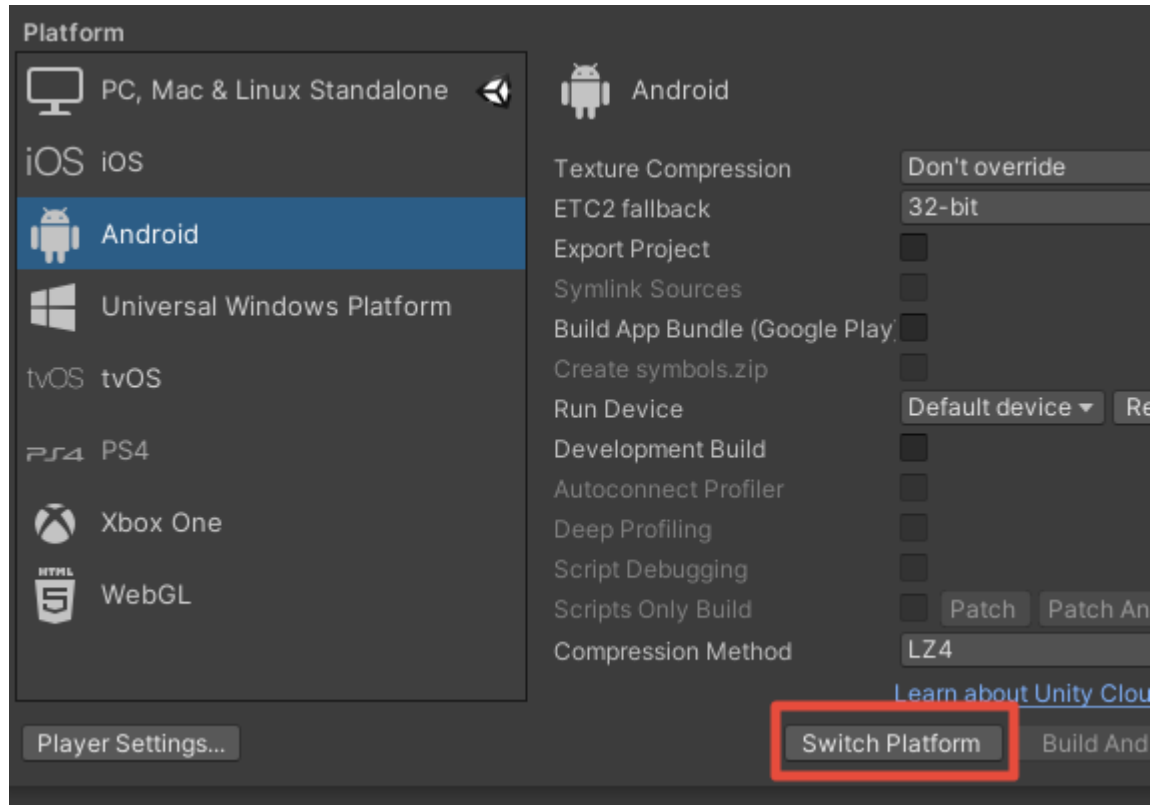


유니티 새 프로젝트를 만든다




Firestore 콘솔 프로젝트를 생성

× 프로젝트 만들기(1/3단계)

프로젝트® 이름을 지정하여
시작하기

프로젝트 이름

FirestoreDatabase



 fir-database-5d53c



계속



Firebase 프로젝트를 위한 Google 애널리틱스



무제한 무료 애널리틱스 솔루션인 Google 애널리틱스를 사용하면 Firebase Crashlytics, 클라우드 메시징, 인앱 메시지, 원격 구성, A/B 테스트, 예측, Cloud Functions에서 타겟팅, 보고 등을 이용할 수 있습니다.



Google 애널리틱스를 통해 다음 기능을 이용할 수 있습니다.



 A/B 테스트 

 장애가 발생하지 않은 사용자 

 Firebase 제품 전반에서 사용자 세분화 
및 타겟팅

 이벤트 기반 Cloud Functions 트리거 

 사용자 행동 예측 

 제한 없는 무료 보고 

☒ 이 프로젝트에서 Google 애널리틱스 사용 설정
권장

[이전](#)

[계속](#)

Google 애널리틱스 구성

Google 애널리틱스 계정 선택 또는 만들기 ⓘ

FirebaseDatabase ▼

애널리틱스 위치 ⓘ

대한민국 ▼

데이터 공유 설정 및 Google 애널리틱스 약관

- ☒ Google 애널리틱스 데이터 공유에 기본 설정을 사용합니다. [Learn more](#)
 - ✓ Google 제품과 서비스의 개선을 위해 Google 애널리틱스 데이터를 Google과 공유합니다.
 - ✓ 벤치마킹을 사용하기 위해 Google 애널리틱스 데이터를 Google과 공유합니다.
 - ✓ 기술 지원을 받기 위해 애널리틱스 데이터를 Google과 공유합니다.
 - ✓ Google 애널리틱스 데이터를 Google 계정 전문가와 공유합니다.
- ☒ [측정 컨트롤러 간 데이터 보호 약관](#)에 동의하며 [EU 최종 사용자 동의 정책](#)이 적용됨을 확인합니다. Google 제품과 서비스의 개선을 위해 Google 애널리틱스 데이터를 공유하는 경우에 이 약관에 동의해야 합니다. [자세히 알아보기](#)
- ☒ [Google 애널리틱스 약관](#)에 동의합니다.

프로젝트를 만들면 새 Google 애널리틱스 속성이 생성되고 Firebase 프로젝트에 연결됩니다. 이 연결을 통해 제품 간에 데이터 흐름이 활성화됩니다. Google 애널리틱스 속성에서 Firebase로 내보낸 데이터에는 Firebase 서비스 약관이 적용되지만 Google 애널리틱스로 가져온 Firebase 데이터에는 Google 애널리틱스 서비스 약관이 적용됩니다. [자세히 알아보기](#)

[이전](#)


프로젝트 만들기




FirebaseDatabase

✓ 새 프로젝트가 준비되었습니다.


계속


 **Firebase**


[프로젝트 개요](#)





빌드


 Authentication


 Cloud Firestore

 Realtime Database


 Storage


 Hosting


 Functions


 Machine Learning

출시 및 모니터링


 Crashlytics


 Performance


 Test Lab


 App Distribution

애널리틱스

 Dashboard

 Realtime

 Events

 Conversions

FirestoreDatabase ▾


FirestoreDatabase

Spark 요금제



앱에 Firebase를 추가하여 시작하기

iOS

앱을 추가하여 시작하기



순식간에 앱 데이터 저장 및 동기화



× Unity 앱에 Firebase 추가

1 앱 등록

iOS 및 Android 모두에서 게임을 출시하시나요? 이 경우 동일한 Firebase 프로젝트로 Unity 앱의 빌드 대상을 둘 다 등록해야 합니다. 등록 후 Firebase 프로젝트에서 Unity 앱이 Android 또는 iOS 앱으로 표시됩니다.

참고: 지금 iOS 및 Android 빌드 대상을 함께 등록할 수 있습니다.

또는 필요한 경우 지금 빌드 대상 하나를 등록한 후 나중에 이 설정 마법사로 돌아와서 다른 빌드 대상을 등록할 수 있습니다.

☐ iOS 앱으로 등록

iOS 번들 ID ⓘ

com.company.appname

앱 닉네임 (선택사항) ⓘ

내 iOS 앱

☒ Android 앱으로 등록

Android 패키지 이름 ⓘ

com.smilejsu.FirebaseDatabase

앱 닉네임 (선택사항) ⓘ

내 Android 앱

앱 등록



× Unity 앱에 Firebase 추가

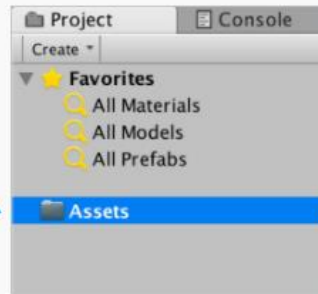
✓ 앱 등록

2 구성 파일 다운로드

📄 google-services.json 다운로드

Unity 프로젝트의 프로젝트 창을 연 후 다운로드한 Firebase 구성 파일을 애셋 폴더로 이동하세요.

참고: 애셋 폴더 내의 원하는 위치에 Firebase 구성 파일을 배치할 수 있습니다.

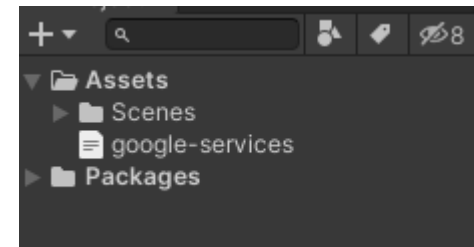


이전

다음

3 Firebase SDK 추가

4 다음 단계



× Unity 앱에 Firebase 추가



앱 등록



구성 파일 다운로드



Firebase SDK 추가



Firebase Unity SDK(Zip) 다운로드

1. 다운로드한 SDK의 압축을 편리한 위치에 풉니다.
2. Unity 프로젝트를 열고 '애셋 > 패키지 가져오기 > 맞춤 패키지'로 이동합니다.
3. 압축을 푼 SDK에서 선택하여 애널리틱스 SDK(FirebaseAnalytics.unitypackage)를 가져옵니다. 원하는 다른 [Firebase 제품](#)의 SDK도 추가합니다.
4. Unity 패키지 가져오기 창에서 가져오기를 클릭합니다.

이전

다음



다음 단계

× Unity 앱에 Firebase 추가

- ✓ 앱 등록
- ✓ 구성 파일 다운로드
- ✓ Firebase SDK 추가
- 4 다음 단계

설정이 끝났습니다.

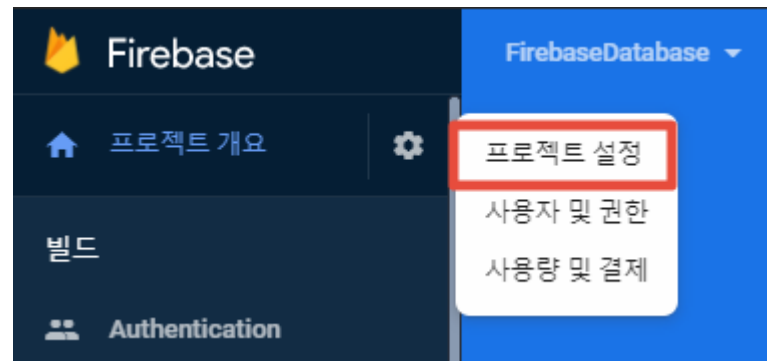
앱에서 사용할 각 Firebase 제품을 시작하는 방법은 [문서](#)를 확인하세요.

[샘플 Firebase 앱](#)도 살펴볼 수 있습니다.

또는 Console로 이동하여 Firebase를 살펴보세요.

[이전](#)

[콘솔로 이동](#)



Android 앱



com.smilejsu.FirebaseDatabase

SDK 설정 및 구성

앱의 Firebase SDK를 다시 구성해야 하나요? SDK 설정 안내를 다시 살펴보거나 앱의 키 및 식별자가 포함된 구성 파일을 다운로드하세요.



SDK 안내 보기



google-services.json

앱 ID ⓘ

1:698529635002:android:747253a79b795146a84083

앱 닉네임

닉네임 추가 ✎

패키지 이름

com.smilejsu.FirebaseDatabase

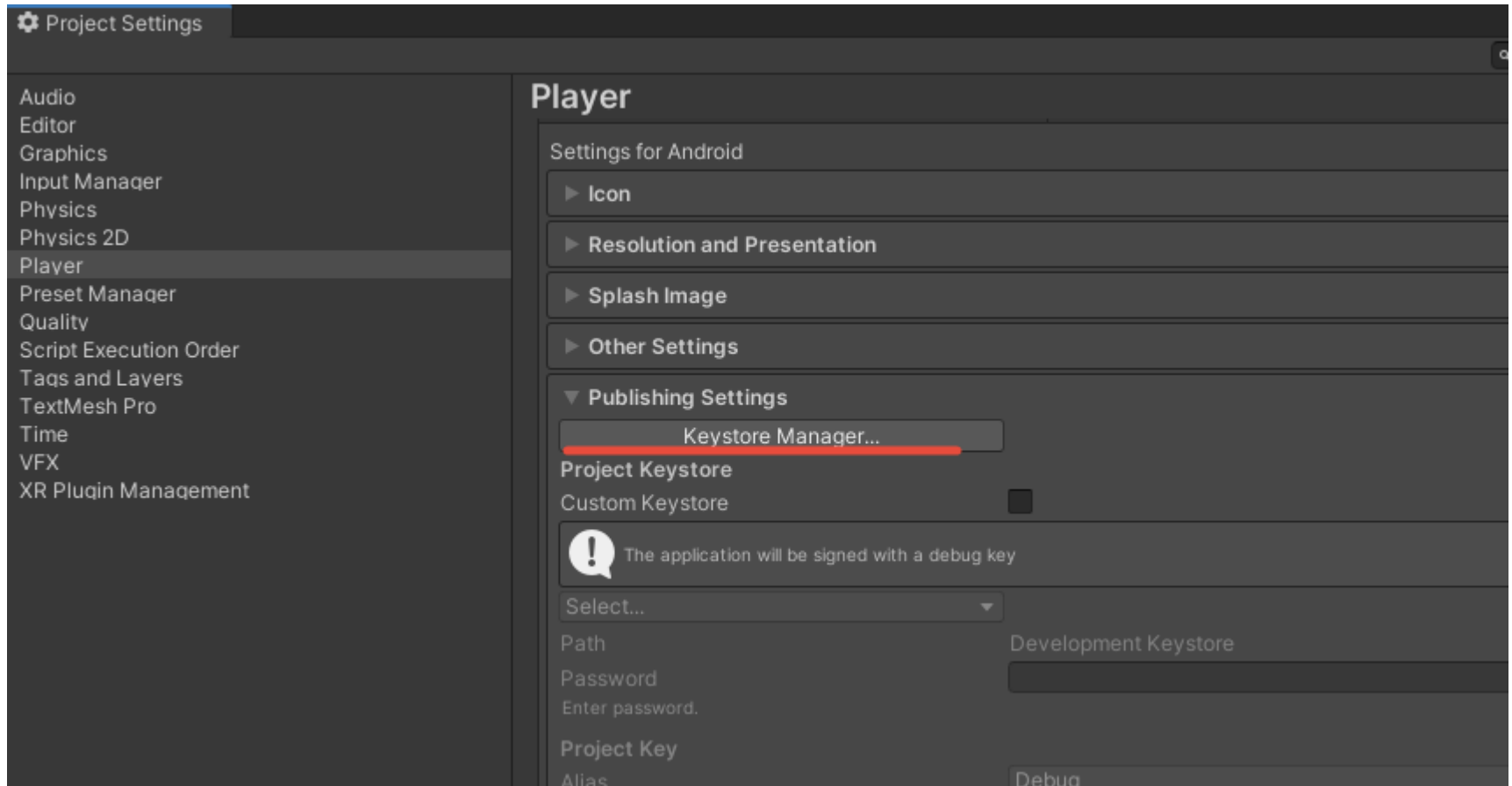
SHA 인증서 지문 ⓘ

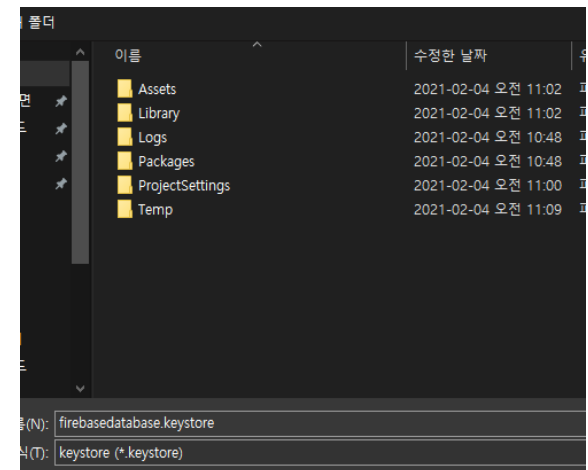
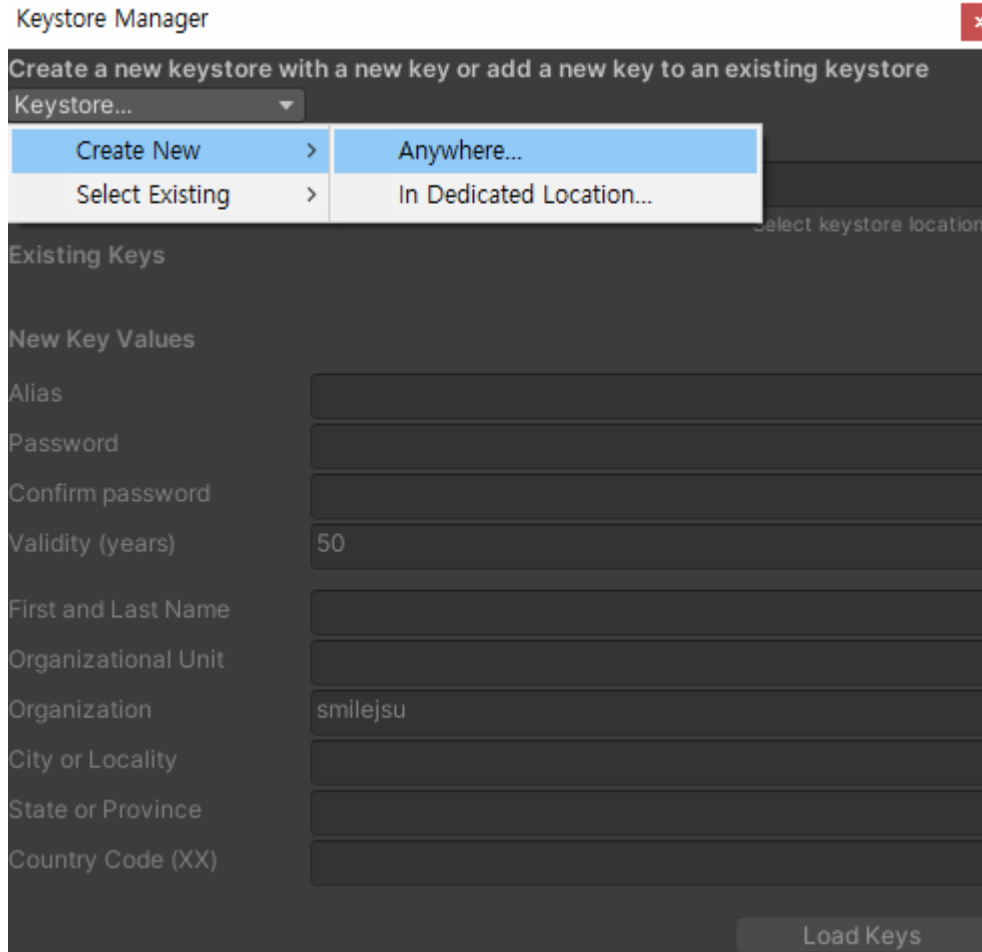
유형 ⓘ

디지털 지문 추가

앱 삭제

keystore를 만든다





Keystore Manager

Create a new keystore with a new key or add a new key to an existing keystore

Keystore... ▾

C:/Users/smile/workspace/unity/FirebaseDatabase/firebasedatabase.keystore

Password *****

Confirm password *****

New Key Values

Alias smilejsu

Password *****

Confirm password *****

Validity (years) 50

First and Last Name

Organizational Unit

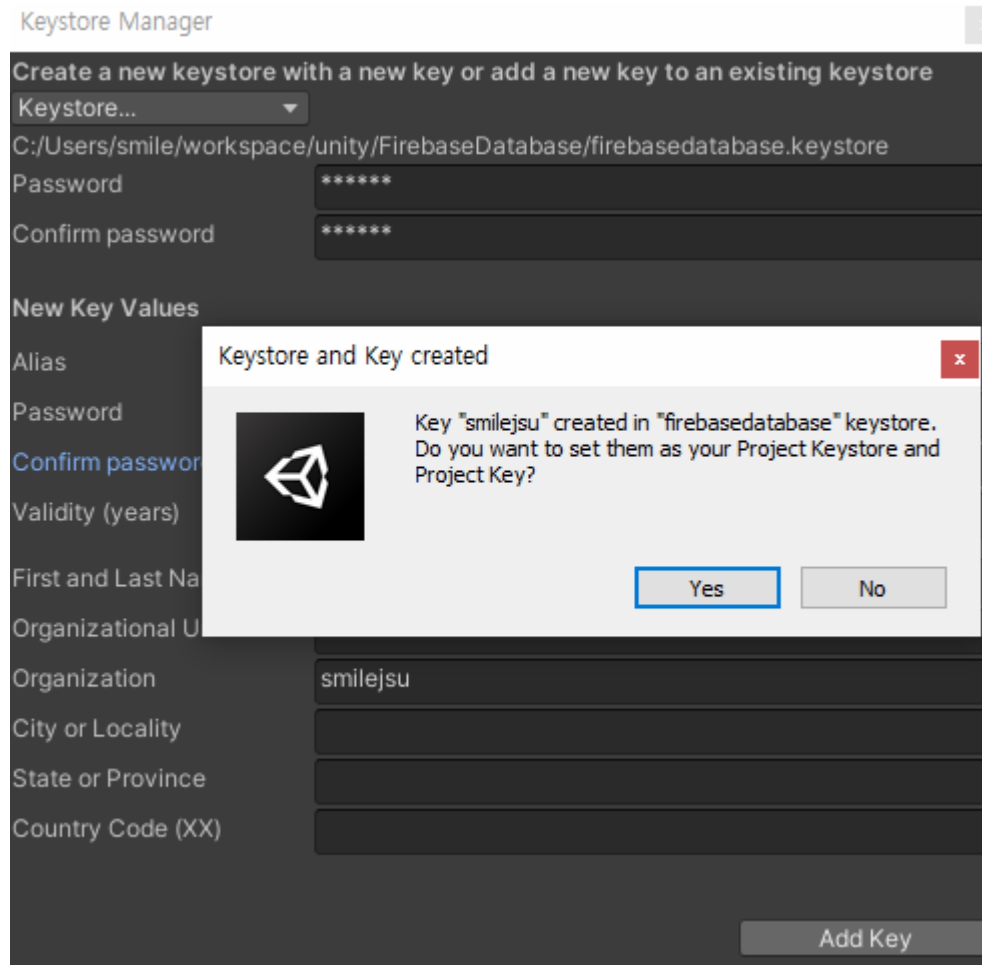
Organization smilejsu

City or Locality

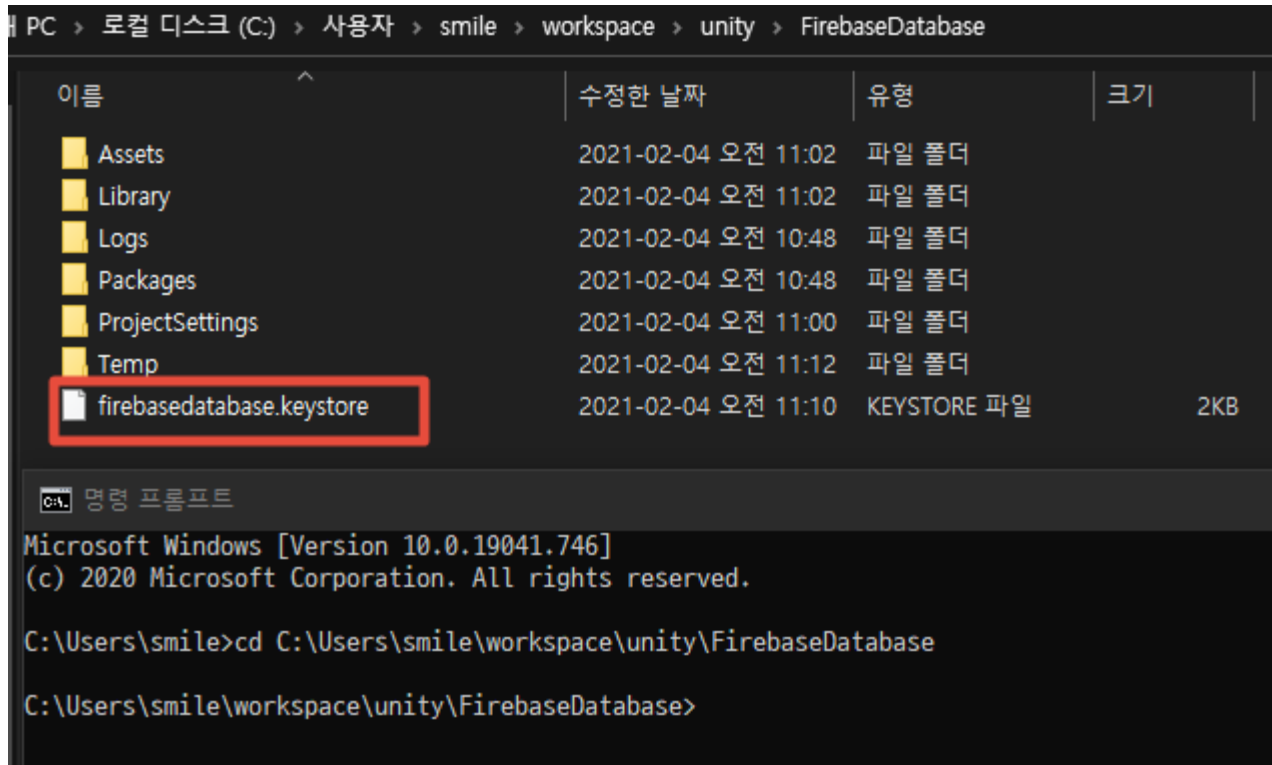
State or Province

Country Code (XX)

Add Key



keystore가 있는 위치에서 cmd창 실행



keytool -list -v -keystore {keystore path}

```
C:\Users\smile\workspace\unity\FirebaseDatabase>keytool -list -v -keystore ./firebasedatabase.keystore
키 저장소 비밀번호 입력:
키 저장소 유형: JKS
키 저장소 제공자: SUN

키 저장소에 1개의 항목이 포함되어 있습니다.

별칭 이름: smilejsu
생성 날짜: 2021. 2. 4
항목 유형: PrivateKeyEntry
인증서 체인 길이: 1
인증서[1]:
소유자: O=smilejsu
발행자: O=smilejsu
일련 번호: 29a10ce3
적합한 시작 날짜: Thu Feb 04 11:10:25 KST 2021 종료 날짜: Fri Jan 23 11:10:25 KST 2071
인증서 지문:
    MD5: 4A:00:EA:10:54:68:55:66:CF:70:48:89:D1:D8:EE:06
    SHA1: A4:D3:DC:54:10:92:35:37:05:5D:55:D7:91:F1:4F:87:55:A6:DF:32
    SHA256: CA:CD:59:38:30:2A:5D:E9:7E:F0:02:A3:25:75:C8:AB:BC:7F:A6:2B:A1:48:7C:A5:A9:3D:23:DC:EE:1B:50:D5
서명 알고리즘 이름: SHA1withRSA
주체 공용 키 알고리즘: 2048비트 RSA 키
버전: 3

*****
*****

Warning:
JKS 키 저장소는 고유 형식을 사용합니다. "keytool -importkeystore -srckeystore ./firebasedatabase.keystore -destkeystore
./firebasedatabase.keystore -deststoretype pkcs12"를 사용하는 산업 표준 형식인 PKCS12로 이전하는 것이 좋습니다.

C:\Users\smile\workspace\unity\FirebaseDatabase>
```

keytool -list -v -keystore ./firebasedatabase.keystore

sha1또는 sha256 둘 중 하나 복사해서 인증서 지문에 입력

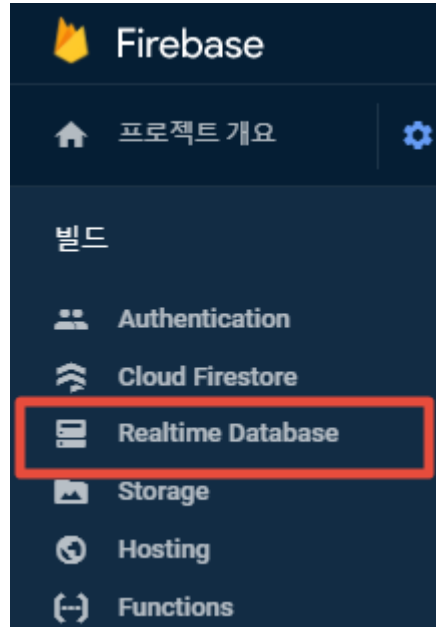
인증서 지문

A4:D3:DC:54:10:92:35:37:05:5D:55:D7:91:F1:4F:87:55:A6:DF:32

SHA1 SHA256

취소 저장

인증서 지문:
MD5: 4A:00:EA:10:54:68:55:66:0E:70:48:89:D1:D8:FF:06
SHA1: A4:D3:DC:54:10:92:35:37:05:5D:55:D7:91:F1:4F:87:55:A6:DF:32
SHA256: CA:CD:59:38:30:2A:5D:E9:7E:F0:02:A3:25:75:C8:AB:BC:7F:A6:2B:A1:48:7C:A5:A9:3D:23:DC:EE:1B:50:D5
서명 알고리즘 이름 SHA1withRSA
주체 공용 키 약리크 2048비트 RSA 키



Realtime Database

실시간으로 데이터 저장 및 동기화

데이터베이스 만들기

데이터베이스 설정



1 데이터베이스 옵션 — 2 보안 규칙

위치 설정은 RTDB 데이터가 저장되는 위치를 결정합니다.

실시간 데이터베이스 위치

미국(us-central1)



취소

다음

1 데이터베이스 옵션 ————— 2 보안 규칙

[자세히 알아보기](#)

- ```
{
 "rules": {
 ".read": false,
 ".write": false
 }
}
```

취소

사용 설정

The screenshot shows the Firebase Realtime Database console. In the left sidebar, the 'Realtime Database' option is highlighted. The main panel displays the 'Rules' tab, which is also highlighted. The 'Publish' button is visible and highlighted. The rules configuration is shown as follows:

```

1 {
2 "rules": {
3 ".read": true,
4 ".write": true
5 }
6 }

```



내 앱


앱 추가


Android 앱

 com.smilejsu.FirebaseDatabase

### SDK 설정 및 구성

앱의 Firebase SDK를 다시 구성해야 하나요? SDK 설정 안내를 다시 살펴보거나 앱의 키 및 식별자가 포함된 구성 파일을 다운로드하세요.


 SDK 안내 보기

 google-services.json

앱 ID ⓘ

1:698529635002:android:747253a79b795146a84083

앱 닉네임

닉네임 추가 

패키지 이름

com.smilejsu.FirebaseDatabase

SHA 인증서 지문 ⓘ





유형 ⓘ

a4:d3:dc:54:10:92:35:37:05:5d:55:d7:91:f1:4f:87:55:a6:df:32

SHA-1

디지털 지문 추가

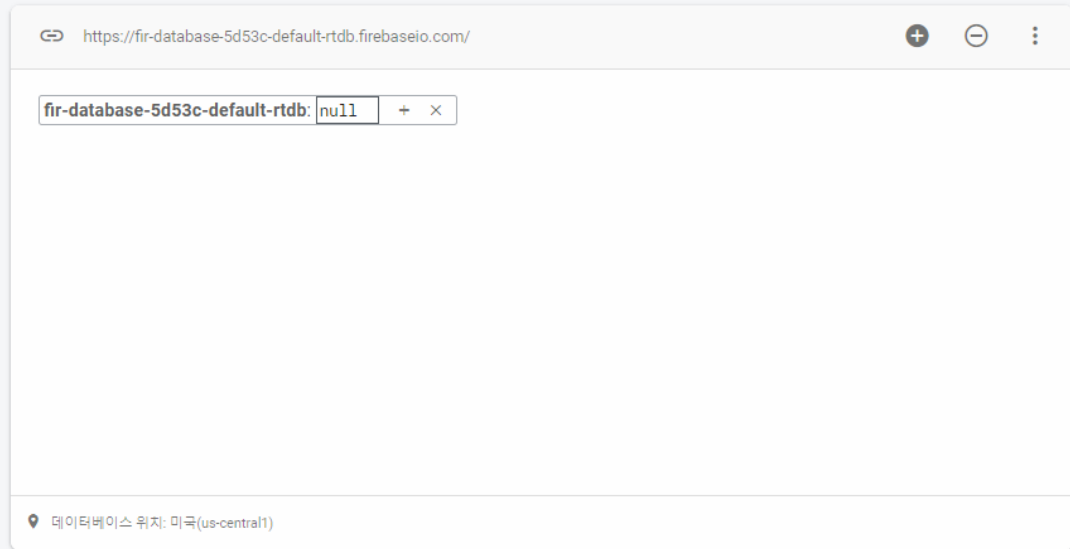
앱 삭제

 Assets  
 Scenes  
 google-services  
 Packages

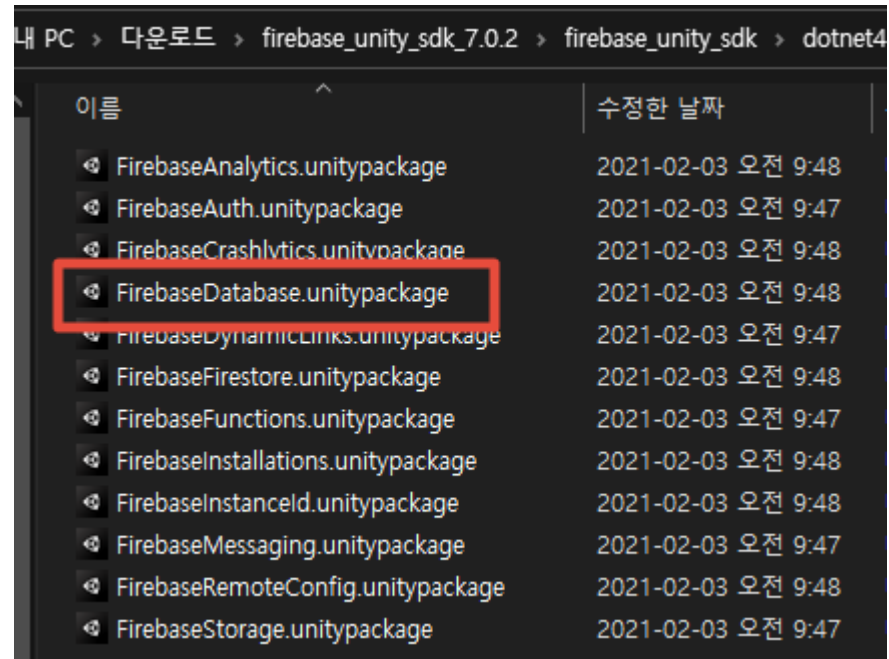
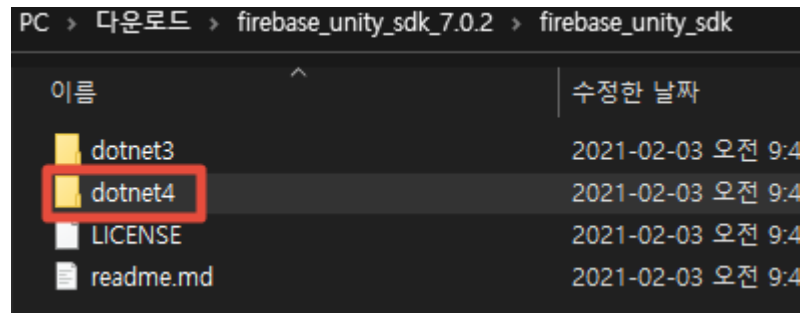
FirestoreDatabase ▾

## Realtime Database

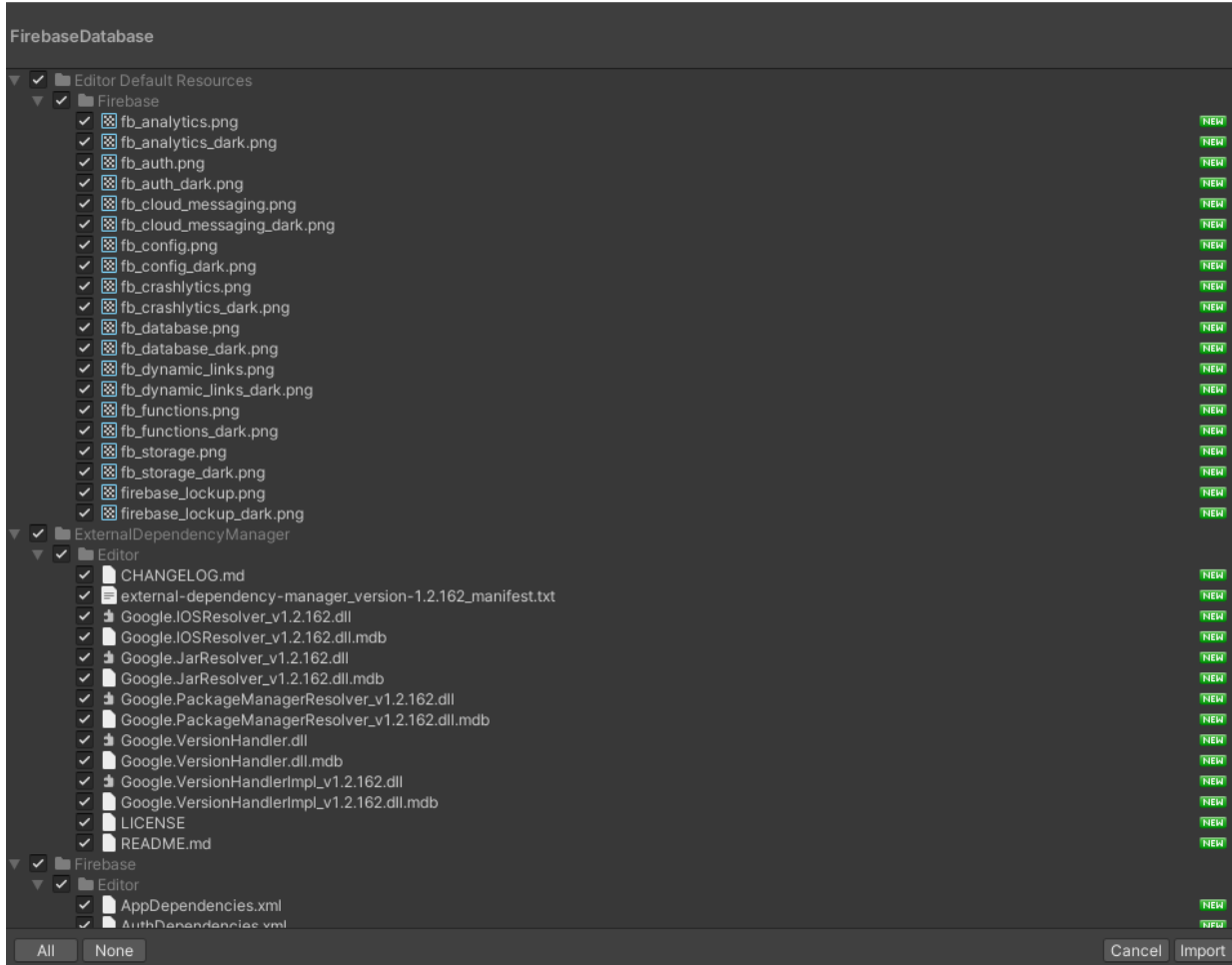
데이터 규칙 백업 사용량

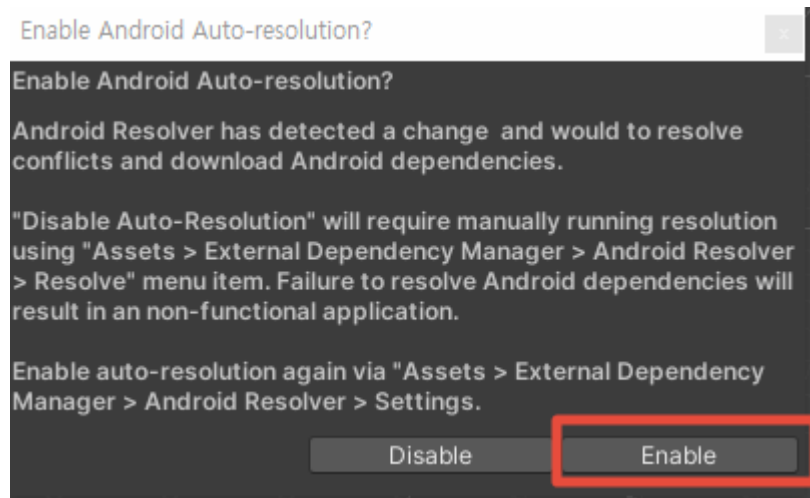


dotnet4 > FirebaseDatabase.untypackage import



# Import Unity Package





## Resolving Android Dependencies...

32%...

Executing command:

```
C:\Users\smile\workspace\unity\FirebaseDatabase\Temp\PlayServicesResolverGradle\gradlew.bat
--no-daemon -b
"C:\Users\smile\workspace\unity\FirebaseDatabase\Temp\PlayServicesResolverGradle\PlayServices
Resolver\scripts\download_artifacts.gradle"
"-PANDROID_HOME=C:/Users/smile/AppData/Local/Android/Sdk"
"-PTARGET_DIR=C:\Users\smile\workspace\unity\FirebaseDatabase\Assets\Plugins\Android"
"-PMAVEN_REPOS=file:///C:/Users/smile/workspace/unity/FirebaseDatabase/Assets/Firebase/m2repo
sitory"
"-PPACKAGES_TO_COPY=com.google.firebase:firebase-common:19.3.1;com.google.firebase:firebase-
```

Executing command:

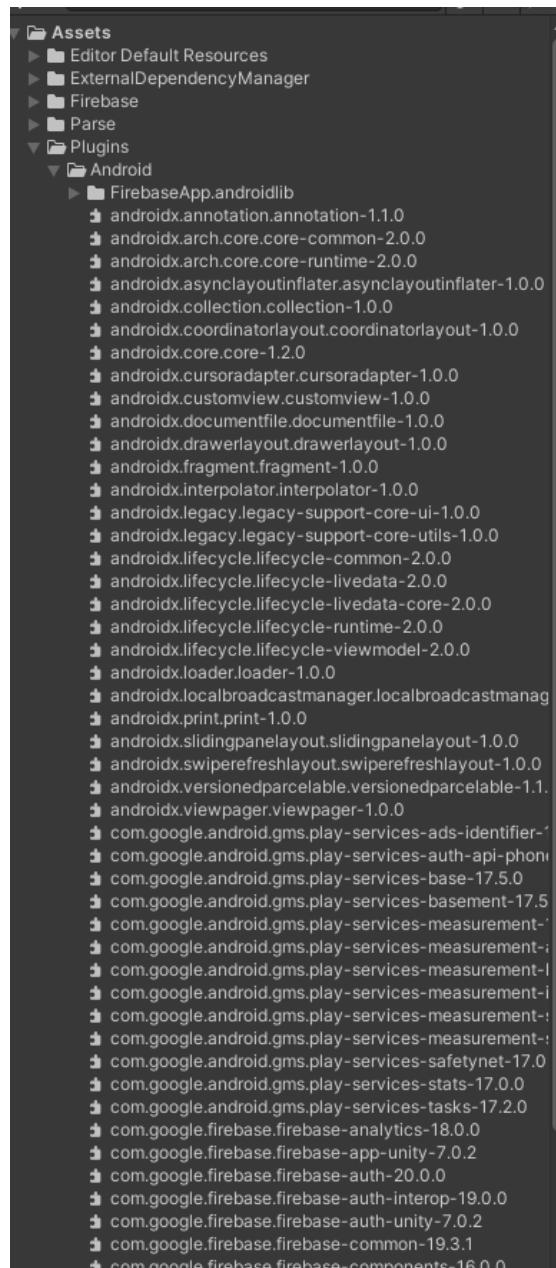
```
C:\Users\smile\workspace\unity\FirebaseDatabase\Temp\PlayServicesResolverGradle\gradlew.bat
--no-daemon -b
"C:\Users\smile\workspace\unity\FirebaseDatabase\Temp\PlayServicesResolverGradle\PlayServices
Resolver\scripts\download_artifacts.gradle"
"-PANDROID_HOME=C:/Users/smile/AppData/Local/Android/Sdk"
"-PTARGET_DIR=C:\Users\smile\workspace\unity\FirebaseDatabase\Assets\Plugins\Android"
"-PMAVEN_REPOS=file:///C:/Users/smile/workspace/unity/FirebaseDatabase/Assets/Firebase/m2repo
sitory"
"-PPACKAGES_TO_COPY=com.google.firebase:firebase-common:19.3.1;com.google.firebase:firebase-
analytics:18.0.0;com.google.android.gms:play-services-base:17.5.0;com.google.firebase:firebase-app-
unity:7.0.2;com.google.firebase:firebase-auth:20.0.0;com.google.firebase:firebase-auth-unity:7.0.2;co
m.google.firebase:firebase-database:19.5.1;com.google.firebase:firebase-database-unity:7.0.2"
```

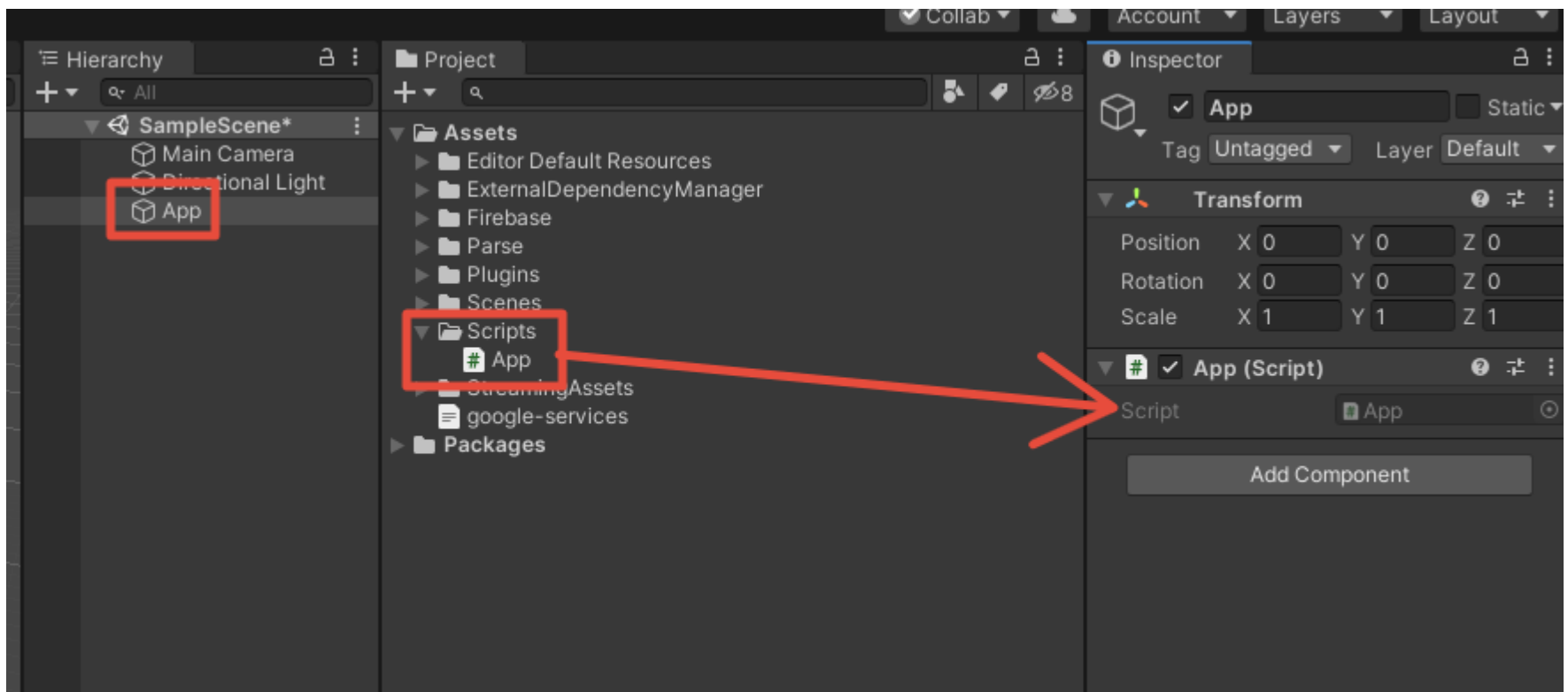
## Resolving Android Dependencies

## Resolving Android Dependencies...

30%...

```
"C:\Users\smile\workspace\unity\FirebaseDatabase\Temp\PlayServicesResolverGradle\PlayServices
Resolver\scripts\download_artifacts.gradle"
"-PANDROID_HOME=C:/Users/smile/AppData/Local/Android/Sdk"
"-PTARGET_DIR=C:\Users\smile\workspace\unity\FirebaseDatabase\Assets\Plugins\Android"
"-PMAVEN_REPOS=file:///C:/Users/smile/workspace/unity/FirebaseDatabase/Assets/Firebase/m2repo
sitory"
"-PPACKAGES_TO_COPY=com.google.firebase:firebase-common:19.3.1;com.google.firebase:firebase-
analytics:18.0.0;com.google.android.gms:play-services-base:17.5.0;com.google.firebase:firebase-app-
unity:7.0.2;com.google.firebase:firebase-auth:20.0.0;com.google.firebase:firebase-auth-unity:7.0.2;co
m.google.firebase:firebase-database:19.5.1;com.google.firebase:firebase-database-unity:7.0.2"
"-PUSE_JETIFIER=0" "-PDATA_BINDING_VERSION=3.4.0"
Executing command:
C:\Users\smile\workspace\unity\FirebaseDatabase\Temp\PlayServicesResolverGradle\gradlew.bat
--no-daemon -b
"C:\Users\smile\workspace\unity\FirebaseDatabase\Temp\PlayServicesResolverGradle\PlayServices
Resolver\scripts\download_artifacts.gradle"
"-PANDROID_HOME=C:/Users/smile/AppData/Local/Android/Sdk"
"-PTARGET_DIR=C:\Users\smile\workspace\unity\FirebaseDatabase\Assets\Plugins\Android"
"-PMAVEN_REPOS=file:///C:/Users/smile/workspace/unity/FirebaseDatabase/Assets/Firebase/m2repo
sitory"
"-PPACKAGES_TO_COPY=com.google.firebase:firebase-common:19.3.1;com.google.firebase:firebase-
analytics:18.0.0;com.google.android.gms:play-services-base:17.5.0;com.google.firebase:firebase-app-
unity:7.0.2;com.google.firebase:firebase-auth:20.0.0;com.google.firebase:firebase-auth-unity:7.0.2;co
```







|                               |   |
|-------------------------------|---|
| 매달리닉스                         | ▼ |
| 확장 프로그램 β                     | ▼ |
| 개발                            |   |
| 인증                            | ▼ |
| 실시간 데이터베이스                    | ▲ |
| 소개                            |   |
| 데이터베이스 선택                     |   |
| ▶ iOS                         |   |
| ▶ Android                     |   |
| ▶ 웹                           |   |
| ▶ 관리                          |   |
| ▶ REST                        |   |
| ▶ C++                         |   |
| ▼ Unity                       |   |
| 시작하기                          |   |
| 데이터 구조화                       |   |
| 데이터 저장                        |   |
| 데이터 검색                        |   |
| ▶ 보안 및 규칙                     |   |
| ▶ 사용량 및 성능                    |   |
| 자동 백업                         |   |
| Cloud 함수로 확장                  |   |
| 동영상 시리즈: SQL 개발자를 위한 Firebase |   |

`UpdateChildrenAsync()`    정의된 경로에서 모든 데이터를 대체하지 않고 일부 키를 업데이트합니다.

`RunTransaction()`    동시 업데이트에 의해 손상될 수 있는 복잡한 데이터를 업데이트합니다.

★ **참고:** 기본적으로 인증된 사용자만 데이터를 읽고 쓸 수 있도록 데이터베이스에 대한 읽기 및 쓰기 액세스 권한이 제한됩니다. 공개 액세스 규칙을 구성하면 인증을 설정하지 않고 시작할 수 있습니다. 다만 이렇게 하면 앱을 사용하지 않는 사람을 포함하여 모두에게 데이터베이스가 공개되므로 인증을 설정할 때 데이터베이스를 다시 제한해야 합니다.

## DatabaseReference 가져오기

데이터베이스에 데이터를 쓰려면 `DatabaseReference`의 인스턴스가 필요합니다.

```
using Firebase;
using Firebase.Database;
using Firebase.Unity.Editor;

public class MyScript: MonoBehaviour {
 void Start() {
 // Get the root reference location of the database.
 DatabaseReference reference = FirebaseDatabase.DefaultInstance.RootReference;
 }
}
```

```
App
using Firebase.Database;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

Unity 스크립트 | 참조 0개
public class App : MonoBehaviour
{
 // Start is called before the first frame update
 Unity 메시지 | 참조 0개
 void Start()
 {
 DatabaseReference reference = FirebaseDatabase.DefaultInstance.RootReference;
 Debug.Log(reference);
 }

 // Update is called once per frame
 Unity 메시지 | 참조 0개
 void Update()
 {
 }
}
```

```
Console
Clear Collapse Clear on Play Clear on Build Error Pause Editor ▼
[11:40:09] https://fir-database-5d53c-default-rtdb.firebaseio.com/
UnityEngine.Debug.Log(Object)
```

## 참조 위치에서 데이터 쓰기, 업데이트 또는 삭제

### 기본 쓰기 작업

기본 쓰기 작업은 `SetValueAsync()` 코드를 사용하여 지정된 참조에 데이터를 저장하고 해당 경로의 기존 데이터를 모두 바꿉니다. 이 메서드를 통해 사용 가능한 JSON 유형에 해당하는 다음과 같은 유형을 전달할 수 있습니다.

- `string`
- `long`
- `double`
- `bool`
- `Dictionary<string, Object>`
- `List<Object>`

형식이 지정된 C# 객체를 사용하는 경우, 기본 제공된 `JsonUtility.ToJson()` 을 사용하여 객체를 원시 JSON으로 변환하고 `SetRawJsonValueAsync()` 를 호출할 수 있습니다. 예를 들어 다음과 같은 User 클래스가 있다고 가정해 보겠습니다.

```
public class User {
 public string username;
 public string email;

 public User() {
 }

 public User(string username, string email) {
 this.username = username;
 this.email = email;
 }
}
```

다음과 같이 `SetRawJsonValueAsync()` 로 사용자를 추가할 수 있습니다.

```
private void writeNewUser(string userId, string name, string email) {
 User user = new User(name, email);
 string json = JsonUtility.ToJson(user);

 mDatabaseRef.Child("users").Child(userId).SetRawJsonValueAsync(json);
}
```



1/1



## JSON .NET For Unity

 parentElement, LLC ★★★★★ 5 | [225 Reviews](#)

**FREE**

Taxes/VAT calculated at checkout

File size 1.4 MB

Latest version 2.0.1

Latest release date Nov 9, 2017

Supported Unity versions 4.6.0 or higher

Support [Visit site](#)

**Import**

[View Full Details](#)

```
using Firebase.Database;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Newtonsoft.Json;
```

```
public class User
{
 public string username;
 public string email;

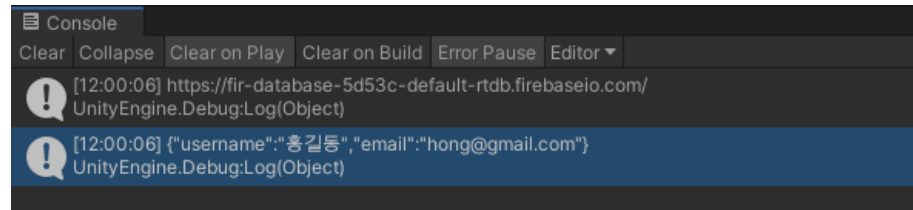
 public User()
 {
 }

 public User(string username, string email)
 {
 this.username = username;
 this.email = email;
 }
}
```

```
public class App : MonoBehaviour
{
 // Start is called before the first frame update
 void Start()
 {
 DatabaseReference reference = FirebaseDatabase.DefaultInstance.RootReference;
 Debug.Log(reference);

 string userId = System.Guid.NewGuid().ToString();
 string name = "홍길동";
 string email = "hong@gmail.com";
 this.WriteNewUser(userId, name, email);
 }

 private void WriteNewUser(string userId, string name, string email)
 {
 User user = new User(name, email);
 string json = JsonConvert.SerializeObject(user);
 Debug.Log(json);
 //mDatabaseRef.Child("users").Child(userId).SetRawJsonValueAsync(json);
 }
}
```




# Realtime Database

데이터

규칙

백업

사용량

 <https://fir-database-5d53c-default-rtdb.firebaseio.com/>

**fir-database-5d53c-default-rtdb**

 **users**

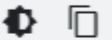
 **2d10254a-0f81-4be9-a267-78a6c29c43b5**

email: "hong@gmail.com"

username: "홍길동"

## 데이터 한 번 읽기

`GetValueAsync` 메서드를 사용하여 지정된 경로에서 콘텐츠의 정적 스냅샷을 읽을 수 있습니다. 작업 결과에 하위 데이터 등 해당 위치의 모든 데이터를 포함하는 스냅샷이 포함됩니다. 데이터가 없을 때 반환되는 스냅샷은 `null` 입니다.



```
FirebaseDatabase.DefaultInstance
 .GetReference("Leaders")
 .GetValueAsync().ContinueWith(task => {
 if (task.IsFaulted) {
 // Handle the error...
 }
 else if (task.IsCompleted) {
 DataSnapshot snapshot = task.Result;
 // Do something with snapshot...
 }
 });
```



참조 1개

```
public void GetValueAsync()
{
 FirebaseDatabase.DefaultInstance.GetReference("users").GetValueAsync().ContinueWith(task =>
 {
 Debug.Log(task.Status);

 if (task.IsFaulted)
 {
 // Handle the error...
 }
 else if (task.IsCompleted)
 {
 DataSnapshot snapshot = task.Result;
 Debug.LogFormat("{0}, {1}", snapshot.Key, snapshot.Value);
 // Do something with snapshot...
 }
 });
}
```

<https://fir-database-5d53c-default-rtdb.firebaseio.com/>

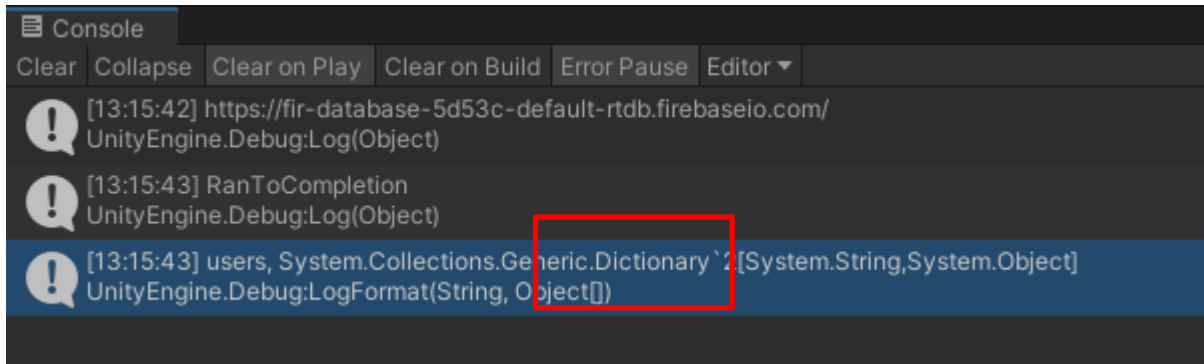
fir-database-5d53c-default-rtdb

users

2d10254a-0f81-4be9-a267-78a6c29c43b5

email: "hong@gmail.com"

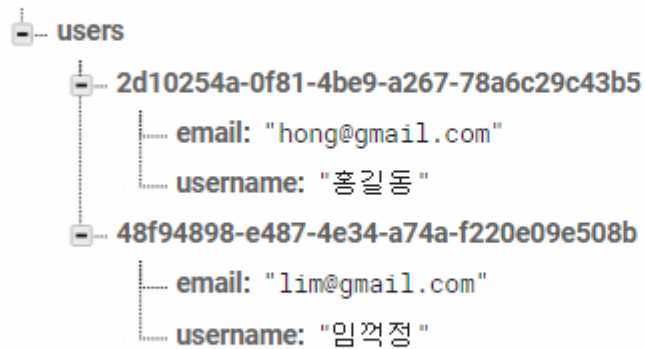
username: "홍길동"



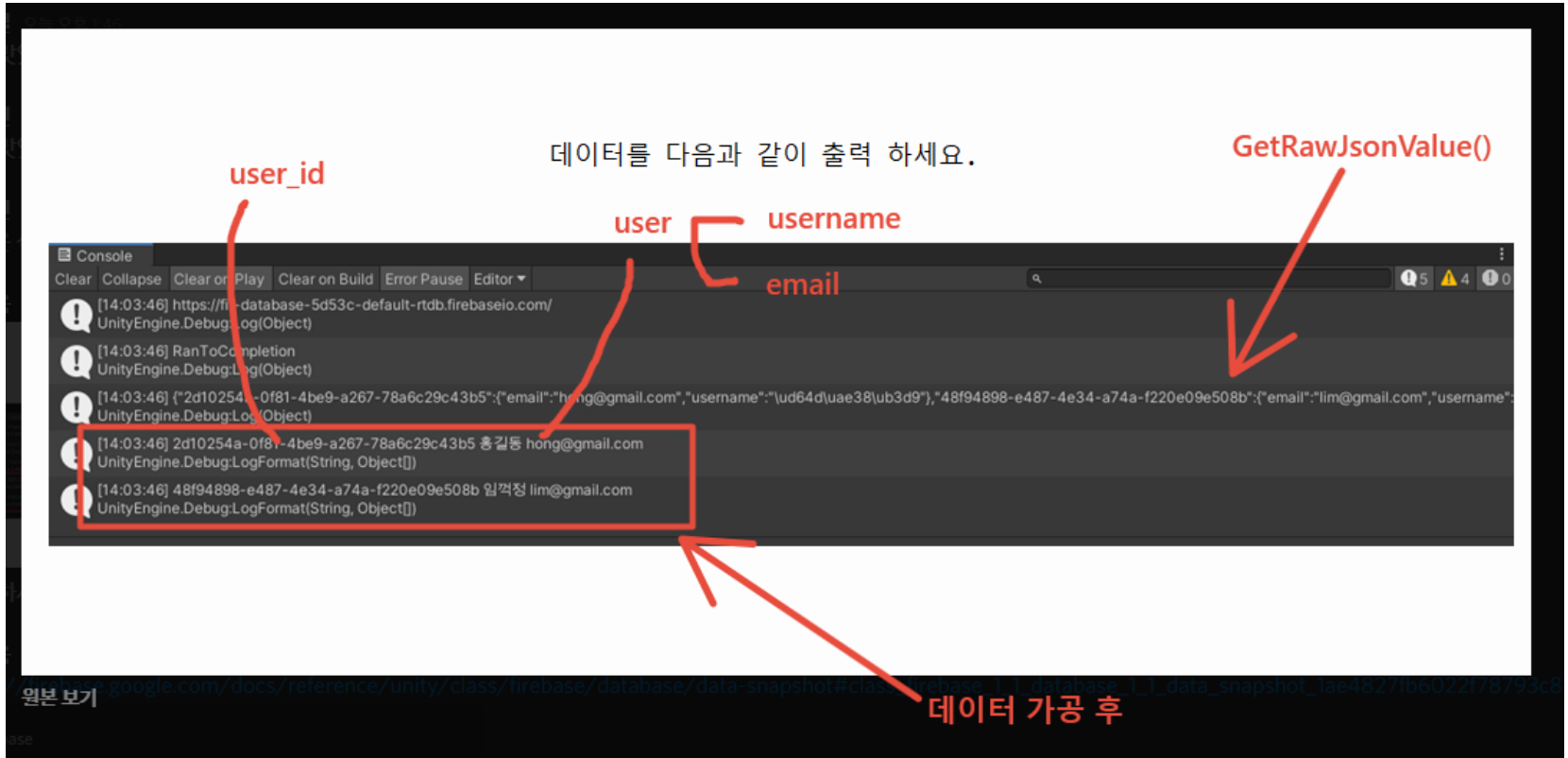
```
string userId = System.Guid.NewGuid().ToString();
string name = "임걱정";
string email = "lim@gmail.com";
this.WriteNewUser(userId, name, email);
```

<https://fir-database-5d53c-default-rtdb.firebaseio.com/>

#### **fir-database-5d53c-default-rtdb**



데이터를 다음과 같이 출력 하세요.



```

public void GetValueAsync()
{
 FirebaseDatabase.DefaultInstance.GetReference("users").GetValueAsync().ContinueWith(task =>
 {
 Debug.Log(task.Status);

 if (task.IsFaulted)
 {
 // Handle the error...
 }
 else if (task.IsCompleted)
 {
 DataSnapshot snapshot = task.Result;
 // Do something with snapshot...

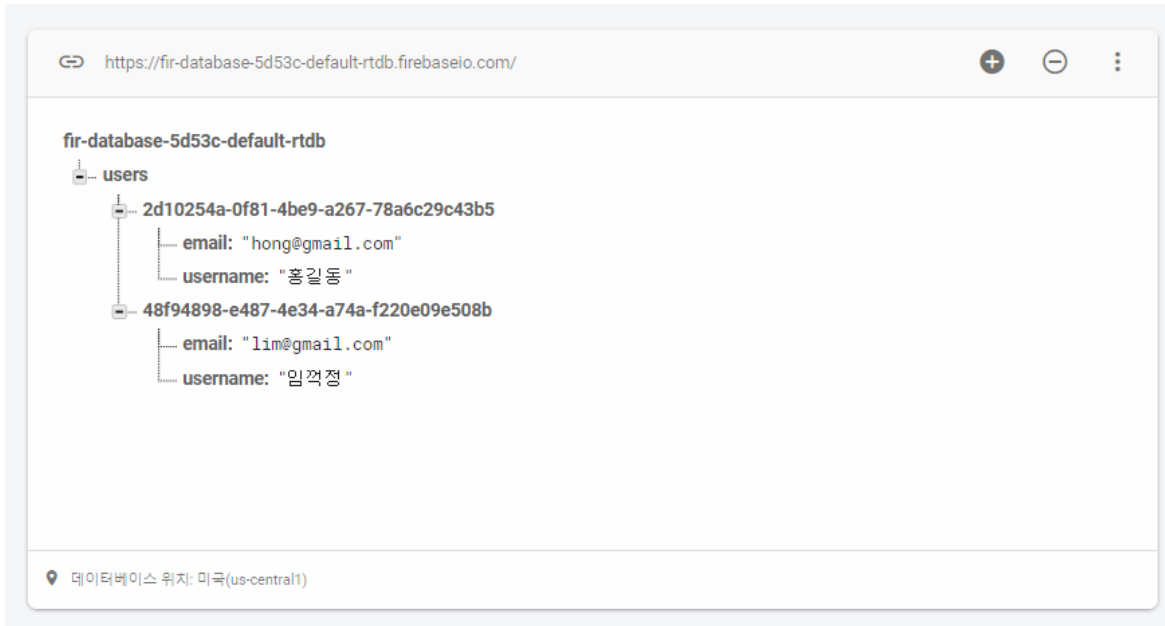
 var json = snapshot.GetRawJsonValue();
 Debug.Log(json);

 var children = snapshot.Children;
 var enumerator = children.GetEnumerator();
 while (enumerator.MoveNext()) {
 var user = JsonConvert.DeserializeObject<User>(enumerator.Current.GetRawJsonValue());
 var userId = enumerator.Current.Key;
 Debug.LogFormat("{0} {1} {2}", userId, user.username, user.email);
 }
 }
 });
}

```

# 챌린지

데이터를 다음과 리스트로 보여 주세요



<https://stackoverflow.com/questions/65224002/for-loop-stops-when-calling-function-in-firebase-for-unity-c-sharp>