# 第八周 Firebase RealTime Database

Danny Shen

# Firebase 的資料庫

Firebase 有兩種資料庫，一種叫 Realtime Database，一種是 Cloud Firestore。技術上這是兩個不太相同的資料庫 Realtime Database 是較早期開發的，效率好，簡單，方便。Cloud Firestore 是較後期開發的，本來是為了加強改進 Realtime Database 的缺點，但結構上較為大而複雜，相對的擴展與延伸性都比較好。這兩種資料庫雖然很多特性類同，但使用的方法，資料內容都無法互通，未來也會同時存在兩種資料庫，不是取代。

我們要教的是 RealTime Database

# 資料結構

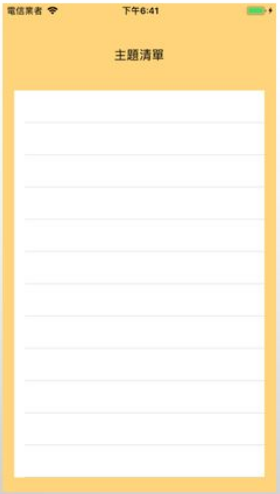
第一個重要的概念，Firebase 的資料是沒有欄位設定的，而是一個類似 JSON 樹的資料庫。傳統資料庫是一個表單，每個欄位有自己的 schema 設定，不能任意修改或新增刪除，但 Realtime Database 是以鍵值為中心，所以每一個資料都是任意格式的

物件Object是由一個或多個名稱與值與組成，放在一組大括號中，名稱與值中間放冒號，每一組之間用逗號區隔，也就是【{key:value,key:value}】名稱通常是字串，值就可能是字串，數值，佈林，或另一個物件或陣列

# 設計目標

即時匿名討論區 APP


登入頁面，若完成輸入暱稱，且為連線狀態，就按圖示進入討論清單




討論清單列表，可按列表進入討論，或上一頁修改暱稱


討論內容頁，，或上一頁修改暱稱

# 資料庫啟用與設定

進入即時資料庫設定頁面，有四個主要的選項，分別是【資料】,【規則】,【備份】,【用量】四個選項。

【資料】就是實際的資料庫內容，你可以在這兒看看是否資料與正確寫入修改，若有一些測試資料，也可在這兒手動加入，匯出備份或重新匯入資料等。在資料修改的同時，這個畫面也會動態的更新資料，並以不同的顏色來表示新增修改刪除移動等狀態

【規則】是資料庫的安全設定，以特定的 JSON 語法來控制誰可以寫入，本書後面會深入討論這個問題，預設值是只能由已驗證的使用者才可以存取。

【備份】是自動化的備份和安全性規則，目前只支援付費使用者使用，而且需要設定一些備份的流程，本書暫時不會討論這個設定。

【用量】則是看一下在一定期限內，有多少人上線，存取量多少，使用多少空問和負戴尖峯最高值等數字，用來評估自己的應用是否該升級或分流時的參考。

# 安全性規則設定

// 任何有登入的使用者都可以讀寫資料
```
{
  "rules": {
    ".read": "auth != null",
    ".write": "auth != null"
  }
}
```

//開放為唯讀資料庫
```
{
  "rules": {
    ".read": true,
    ".write": false
  }
}
```

//全開放不限制資料庫
```
{
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

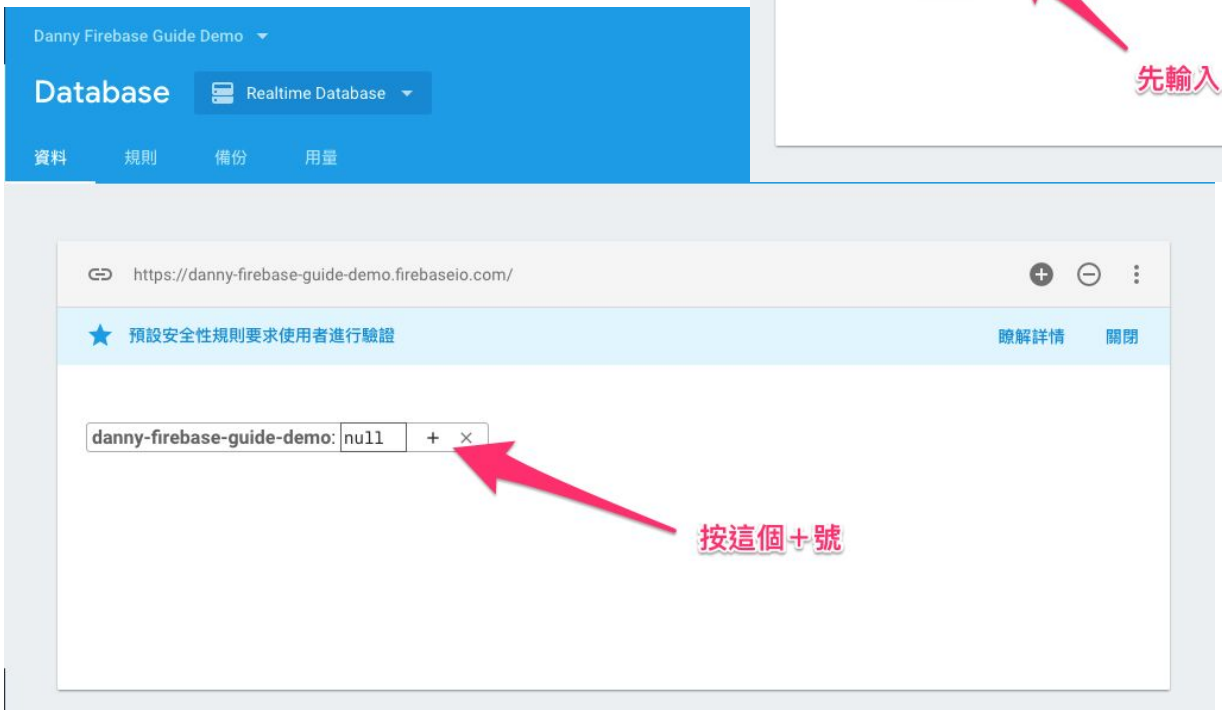
//個人資料個人擔
```
{
  "rules": {
    "users": {
      "$uid": {
        ".read": "$uid === auth.uid",
        ".write": "$uid === auth.uid"
      }
    }
  }
}
```

//開放特定資料匣給登入的使用者
```
{
  "rules": {
    "users": {
          ".read": "auth != null",
          ".write": "auth != null"
    }

  }
}
```

# 手動設定第一筆資料


https://danny-firebase-guide-demo.firebaseio.com/
預設安全性規則要求使用者進行驗證
瞭解詳情
關閉
danny-firebase-guide-demo: null
名稱 appdefault
值 值
取消
新增
先輸入 appdefault
再按下【+】


Danny Firebase Guide Demo
Database
Realtime Database
資料
規則
備份
用量
https://danny-firebase-guide-demo.firebaseio.com/
預設安全性規則要求使用者進行驗證
瞭解詳情
關閉
danny-firebase-guide-demo: null
按這個＋號

# 資料備份的匯入與匯出

你可在測試階段，使用 Web 匯入或匯出你的資料。

Realtime Database 不是一個設計成大量不斷累積的資料庫，而是短暫快速反應，有效利用的的資料庫

單一資料庫的物理限制是有的，雖然可能看起來很高，但一開始良好的規劃，可幫你的 APP 永續經營.

一般來說，在使用者不多的情況下，每個資料庫免費同時連線 100人，其他各項限制:

https://firebase.google.com/docs/database/usage/limits

# 寫入資料庫

1. 先取得資料庫實體
2. 再取得位置參考節點
3. 對節點進行寫入，必要時加入節點 (注意權限)
4. 可寫入字串, 數字, 與字串/數字的集合資料
5. 由後台檢查寫入的結果(觀查陣列或字典寫入的結果)
6. 可寫入字串，數字，陣列，字典等，看看會有什麼不同的寫入結果

```
var ref: DatabaseReference!
ref = Database.database().reference()
ref.child("[節點]").setValue([變數名稱])
```

# 讀取一次資料

1. 同寫入資料
2. 使用 .getData 也可使用 .observeSingleEventOfType 差別在取資料的方式內容不同
3. 非同步 CallBack 回傳 Error 與 DataSnapshot
4. 解析 DataSnapshot 的值，並轉回合適的資料格式

```
ref.child("users").child(userID!).observeSingleEvent(of: .value, with: { snapshot in
  let value = snapshot.value as? NSDictionary
  let username = value?["username"] as? String ?? ""
  let user = User(username: username)
}) { error in
  print(error.localizedDescription)}
```

# 伺服器時間參照

寫入資料時，有一種特別的資料，就是寫入當下的伺服器時間。在寫入資料時，我們經常需要資料寫入的順序，最常見的就是寫入的時間先後序，而這個資料並不適合由 iOS / Android 取得，因為手機的時間，多少會有一點誤差，而且是可以自己調整的，所以Firebase 在寫資料時，有一個專用的常數，在寫入資料庫後，會轉成伺服器當下的時間。要做這件事的方法很簡單，只要把常數物件寫在 .setValue 的參數就可以了。當然，要放在合適的節點

.setValue(ServerValue.timestamp())

寫入的格是會是 64 bit 的整數，也就是 Java的 long 與 Swift 的 Int，自1970年1月1日起的 ms (千份之一秒)，由於只有一種格式所以使用時必需自行轉換，Android 可以直接讀取後轉成時間，iOS 要換成除以1000 的 Double 再轉換成時間的格式。

# 建立唯一鍵值

雖然 Firebase 即時資料庫是一個 JSON Tree，但許多時候，我們有類似SQL的記錄一樣，有寫一筆記錄的需求。比如說，以之前面提到的案例，一個討論區下面可能有許多筆留言，每筆留言都有，內容，時間，留言者等資料。這種情形就需要建立唯一鍵值了當作 key 了。

這種時後，我們就需要在資料庫參考後面，加上自動ID 再相關的資料存到資料庫。要使用這個功能，iOS 方法叫做 .childByAutoID()

ref.childByAutoId().setValue(member)

# 持續觀察

如同一次性讀取，我們可以用 .observe 方法來持續觀察一個節點內資料的新增/變化/刪除等，在每一次有變化時，就 CallBack 一次。可用於如討論區或即時通等資料的更新。

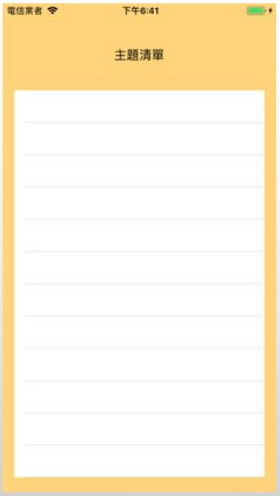
但每一個 observe 在未中斷之前，都算一個持續使用的使用者，節點內的資料，也會依妳的 observe 一直更新，所以資源使用不小，若不能好好設計，會使費用變的非常巨大

至少，在離開頁面時，使用 .removeAllObervers() 或 .removeObserver(withHandle:) 解除設定，不能看了十個討論區，就不停的接受這十個討論區的資料，只接受目前的討論區即可。

# 再看設計目標

即時匿名討論區 APP


登入頁面，若完成輸入暱稱，且為連線狀態，就按圖示進入討論清單




討論清單列表，可按列表進入討論，或上一頁修改暱稱


討論內容頁，，或上一頁修改暱稱

# 登入頁

1. 實作匿名登入
2. 回傳網路與登入狀態
3. 加入導欄頁
4. 檢查使用者的暱稱格式與預存

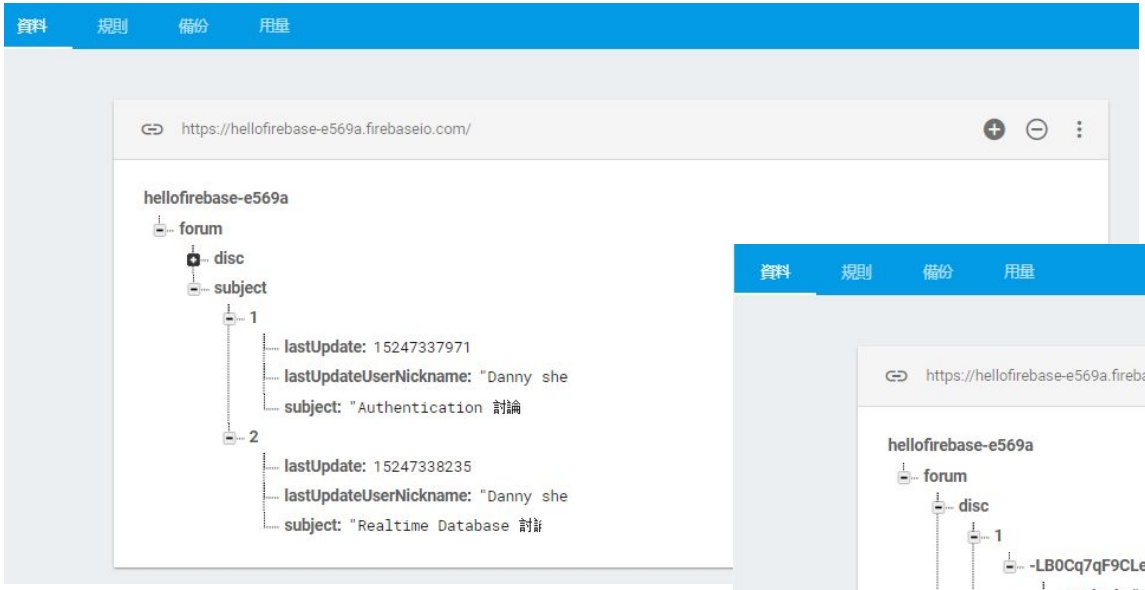
# 討論區列表頁

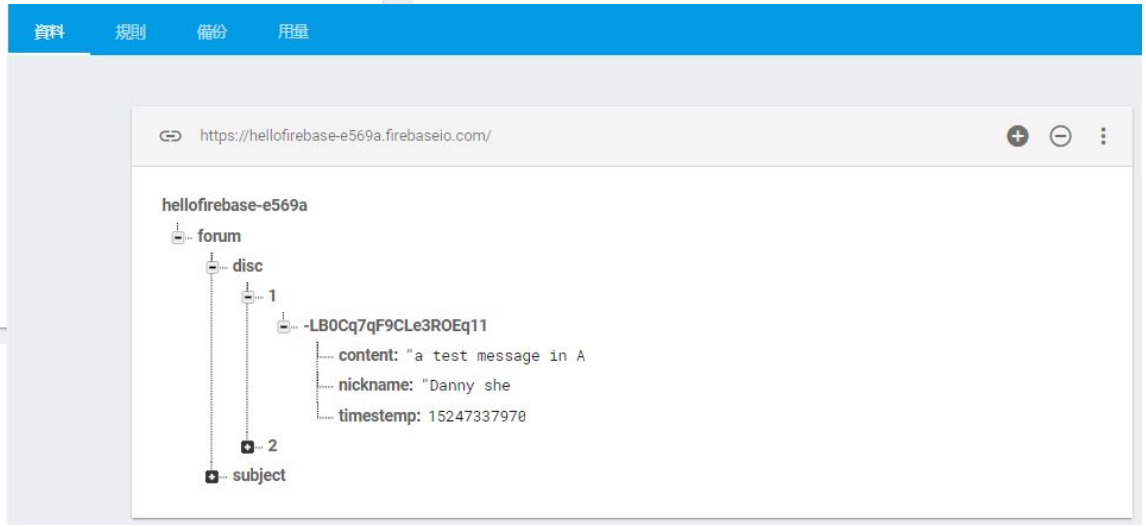
一次性讀取資料庫

使用 Table View 顯示在畫面

必要時更新時間記錄(最後更新時間是選擇性的)

# 資料結構設計


資料
規則
備份
用量
https://hellofirebase-e569a.firebaseio.com/
hellofirebase-e569a
forum
disc
subject
1
lastUpdate: 15247337971
lastUpdateUserNickname: "Danny she
subject: "Authentication 討論
2
lastUpdate: 15247338235
lastUpdateUserNickname: "Danny she
subject: "Realtime Database 討論


資料
規則
備份
用量
https://hellofirebase-e569a.firebaseio.com/
hellofirebase-e569a
forum
disc
1
-LB0Cq7qF9CLe3ROEq11
content: "a test message in A
nickname: "Danny she
timestemp: 15247337970
2
subject

# 不良設計範例

若討論內容與主題，放在同一節點時，由於我們讀取資料時，需以節點為存取點，會讀取並不一定需要的資料

雖然資料量不足以影響程式運作，但會造成不明的流量損失，成本上揚，類似的設計累計起來，就是 APP 順與不順的差別

# 討論內容頁頁

可新增留言

記錄留言者與留言時間

必要時排序

思考輸入的 UI, 重新設計

# 完善討論區 APP

記錄討論區留言時間

按讚功能

刪除留言

只保留最後十則留言