

KAIST Spring 2018

CS374: Intro to HCI

<https://kixlab.org/courses/hci/>

Tutorial #3: Working with Data

2018.04.24

Hyeungshik Jung

Working with Data?

- Why?
 - Our application should be available on any client
 - We have to store 'state' of our web application
- Topics we will cover today
 - Firebase Database
 - How to store data in Firebase Database
 - How to retrieve data from Firebase and update UI

bit.ly/cs374-tut3

Learning Objective

- We will build a comment board

name

comment

name

comment

Hyeungshik: Hello CS374

Firebase



- Firebase is a set of services for building web / mobile applications easily
- Authentication, ***Database***, Storage, Functions, Crash Analytics, User Analysis, etc
- Why Firebase?
 - Easy to start

Getting started with Firebase



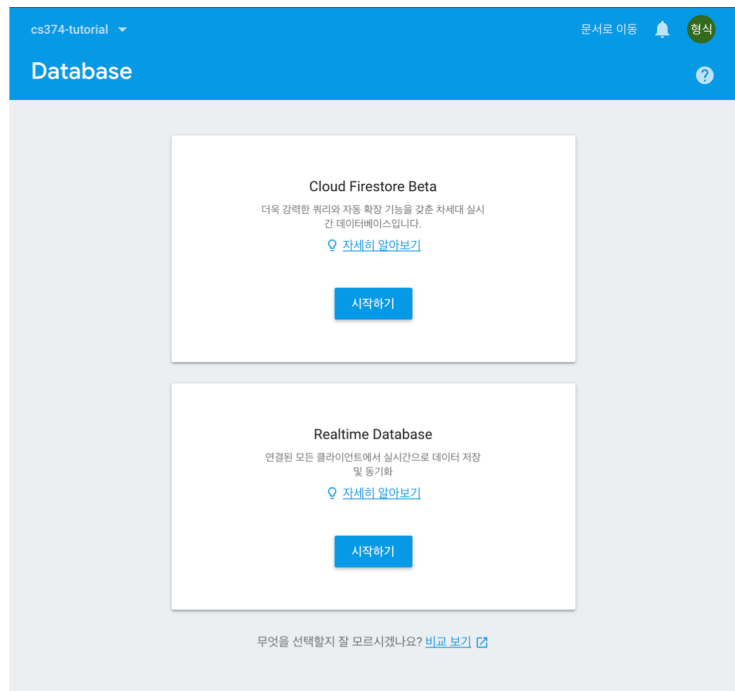
- Prepare a Google account
- Go to <http://firebase.google.com>
- Start new project

The screenshot shows the Firebase console interface. At the top, there's a header with the Firebase logo and user information. Below the header, a welcome message reads "Firebase에 오신 것을 환영합니다." (Welcome to Firebase). A main section titled "최근 프로젝트" (Recent Projects) displays a grid of project cards, including "dynamic-slide", "cs374-2018-pr3", and "dynamic-player". A modal dialog box titled "프로젝트 추가" (Add Project) is open in the foreground. It contains the following fields: "프로젝트 이름" (Project Name) with a dropdown menu set to "cs374-tutorial"; "프로젝트 ID" (Project ID) with the value "cs374-tutorial-52f70"; and "국가/지역" (Country/Region) with a dropdown menu set to "미국" (USA). There are "취소" (Cancel) and "프로젝트 만들기" (Create Project) buttons at the bottom of the dialog. A footer at the bottom of the page reads "약관 · 개인정보처리방침" (Terms · Privacy Policy).

Firestore Database



- Two types of database (Cloud Firestore vs Realtime Database)
- We will use Realtime Database



Firestore Database - Permission



- For this tutorial and homework, let's use the test mode

실시간 데이터베이스 보안 규칙

데이터 구조를 정의한 후 규칙을 작성해 데이터를 보호해야 합니다.
[자세히 알아보기](#)

잠금 모드로 시작
모든 읽기 및 쓰기를 거부하여 데이터베이스를 비공개로 설정하세요.

테스트 모드로 시작
데이터베이스에 대한 모든 읽기 및 쓰기를 허용하여 빠르게 설정하세요.

```
{  
  "rules": {  
    ".read": true,  
    ".write": true  
  }  
}
```

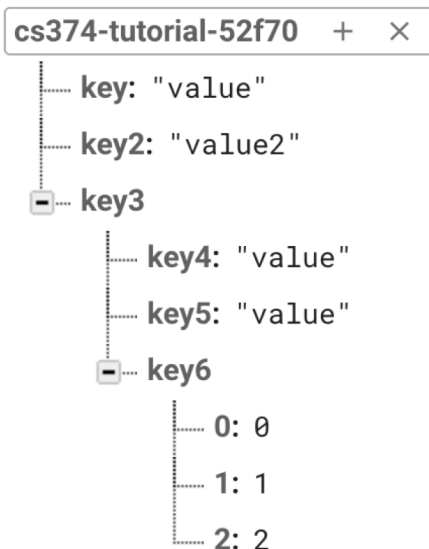
! 데이터베이스 참조를 사용하는 누구나 데이터베이스를 읽고 쓸 수 있습니다.

취소 사용 설정

Firestore Database - Structure



- Data should be stored in Key / Value structure
- Define your application state into Key / Value



Storing Data of Comment Board



name

comment

Hyeungshik: Hello CS374

Hyungyu: I miss you



Firestore Database - Realtime



- When the data change
 - Firestore emits event noticing that data changed
- We will use this property to keep our UI up to date.

Data Flow



name

comment



Data Flow



name

comment

Hyeungshik: Hello CS374

Fetch ←



Data Flow



name

comment

Hyeungshik: Hello CS374

Fetch ←



Data Flow



name

comment

Hyeungshik: Hello CS374

Fetch ←

Update →



Data Flow



name

comment

Hyeungshik: Hello CS374

Fetch ←

Update →

Fetch ←



Data Flow

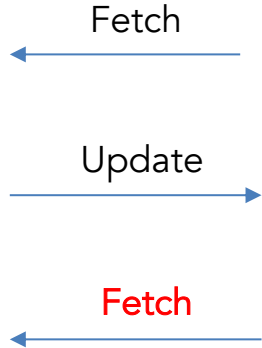


name

comment

Hyeungshik: Hello CS374

Hyungyu: I miss you



TODO1. Bind UI to data



name

comment

Hyeungshik: Hello CS374

Hyungyu: I miss you

← BIND

Update in data
> Update in UI



TODO2. Update FB database



name

comment

Hyeungshik: Hello CS374

Update →



Setting Up Firebase

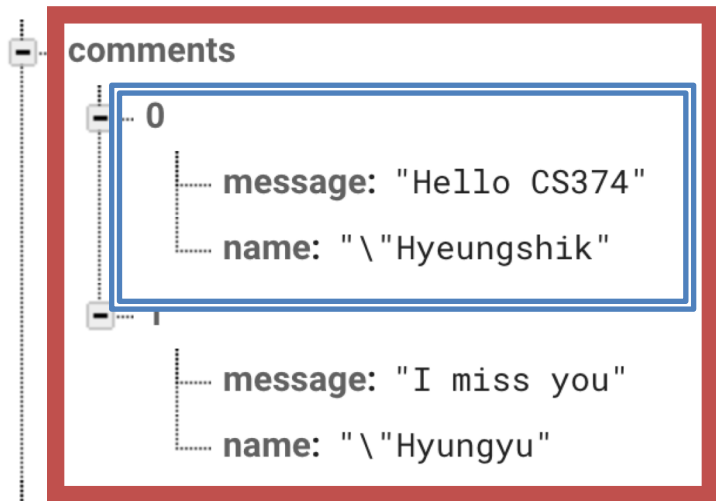
- Import Firebase JS SDK in codepen.io
 - `<script src="https://www.gstatic.com/firebasejs/4.0.0/firebase.js" ></script>`
 - jQuery
- <https://firebase.google.com/docs/web/setup>
 - Web API Key
 - Database URL

Writing HTML

Accessing Firebase Database

- <https://firebase.google.com/docs/database/web/start>
- We need 'ref' <- Reference to subset of whole database

cs374-tutorial-52f70



```
firebase.database().ref(comments)
```

```
firebase.database().ref(comments/0)
```

Subscribing to Ref

- `Ref.on('value', callback)`
 - Call callback when ref is updated
 - We will update comments everytime ref is updated
- Called once when ref created

cs374-tutorial-52f70 update



render

Hyeungshik: Hello CS374

Hyungyu: I miss you

Updating UI with Data

```
- Object {  
  -LApxA9uXxyeE10SCfo2: Object {  
    message: "Hello CS374",  
    name: "Hyeungshik"  
  },  
  -LApXHktPuX2fcffYUqW: Object {  
    message: "I miss you",  
    name: "Hyungyu"  
  }  
}
```



```
[Object {  
  message: "Hello CS374",  
  name: "Hyeungshik"  
}, Object {  
  message: "I miss you",  
  name: "Hyungyu"  
}]
```



```
["  
  <div>  
    <b>Hyeungshik</b> : <i>Hello CS374</i>  
  </div>  
  , "  
  <div>  
    <b>Hyungyu</b> : <i>I miss you</i>  
  </div>  
"]
```

Push data

- `Ref.push(value)`
 - Add new entry with new key and value
 - 'value' event in `Ref.on('value', callback)` will fire and 'callback' will be called
 - Our UI will be rendered again with new data

What we learned

- How to start Firebase project and database
- How to listen to firebase database and bind UI
- How to add item to firebase