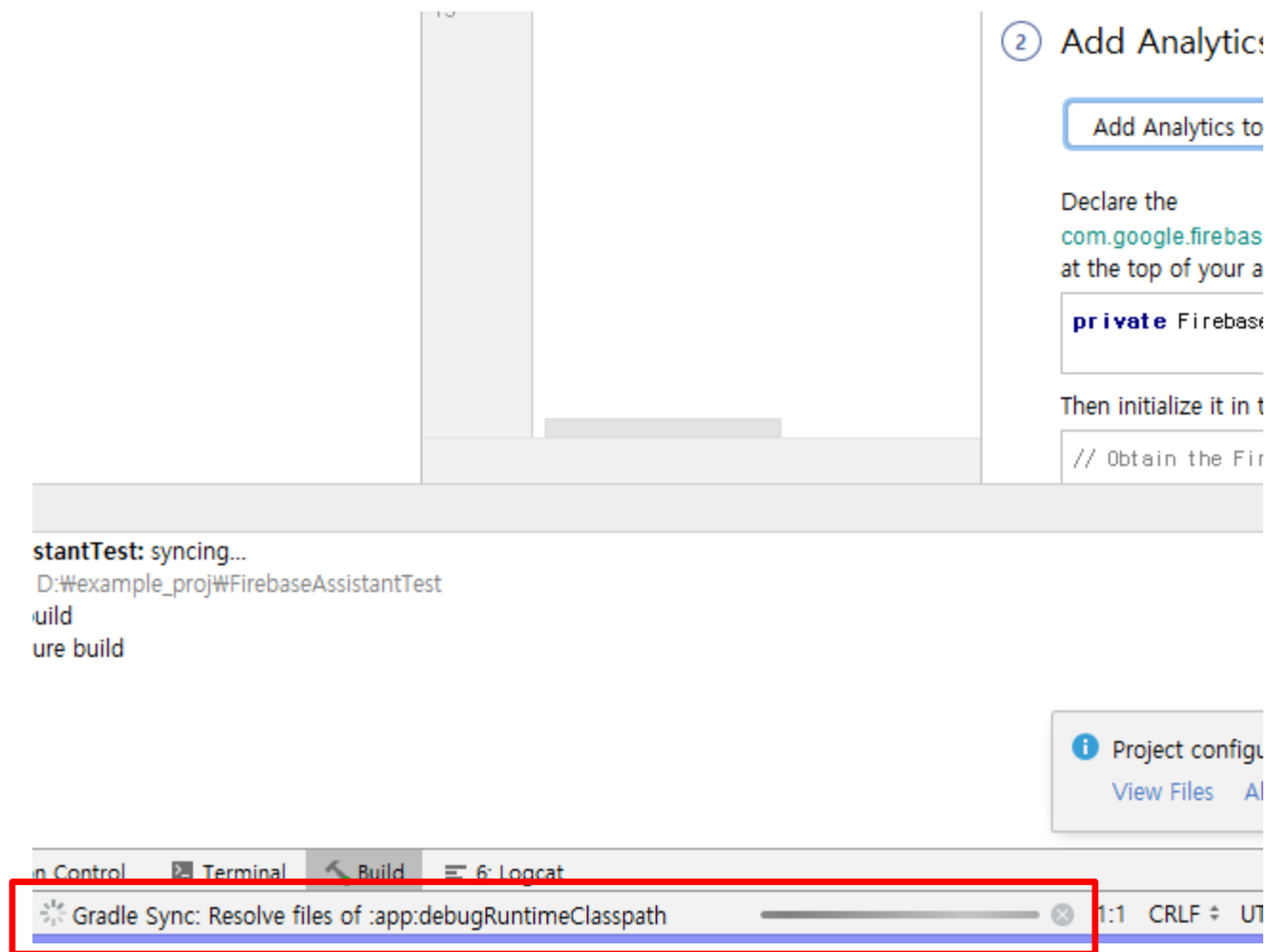
Set up Firebase Cloud Messa...

Firebase Cloud Messaging lets you receive and send messages from your app to the server and other clients. This tutorial explains how to set up FCM and enable your app to receive notifications.

[Launch in browser](#)

- 1 Connect your app to **Firebase**
✓ Connected
- 2 Add FCM to your app
Add FCM to your app
- 3 Handle messages
If you wish to do any message handling beyond receiving notifications on apps in the background



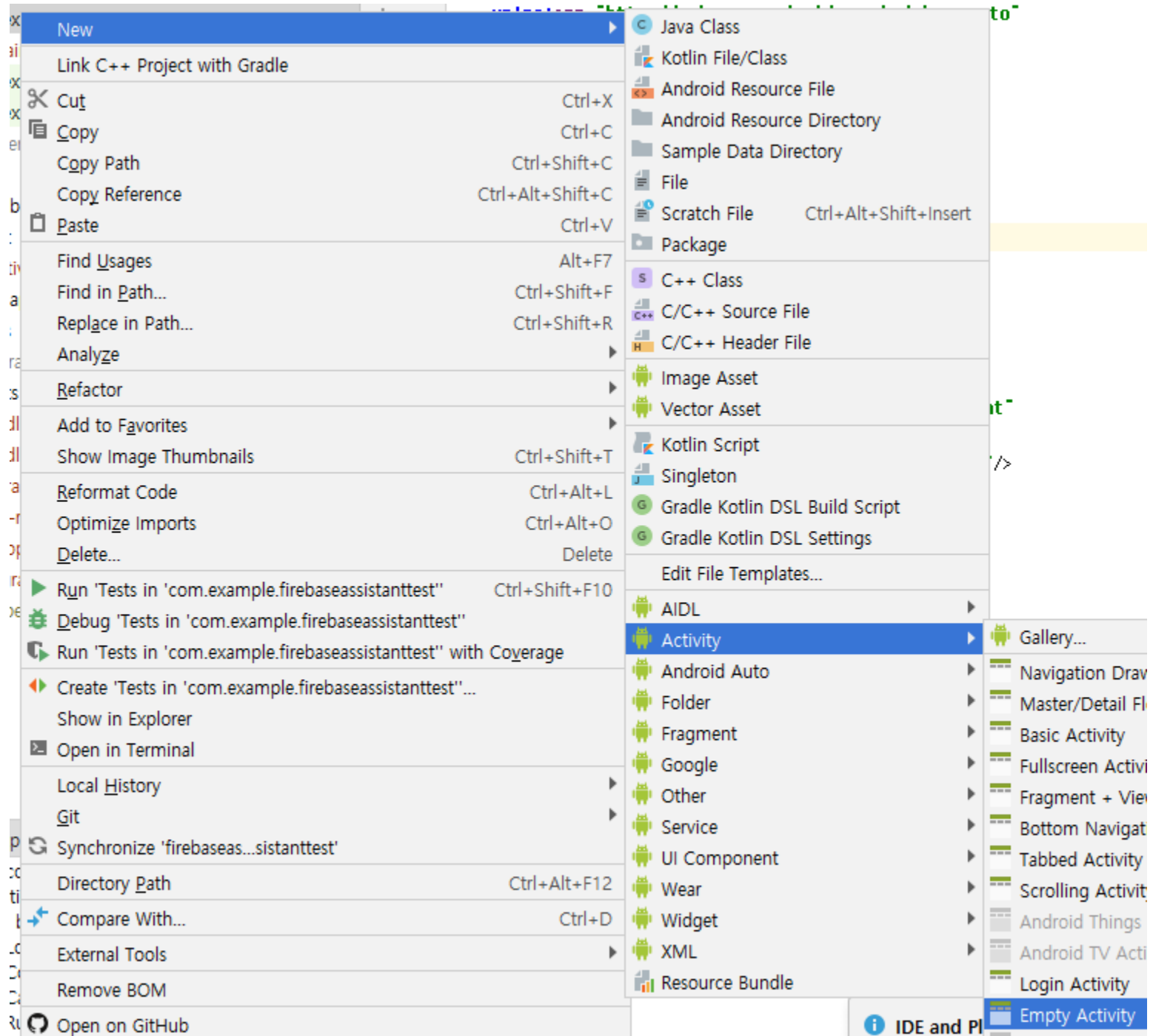


The screenshot shows the Android Studio IDE with the Firebase Assistant tab active. The breadcrumb navigation shows 'Firebase' > 'Cloud Messaging'. The main content area displays the title 'Set up Firebase Cloud Messa...' and a description: 'Firebase Cloud Messaging lets you receive and send messages from your app to the server and other clients. This tutorial explains how to set up FCM and enable your app to receive notifications.' Below this is a link 'Launch in browser'. A list of three steps is shown, with the first step highlighted in yellow:

- 1 Connect your app to Firebase
✓ Connected
- 2 Add FCM to your app
✓ Dependencies set up correctly
- 3 Handle messages
If you wish to do any message handling beyond receiving notifications on apps in the background, create a new Service (**File** > **New** > **Service** >

The right sidebar contains tabs for 'Gradle' and 'Assistant', with 'Assistant' currently selected.

```
activity_main.xml x MainActivity.java x activity_cloud_message.xml x C
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity">
9
10    <TextView
11        android:id="@+id/firebaseCloudMessagingBtn"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:text="Firebase Cloud Messaging"
15        app:layout_constraintBottom_toBottomOf="parent"
16        app:layout_constraintLeft_toLeftOf="parent"
17        app:layout_constraintRight_toRightOf="parent"
18        app:layout_constraintTop_toTopOf="parent" />
19
20 </androidx.constraintlayout.widget.ConstraintLayout>
```



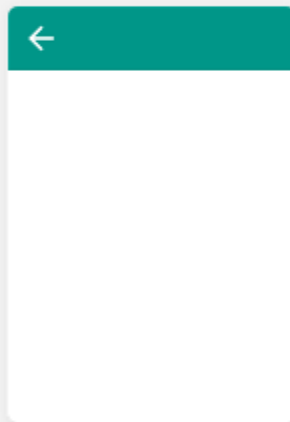


Configure Activity

Android Studio



Creates a new empty activity



Activity Name:

CloudMessageActivity

☒ Generate Layout File

Layout Name:

activity_cloud_message

☐ Launcher Activity

Package name:

com.example.firebasepushapp

Source Language:

Java

The name of the activity class to create

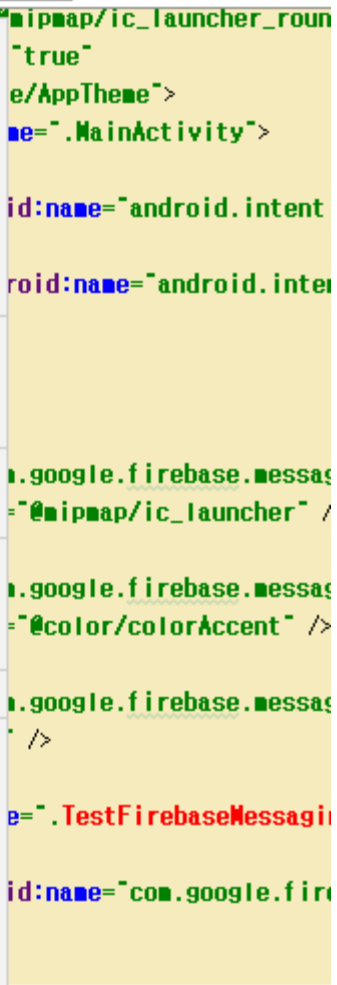
Previous

Next

Cancel

Finish


```
activity_main.xml x MainActivity.java x activity_cloud_message.xml x C
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".CloudMessageActivity">
9
10    <Button
11        android:id="@+id/tokenBtn"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:text="Create Token"
15        app:layout_constraintTop_toTopOf="parent"
16        app:layout_constraintBottom_toBottomOf="parent"
17        app:layout_constraintLeft_toLeftOf="parent"
18        app:layout_constraintRight_toRightOf="parent"/>
19
20 </androidx.constraintlayout.widget.ConstraintLayout>
```





Configure Component

Android Studio

Creates a new service component and adds it to your Android manifest.

Class Name:

TestFirebaseMessagingService

☒ Exported

☒ Enabled

Source Language:

Java

Target Source Set:

main

The source set within which to generate new project files.

If you specify a source set that does not yet exist on disk, a folder will be created for it.

Previous

Next

Cancel

Finish

```
activity_main.xml x MainActivity.java x activity_cloud_message.xml x CloudMessageActivity.java x
1 package com.example.firebasepushapp;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.content.Intent;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.widget.Button;
9
10 public class MainActivity extends AppCompatActivity implements View.OnClickListener{
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16
17         Button firebasecloudmessagingbtn = (Button)findViewById(R.id.firebaseCloudMessagingBtn);
18         firebasecloudmessagingbtn.setOnClickListener(this);
19     }
20
21     @Override
22     public void onClick(View view) {
23         Intent i = null;
24         switch (view.getId()) {
25             case R.id.firebaseCloudMessagingBtn:
26                 i = new Intent( packageContext this, CloudMessageActivity.class);
27                 startActivity(i);
28                 break;
29             default:
30                 break;
31         }
32     }
33 }
```

```
activity_main.xml x activity_cloud_message.xml x CloudMessageActivity.java x AndroidManifest.xml
1 package com.example.firebasepushapp;
2
3 import androidx.annotation.NonNull;
4 import androidx.appcompat.app.AppCompatActivity;
5
6 import android.os.Bundle;
7 import android.util.Log;
8 import android.view.View;
9 import android.widget.Button;
10 import android.widget.Toast;
11
12 import com.google.android.gms.tasks.OnCompleteListener;
13 import com.google.android.gms.tasks.Task;
14 import com.google.firebase.iid.FirebaseInstanceId;
15 import com.google.firebase.iid.InstanceIdResult;
16
17 public class CloudMessageActivity extends AppCompatActivity implements View.OnClickListener {
18
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_cloud_message);
23
24         Button tokenbtn = (Button)findViewById(R.id.tokenBtn);
25         tokenbtn.setOnClickListener(this);
26     }
27 }
```

```

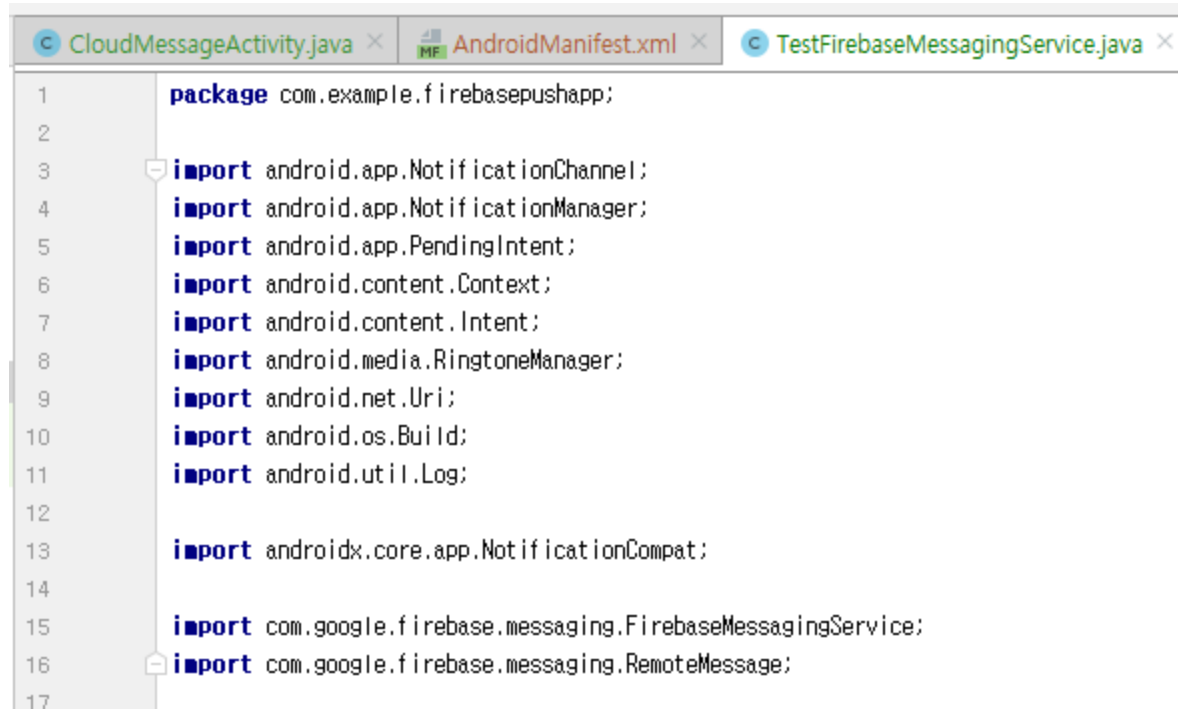
27
28
29 ① ↑ @
30
31
32
33
34
35 ① ↑
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

@Override
public void onClick(View view) {
    switch (view.getId()) {
        case R.id.tokenBtn:
            FirebaseInstanceId.getInstance().getInstanceId()
                .addOnCompleteListener(new OnCompleteListener<InstanceIdResult>() {
                    @Override
                    public void onComplete(@NonNull Task<InstanceIdResult> task) {
                        if (!task.isSuccessful()) {
                            Log.d( tag: "test", msg: "getInstanceId failed", task.getException());
                            return;
                        }

                        // Get new Instance ID token
                        String token = task.getResult().getToken();

                        // Log and toast
                        String msg = "InstanceID Token: " + token;
                        Log.d( tag: "test", msg);
                        Toast.makeText( context: CloudMessageActivity.this, msg, Toast.LENGTH_SHORT).show();
                    }
                });
            break;
        default:
            break;
    }
}

```



```
1 package com.example.firebasepushapp;
2
3 import android.app.NotificationChannel;
4 import android.app.NotificationManager;
5 import android.app.PendingIntent;
6 import android.content.Context;
7 import android.content.Intent;
8 import android.media.RingtoneManager;
9 import android.net.Uri;
10 import android.os.Build;
11 import android.util.Log;
12
13 import androidx.core.app.NotificationCompat;
14
15 import com.google.firebase.messaging.FirebaseMessagingService;
16 import com.google.firebase.messaging.RemoteMessage;
17
```

```

17
18 public class TestFirebaseMessagingService extends FirebaseMessagingService {
19     public TestFirebaseMessagingService() { }
20
21     @Override
22     public void onMessageReceived(RemoteMessage remoteMessage) {
23         if (remoteMessage.getData().size() > 0) {
24             Log.d( tag: "test", msg: "Message data payload: " + remoteMessage.getData());
25
26             if (false) {
27                 // For long-running tasks (10 seconds or more) use WorkManager.
28                 scheduleJob();
29             } else {
30                 // Handle message within 10 seconds
31                 handleNow(remoteMessage);
32             }
33
34             // Check if message contains a notification payload.
35             if (remoteMessage.getNotification() != null) {
36                 Log.d( tag: "test", msg: "Message Notification Body: " + remoteMessage.getNotification().getBody());
37             }
38         }
39     }
40 }

```



```

40
41
42  
43
44
45
46
47
48
49
50
51 @
52
53
54
55
56
57
58
59

```

```

@Override
public void onNewToken(String token) {
    Log.d( tag: "test", msg: "onNewToken token = " + token);
    sendRegistrationToServer(token);
}

private void scheduleJob() {
    Log.d( tag: "test", msg: "scheduleJob in");
}

private void handleNow(RemoteMessage remoteMessage) {
    Log.d( tag: "test", msg: "handleNow in");
    sendNotification(remoteMessage.getData().get("message"));
}

private void sendRegistrationToServer(String token) {
    Log.d( tag: "test", msg: "onNewToken token = " + token);
}

```

```

59
60 private void sendNotification(String messageBody) {
61     Intent intent = new Intent( packageContext: this, MainActivity.class);
62     intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
63     PendingIntent pendingIntent = PendingIntent.getActivity( context: this, requestCode: 0 /* Request code */, intent,
64         PendingIntent.FLAG_ONE_SHOT);
65
66     String channelId = "1000";
67     Uri defaultSoundUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
68     NotificationCompat.Builder notificationBuilder =
69         new NotificationCompat.Builder( context: this, channelId)
70         .setSmallIcon(R.mipmap.ic_launcher)
71         .setContentTitle(getString(R.string.app_name))
72         .setContentText(messageBody)
73         .setAutoCancel(true)
74         .setSound(defaultSoundUri)
75         .setContentIntent(pendingIntent);
76
77     NotificationManager notificationManager =
78         (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
79
80     // Since android Oreo notification channel is needed.
81     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
82         NotificationChannel channel = new NotificationChannel(channelId,
83             name: "Channel human readable title",
84             NotificationManager.IMPORTANCE_DEFAULT);
85         notificationManager.createNotificationChannel(channel);
86     }
87
88     notificationManager.notify( id: 0 /* ID of notification */, notificationBuilder.build());
89 }
90 }

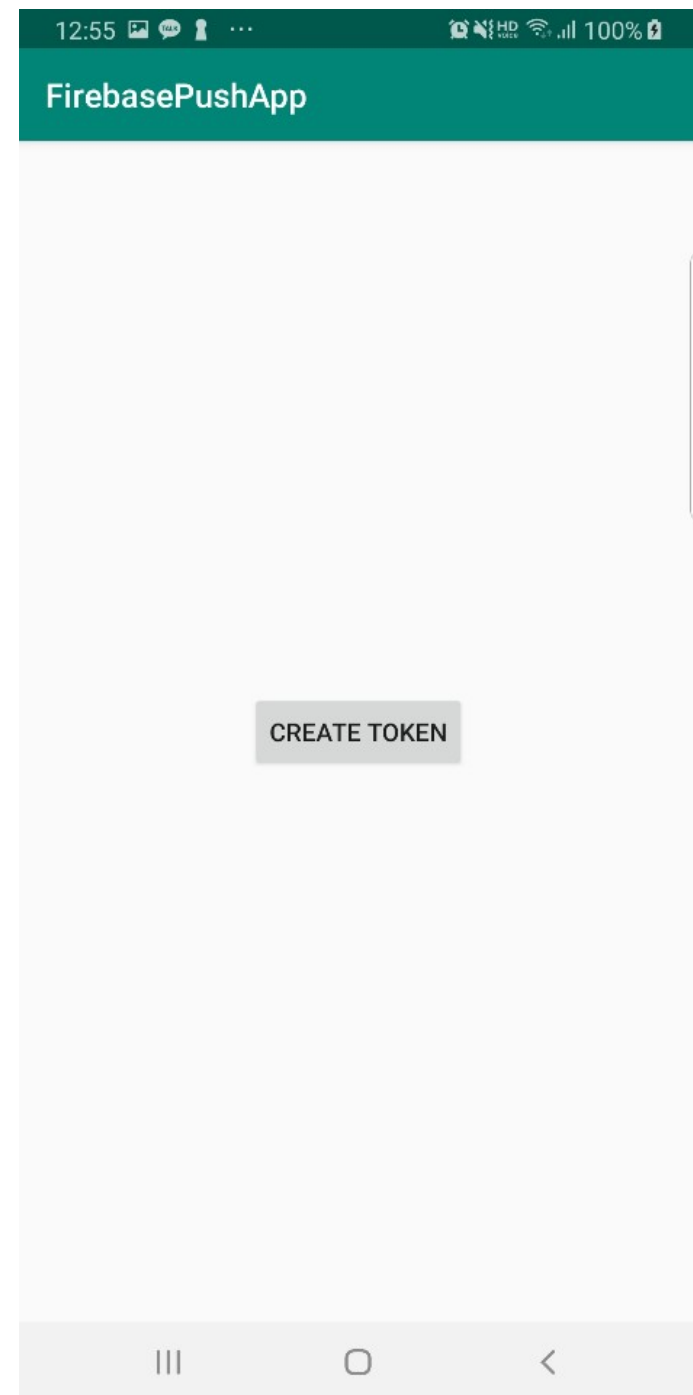
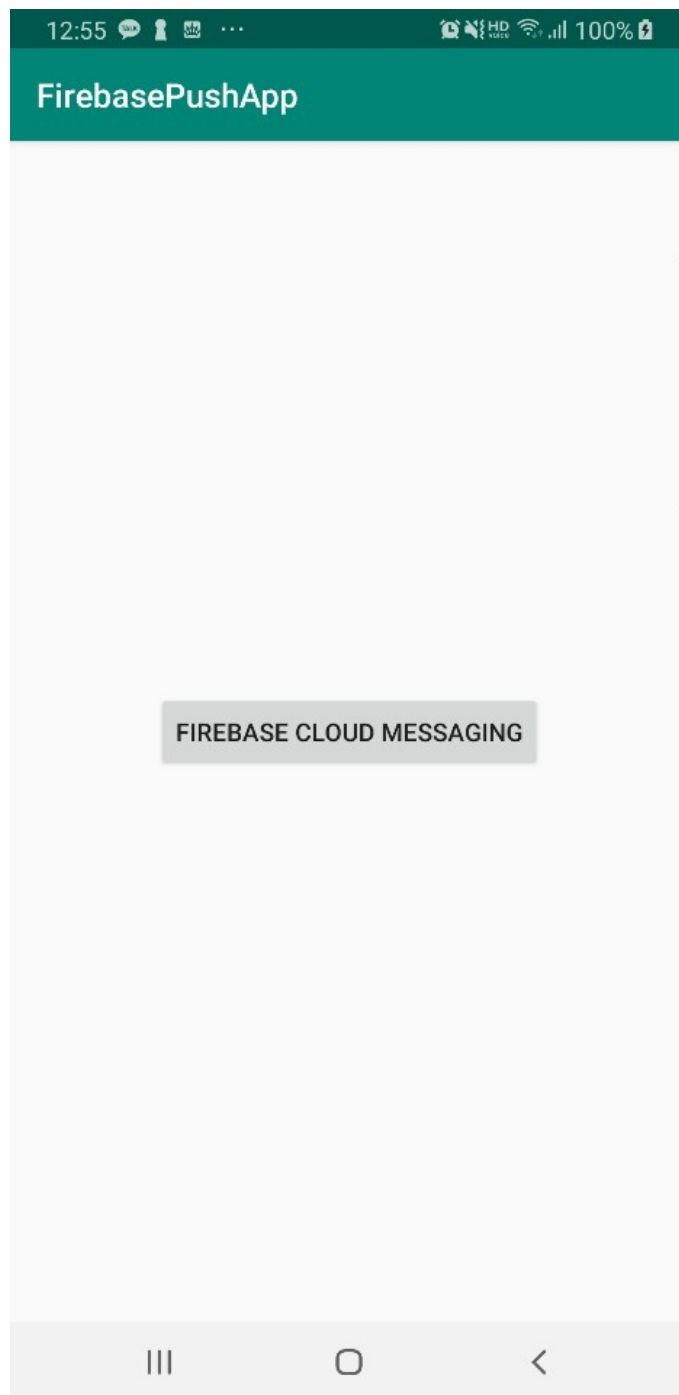
```

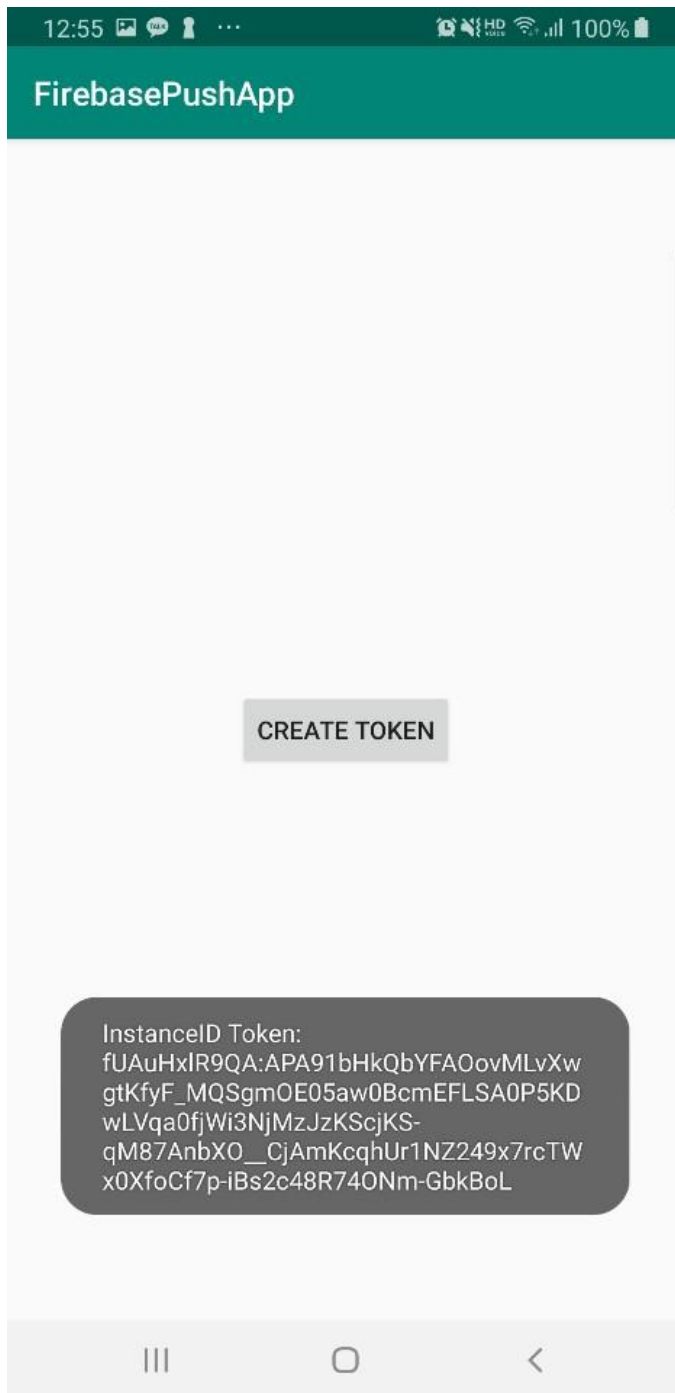
```
CloudMessageActivity.java x AndroidManifest.xml x strings.xml x FirebasePushApp x
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.firebasepushapp">
4
5     <uses-permission android:name="android.permission.INTERNET" />
6     <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
7
8     <application
9         android:allowBackup="true"
10        android:icon="@mipmap/ic_launcher"
11        android:label="@string/app_name"
12        android:roundIcon="@mipmap/ic_launcher_round"
13        android:supportRtl="true"
14        android:theme="@style/AppTheme">
15        <activity android:name=".CloudMessageActivity"></activity>
16    </application>
```

```

16
17
18     <service
19         android:name=".TestFirebaseMessagingService">
20         <intent-filter>
21             <action android:name="com.google.firebase.MESSAGING_EVENT" />
22         </intent-filter>
23     </service>
24
25     <activity android:name=".MainActivity">
26         <intent-filter>
27             <action android:name="android.intent.action.MAIN" />
28             <category android:name="android.intent.category.LAUNCHER" />
29         </intent-filter>
30     </activity>
31
32     <meta-data
33         android:name="com.google.firebase.messaging.default_notification_icon"
34         android:resource="@mipmap/ic_launcher" />
35     <meta-data
36         android:name="com.google.firebase.messaging.default_notification_color"
37         android:resource="@color/colorAccent" /> <!-- fcm default notification channel id -->
38     <meta-data
39         android:name="com.google.firebase.messaging.default_notification_channel_id"
40         android:value="1" />
41 </application>
42 </manifest>

```







getpostman.com/downloads/



POSTMAN

Get Postman for Wind

Join 8 million developers and download the **ONLY** complete API Develop

 **Download** ^

[Windows 32-bit](#)

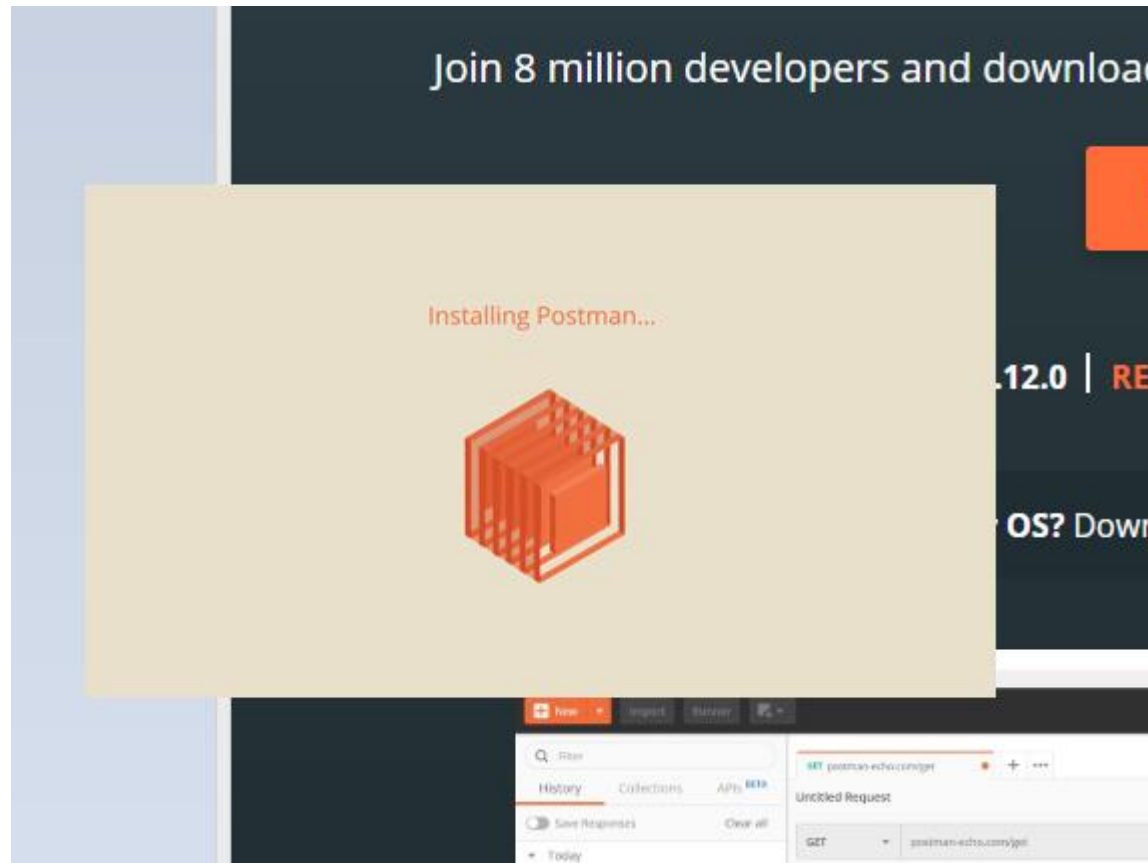
[Windows 64-bit](#)

Version 7.12.0

PRODUCT ROADMAP

Click to download the latest Post

Not your OS? Download for **macOS** or Linux (**x64**)





POSTMAN

Why Sign Up?

- Organize all your API development within Postman Workspaces
- Sync your Postman data across devices
- Backup your data to the Postman cloud
- It's free!



Create Account

[Sign In](#) instead?

Email

Username

Password

[SHOW](#)☐ I agree to the [Terms of Use](#).☒ Keep me signed in[Create free account](#)

or

[Sign up with Google](#)



하나의 계정으로 모든 Google 서비스를

Google 계정으로 로그인



다음

[내 계정 찾기](#)

이렇게 하면 getpostman.com에서 다음 작업을 할 수 있습니다.



개인정보(공개로 설정한 개인정보 포함) 보기



내 이메일 주소를 봅니다.



허용을 클릭하면 이 앱과 Google에서 서비스 약관 및 개인정보처리방침에 따라 내 정보를 사용하도록 허용하게 됩니다. 언제든지 이 권한 및 다른 [계정 권한](#)을 변경할 수 있습니다.

거부

허용

Sign in / Sign up through email instead



1

Set Up Your Workspace with the Right Tools

Which tasks do you want to accomplish with Postman?

- | | |
|--|--|
| <input checked="" type="checkbox"/> Debugging and manual testing | <input checked="" type="checkbox"/> Designing and mocking APIs |
| <input checked="" type="checkbox"/> Automated testing | <input checked="" type="checkbox"/> API documentation |
| <input checked="" type="checkbox"/> Monitoring | <input checked="" type="checkbox"/> Publishing APIs |

2

Custom

Personaliz

Theme

☒ Light

☐ Dark

Set Up Later

Save My Preferences

Set up your Postman team

Bring your entire team to Postman to get most out of Postman's collaboration features.

Add people to your team

Enter an email address

or, share an invite link

Get a link which you can share with your colleagues to join your team.

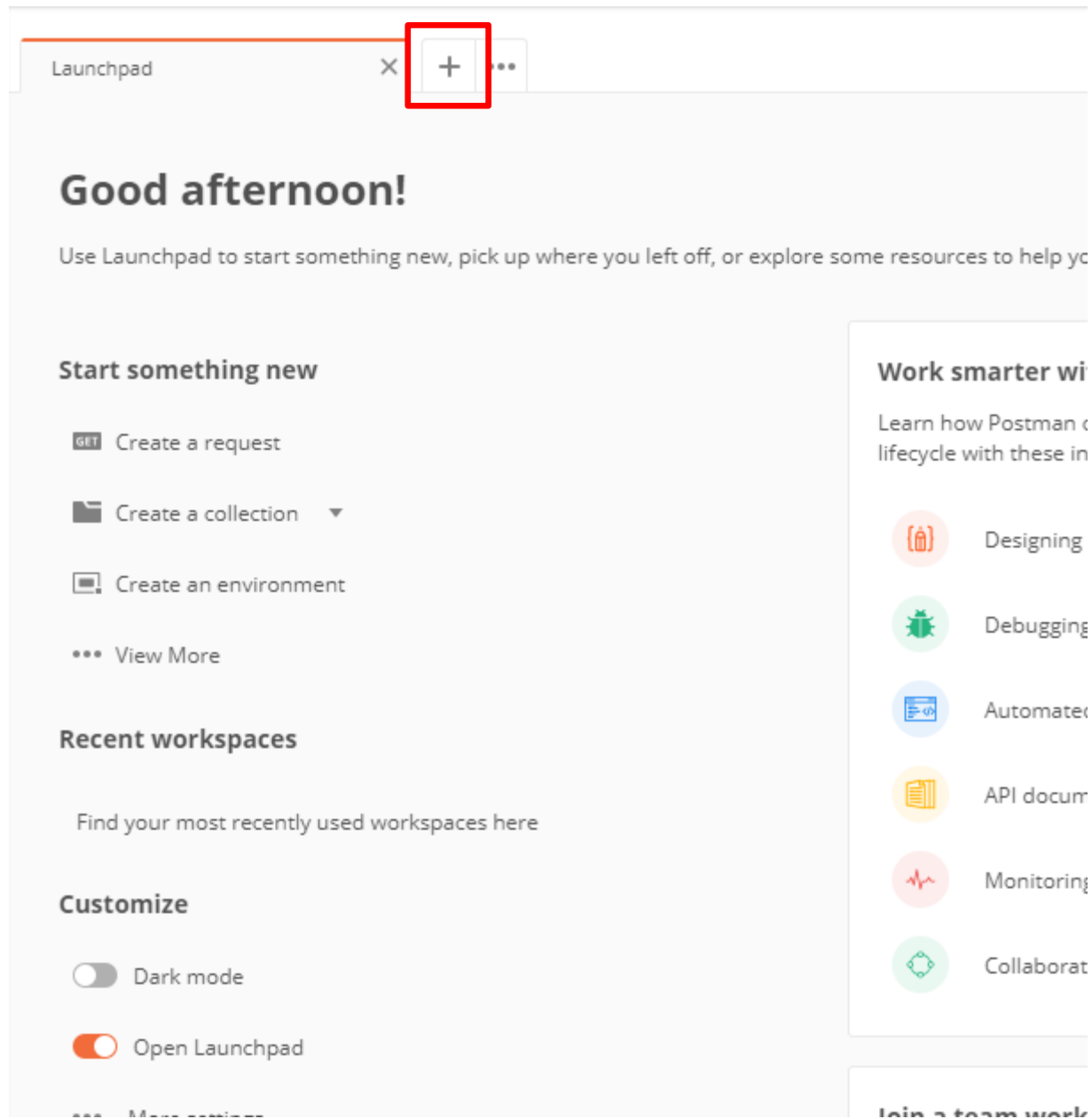
Get Link

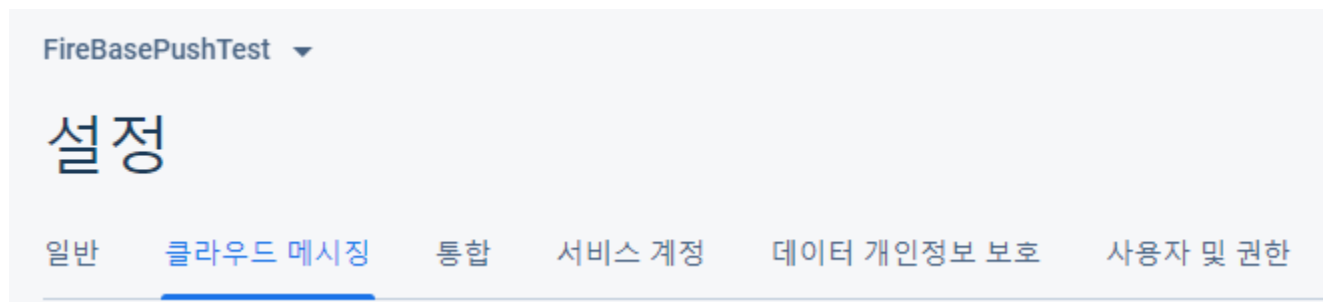
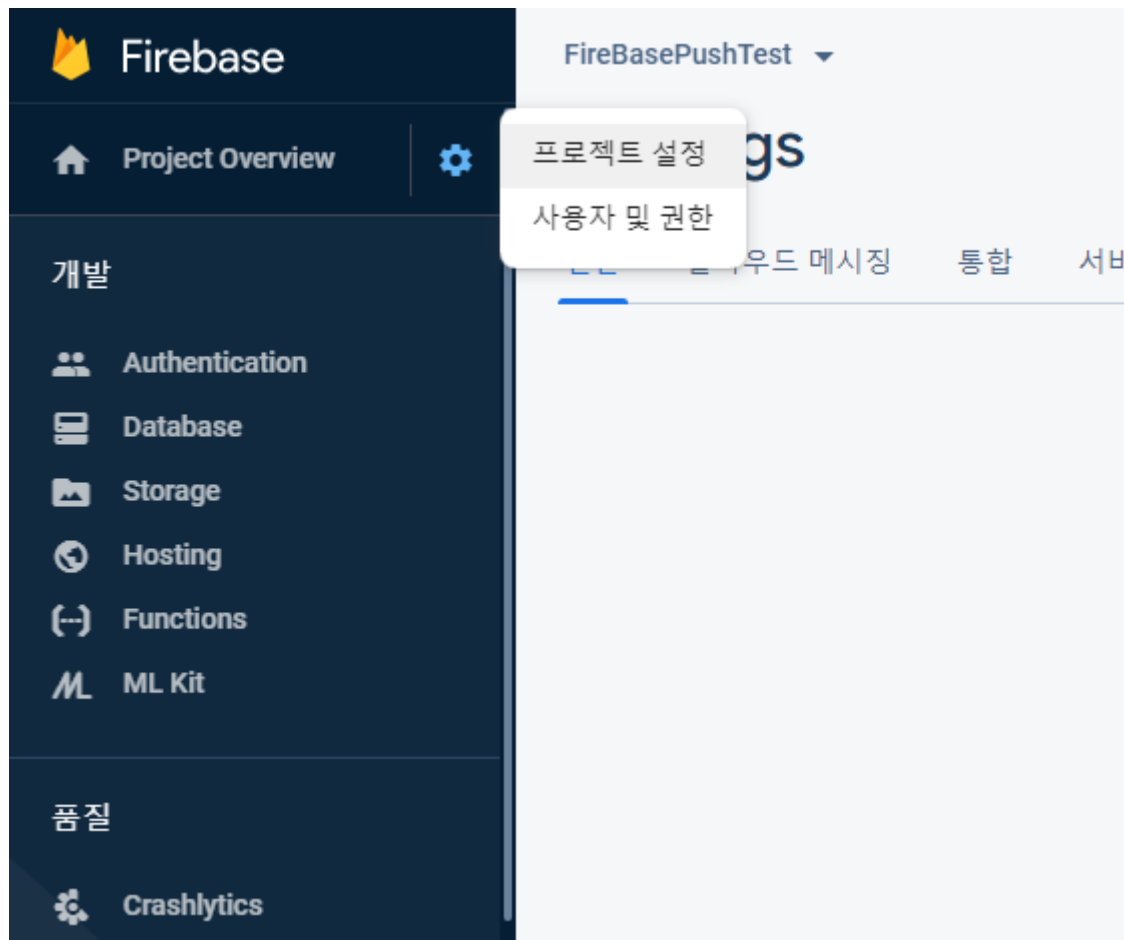
https://app.getpostman.com/join?invite_code=1a2b3c4d



Create team

Skip for now





MANAGE HEADER PRESETS

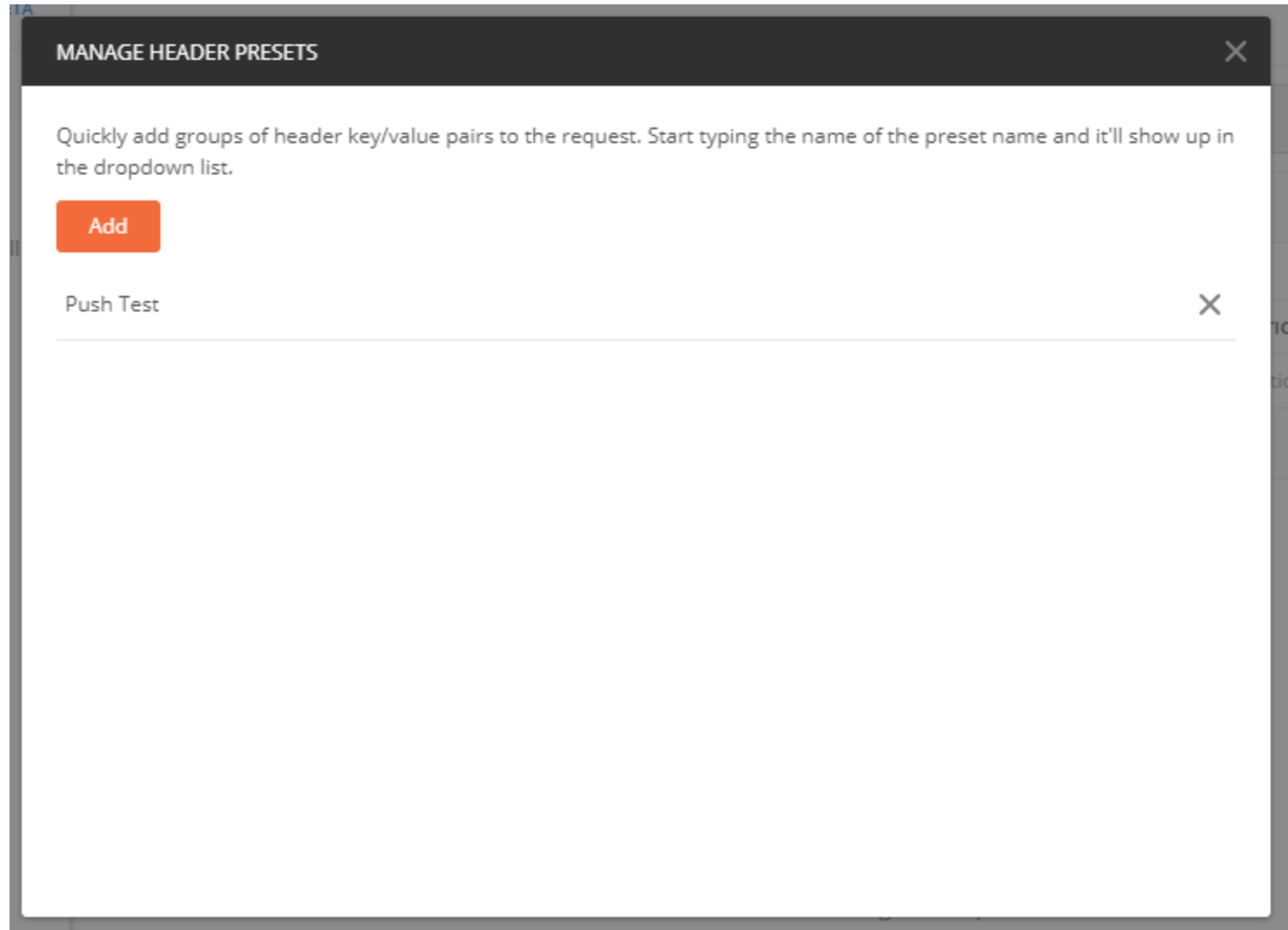
Add Header Preset

Push Test

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	Authorization	key=AAAANIFt02k:APA91bEZlrRiq...			
<input checked="" type="checkbox"/>	Content-Type	application/json			
	Key	Value	Description		

Cancel

Add



Launchpad

POST https://fcm.googleapis.com/fc... + ...

Untitled Request


POST https://fcm.googleapis.com/fcm/send

Params Authorization Headers (2) Body Pre-request Script Tests Settings

▼ Headers (2)

	KEY	VALUE
<input checked="" type="checkbox"/>	Authorization	key=AAAAANIFt02k:APA91bEZlrRiqjDrsOp6JDNciW...
<input checked="" type="checkbox"/>	Content-Type	application/json
	Key	Value

Response



Hit Send to get a response

Build Variants
Structure
Favorites
Layout Captures

```

75
76
77
78
79
80
// Since android Oreo notification
TestFirebaseMessagingService > sendNotificati

```

Run: app
✕

D/ViewRootImpl@9b71fe4[Toast]: dispatchDetachedFromWindow

D/ViewRootImpl@9b71fe4[Toast]: Surface release. android.view.ViewRootImpl.doDie:7954 android.view.ViewRootImpl

D/InputTransport: Input channel destroyed: fd=75

D/ViewRootImpl@946bcfa[CloudMessageActivity]: ViewPostIme pointer 0

D/ViewRootImpl@946bcfa[CloudMessageActivity]: ViewPostIme pointer 1

D/test: InstanceID Token: fUAuHxIR9QA:APA91bHkQbYFA0ovMLvXwgtKfyF_MQSGmOE05aw0BcmEFLSAOP5KDwLVqaOfjWl3NjM

D/InputTransport: Input channel constructed: fd=75

D/ViewRootImpl@a80405a[Toast]: setView = android.widget.LinearLayout{bd3028b V.E..... 0,0-0,0}

V/Toast: Text: Jnst in android.widget.Toast\$TN@6209f68

D/ViewRootImpl@a80405a[Toast]: dispatchAttachedToWindow

D/ViewRootImpl@a80405a[Toast]: Relayout returned: old=[0,63][1080,2094] new=[86,1579][993,1926] result=0x

D/mali_winsys: EGLint new_window_surface(egl_winsys_display *, void *, EGLSurface, EGLConfig, egl_winsys_

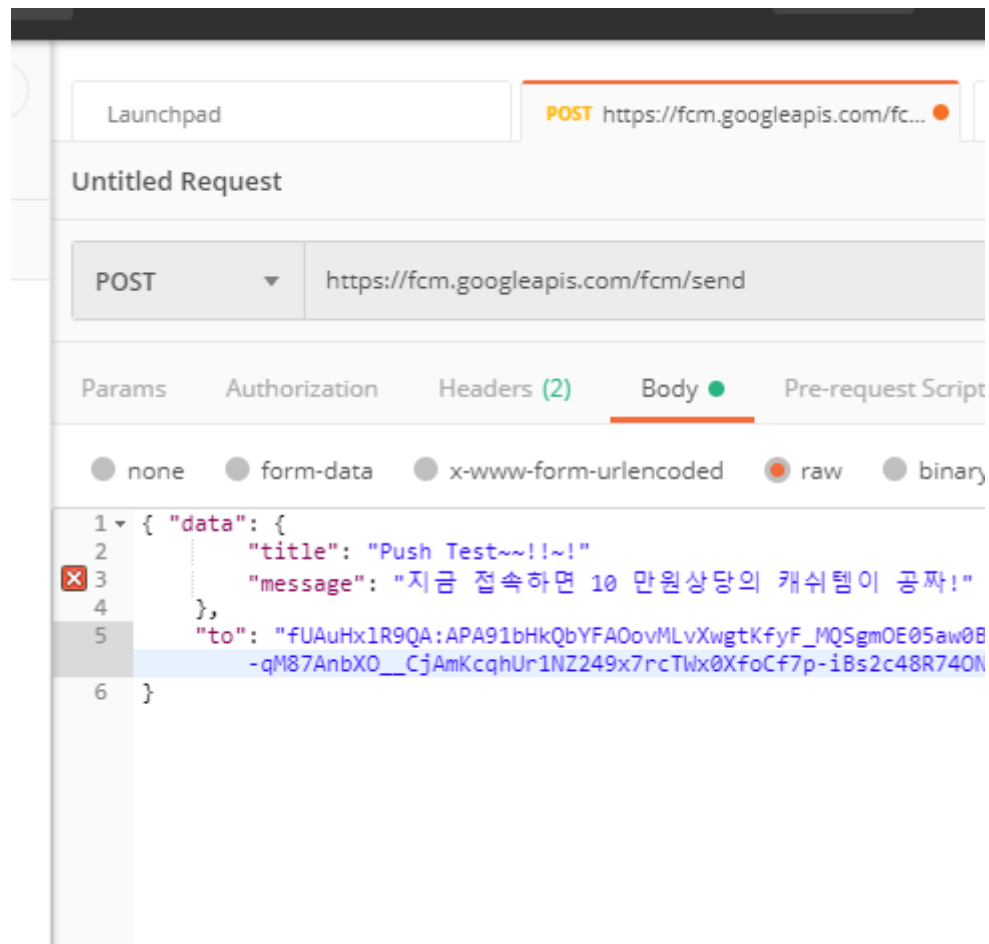
D/OpenGLRenderer: eglCreateWindowSurface = 0x756b0bf400, 0x7579760010

D/ViewRootImpl@a80405a[Toast]: MSG_RESIZED: frame=Rect(86, 1579 - 993, 1926) ci=Rect(0, 0 - 0, 0) vi=Rect

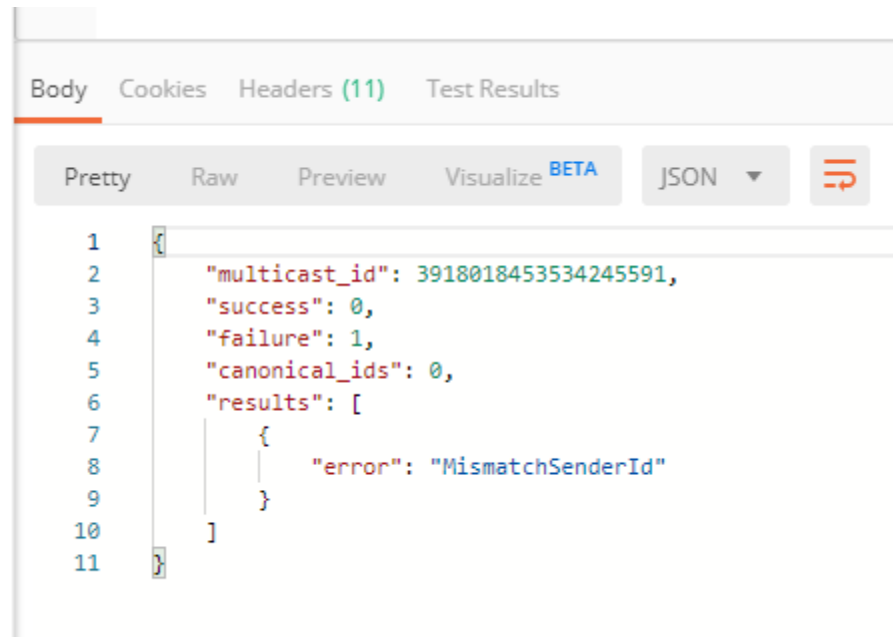
W/libEGL: EGLNativeWindowType 0x7579760010 disconnect failed

TODO
g: Version Control
Terminal
Build
Logcat
Profiler
4: Run

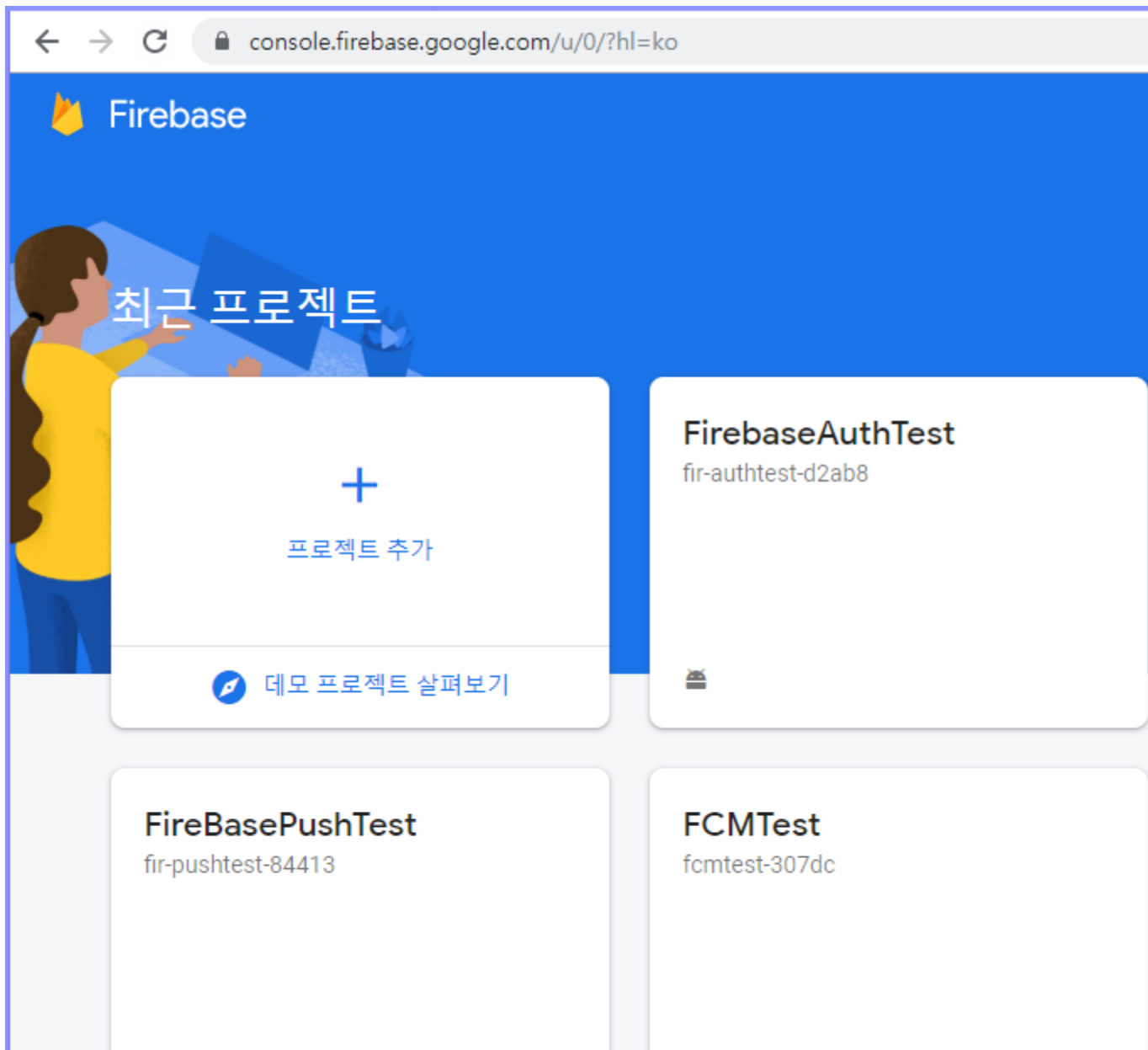
Install successfully finished in 319 ms: App restart successful without requiring a re-install (3 minute ago)

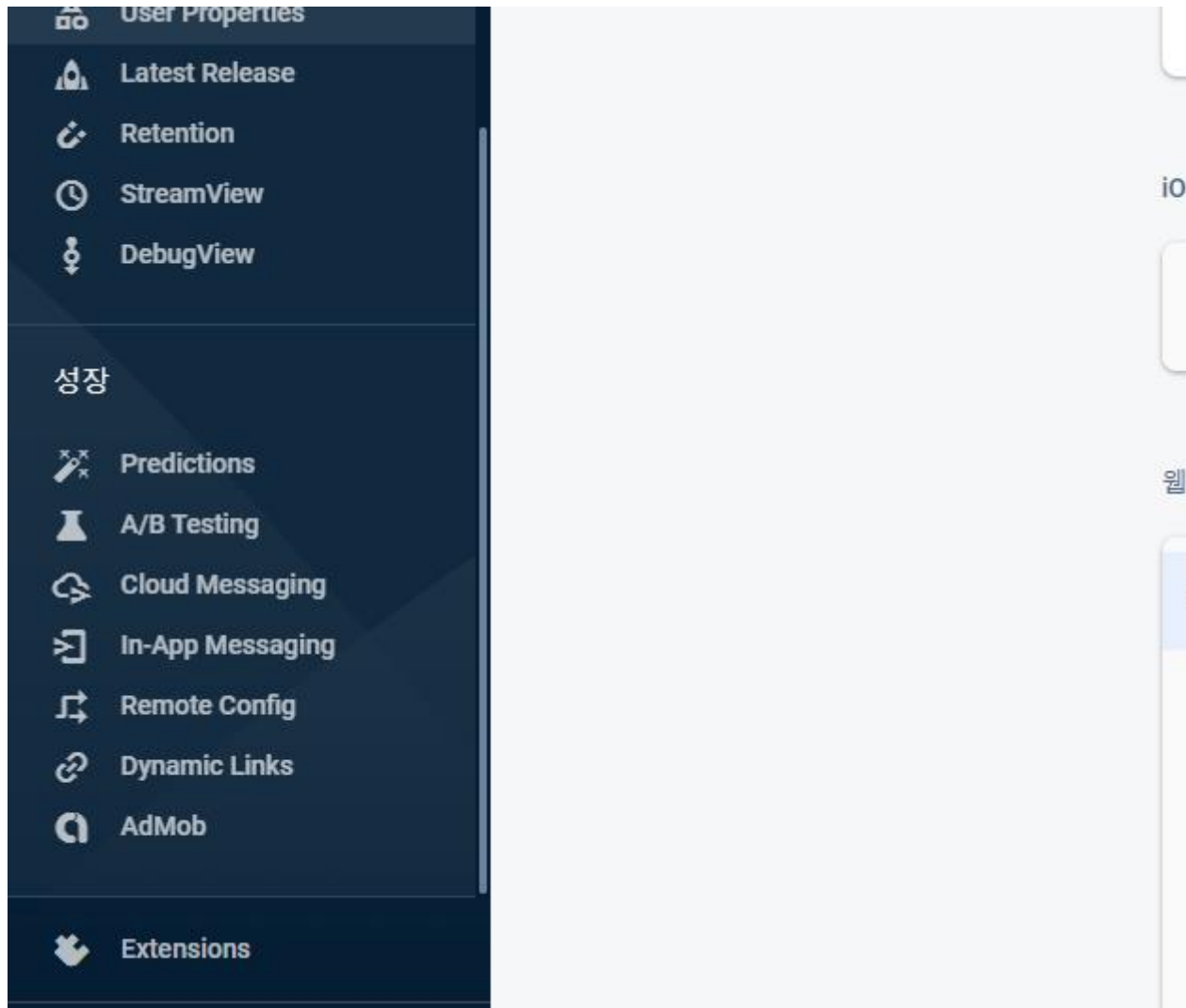


서버키는 Firebase To 는 기기 토큰



```
Body Cookies Headers (11) Test Results
Pretty Raw Preview Visualize BETA JSON ↻
1 {
2   "multicast_id": 3918018453534245591,
3   "success": 0,
4   "failure": 1,
5   "canonical_ids": 0,
6   "results": [
7     {
8       "error": "MismatchSenderId"
9     }
10  ]
11 }
```







5 추가 옵션(선택사항)

메시지 검토


알림 콘텐츠

 지금 접속하면 10 만원 상당의 캐쉬템이 공짜!

대상

 타겟팅 기준 1개와 일치하는 사용자 세그먼트

예약

 지금 보내기

취소

게시



데이터 공유 설정 선택

'FirebasePushApp' 프로젝트의 데이터 공유 설정을 업데이트해야 합니다.

☒ Firebase용 Google 애널리틱스 데이터 공유에 기본 설정을 사용합니다.

- ✓ 모든 Firebase 기능과 애널리틱스 데이터 공유
- ✓ Google 제품과 서비스의 개선을 위해 Google 애널리틱스 데이터를 Google과 공유합니다.
- ✓ 기술 지원을 받기 위해 Google 애널리틱스 데이터를 Google과 공유합니다.
- ✓ 벤치마킹을 사용하기 위해 Google 애널리틱스 데이터를 Google과 공유합니다.
- ✓ Google 애널리틱스 데이터를 Google 계정 전문가와 공유합니다.

☒ [컨트롤러 간 약관](#)에 동의합니다. Google 제품과 서비스의 개선을 위해 Google 애널리틱스 데이터를 공유하는 경우에 이 약관에 동의해야 합니다.
[자세히 알아보기](#)

나중에

완료

1 알림

알림 제목 ②

최고급템 증정 이벤트

알림 텍스트

지금 접속하면 10 만원 상당의 캐쉬템이 공짜!

알림 이미지(선택사항) ②

예: <https://yourapp.com/image.png>



알림 이름(선택사항) ②

이름 입력(선택사항)

기기 미리보기

이 미리보기는 휴대
여줍니다. 실제 메
를 보려면 실제 기기

테스트 메시지

초기 상태

최고급템 증정
지금 접속하면



알림

지금 접속하면 10 만원 상당의 캐쉬템이 공짜!

2

타겟

사용자 세그먼트

주제

사용자 타겟팅 조건...

앱



com.example.firebasepushtest

Target another app

잠재 사용자의 100%가 이 캠페인의 대상 요건 충족:1 ?

다음



알림

지금 접속하면 10 만원 상당의 캐쉬템이 공짜!



타겟

타겟팅 기준 1개와 일치하는 사용자 세그먼트



3 예약

대상 사용자에게 전송

지금

다음

	값
--	---

초안으로 저장

검토

FirestorePushApp ▼

클라우드 메시징

알림

보고서

알림

상태 ?



I'm babo

✓ 완료됨



0/10 Recurring notifications ?



BigQuery에서 지

