

# 구글 Firebase 활용한 Javascript CRUD 프로그램

자바스크립트로 웹 프로그래밍 하기

2021-11-24

# Firebase 버전 확인

Firebase 에서 nodejs 10이상 버전을 사용하면 신용카드 인증을 해야 한다.

그래서 하위 버전의 Nodejs와 firebase 버전을 사용하도록 한다.

설치가 잘 되지 않으면 CMD를 관리자 권한으로 진행 보시길 바랍니다.

nvm에서 node.js 버전 10이상 버전은 미리 지워 주세요.

firebase-tools 는 서버관련 도구이고

firebase 모듈은 파이어베이스 api를 사용 하도록 도와주는 라이브라이입니다.(혼동하지 말것)

# Firebase 기본 CLI 명령어

1. nvm list에서 현재 노드 확인

```
> nvm list
```

2. Nodejs 버전 8.17.0으로 변경

```
> nvm use v8.17.0
```

3. npm 업데이트하기

```
> npm install -g npm
```

4. firebase-tools v8.20.0 설치

```
> npm install -g firebase-tools@^8
```

5. firebase 버전 확인

```
> firebase --version
```

6. firebase 로그인

```
> firebase login
```

7. firebase 홈디렉토리 설치

```
> firebase init
```

8. 테스트 서버 실행

```
> firebase serve
```

# Firebase SDK CDN 설정

<!-- update the version number as needed -->

<script defer src="/\_\_/firebase/8.9.1/firebase-app.js"></script>

<!-- include only the Firebase features as you need -->

<script defer src="/\_\_/firebase/8.9.1/firebase-auth.js"></script>

<script defer src="/\_\_/firebase/8.9.1/firebase-firestore.js"></script>

<script defer src="/\_\_/firebase/8.9.1/firebase-storage.js"></script>

# Firestore 초기화 - body 하단에 추가

```
<script>
```

```
// Your web app's Firebase configuration
```

```
var firebaseConfig = {  
  apiKey: "AlzaSyBwUyQSI4o_-6lnbXiKaZdbBgmSvNzGbZk",  
  authDomain: "comstudy-lab.firebaseio.com",  
  projectId: "comstudy-lab",  
  storageBucket: "comstudy-lab.appspot.com",  
  messagingSenderId: "755551809117",  
  appId: "1:755551809117:web:68bc59d02eb7af758767b4"  
};
```

```
// Initialize Firebase
```

```
firebase.initializeApp(firebaseConfig);
```

```
</script>
```

# firebase 사용 - 데이터 읽어오기

```
<script>
```

```
document.addEventListener('DOMContentLoaded', function() {
```

```
  var db = firebase.firestore();
```

```
  db.collection("car").get().then((querySnapshot) => {
```

```
    querySnapshot.forEach((doc) => {
```

```
      //console.log(`${doc.id} => ${doc.data().name} | ${doc.data().company}`);
```

```
      $('#load').append(`<li>${doc.data().name} | ${doc.data().company}</li>`);
```

```
    });
```

```
  });
```

```
});
```

```
</script>
```

# 입력 폼 준비

```
<div>

  <h1>범준샘 코딩교실</h1>

  <fieldset>

    제품명: <input type="text" id="name" value="KOSMO"><br>

    가격: <input type="text" id="price" value="20000"><br>

    메이커: <input type="text" id="company" value="KOREA"><br>

    연식: <input type="text" id="year" value="2011"><br>

    사진: <input type="file" class="image">

  <hr>

  <button id="saveBtn">저장</button>

</fieldset>

<table id="load" border="1" width="99%"></table>

</div>
```

제품명:	<input type="text" value="KOSMO2"/>
가격:	<input type="text" value="20000"/>
메이커:	<input type="text" value="KOREA2"/>
연식:	<input type="text" value="2011"/>
사진:	<input type="button" value="파일 선택"/> 20210629_142545.png

---

# firestore에서 데이터 불러 오기

```
var storage = null;

var db = null;

function loadData (db) {

    db.collection("car").get().then((querySnapshot) => {

        // querySnapshot 반복

        querySnapshot.forEach((doc) => {

            });

        });

    });
}
```



## querySnapshot을 반복해서 행 추가

```
querySnapshot.forEach((doc) => {  
    $('#load').append(`  
        <td></td>  
        <td>${doc.id}</td>  
        <td>${doc.data().name}</td>  
        <td>${doc.data().price}</td>  
        <td>${doc.data().company}</td>  
        <td>${doc.data().year}</td>  
        <td><input data-img="${doc.data().image}" data-id="${doc.id}"  
            type="button" class="delBtn" value="DELETE"> </td>  
    </tr>`);  
});
```

# 삭제 버튼 기능 구현

// 삭제 버튼 기능

```
$('#input.delBtn').click(function(event) {  
    var dataId = $(this).attr("data-id");  
    var dataImg = $(this).attr("data-img");  
    db.collection("car").doc(dataId).delete().then(() => {  
        var storageRef = storage.ref();  
        var delImgRef = storageRef.child('photo/' + dataImg);  
        delImgRef.delete().then(function() {  
            loadData(db); // 삭제 성공 후 재귀호출.  
        }).catch(function() {  
            console.log("delete img error!");  
        })  
    }).catch((error) => {  
        console.error("Error removing document: ", error);  
    });  
});  
// 삭제 버튼 기능 끝
```

# 사진 저장 기능을 위해 storage 준비

```
document.addEventListener('DOMContentLoaded', function() {  
    //console.dir(firebase);  
    db = firebase.firestore();  
    storage = firebase.storage();  
});
```


# 사진 저장 기능



```
$("#saveBtn").click(function(event) {
  var timestamp = new Date().getTime();
  var fileData = $(".image")[0].files[0];
  var fileName = fileData.name;
  var suffix = fileName.substr(fileName.lastIndexOf("."));
  var newFileName = timestamp+suffix;
  console.log(newFileName);
  // 저장 경로
  var uploadRef = storage.ref();
  var fileSavePath = uploadRef.child('photo/'+ newFileName);
  //console.log('fileSavePath>>> ', fileSavePath);
  var result = fileSavePath.put(fileData).then(function(snapshot) {
    console.log("사진 저장 ...");
    // 먼저 사진을 저장 하고
    snapshot.ref.getDownloadURL().then(function(downloadUrl) {
      // 사진 저장 완료 되면 firestore에 정보 기록
      var saveData = {
        name: $('#name').val(),
        price: $('#price').val(),
        company: $('#company').val(),
        year: $('#year').val(),
        image: newFileName,
        downloadUrl: downloadUrl
      }
      // firestore에 데이터 저장
      db.collection("car").add(saveData).then((docRef) => {
        console.log("Document written with ID: ", docRef.id);
        // 저장 성공 후에 목록을 다시 로드 한다.
        loadData(db);
      }).catch((error) => {
        console.error("Error adding document: ", error);
      });
    });
  });
}); // 사진 저장 버튼 기능 끝
```

# 사람정보 관리기능 화면 구현

Comstudy21

사람 정보 관리 프로그램

아이디	<input type="text" value="KIM"/>	
성명	<input type="text" value="김길동"/>	
이메일	<input type="text" value="kim@coms21.lab"/>	
사진	<input type="button" value="파일 선택"/> cat02.jpg	
<input type="button" value="저장하기"/> <input type="button" value="수정하기"/>		

번호	사진	아이디	성명	이메일	삭제
0		HONG	홍길동	hong@coms21.lab	<input type="button" value="삭제"/>
1		CHOI	최길동	choi@coms21.lab	<input type="button" value="삭제"/>

Firestore SDK loaded with auth, firestore, storage

# 사람정보 관리기능 화면 구현

```
<h2>Comstudy21</h2>
<h1>사람 정보 관리 프로그램</h1>
<hr>
<table class="table table-bordered">
  <tr>
    <th>아이디</th><td><input type="text" id="user_id" value="KIM"></td>
    <td rowspan="5"></td>
  </tr>
  <tr>
    <th>성명</th><td><input type="text" id="user_name" value="김길동"></td>
  </tr>
  <tr>
    <th>이메일</th><td><input type="text" id="email" value="kim@coms21.lab"></td>
  </tr>
  <tr>
    <th>사진</th><td><input type="file" id="photo"></td>
  </tr>
  <tr>
    <th colspan="2">
      <button id="saveBtn">저장하기</button>
      <button id="updateBtn">수정하기</button>
    </th>
  </tr>
</table>
<hr>
<table id="list" class="table table-hover"></table>
</div>
<p id="load">Firebase SDK Loading&hellip;</p>
```

# 이미지 미리 보기 구현(Step01)

```
<body>  
  <form action="">  
    <input type="file" class="file" id="imageSelector" name="imageSelector"  
      accept="image/jpeg, image/jpg, image/png" multiple />  
  </form>  
  <div id="preview-image"></div>  
</body>
```

## 이미지 미리 보기 구현(Step02)

```
// FileReader를 이용한 방식
function readImage(input) {
  // 인풋 태그에 파일이 있는 경우
  if (input.files && input.files[0]) {
    // 이미지 파일인지 검사 (생략)
    // FileReader 인스턴스 생성
    const reader = new FileReader()
    // 이미지가 로드된 경우
    reader.onload = e => {
      //console.log(e.target.result);
      let readFileURL = e.target.result;
      const previewImageDiv = document.getElementById("preview-image");
      $(previewImageDiv).append('');
    }
    // reader가 이미지 읽도록 하기
    reader.readAsDataURL(input.files[0])
  }
}
```



## 이미지 미리 보기 구현(Step03)

```
// URL.createObjectURL 을 이용하는 방식
function readImagePreview (elem) {
    let readDataURL = URL.createObjectURL (elem.files [0]); //파일 객체에서 이미지 데이터 가져옴.
    console.log(readDataURL);

    const previewImageDiv = document.getElementById ("preview-image");
    $(previewImageDiv).append('');

    //이미지 로딩 후 객체를 메모리에서 해제
    $('img', previewImageDiv).on('load', function () {
        URL.revokeObjectURL ($(this).attr('src'));
    });
}
```

## 이미지 미리 보기 구현(Step03) - 호출

```
$('input.file').change((event) => {  
    readImage(event.target);  
    readImagePreview(event.target);  
});
```

## 이미지 미리 보기 구현(Step05) - 다중파일

```
<body>
  <p>한번에 여러 이미지파일 선택 가능 합니다.</p>
  <form>
    <input type="file" class="file" id="imageSelector" name="imageSelector"
      accept="image/jpeg, image/jpg, image/png" multiple />
  </form>
  <div id="preview-image"></div>
</body>
```

## 이미지 미리 보기 구현(Step06)

```
// URL.createObjectURL 을 이용하는 방식 - 파일의 경로를 읽어오는 방식
function readImagePreview (elem) {
  console.log(elem.files.length)
  // 이미지 저장 위치
  const previewImageDiv = document.getElementById("preview-image");
  $(previewImageDiv).html("");
  for (var i = 0; i < elem.files.length; i++) {
    let readFileURL = URL.createObjectURL (elem.files[i]); //파일 객체에서 이미지 데이터
    가져옴.

    console.log(readFileURL);
    $(previewImageDiv).append('');





    //이미지 로딩 후 객체를 메모리에서 해제
    $('img', previewImageDiv).on('load', function () {
      URL.revokeObjectURL ($(this).attr('src'));
    });
  }
}
```

## 파일 선택 시 함수 호출

```
$('#input.file').change((event) => {  
    readImagePreview(event.target);  
});
```

# 사람 정보 관리 프로그램 화면

ObjectID		
아이디	<input type="text"/>	
성명	<input type="text"/>	
이메일	<input type="text"/>	
사진	<input type="button" value="파일 선택"/> <input type="button" value="선택된 파일 없음"/>	상세정보:
비밀번호:	<input type="text"/> <input type="button" value="저장하기"/> <input type="button" value="수정하기"/>	

번호	사진	아이디	성명	이메일	삭제
0		HO	호연지기	ho@coms21.lab	<input type="button" value="비밀번호"/> <input type="button" value="삭제"/>
1		TIGER	호랭이	tiger@naver.com	<input type="button" value="비밀번호"/> <input type="button" value="삭제"/>
2		KIM	김범준	beomjoon@coms21.lab	<input type="button" value="비밀번호"/> <input type="button" value="삭제"/>
3		EUNG	응무소주	eung@coms21.lab	<input type="button" value="비밀번호"/> <input type="button" value="삭제"/> <input type="button" value="원문 저장하기"/>

# 삭제 기능

```
// 생성된 순서의 역순으로 삭제한다 (doc삭제 -> file삭제)
db.collection('saram').doc(docID).get().then((docRef) => {
  var passwd = docRef.data().passwd == undefined ? "" : docRef.data().passwd;
  if (passwd !== confirmPasswd) {
    alert("비밀 번호가 일치 하지 않습니다!");
    $(this).parent().find('input.passwd').focus();
    return;
  }
  db.collection('saram').doc(docID).delete().then(() => {
    var storageRef = storage.ref();
    var delImgRef = storageRef.child('images/' + $(THIS).attr('data-img'));
    delImgRef.delete().then(() => {
      // 삭제 완료 후 목록 갱신
      showList();
      // 입력 창 초기화
      clearInputArea();
      alert("항목 삭제 완료!");
    }).catch((deleteFileErr) => {
      console.log('삭제 실패!', deleteFileErr);
    });
  }); // end of delete
});
```

## 상세 보기

```
var objId = $(this).attr('data-doc-id');
db.collection('saram').doc(objId).get().then((sm) => {
  var item = sm.data();
  var image = item.image;
  $('#object_id_show').text(sm.id);
  $('#object_id').val(sm.id);
  $('#user_id').val(item.user_id);
  $('#user_name').val(item.user_name);
  $('#email').val(item.email);
  $('#photo').val("");
  $('#imageView').attr('src', item.fileURL);
  $('#imageView').attr('title', item.orginFileName);
  $('#imageView').attr('alt', item.image);
  $('#passwd').val("");
});
```



# 수정 기능

```
// 만약 새 파일이 있다면 기존 파일은 지우고 새 파일로 교체 한다.
if (objectId) {
  db.collection('saram').doc(objectID).get().then((docRef) => {
    var doc = docRef.data();
    var passwd = doc.passwd == undefined ? "" : doc.passwd;
    if (passwd !== confirmPasswd) {
      alert("비밀 번호가 일치 하지 않습니다!");
      $(this).parent().find('input.passwd').focus();
      return;
    }
    //console.log("doc >>> ", doc);
    try {
      var saveFile = $('#photo')[0].files[0];
      var orginFileName = saveFile.name;
      var storageRef = storage.ref();
      //console.log(">>>> 파일 있다 : ", objectId);
      storageRef.child('images/' + doc.image).delete().then(() => {
        //console.log("기존 파일 삭제 완료!");
        saveDataWithFile(doc, mode, saveFile, orginFileName, storageRef, objectId);
      });
    } catch (e) {
      //console.log('>>>> 파일이 없다 : ');
      updateDataWithoutFile(doc, mode, objectId);
    }
  });
} else {
  alert("Object ID가 존재 하지 않음. 먼저 수정 할 항목을 선택 하세요.");
}
```