

Firebase



요즘 누가 유료로 웹서비스 구축 하나, 파이어 베이스의 모든 것-JAVASCRIPT

David . 2021.02.01



Firestore Authentication 실습

startup.sh 생성

```
vi startup.sh  
chmod +x ./startup.sh
```

```
cd /Users/David/firebase-deploy/david-zzang/public  
python -m SimpleHTTPServer
```


Firebase Authentication 실습

```
<!-- Insert these scripts at the bottom of the HTML, but before you use any Firebase services -->
<!-- Firebase App (the core Firebase SDK) is always required and must be listed first -->

<script src="https://www.gstatic.com/firebasejs/8.2.2/firebase-app.js"></script>

<!-- If you enabled Analytics in your project, add the Firebase SDK for Analytics -->

<script src="https://www.gstatic.com/firebasejs/8.2.2/firebase-analytics.js"></script>

<!-- Add Firebase products that you want to use -->

<script src="https://www.gstatic.com/firebasejs/8.2.2/firebase-auth.js"></script>

<script src="https://www.gstatic.com/firebasejs/8.2.2/firebase-firestore.js"></script>
```

Firestore Authentication 실습

```
// Firestore 초기화 (mycutom.js)

// For Firestore JavaScript SDK v7.20.0 and later, `measurementId` is an optional field

var firestoreConfig =
{
  apiKey: "API_KEY",
  authDomain: "PROJECT_ID.firebaseio.com",
  databaseURL: "https://PROJECT_ID.firebaseio.com",
  projectId: "PROJECT_ID",
  storageBucket: "PROJECT_ID.appspot.com",
  messagingSenderId: "SENDER_ID",
  appId: "APP_ID",
  measurementId: "G-MEASUREMENT_ID",
};

// Initialize Firestore
firebase.initializeApp(firestoreConfig);
```


Firestore Authentication 실습

```
$(document).ready(function ($) {
```

```
    firebase.auth().onAuthStateChanged(function (user) {  
        if (user) {}  
        else{}  
    });
```

```
});
```

} 비동기 Callback함수

Firestore Authentication 실습

```
firebase.auth().signInWithEmailAndPassword(email, password).then(function (result) {
```

```
    //DO-NOTHING
```

```
}).catch(function (error) {
```

```
    // Handle Errors here.
```

```
    // console.log('err',error);
```

```
    var errorCode = error.code;
```

```
    if (errorCode == 'auth/invalid-email') {
```

```
        showAlert("잘못된 메일 형식 입니다.");
```

```
    }
```

```
    else if (errorCode == 'auth/wrong-password') {
```

```
        showAlert("패스 워드 혹은 이메일 정보를 다시 확인 부탁드립니다.");
```

```
    }
```

```
    else if (errorCode == 'auth/user-not-found') {
```

```
        showAlert("등록되지 않은 사용자 입니다.");
```

```
    }
```

```
    else {
```

```
        showAlert("[ " + errorCode + " : " + error.message + " ]");
```

```
    }
```

```
});
```


Firebase Authentication 실습

```
firebase.auth().createUserWithEmailAndPassword(email, password).then(function (user) {
```

```
  }).catch(function (error) {  
    // Handle Errors here.  
    var errorCode = error.code;  
    var errorMessage = error.message;  
  });
```

```
firebase.auth().signOut().then(function () {
```

```
  }, function (error) {  
    //DO  
  });
```

Firestore Authentication 실습

```
var cu = window.location.href;

var n1 = cu.indexOf('login');

if (user) {
    if (n1 > 1) {
        window.open('../..', '_self', false);
    }
} else {
    if (n1 < 1) {
        window.open('./auth/login/', '_self', false);
    }
}
```


Firebase Authentication 실습

```
function googlelogIn() {
  var provider = new firebase.auth.GoogleAuthProvider();
  provider.addScope("https://www.googleapis.com/auth/plus.login");
  provider.setCustomParameters({
    prompt: "select_account"
  });

  firebase.auth().signInWithRedirect(provider).then(function (result) {
    firebase.auth()
      .setPersistence(firebase.auth.Auth.Persistence.LOCAL)
      .then(() => {
        // This gives you a Google Access Token. You can use it to access the Google API.
        var token = result.credential.accessToken;
        // The signed-in user info.
        var user = result.user;
      })
      .catch(function (error) { // Handle Errors here.
        var errorCode = error.code;
        // The email of the user's account used.
        var email = error.email;
      });
  });
}
```

Firebase Authentication 실습

```
function facebooklogIn() {
var provider = new firebase.auth.FacebookAuthProvider();
firebase
.auth()
.signInWithRedirect(provider)
.then(function(result) {
firebase
.auth()
.setPersistence(firebase.auth.Auth.Persistence.LOCAL)
.then(() => {
// This gives you a Google Access Token. You can use it to access the Google API.
var token = result.credential.accessToken;
// The signed-in user info.
var user = result.user;
});
})
.catch(function(error) {
// Handle Errors here.
var errorCode = error.code;
var errorMessage = error.message;
});
},
```


Firestore Authentication 실습

```
function onSendCode() {
  firebase.auth().languageCode = 'ko';

  var appVerifier = window.recaptchaVerifier = new firebase.auth.RecaptchaVerifier(
    "recaptcha-container",
    {
      size: "invisible",
      callback: function (confirmationResult) {
        window.confirmationResult = confirmationResult;
      }
    }
  );

  var _phoneNumber = '+821012341234';
  // create user using firebase
  firebase.auth().signInWithPhoneNumber(_phoneNumber, appVerifier)
    .then((confirmationResult) => {
      window.confirmationResult = confirmationResult;

    }, (error) => {
    })
}
```

Firestore Authentication 실습

1. 환경 셋팅 후 console.log 확인
2. 회원가입 예제 Copy
3. Social Login(회원가입과 로그인 구분 없음)
4. email Login(회원가입/로그인)
5. 전화번호 Login(회원가입/로그인)

```
const errorCodeSet = [
  { no: "E-001", code: 'auth/claims-too-large', message: '구글에서 제공되는 페이로드가 허용키인 1000 바이트를 초과합니다.' },
  { no: "E-002", code: 'auth/email-already-exists', message: '해당 이메일은 이미 기존 사용자가 사용 중입니다.' },
  { no: "E-003", code: 'auth/id-token-expired', message: '세션 토큰이 만료되었습니다.' },
  { no: "E-004", code: 'auth/id-token-revoked', message: '세션 토큰이 회수되었습니다.' },
  { no: "E-005", code: 'auth/insufficient-permission', message: '해당 리소스에 접근할 권한이 없습니다.' },
  { no: "E-006", code: 'auth/internal-error', message: '요청을 처리하는 동안 인증 서버에서 예기치 않은 오류가 발생했습니다.' },
  { no: "E-007", code: 'auth/invalid-argument', message: '인증 방법에 잘못된 인수가 언급되었습니다.' },
  { no: "E-008", code: 'auth/invalid-claims', message: '사용자 지정 클레임 속성이 잘못되었습니다.' },
  { no: "E-009", code: 'auth/invalid-continue-uri', message: '유효한 URL 문자열이어야 합니다.' },
  { no: "E-010", code: 'auth/invalid-creation-time', message: '생성 시간은 유효한 UTC 형식이어야 합니다.' },
  { no: "E-011", code: 'auth/invalid-credential', message: '해당 사용자는 인증 작업을 수행하는 데 사용할 수 없습니다.' },
  { no: "E-012", code: 'auth/invalid-disabled-field', message: '비활성화 된 사용자 속성에 제공된 값이 잘못되었습니다.' },
  { no: "E-013", code: 'auth/invalid-display-name', message: '해당 사용자의 이름 값이 누락되었습니다. 사용자 이름은 필수입니다.' },
  { no: "E-014", code: 'auth/invalid-dynamic-link-domain', message: '제공된 동적 링크 도메인이 현재 프로젝트에 승인되지 않았습니다.' },
  { no: "E-015", code: 'auth/invalid-email', message: '잘못된 이메일 주소입니다.' },
  { no: "E-016", code: 'auth/invalid-email-verified', message: '이메일 확인 속성이 잘못 되었습니다.' },
  { no: "E-017", code: 'auth/invalid-hash-algorithm', message: '해시 알고리즘이 잘못되었습니다.' },
  { no: "E-018", code: 'auth/invalid-hash-block-size', message: '해시 블록 사이즈가 숫자가 아닙니다.' },
  { no: "E-019", code: 'auth/invalid-hash-derived-key-length', message: '해시 블록 사이즈가 잘못 되었습니다.' },
  { no: "E-020", code: 'auth/invalid-hash-key', message: '유효한 해시 키 버퍼가 아닙니다.' },
  { no: "E-021", code: 'auth/invalid-hash-memory-cost', message: '유효한 해시 메모리 코스트 숫자가 아닙니다.' },
  { no: "E-022", code: 'auth/invalid-hash-parallelization', message: '유효한 해시 병렬 사이즈가 아닙니다.' },
  { no: "E-023", code: 'auth/invalid-hash-rounds', message: '유효한 해시 회차가 아닙니다.' },
  { no: "E-024", code: 'auth/invalid-hash-salt-separator', message: '유효한 해시 알고리즘 구분자입니다.' },
  { no: "E-025", code: 'auth/invalid-id-token', message: '제공된 ID 토큰은 유효한 Firebase ID 토큰이 아닙니다.' },
  { no: "E-026", code: 'auth/invalid-last-sign-in-time', message: '마지막 로그인 시간은 유효한 UTC 날짜 문자열이어야 합니다.' },
  { no: "E-027", code: 'auth/invalid-page-token', message: '페이지 토큰이 유효하지 않습니다.' },
  { no: "E-028", code: 'auth/invalid-password', message: '비밀번호가 올바르지 않습니다.' },
  { no: "E-028", code: 'auth/wrong-password', message: '비밀번호가 올바르지 않습니다.' },
  { no: "E-029", code: 'auth/invalid-password-hash', message: '암호 해시는 유효한 바이트 버퍼 여야 합니다.' },
  { no: "E-030", code: 'auth/invalid-password-salt', message: '유효하지 않은 값입니다.' },
  { no: "E-031", code: 'auth/invalid-phone-number', message: '전화번호가 올바르지 않습니다.' },
  { no: "E-032", code: 'auth/invalid-photo-url', message: '프로필 이미지 URL 값이 잘못되었습니다.' },
  { no: "E-033", code: 'auth/invalid-provider-data', message: '유효한 개체 배열이 아닙니다.' },
  { no: "E-034", code: 'auth/invalid-provider-id', message: '유효한 공급자 식별 코드가 아닙니다.' },
  { no: "E-035", code: 'auth/invalid-session-cookie-duration', message: '세션 쿠키 기간은 5 분에서 2 주 사이의 유효한 밀리 초 단위 여야 합니다.' },
  { no: "E-036", code: 'auth/invalid-uid', message: '제공된 uid 는 최대 128 자의 비어 있지 않은 문자열이어야 합니다.' },
  { no: "E-037", code: 'auth/invalid-user-import', message: '가져올 사용자 레코드가 잘못되었습니다.' },
  { no: "E-038", code: 'auth/maximum-user-count-exceeded', message: '가져올 수 있는 최대 사용자 수를 초과했습니다.' },
  { no: "E-039", code: 'auth/missing-android-pkg-name', message: 'Android 앱을 설치해야하는 경우 Android 패키지 이름을 제공해야 합니다.' },
  { no: "E-040", code: 'auth/missing-continue-uri', message: '유효한 URL 이 아닙니다.' },
  { no: "E-041", code: 'auth/missing-hash-algorithm', message: '암호 해시 알고리즘이 누락되었습니다.' },
  { no: "E-042", code: 'auth/missing-ios-bundle-id', message: '요청에 iOS 번들 ID 가 없습니다.' },
```