

자기주도 PJT

Firebase 로 key-value 기반 DB 사용해보기

목차

1. 과제 개요.....	3
2. 과제 목표.....	4
3. 필수 지식 학습.....	5
4. 기능 명세.....	6
5. 산출물 제출.....	11

1. 과제 개요

본 과제는 구글의 클라우드 자동화 환경 Firebase 를 이용하여 클라우드 인프라 환경의 경험을 해보고 더 나아가 Key-value 기반의 DB 를 사용 해봄으로써 2 학기 과제를 수행하는데 어려움을 개선해주는 목적을 가지고 있다. 1 학기 교육 과정에서 이미 학습한 Vue.js 를 토대로 key-value 기반의 Firestore DB 를 구성하고, 데이터를 주고받는 과정을 학습함으로써 모든 프로젝트의 기본을 다지는데 도움이 될 것 입니다.

2 학기 프로젝트에는 회원기반의 프로젝트로 지금까지의 대부분의 프로젝트가 구성되어 있습니다. 이에 본 과제에서는 인프라 작업과 백엔드 작업을 줄여줄 수 있는 과정을 수행해보고, 구글의 최신 클라우드인 Firebase 를 사용하는 것이 실제 2 학기의 과정에도 포함되어 있습니다. DB 기능을 사용해보고 Firebase 와의 프로세스과정을 이해해야 하며, Mysql 같은 RDB 와의 차이점을 이해할 수 있어야 합니다.

2. 과제 목표

1) Firebase Vue(Javascript) 프로젝트 생성

Android , iOS , 웹, Unity, C++ 중 웹프로젝트로 진행 및 기본적인 세팅을 학습

2) Firebase Firestore DB insert,delete,select 실행

데이터를 insert 해보고 select,delete 해보면서 단순한 기능들을 익혀본다

3. 필수 지식 학습

구글의 Firebase 를 학습하기 위해서는 클라우드 개념과 인프라, 백엔드의 개념을 알고 있어야 기본적인 기능을 사용할 수 있습니다. Firestore 의 경우 key-value 기반의 구조를 이해하고 있어야 하며 collection, document 개념을 우선 이해하여야 합니다.

API 서버의 이해와 구현을 위한 참고 자료

구분	제목	링크
이해	Firebase	https://firebase.google.com/?authuser=0
이해	Firestore	https://firebase.google.com/products/firestore?authuser=0
이해	Functions	https://firebase.google.com/products/functions?authuser=0
이해	Login	https://firebase.google.com/products/auth?authuser=0
이해	Storage	https://firebase.google.com/products/storage?authuser=0
이해	hosting	https://firebase.google.com/products/hosting?authuser=0

4. 기능 명세

1. 기능/과제 목록

Req.	Category
1	Firebase 프로젝트 생성 및 환경설정 세팅
2	Firebase Firestore 설정 및 insert, select, delete 수행

2. 기능/과제 상세

Req. 1. Firebase 프로젝트 생성 및 환경설정 세팅

Req. 1-1	프로젝트를 생성하고 config 코드를 프로젝트내에 입력하여 연동시킨다
기능 상세	<p>Firebase cli , npm 등 필요한 라이브러리를 설치한다 Firebase 앱에서 웹을 생성한다 생성된 앱에 대한 정보를 보고 config 및 설정코드들을 학습한다 프로젝트에 코드를 넣어보고 연동이 되는지 확인한다</p> <p>Main.js 에 다음과 같은 본인의 설정 코드를 추가한다 (콘솔창에서 직접 확인할 수 있다)</p> <pre>import firebase from 'firebase' var firebaseConfig = { apiKey: "AIzaSyBjoGqgAS08Exxxxxxxxxxxxxxxx", authDomain: "ssafy-be63a.firebaseio.com", projectId: "ssafy-be63a", storageBucket: "ssafy-be63a.appspot.com", messagingSenderId: "197568576359", appId: "1:197568576359:web:599ef9xxxxxc2", measurementId: "G-3RC3JL9R9S" }; firebase.initializeApp(firebaseConfig)</pre>

Req. 2. Firebase Firestore 설정 및 DB query 실행

Req. 2-1	Firestore 를 설정하고 select,insert,delete 를 실행시켜본다
기능 상세	<p>Firestore 를 생성한다.</p> <pre>const db = firebase.firestore()</pre>

	<p>DB 상의 경로를 최대한 단순하게 생각하고 진행한다.</p> <p>Collection - doc - collection - doc - 순으로 계속 구조를 생각해본다</p> <pre>await db.collection("test").doc("test2").collection("test3")</pre> <p>insert 는 다음과 같은 방식으로 진행한다</p> <pre>await db.collection("health_chat_users").add({ "ssafy" : "test 입니다", }).then(()=>{ console.log("ssafy test is done") return something })</pre> <p>Select 는 다음과 같은 방식으로 진행한다.</p> <pre>await db.collection("health_chat_users").where("UID", "==", UID).get().then(async (querySnapshot) => { print(querySnapshot) }) .catch((error)=>{ return error });</pre> <p>Delete 는 다음과 같은 방식으로 진행한다</p> <pre>await deleteDoc(doc(db, "cities", "DC"));</pre>

5. 산출물 제출

산출물 제출	<p>생성한 프로젝트를 제출한다.</p> <p>산출물 제출 가이드 : https://lab.ssafy.com/s10-study/self-project/</p> <p>주의사항 : 프로젝트를 git 에서 올려야 node_modules 제외</p>
--------	---