

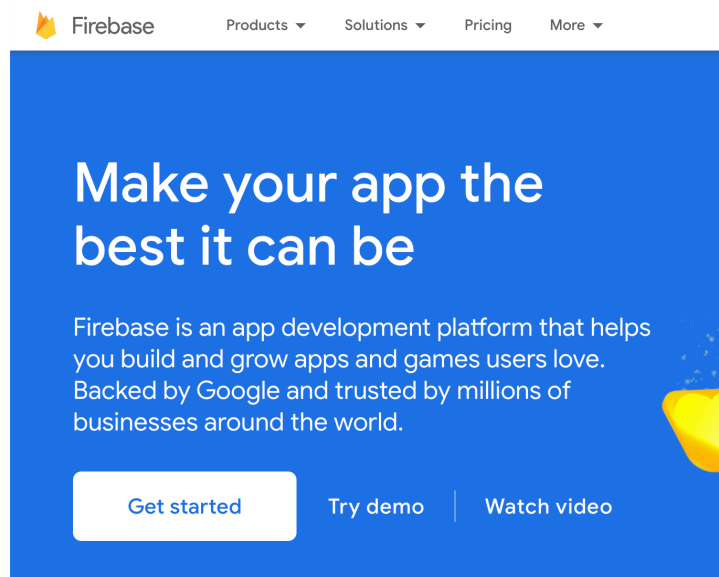
# Read/Write iOS app data with firebase database

## Store app data in firebase realtime database (json, nosql)

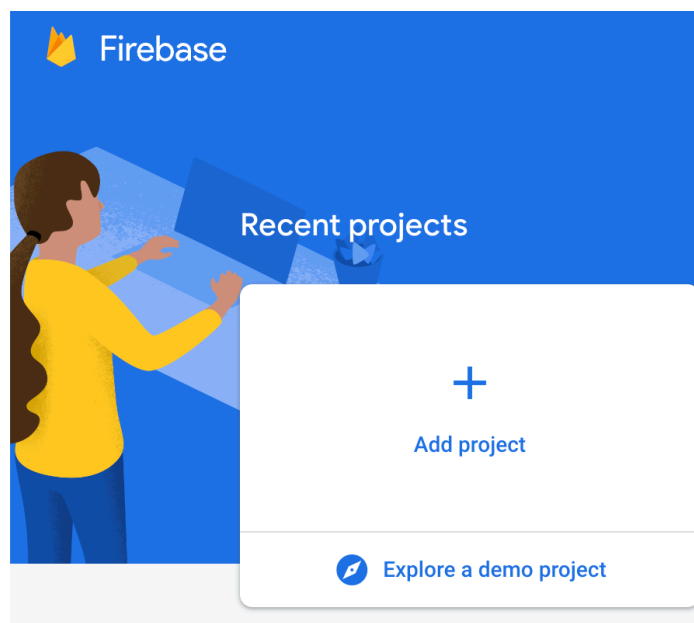
---

### Steps:

1. Create a new Xcode project or using existing project.
2. Create firebase account.



3. Add a new project




## 4. Follow the steps

× Create a project (Step 1 of 3)

Let's start with a name for your project <sup>?</sup>

Project name

 ios-attr


Continue


× Create a project (Step 2 of 3)


Google Analytics  
for your Firebase project


Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, and Cloud Functions.


Google Analytics enables:

 A/B testing <sup>?</sup>

 Crash-free users <sup>?</sup>

 User segmentation & targeting across <sup>?</sup>  
Firebase products

 Event-based Cloud Functions triggers <sup>?</sup>

 Free unlimited reporting <sup>?</sup>


☒ Enable Google Analytics for this project  
Recommended


Previous Continue


× Create a project (Step 3 of 3)

## Configure Google Analytics

Choose or create a Google Analytics account ?

 ios

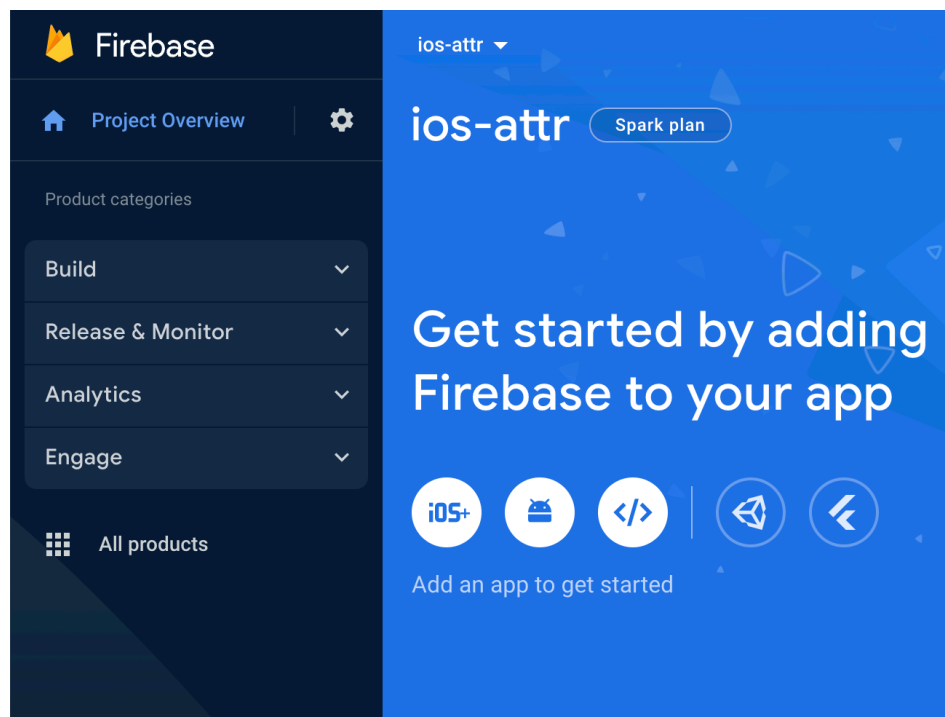
Automatically create a new property in this account 

Upon project creation, a new Google Analytics property will be created in your chosen Google Analytics account and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the [Firebase terms of service](#), while Firebase data imported into Google Analytics is subject to the [Google Analytics terms of service](#). [Learn more](#) .

[Previous](#)[Create project](#)

5. Open your new Firebase project

6. Tap on iOS button



7. In Apple bundle ID: enter Bundle Identifier from your Xcode project

× **Add Firebase to your Apple app**

1

Register app

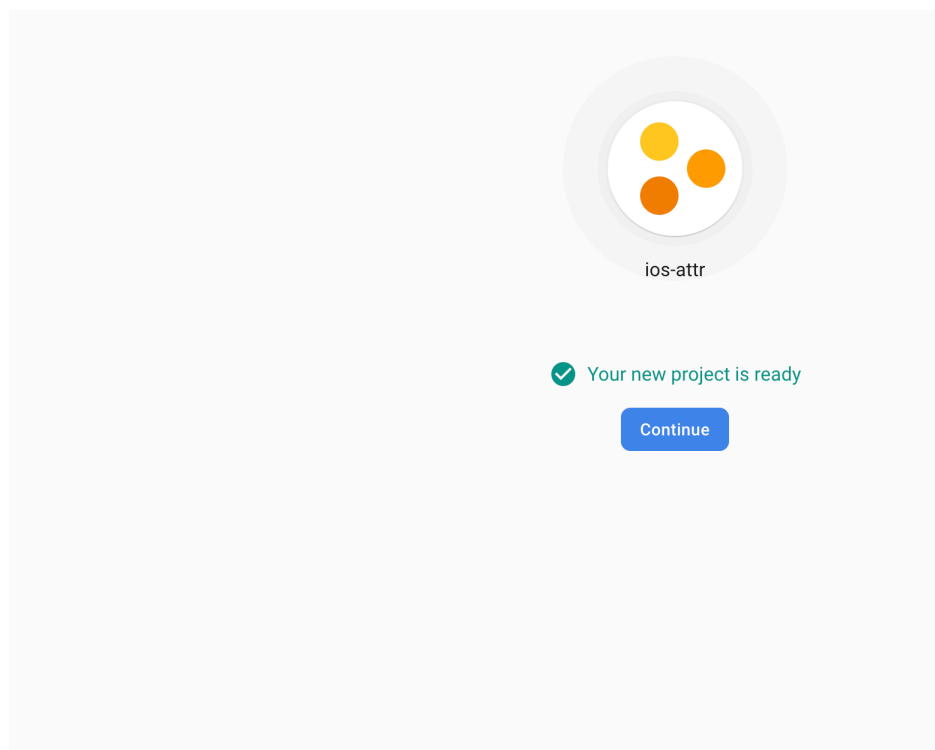
Apple bundle ID ⓘ

App nickname (optional) ⓘ

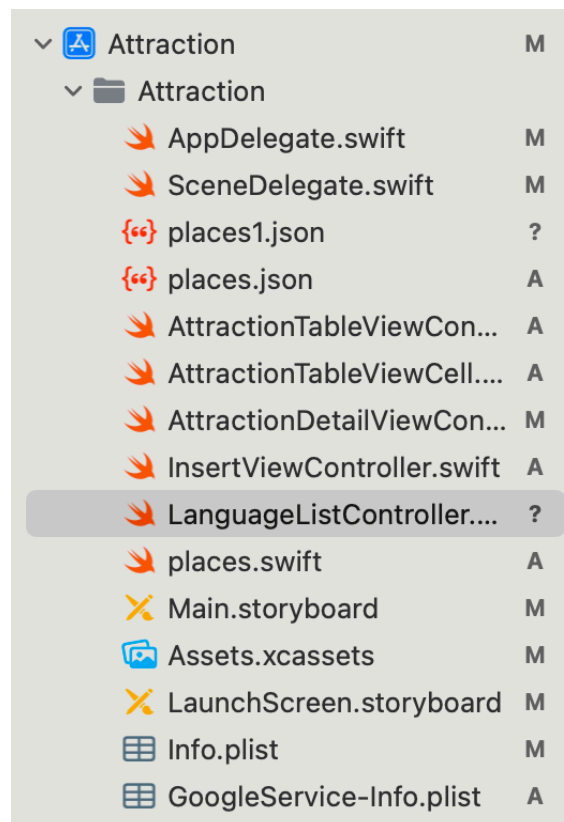
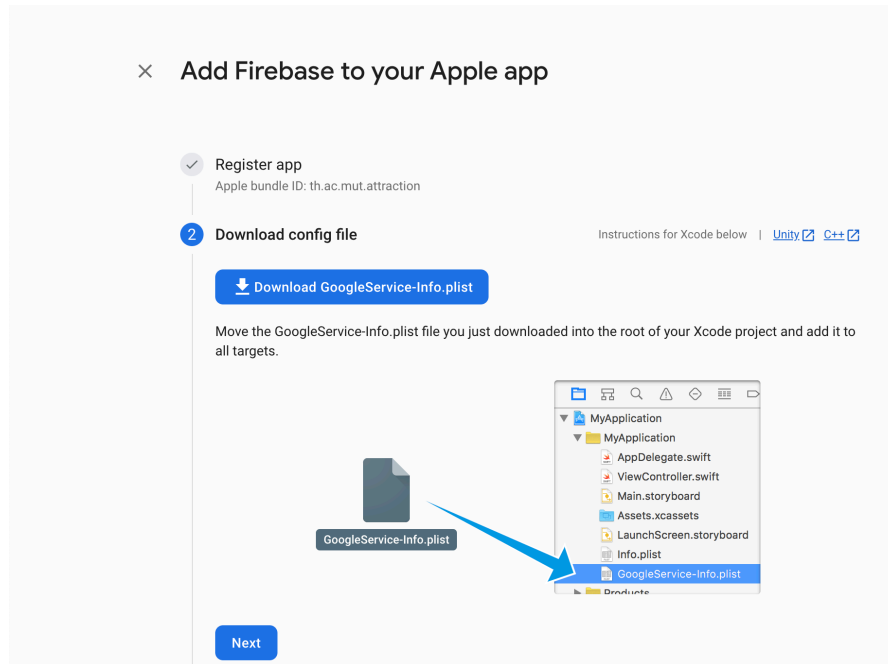
App Store ID (optional) ⓘ

Register app

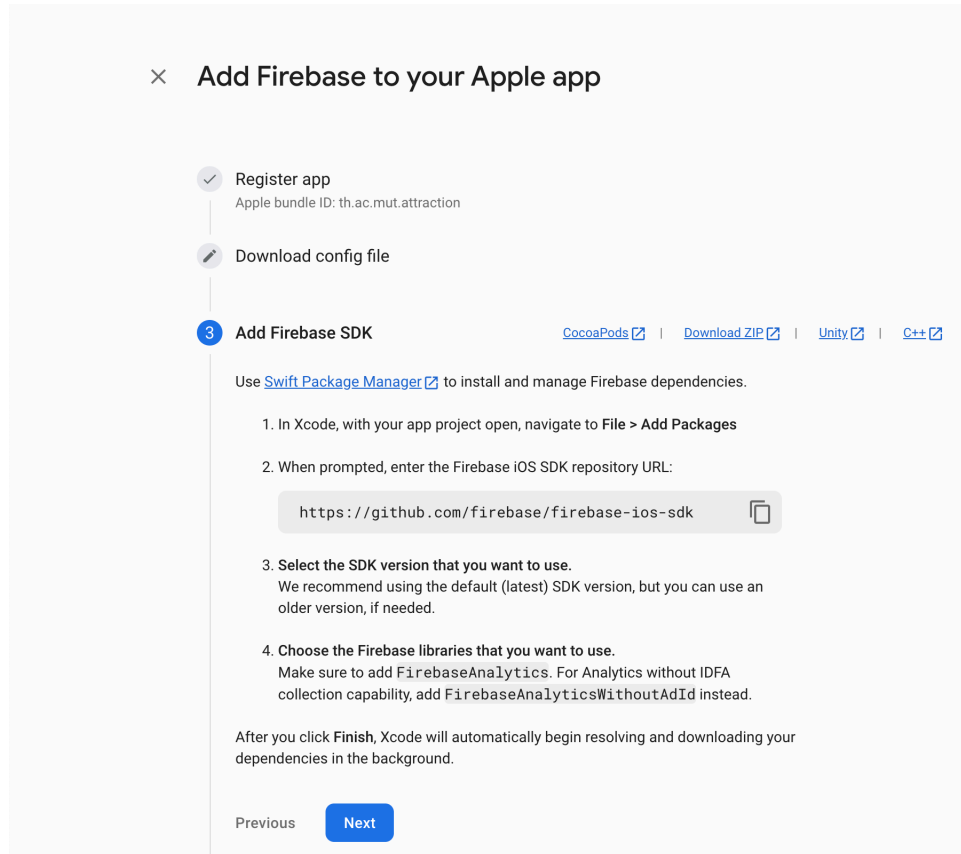
8. Press Register app



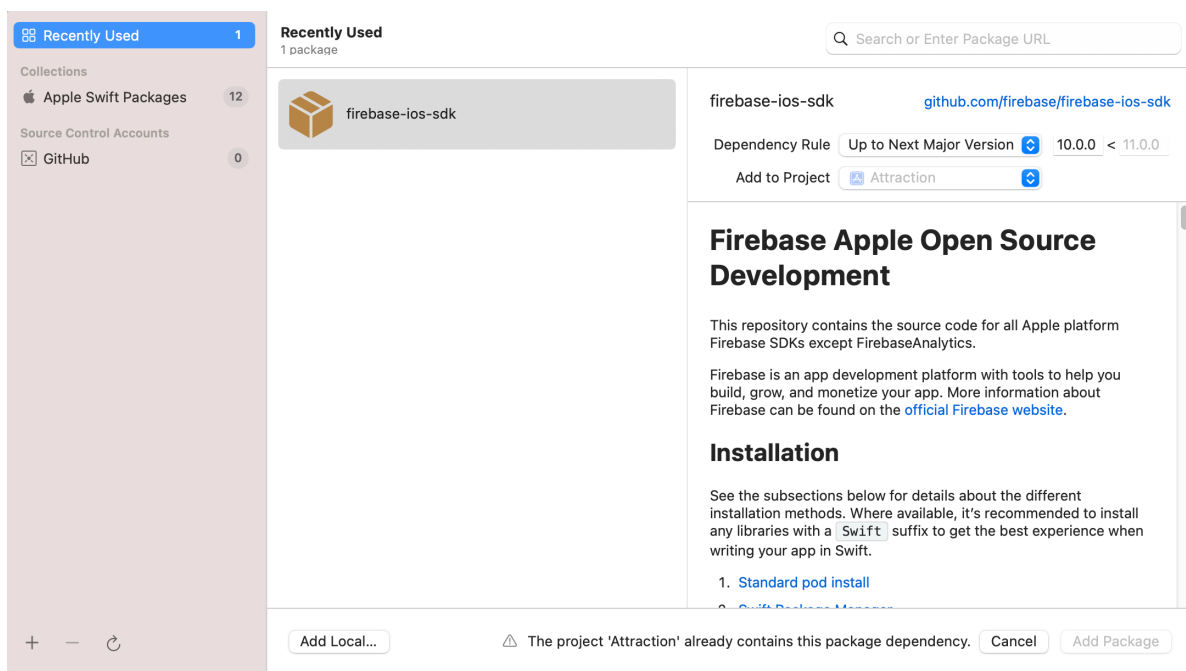
9. Download GoogleService-Info.plist file and paste it into your Xcode project's root folder. Make sure that the name of the download file is exactly GoogleService-Info.plist



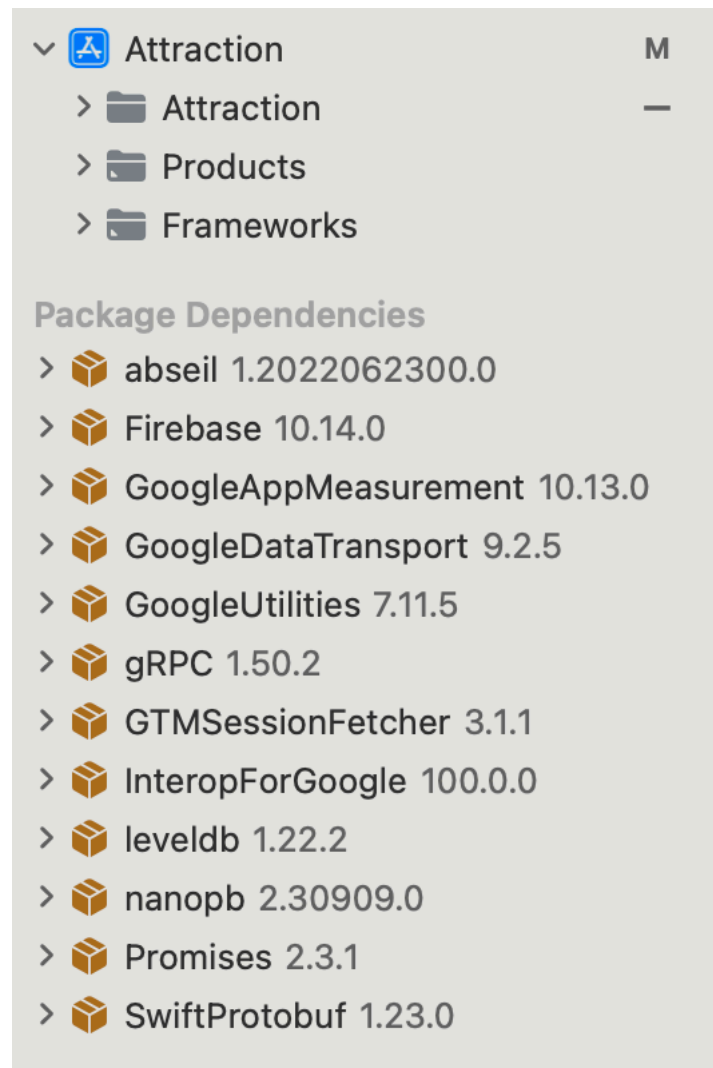
## 10. Add firebase-ios-sdk in Xcode menu File—&gt; Add Packages...



## 11. When prompt, select Firebase Database module



12. Xcode will download all firebase library and add to project.



13. In Xcode, Open AppDelegate.swift and add Firebase to your project:

```
import Firebase
```

14. In AppDelegate.swift, Add the following line in didFinishLaunchingWithOptions method (before return true):

FirebaseApp.configure()

```

8  import UIKit
9  import Firebase
10 @main
11 class AppDelegate: UIResponder, UIApplicationDelegate {
12
13
14
15     func application(_ application: UIApplication, didFinishLaunchingWithOptions
        launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
16         // Override point for customization after application launch.
17         FirebaseApp.configure()
18         return true
19     }
20 }

```

15. Go to the ViewController.swift file and import firebase.

import Firebase

16. Add the following variable in class ViewController:

var ref: DatabaseReference!

17. In method viewDidLoad() create the database reference:

ref = Database.database().reference(withPath: "users")

18. Write data (using dictionary key:value) into database.

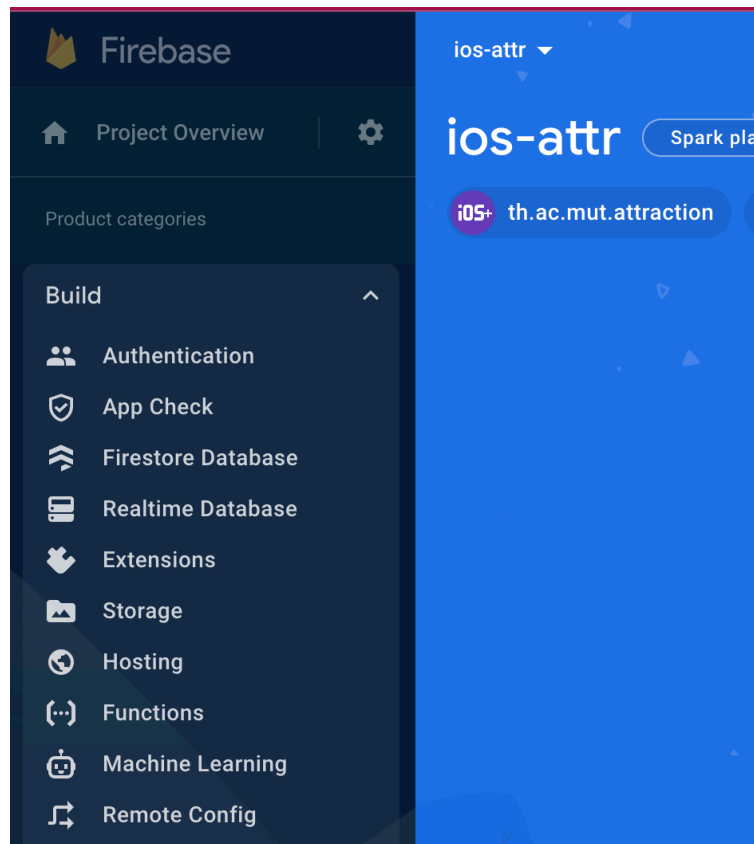
```

/
8  import UIKit
9  import Firebase
10 class ViewController: UIViewController {
11
12     var ref: DatabaseReference!
13     override func viewDidLoad() {
14         super.viewDidLoad()
15         // Do any additional setup after loading the view.
16         ref = Database.database().reference(withPath: "users")
17         self.ref.child("user_id").setValue(123456)
18         self.ref.child("name").setValue("m munlin")
19         self.ref.child("user").setValue("mmunlin")
20         self.ref.child("pass").setValue("pass")

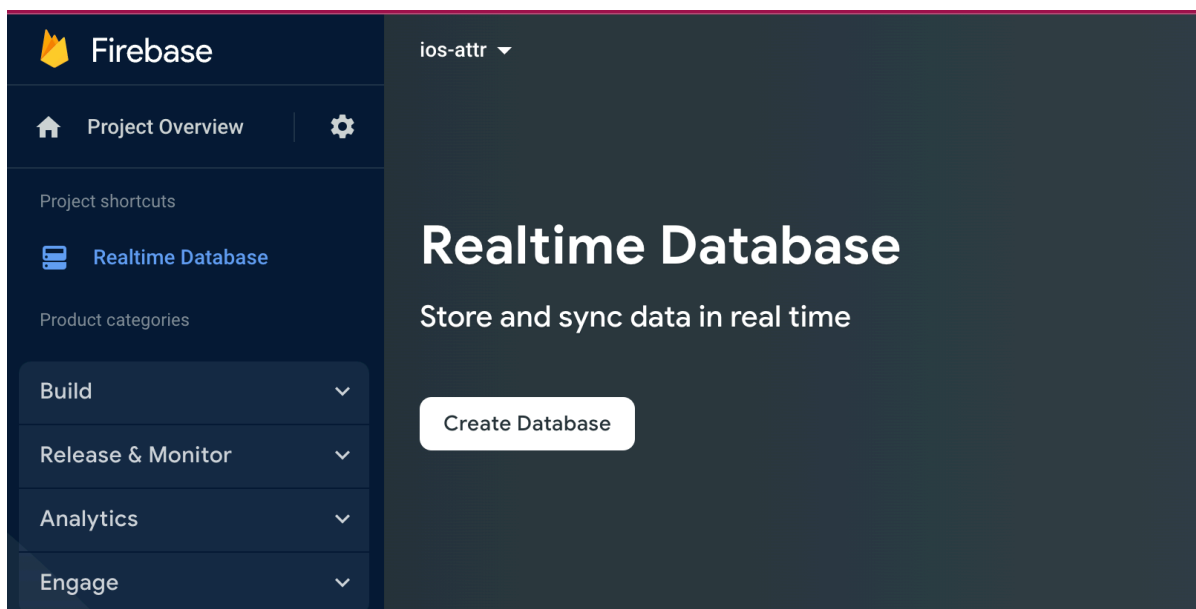
```



19. In Firebase, go to the firebase console, select project and select Realtime Database section in the left column under button Build.



20. Clcik Create Database



### Set up database

1 Database options

2 Security rules

Your location setting is where your Realtime Database data will be stored.

Realtime Database location

United States (us-central1)

CancelNext

21. Select Start in test mode and click Next.

### Set up database

Database options

2 Security rules

Once you have defined your data structure **you will have to write rules to secure your data.**  
[Learn more](#)

☐ **Start in locked mode**  
Your data is private by default. Client read/write access will only be granted as specified by your security rules.

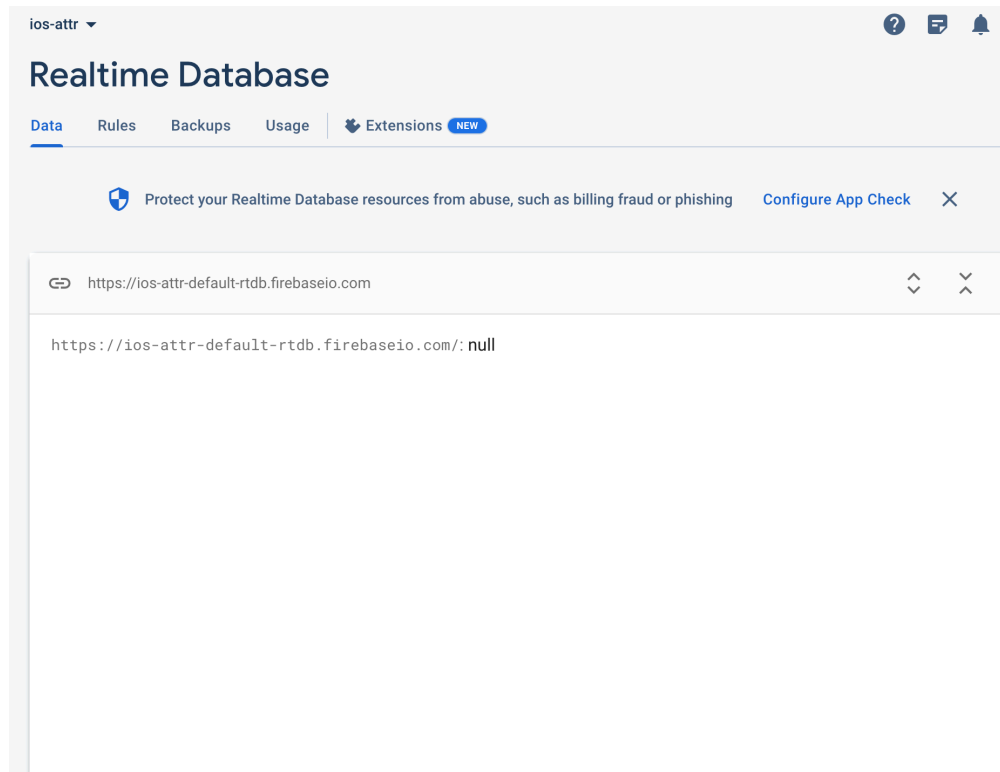
☒ **Start in test mode**  
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
{
  "rules": {
    ".read": "now < 1697130000000", // 2023-10-13
    ".write": "now < 1697130000000", // 2023-10-13
  }
}
```

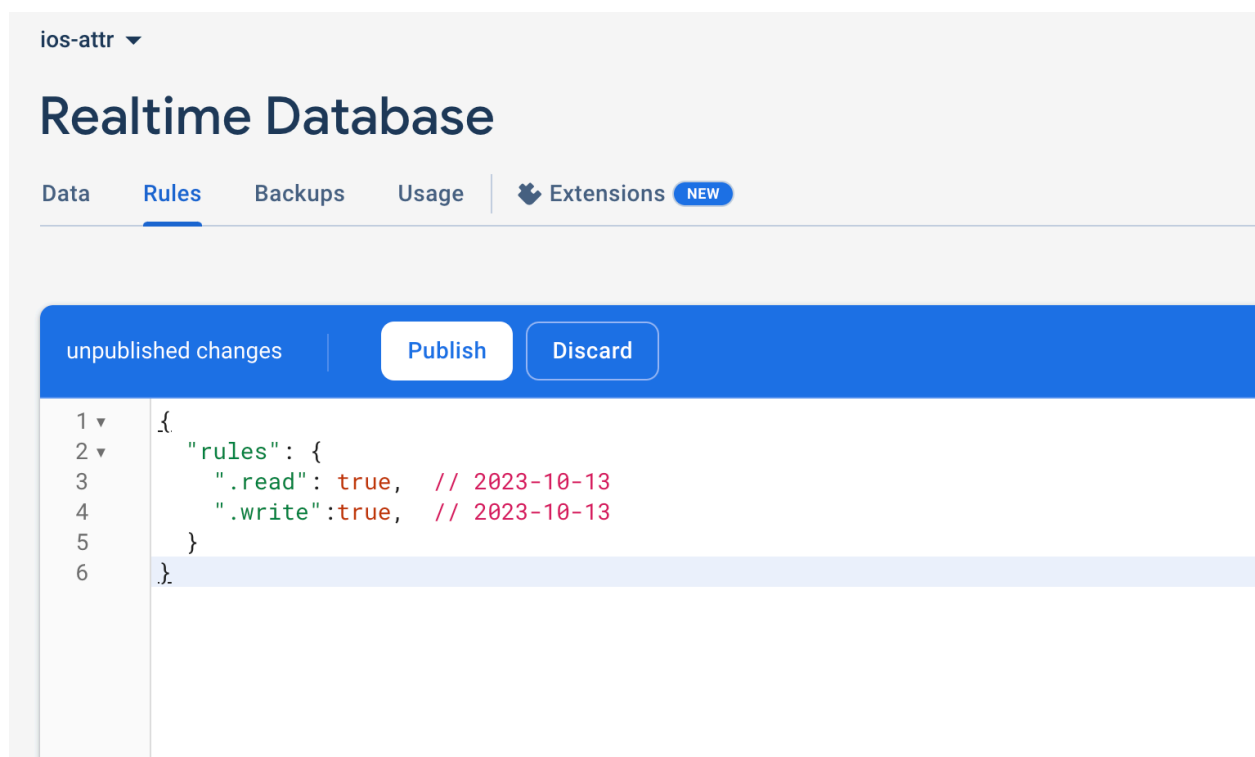
**The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days**

CancelEnable

## 22. Click Enable



## 23. Click Rules and set the read/write to true.



24. Click Publish.

25. In Xcode, run the app (commnad-R). See the rersult in firebase console.

```

/
8 import UIKit
9 import Firebase
10 class ViewController: UIViewController {
11
12     var ref: DatabaseReference!
13     override func viewDidLoad() {
14         super.viewDidLoad()
15         // Do any additional setup after loading the view.
16         ref = Database.database().reference(withPath: "users")
17         self.ref.child("user_id").setValue(123456)
18         self.ref.child("name").setValue("m munlin")
19         self.ref.child("user").setValue("mmunlin")
20         self.ref.child("pass").setValue("pass")

```

## Realtime Database

[Data](#) [Rules](#) [Backups](#) [Usage](#) [Extensions](#) [NEW](#)

<https://fb-ex-e9219-default-rtdb.firebaseio.com>

<https://fb-ex-e9219-default-rtdb.firebaseio.com/>

▼ — users

```

├── name: "m munlin"
├── pass: "pass"
├── user: "mmunlin"
└── user_id: 123456

```

26. Add the followin code in viewDidLoad( ) after the last line.

```
// Retrieving Data
ref.observe(.value, with: { snapshot in
    print("snapshot")
    print((snapshot.value! as Any))
})
```

27. run the app (commnad-R). See the rersult in Xcode console.

```
snapshot
{
    name = "m munlin";
    pass = pass;
    user = mmunlin;
    userid = 123456;
}
```

28. Add the followin code in viewDidLoad( ) after the last line to retrieve a single value. See the rersult in Xcode console.

```
self.ref.child("user_id").observeSingleEvent(of: .value, with: {
    (snapshot) in
    if let id = snapshot.value as? Int {
        print("The value from the database: \(id)")
    }
})
```

**The value from the database: 123456**

29. See the following reference to read and write objects into firebase database as well as perform user authentication.

<https://www.kodeco.com/21133526-firebase-tutorial-getting-started?page=1#toc-anchor-001>