



Firebase for Web

with ABCD

Created at 2016. 08. / 장승빈

오늘의 진행

- 실습 자료 준비
- Firebase Database
 - NoSQL와 Realtime (장점소개)
 - Data구조 (NoSQL의 특징)
- 간단한 DB 사용
- 실시간 접속자

실습 자료 다운로드

- 실습 자료를 다운로드 받습니다.
 - https://github.com/denlyou/study_firebase4web_abcd
- Firebase project로 초기화 되어 있는 폴더에 public 폴더의 파일을 복사합니다
 - (지난 실습 파일들과 통합하였습니다)



Firebase Database

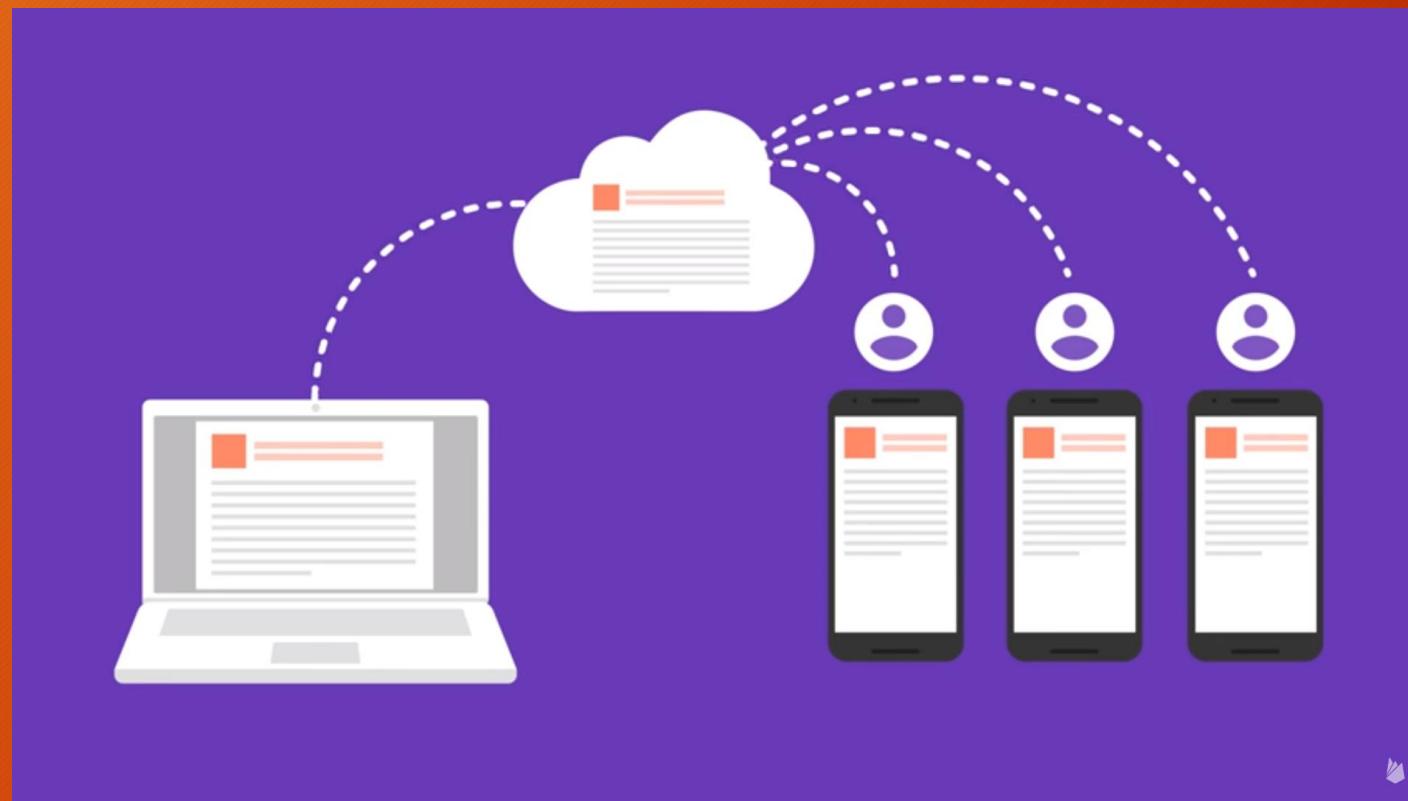
The screenshot shows the Firebase Database landing page. At the top, there are three dark grey rectangular boxes representing servers, each with a small blue and yellow status indicator. Below the servers is a yellow background area containing a white wavy line graphic. The word "Database" is centered in a large, bold, dark font. Below it, a paragraph of text in Korean reads: "연결된 모든 클라이언트에서 실시간으로 데이터를 저장하고 동기화할 수 있습니다." At the bottom of the page, there are two blue buttons: "자세히 알아보기" on the left and "시작하기" on the right.

Database

연결된 모든 클라이언트에서 실시간으로 데이터를 저장하고 동기화할 수 있습니다.

[자세히 알아보기](#) [시작하기](#)

Firebase Database?



https://youtu.be/U5aeM5dvUpA?list=PLl-K7zZEzYLmOF_07IayrTntevxtbUxDL

NoSQL적인 특징

- 대용량 데이터 처리, 분산 시스템 구성에 유리
- 데이터를 Key-Value 쌍으로 저장
 - Value에는 또 하위에 key-value쌍의 데이터가 저장 가능
- RDBMS와 비교
 - 고정된 스키마를 갖지 않아 저장 구조가 유동적
 - Foreign key를 사용하지 않아 join등의 연산이 없음
- 특정 용도에 특화되어 있음
 - 채팅, 메시지, 뉴스 피드와 같은 기능 구현에 매우 유리

Firebase Database의 자료 구조

- JSON의 구조를 기본으로 생각

```
{  
    "key": "value" ,  
    "key": "value" ,  
    "key": "value" ,  
    "key": "value" ,  
    ...  
    "key": { "column": "value", "column": "value", "column": "value" } ,  
    "key": { "column": "value", "column": "value", "column": "value" } ,  
    "key": { "column": "value", "column": "value", "column": "value" } ,  
    "key": { "column": "value", "column": "value", "column": "value" } ,  
}
```

Key	Value
Key	Value

:



Key	Column	Column	Column
	Value	Value	Value
Key	Column	Column	Column
	Value	Value	Value

:

[이미지 출처 : 조대협님 블로그]

데이터의 접근 방식

- 하위 Key들을 '/'로 구분해서 현재 다룰 위치를 ref() 함수로 지정
 - (폴더 or 디렉토리 개념과 유사하게 접근)
- example

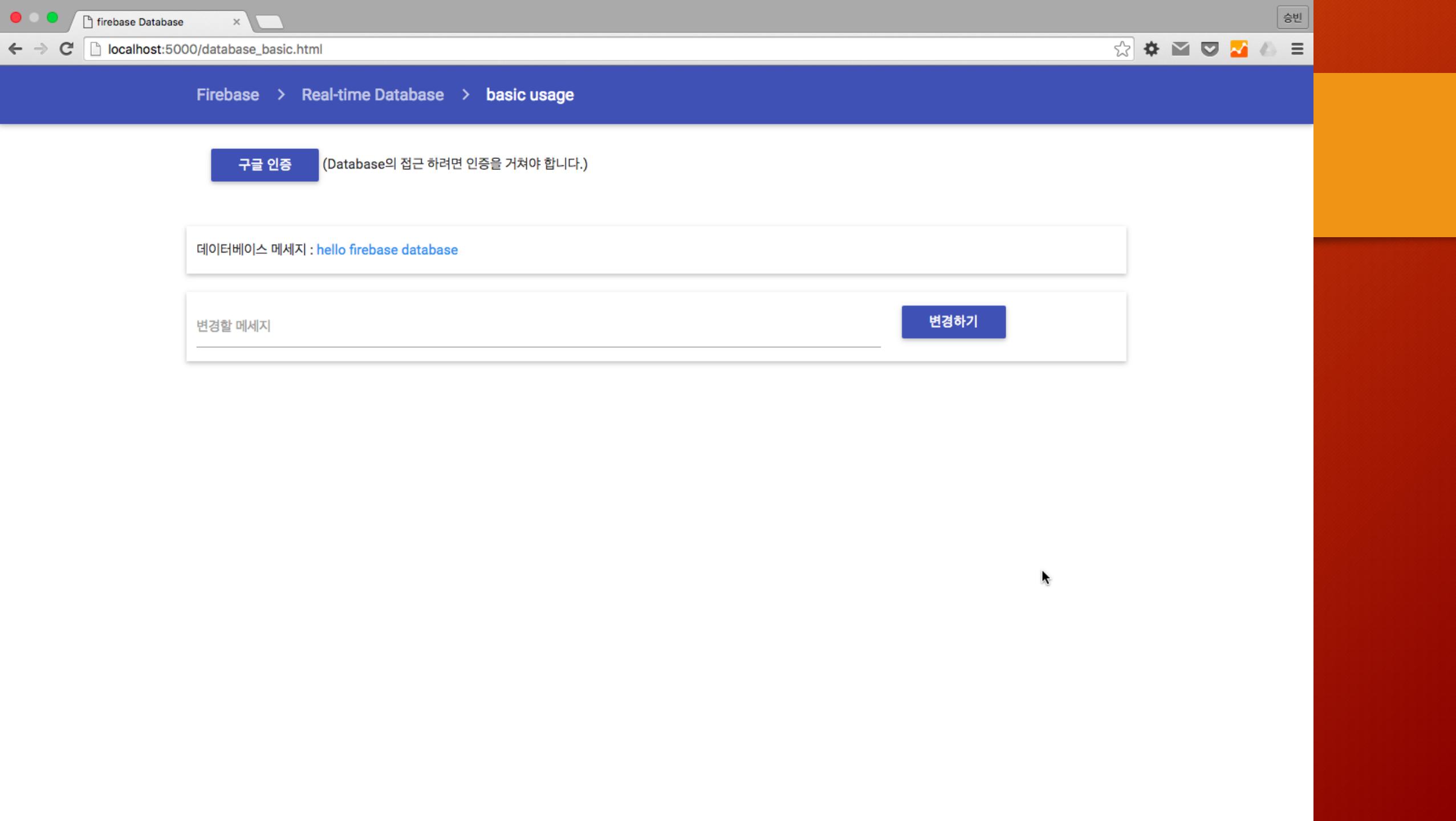
```
hellofirebase4web ..... → firebase.database().ref('/')
  |- Member ..... → firebase.database().ref('/Member')
    |- SBJang ..... → firebase.database().ref('/Member/SBJang')
      |- Gender: "남자" ..... → firebase.database().ref('/Member/SBJang/Gender')
      |- Job: "Developer" ..... → firebase.database().ref('/Member/SBJang/Job')
      |- Name: "장승빈" ..... → firebase.database().ref('/Member/SBJang/Name')
      ...
      메세지: "안녕~ 불판다"
```

기본적인 DB 사용

다음 파일을 편집기로 열어주세요

[project folder]/public/database_basic.html

1. HTML 파일의 UI 확인
2. firebase SDK 링크
3. 구글 인증 넣기
4. DB에 데이터 쓰기
5. 데이터 변경하기
6. 데이터 읽어오기



firebase

public

auth

css

database

- database_basic_step1.html
- database_basic_step2.html
- database_basic_step3.html
- database_basic_step4.html
- database_basic_step5.html
- database_basic_step6.html

font

fonts

images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html

database_basic.html

database.html

fireinit.js

fireinit.js.sample

hosting.html

index.html

index2.html

mail_auth.html

.DS_Store

.firebaserc

database.rules.json

firebase.json

database_basic_step1.html x database_basic.html x

```
73     &copy; 2016 Copyright <a href="http://sbssoft.kr" target="_blank" class="white-text">SBJang</a>
74     <a href="http://abcds.kr/" target="_blank" class="white-text">ABCD</a> (
75       <a href="https://www.facebook.com/groups/aboutCoding/" target="_blank" class="white-text">Fa
76         <a class="white-text right" href="http://materializecss.com/" target="_blank">design by Mate
77           </div>
78         </div>
79       </footer>
80
81      <!-- ////////////////////////////// javascript 스크립트 영역 //////////////////////////////
82
83      <!--Import jQuery before materialize.js-->
84      <script type="text/javascript" src="https://code.jquery.com/jquery-2.1.1.min.js"></script>
85      <script type="text/javascript" src="/js/materialize.min.js"></script>
86
87      <!-- TODO 1. firebase SDK 링크 -->
88      <script type="text/javascript">
89        /* TODO 2. 버튼을 눌렀을 경우 구글 인증 동작하게 */
90
91        /* TODO 3. 데이터 베이스 '/메세지' 쓰기 */
92
93        /* TODO 4. DB에서 '/메세지'값 읽어오기 */
94
95        /* TODO 5. DB 메세지 값 변경 */
96
97      </script>
98
99    </body>
100   </html>
101
```

항상 Firebase SDK 링크를 제일 먼저!

```
firebase
  public
    auth
    css
  database
    database_basic_step1.html
    database_basic_step2.html
    database_basic_step3.html
    database_basic_step4.html
    database_basic_step5.html
    database_basic_step6.html
  font
  fonts
  images
  js
    404.html
    auth_firebase_ui.html
    auth_google.html
    auth_mail.html
    auth_term.html
    auth.html
    database_basic.html
    database.html
    fireinit.js
    fireinit.js.sample
    hosting.html
    index.html
    index2.html
    mail_auth.html
  .DS_Store
  .firebaserc
  database.rules.json
  firebase.json

database_basic_step1.html x database_basic.html x

78      </div>
79      </footer>
80
81      
82
83      <!--Import jQuery before materialize.js--&gt;
84      &lt;script type="text/javascript" src="https://code.jquery.com/jquery-2.1.1.min.js"&gt;&lt;/script&gt;
85      &lt;script type="text/javascript" src="/js/materialize.min.js"&gt;&lt;/script&gt;
86
87      <!-- 1. firebase SDK 링크 --&gt;
88      &lt;script src="https://www.gstatic.com/firebasejs/3.1.0.firebaseio.js"&gt;&lt;/script&gt;
89      &lt;script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-app.js"&gt;&lt;/script&gt;
90      &lt;script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-auth.js"&gt;&lt;/script&gt;
91      &lt;script src="https://www.gstatic.com/firebasejs/3.1.0.firebaseio-database.js"&gt;&lt;/script&gt;
92      &lt;script src="/fireinit.js"&gt;&lt;/script&gt;
93      &lt;script type="text/javascript"&gt;
94          /* TODO 2. 버튼을 눌렀을 경우 구글 인증 동작하게 */
95
96          /* TODO 3. 데이터 베이스 '/메세지' 쓰기 */
97
98          /* TODO 4. DB에서 '/메세지'값 읽어오기 */
99
100         /* TODO 5. DB 메세지 값 변경 */
101
102     &lt;/script&gt;
103
104     &lt;/body&gt;
105
106     &lt;/html&gt;</pre>
```

firebase

public

auth

css

database

font

fonts

images

js

404.html

auth_firebase_ui.html

auth_google.html

auth_mail.html

auth_term.html

auth.html

database_basic.html

database.html

fireinit.js

fireinit.js.sample

hosting.html

index.html

index2.html

mail_auth.html

.DS_Store

.firebaserc

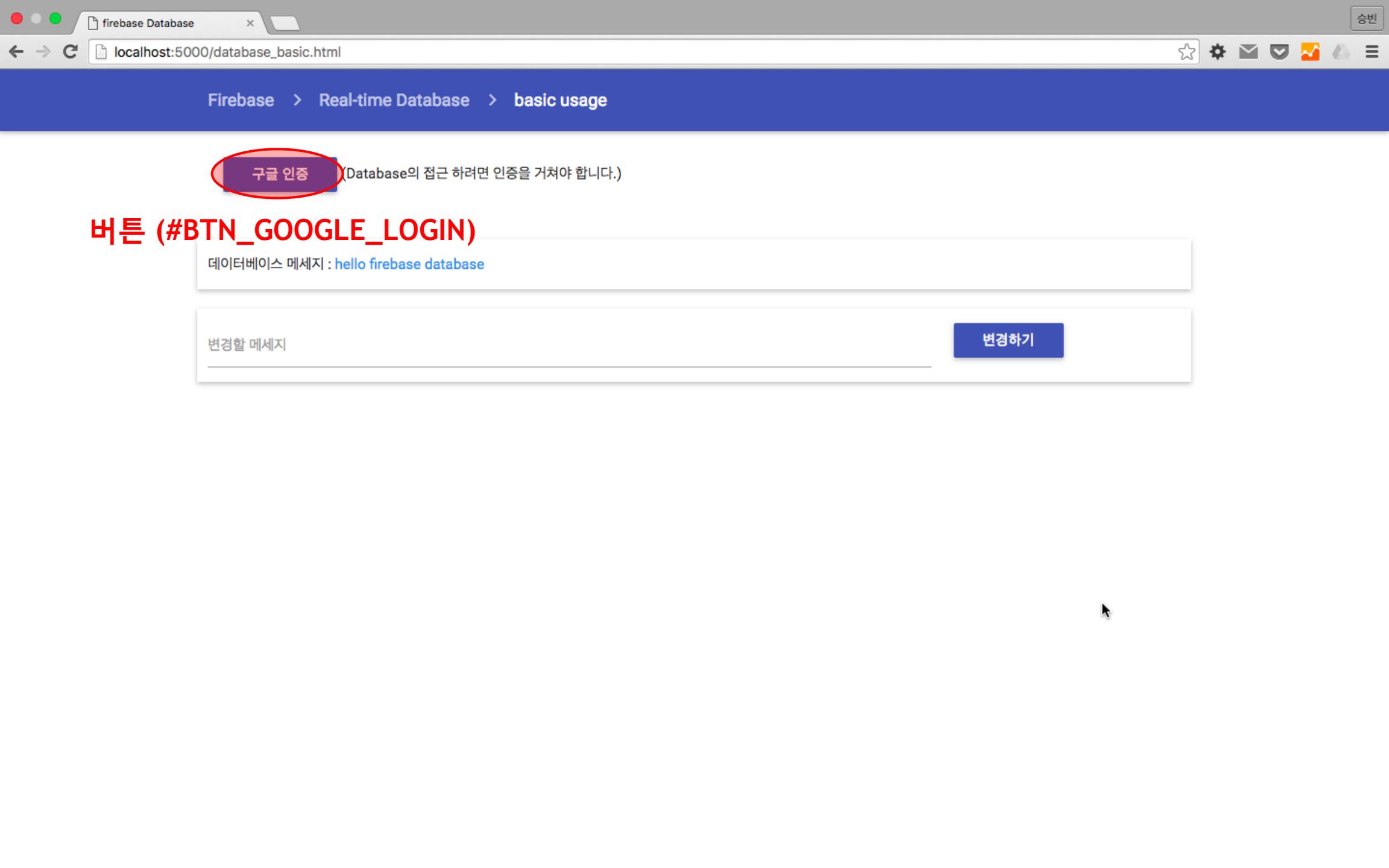
database.rules.json

firebase.json

database_basic.html

```
23     </div>
24     </div>
25 </nav>
26
27 <!-- 내용 부분 -->
28 <div class="container" id="CONTENTS">
29     <br>
30     <!-- TODO 2. 로그인 버튼 확인 -->
31     <div class="row" style="margin: 0; padding: 5px;">
32         <div class="col l12 m12 s12">
33             <button id="BTN_GOOGLE_LOGIN" type="button" class="btn indigo">구글 인증</button>
34             (Database의 기본 접근 하려면 인증을 거쳐야 합니다.)
35         </div>
36     </div>
37     <br><br>
38
39     <div class="row z-depth-1">
40         <div class="col l12 m12 s12">
41             <p>
42                 데이터베이스 메세지 :
43                 <!-- TODO 4. 상태 보기용 span 확인-->
44                 <span id="DB_MESSAGE" class="blue-text">(Loading..)</span>
45             </p>
46         </div>
47     </div>
48
49     <!-- TODO 5. 메세지 수정 폼 확인하기 -->
50     <div class="row z-depth-1">
51         <div class="col l9 m9 s12 input-field">
52             <input id="INPUT_MESSAGE" type="text" class="validate">
```

구글 인증 버튼 태그 확인



firebase

public

auth

css

database

- database_basic_step1.html
- database_basic_step2.html
- database_basic_step3.html
- database_basic_step4.html
- database_basic_step5.html
- database_basic_step6.html

font

fonts

images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting.html
- index.html
- index2.html
- mail_auth.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase.json

database_basic_step1.html x database_basic_step2.html x

```
78      </div>
79    </footer>
80
81    <!-- ////////////////////////////// javascript 스크립트 영역 //////////////////////////////
82
83    <!--Import jQuery before materialize.js-->
84    <script type="text/javascript" src="https://code.jquery.com/jquery-2.1.1.min.js"></script>
85    <script type="text/javascript" src="/js/materialize.min.js"></script>
86
87    <!-- 1. firebase SDK 링크 -->
88    <script src="https://www.gstatic.com/firebasejs/3.1.0.firebaseio.js"></script>
89    <script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-app.js"></script>
90    <script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-auth.js"></script>
91    <script src="https://www.gstatic.com/firebasejs/3.1.0.firebaseio-database.js"></script>
92    <script src="/fireinit.js"></script>
93    <script type="text/javascript">
94      /* TODO 2. 버튼을 눌렀을 경우 구글 인증 동작하게 */
95
96      /* TODO 3. 데이터 베이스 '/메세지' 쓰기 */
97
98      /* TODO 4. DB에서 '/메세지'값 읽어오기 */
99
100     /* TODO 5. DB 메세지 값 변경 */
101
102    </script>
103
104  </body>
105
106
```

지난 시간 배운 인증 코드를 작성 합니다

```
firebase
  public
    auth
    css
  database
    database_basic_step1.html
    database_basic_step2.html
    database_basic_step3.html
    database_basic_step4.html
    database_basic_step5.html
    database_basic_step6.html
  font
  fonts
  images
  js
    404.html
    auth_firebase_ui.html
    auth_google.html
    auth_mail.html
    auth_term.html
    auth.html
    database_basic.html
    database.html
    fireinit.js
    fireinit.js.sample
    hosting.html
    index.html
    index2.html
    mail_auth.html
  .DS_Store
  .firebaserc
  database.rules.json
  firebase.json

database_basic_step2.html x database_basic_step2.html x
85 <script type="text/javascript" src="/js/materialize.min.js"></script>
86
87     <!-- 1. firebase SDK 링크 --&gt;
88     &lt;script src="https://www.gstatic.com/firebasejs/3.1.0.firebaseio.js"&gt;&lt;/script&gt;
89     &lt;script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-app.js"&gt;&lt;/script&gt;
90     &lt;script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-auth.js"&gt;&lt;/script&gt;
91     &lt;script src="https://www.gstatic.com/firebasejs/3.1.0.firebaseio-database.js"&gt;&lt;/script&gt;
92     &lt;script src="/fireinit.js"&gt;&lt;/script&gt;
93     &lt;script type="text/javascript"&gt;
94         /* 2. 버튼을 눌렀을 경우 구글 인증 동작하게 */
95         $('#BTN_GOOGLE_LOGIN').click(function(){
96             var provider = new firebase.auth.GoogleAuthProvider();
97             firebase.auth().signInWithPopup(provider).then(function(result) {
98                 }).catch(function(error) {
99                     alert(error.message)
100                });
101            });
102
103         /* TODO 3. 데이터 베이스 '/메세지' 쓰기 */
104
105         /* TODO 4. DB에서 '/메세지' 값 읽어오기 */
106
107         /* TODO 5. DB 메세지 값 변경 */
108
109     &lt;/script&gt;
110
111     &lt;/body&gt;
112     &lt;/html&gt;
113</pre>
```

DB에 데이터 쓰기

firebase.database.Reference.set()

- <https://firebase.google.com/docs/reference/js/firebase.database.Reference#set>
- Sample Code

```
function writeUserData(userId, name, email, imageUrl) {
  firebase.database().ref('users/' + userId).set({
    username: name,
    email: email,
    profile_picture : imageUrl
  });
}
```

firebase

public

auth

css

database

- database_basic_step1.html
- database_basic_step2.html
- database_basic_step3.html
- database_basic_step4.html
- database_basic_step5.html
- database_basic_step6.html

font

fonts

images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting.html
- index.html
- index2.html
- mail_auth.html

.DS_Store

.firebaserc

database.rules.json

firebase.json

database_basic_step2.html x database_basic_step2.html x

```
85 <script type="text/javascript" src="/js/materialize.min.js"></script>
86
87 <!-- 1. firebase SDK 링크 -->
88 <script src="https://www.gstatic.com/firebasejs/3.1.0/firebase.js"></script>
89 <script src="https://www.gstatic.com/firebasejs/3.1.0.firebaseio-app.js"></script>
90 <script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-auth.js"></script>
91 <script src="https://www.gstatic.com/firebasejs/3.1.0.firebaseio-database.js"></script>
92 <script src="/fireinit.js"></script>
93 <script type="text/javascript">
94 /* 2. 버튼을 눌렀을 경우 구글 인증 동작하게 */
95 $('#BTN_GOOGLE_LOGIN').click(function(){
96     var provider = new firebase.auth.GoogleAuthProvider();
97     firebase.auth().signInWithPopup(provider).then(function(result) {
98         }).catch(function(error) {
99             alert(error.message)
100        });
101    });
102
103 /* TODO 3. 데이터베이스 '/메세지' 쓰기 */
104
105 /* TODO 4. DB에서 '/메세지' 값 읽어오기 */
106
107 /* TODO 5. DB 메세지 값 변경 */
108
109 </script>
110
111 </body>
112 </html>
113
```

DB에 간단히 데이터 쓰는(write) 코드를 작성합니다

firebase

public

auth

css

database

- database_basic_step1.html
- database_basic_step2.html
- database_basic_step3.html**
- database_basic_step4.html
- database_basic_step5.html
- database_basic_step6.html

font

fonts

images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting.html
- index.html
- index2.html
- mail_auth.html

.DS_Store

.firebaserc

database.rules.json

firebase.json

database_basic_step3.html

```
86
87    <!-- 1. firebase SDK 링크 -->
88    <script src="https://www.gstatic.com/firebasejs/3.1.0.firebaseio.js"></script>
89    <script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-app.js"></script>
90    <script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-auth.js"></script>
91    <script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-database.js"></script>
92    <script src="/fireinit.js"></script>
93    <script type="text/javascript">
94        /* 2. 버튼을 눌렀을 경우 구글 인증 동작하게 */
95        $('#BTN_GOOGLE_LOGIN').click(function(){
96            var provider = new firebase.auth.GoogleAuthProvider();
97            firebase.auth().signInWithPopup(provider).then(function(result) {
98                }).catch(function(error) {
99                    alert(error.message)
100                });
101            });
102
103        /* 3. 데이터 베이스 '/메세지' 쓰기 */
104        firebase.database().ref('/메세지').set('안녕하세요!');
105
106        /* TODO 4. DB에서 '/메세지'값 읽어오기 */
107
108        /* TODO 5. DB 메세지 값 변경 */
109
110    </script>
111
112    </body>
113    </html>
114
```

Firebase Console

https://console.firebaseio.google.com/project/hellofirebase4web/database/data

승인

Firebase

HelloFirebase4Web 문서로 이동

Realtime Database

데이터 RULES USAGE

Analytics

DEVELOP

Auth

Database

Storage

Hosting

Remote Config

Test Lab

Crash

GROW

Notifications

Dynamic Links

EARN

AdMob

Spark 무료 업그레이드

Realtime Database

https://hellofirebase4web.firebaseio.com/

★ 기본 권한 규칙을 사용하면 사용자가 인증을 받아야 합니다.

자세히 알아보기 닫기

hellofirebase4web

메세지: "안녕하세요"

Firebase Console에서 실시간으로 변하는 데이터를 볼 수 있습니다

The screenshot shows the Firebase Realtime Database interface. On the left, there's a sidebar with various development tools like Auth, Database, Storage, and Hosting. The 'Database' option is selected. The main area is titled 'Realtime Database' and shows a single node under the root path: 'https://hellofirebase4web.firebaseio.com/'. This node contains a child node 'hellofirebase4web' which has a child node '메세지' (Message) with the value '안녕하세요' (Hello). A large red text overlay at the bottom states 'Firebase Console에서 실시간으로 변하는 데이터를 볼 수 있습니다' (You can see data changing in real-time in the Firebase Console).

Key의 이름 규칙

- UTF-8 인코딩 사용 (한글 및 유니코드 기호 사용 가능)
- 최대 768byte의 길이까지 허용됨
- Key의 이름에는 . \$ # [] / 일부 ASCII문자(0-31, 127)는 사용 불가능



If you create your own keys, they must be UTF-8 encoded, can be a maximum of 768 bytes, and cannot contain ., \$, #, [,], /, or ASCII control characters 0-31 or 127.

DB에서 데이터 읽어오기

`firebase.database.Reference.once()`

- <https://firebase.google.com/docs/reference/js/firebase.database.Reference#once>
- Sample Code

```
// Basic usage of .once() to read the data located at ref.  
ref.once('value')  
  .then(function(dataSnapshot) {  
    // handle read data.  
  });
```

firebase

public

auth

css

database

font

fonts

images

js

404.html

auth_firebase_ui.html

auth_google.html

auth_mail.html

auth_term.html

auth.html

database_basic.html

database.html

fireinit.js

fireinit.js.sample

hosting.html

index.html

index2.html

mail_auth.html

.DS_Store

.firebaserc

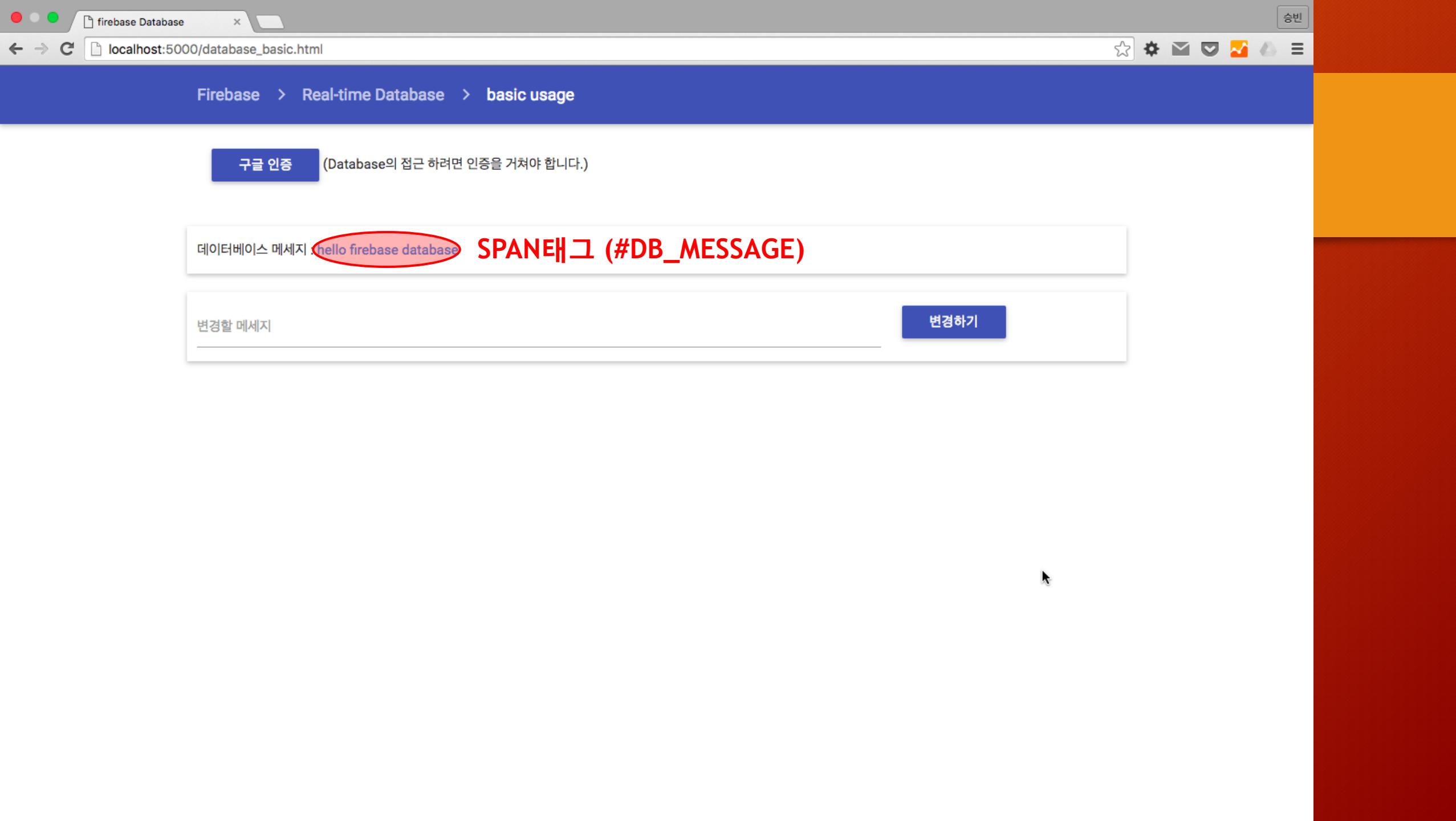
database.rules.json

firebase.json

database_basic.html

```
34         (Database의 기본 접근 하려면 인증을 거쳐야 합니다.)
35     </div>
36 </div>
37 <br><br>
38
39 <div class="row z-depth-1">
40     <div class="col l12 m12 s12">
41         <p>
42             데이터베이스 메세지 :
43             <!-- TODO 4. 상태 보기용 span 확인-->
44             <span id="DB_MESSAGE" class="blue-text">(Loading..)</span>
45         </p>
46     </div>
47 </div>
48
49             <!-- TODO 5. 메세지 수정 폼 확인하기 -->
50 <div class="row z-depth-1">
51     <div class="col l9 m9 s12 input-field">
52         <input id="INPUT_MESSAGE" type="text" class="validate">
53         <label for="INPUT_MESSAGE">변경할 메세지</label>
54     </div>
55     <div class="col l3 m3 s12 input-field">
56         <button id="BTN_UPDATE" type="button" class="btn indigo">변경하기</button>
57     </div>
58 </div>
59
60
61
62 </div>
```

DB에서 읽어올 데이터를 표시할 SPAN 태그



firebase

public

auth

css

database

- database_basic_step1.html
- database_basic_step2.html
- database_basic_step3.html**
- database_basic_step4.html
- database_basic_step5.html
- database_basic_step6.html

font

fonts

images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting.html
- index.html
- index2.html
- mail_auth.html

.DS_Store

.firebaserc

database.rules.json

firebase.json

database_basic_step3.html x database_basic_step3.html x

```
86
87     <!-- 1. firebase SDK 링크 -->
88     <script src="https://www.gstatic.com/firebasejs/3.1.0.firebaseio.js"></script>
89     <script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-app.js"></script>
90     <script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-auth.js"></script>
91     <script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-database.js"></script>
92     <script src="/fireinit.js"></script>
93     <script type="text/javascript">
94         /* 2. 버튼을 눌렀을 경우 구글 인증 동작하게 */
95         $('#BTN_GOOGLE_LOGIN').click(function(){
96             var provider = new firebase.auth.GoogleAuthProvider();
97             firebase.auth().signInWithPopup(provider).then(function(result) {
98                 }).catch(function(error) {
99                     alert(error.message)
100                });
101            });
102
103         /* 3. 데이터 베이스 '/메세지' 쓰기 */
104         firebase.database().ref('/메세지').set('안녕하세요!');
105
106         /* TODO 4. DB에서 '/메세지'값 읽어오기 */
107
108         /* TODO 5. DB 메세지 값 변경 */
109
110     </script>
111
112     </body>
113     </html>
114
```

DB에서 데이터 읽어오는 코드를 작성합니다

firebase

public

auth

css

database

- database_basic_step1.html
- database_basic_step2.html
- database_basic_step3.html
- database_basic_step4.html
- database_basic_step5.html
- database_basic_step6.html

font

fonts

images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting.html
- index.html
- index2.html
- mail_auth.html

.DS_Store

.firebaserc

database.rules.json

firebase.json

database_basic_step4.html

```
88 <script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-auth.js"></script>
89 <script src="https://www.gstatic.com/firebasejs/3.1.0.firebaseio.js"></script>
90 <script src="/fireinit.js"></script>
91 <script type="text/javascript">
92     /* 2. 버튼을 눌렀을 경우 구글 인증 동작하게 */
93     $('#BTN_GOOGLE_LOGIN').click(function(){
94         var provider = new firebase.auth.GoogleAuthProvider();
95         firebase.auth().signInWithPopup(provider).then(function(result) {
96             }).catch(function(error) {
97                 alert(error.message)
98             });
99         });
100
101     /* 3. 데이터 베이스 '/메세지' 쓰기 */
102     // firebase.database().ref('/메세지').set('안녕하세요!');
103
104     /* 4. DB에서 '/메세지'값 읽어오기 */
105     firebase.database().ref('/메세지').once('value', function(snapshot) {
106         var message = snapshot.val();
107         $("#DB_MESSAGE").text(message);
108     });
109
110     /* TODO 5. DB 메세지 값 변경 */
111
112     </script>
113
114 </body>
115
116 </html>
117
118
```

firebase

public

- auth
- css
- database
- font
- fonts
- images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html

database_basic.html

database.html

fireinit.js

fireinit.js.sample

hosting.html

index.html

index2.html

mail_auth.html

.DS_Store

.firebaserc

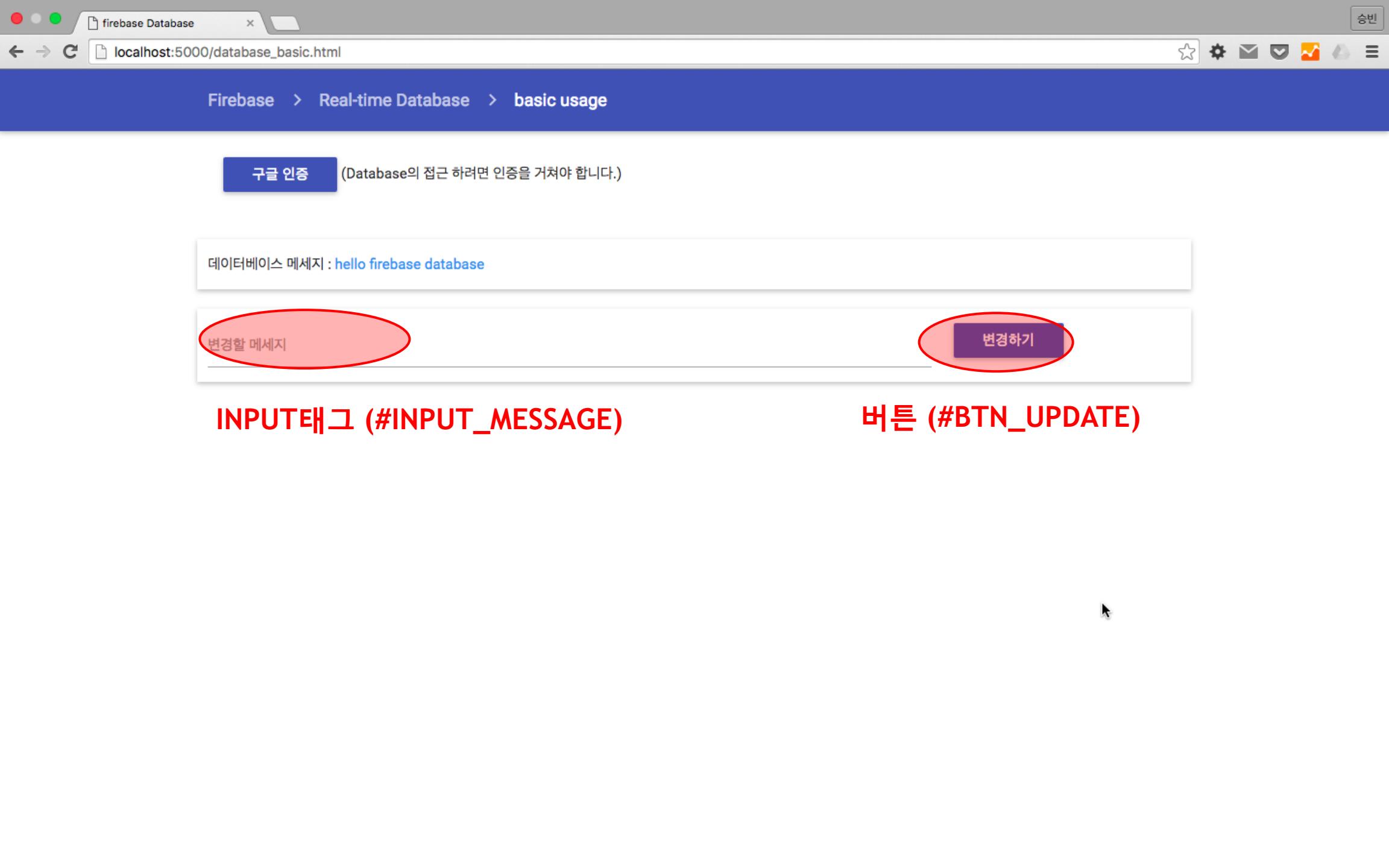
database.rules.json

firebase.json

database_basic.html

```
37     <br><br>
38
39     <div class="row z-depth-1">
40         <div class="col l12 m12 s12">
41             <p>
42                 데이터베이스 메세지 :
43                 <!-- TODO 4. 상태 보기용 span 확인-->
44                 <span id="DB_MESSAGE" class="blue-text">(Loading..)</span>
45             </p>
46         </div>
47     </div>
48
49     <!-- TODO 5. 메세지 수정 폼 확인하기 -->
50     <div class="row z-depth-1">
51         <div class="col l9 m9 s12 input-field">
52             <input id="INPUT_MESSAGE" type="text" class="validate">
53             <label for="INPUT_MESSAGE">변경할 메세지</label>
54         </div>
55         <div class="col l3 m3 s12 input-field">
56             <button id="BTN_UPDATE" type="button" class="btn indigo">변경하기</button>
57         </div>
58     </div>
59
60
61
62     </div>
63
64     <!-- 바닥글 영역 -->
65     <footer class="page-footer indigo">
66         <div class="container">
```

메시지 변경용 Form 태그 확인



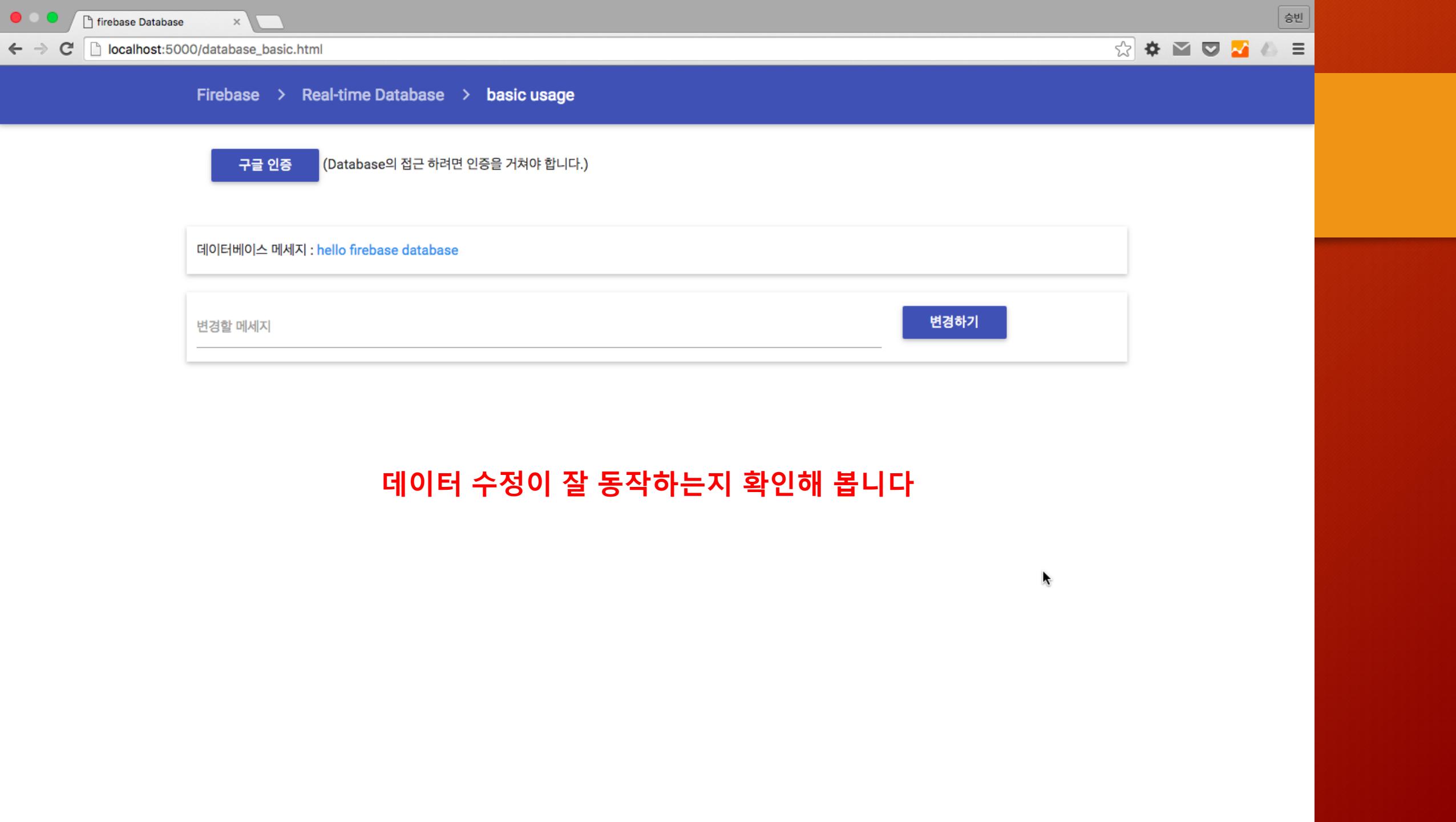
- firebase
- public
- auth
- css
- database
 - database_basic_step1.html
 - database_basic_step2.html
 - database_basic_step3.html
 - database_basic_step4.html
 - database_basic_step5.html
 - database_basic_step6.html
- font
- fonts
- images
- js
 - 404.html
 - auth_firebase_ui.html
 - auth_google.html
 - auth_mail.html
 - auth_term.html
 - auth.html
 - database_basic.html
 - database.html
 - fireinit.js
 - fireinit.js.sample
 - hosting.html
 - index.html
 - index2.html
 - mail_auth.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase.json

```
88 <script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-auth.js"></script>
89 <script src="https://www.gstatic.com/firebasejs/3.1.0.firebaseio.js"></script>
90 <script src="/fireinit.js"></script>
91 <script type="text/javascript">
92     /* 2. 버튼을 눌렀을 경우 구글 인증 동작하게 */
93     $('#BTN_GOOGLE_LOGIN').click(function(){
94         var provider = new firebase.auth.GoogleAuthProvider();
95         firebase.auth().signInWithPopup(provider).then(function(result) {
96             }).catch(function(error) {
97                 alert(error.message)
98             });
99         });
100
101     /* 3. 데이터 베이스 '/메세지' 쓰기 */
102     // firebase.database().ref('/메세지').set('안녕하세요!');
103
104     /* 4. DB에서 '/메세지'값 읽어오기 */
105     firebase.database().ref('/메세지').once('value', function(snapshot) {
106         var message = snapshot.val();
107         $("#DB_MESSAGE").text(message);
108     });
109
110     /* TODO 5. DB 메세지 값 변경 */
111
112     </script>
113
114 </body>
115 </html>
116
117
118
```

데이터를 수정하는 코드를 작성합니다

```
firebase
public
  auth
  css
database
  database_basic_step1.html
  database_basic_step2.html
  database_basic_step3.html
  database_basic_step4.html
  database_basic_step5.html
  database_basic_step6.html
font
fonts
images
js
  404.html
  auth_firebase_ui.html
  auth_google.html
  auth_mail.html
  auth_term.html
  auth.html
  database_basic.html
  database.html
  fireinit.js
  fireinit.js.sample
  hosting.html
  index.html
  index2.html
  mail_auth.html
.DS_Store
.firebaseio
database.rules.json
firebase.json
```

```
database_basic_step5.html *  database_basic_step5.html *  database_basic_step5.html *
94  /* 2. 버튼을 눌렀을 경우 구글 인증 동작하게 */
95  $('#BTN_GOOGLE_LOGIN').click(function(){
96    var provider = new firebase.auth.GoogleAuthProvider();
97    firebase.auth().signInWithPopup(provider).then(function(result) {
98      }).catch(function(error) {
99        alert(error.message)
100      });
101    });
102
103  /* 3. 데이터 베이스 '/메세지' 쓰기 */
104  // firebase.database().ref('/메세지').set('안녕하세요!');
105
106  /* 4. DB에서 '/메세지'값 읽어오기 */
107  firebase.database().ref('/메세지').once('value', function(snapshot) {
108    var message = snapshot.val();
109    $("#DB_MESSAGE").text(message);
110  });
111
112  /* 5. DB 메세지 값 변경 */
113  $("#BTN_UPDATE").click(function(){
114    var new_message = $("#INPUT_MESSAGE").val();
115    firebase.database().ref('/메세지').set(new_message);
116  });
117
118  </script>
119
120  </body>
121  </html>
122 }
```



데이터 수정이 잘 동작하는지 확인해 봅니다

firebase

public

auth

css

database

- database_basic_step1.html
- database_basic_step2.html
- database_basic_step3.html
- database_basic_step4.html
- database_basic_step5.html
- database_basic_step6.html

font

fonts

images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting.html
- index.html
- index2.html
- mail_auth.html

.DS_Store

.firebaserc

database.rules.json

firebase.json

database_basic_step6.html x database_basic_step6.html x

```
98     }).catch(function(error) {
99         alert(error.message)
100    });
101 });
102
103 /* TODO 3. 데이터 베이스 '/메세지' 쓰기 */
104 // firebase.database().ref('/메세지').set('안녕하세요!');
105
106 /* TODO 4. DB에서 '/메세지'값 읽어오기 */
107 // firebase.database().ref('/메세지').once('value', function(snapshot) {
108 firebase.database().ref('/메세지').on('value', function(snapshot) {
109     var message = snapshot.val();
110     $("#DB_MESSAGE").text(message);
111 });
112
113 /* TODO 5. DB 메세지 값 변경 */
114 $("#BTN_UPDATE").click(function(){
115     var new_message = $("#INPUT_MESSAGE").val();
116     var updates = {};
117     updates['/메세지'] = new_message;
118     // updates['/key_ref'] = value; 형식으로
119     firebase.database().ref().update(updates);
120 });
121
122 </script>
123
124 </body>
125 </html>
126
```

사실 수정은 update 메소드

데이터 쓰기 메소드

There are four methods for writing data to the Firebase Realtime Database:

Method	Common uses
<code>set()</code>	Write or replace data to a defined path, such as <code>users/<user-id>/<username></code> .
<code>push()</code>	Add to a list of data. Every time you call <code>push()</code> , Firebase generates a unique key that can also be used as a unique identifier, such as <code>user-posts/<user-id>/<unique-post-id></code> .
<code>update()</code>	Update some of the keys for a defined path without replacing all of the data.
<code>transaction()</code>	Update complex data that could be corrupted by concurrent updates.

★ Note: By default, read and write access to your database is restricted so only authenticated users can read or write data. To get started without setting up [Authentication](#), you can [configure your rules for public access](#). This does make your database open to anyone, even people not using your app, so be sure to restrict your database again when you set up authentication.

<https://firebase.google.com/docs/database/web/save-data>

firebase

public

auth

css

database

- database_basic_step1.html
- database_basic_step2.html
- database_basic_step3.html
- database_basic_step4.html
- database_basic_step5.html
- database_basic_step6.html

font

fonts

images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting.html
- index.html
- index2.html
- mail_auth.html

.DS_Store

.firebaserc

database.rules.json

firebase.json

database_basic_step6.html x database_basic_step6.html x

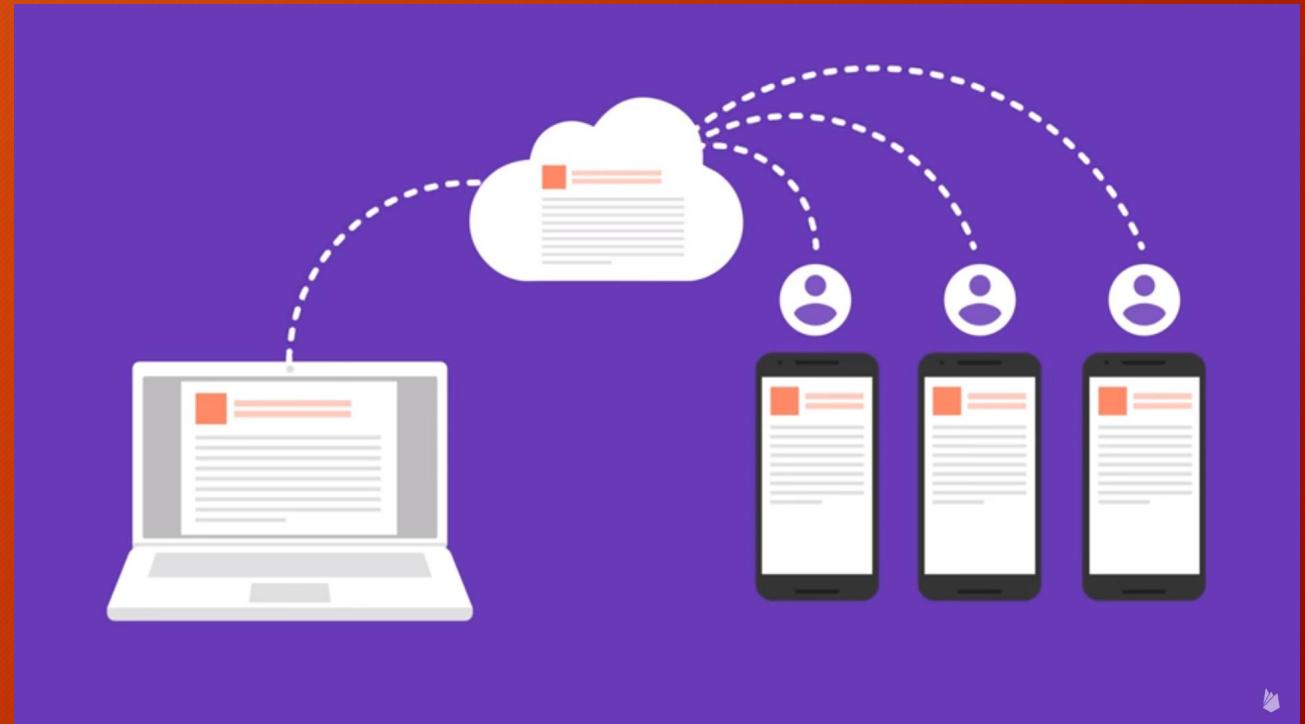
```
98     }).catch(function(error) {
99         alert(error.message)
100    });
101 });
102
103 /* TODO 3. 데이터베이스 '/메세지' 쓰기 */
104 // firebase.database().ref('/메세지').set('안녕하세요!');
105
106 /* TODO 4. DB에서 '/메세지' 값 읽어오기 */
107 // firebase.database().ref('/메세지').once('value', function(snapshot) {
108 firebase.database().ref('/메세지').on("value", function(snapshot) {
109     var message = snapshot.val();
110     $("#DB_MESSAGE").text(message);
111 });
112
113 /* TODO 5. DB 메세지 값 변경 */
114 $("#BTN_UPDATE").click(function(){
115     var new_message = $("#INPUT_MESSAGE").val();
116     var updates = {};
117     updates['/메세지'] = new_message;
118     // updates['/key_ref'] = value; 형식으로
119     firebase.database().ref().update(updates);
120 });
121
122 </script>
123
124 </body>
125 </html>
126
```

once -> on로 변경해봅니다

CRLF UTF-8 HTML 2 updates

Real-Time Database

- Database와 connection
- 데이터가 변하면 connection된 모든 사용자에게 notification이 감 (board casting)
- 웹에서는 Firebase DB SDK가 변화를 감지해서 특정 메소드를 수행



데이터 읽는 방법

- 기본적으로 이벤트 리스너를 통해서만 가능

(앞에서 사용한 once()는 내부적으로 on()을 한번 수행한 후 리스너를 제거하는 off()를 수행하도록 구현되어 있음)

- 데이터 변화 감지 메소드

`firebase.database.Reference.on()`

- <https://firebase.google.com/docs/reference/js/firebase.database.Reference#on>

- Sample Code

```
firebase.database().ref('posts/' + postId + '/starCount').on('value', function(snapshot) {  
  updateStarCount(postElement, snapshot.val());  
});
```

데이터 읽는 방법

Listen for events

You can listen for these types of events that retrieve data:

Event	Typical usage
<code>value</code>	Read and listen for changes to the entire contents of a path.
<code>child_added</code>	Retrieve lists of items or listen for additions to a list of items. Suggested use with <code>child_changed</code> and <code>child_removed</code> to monitor changes to lists.
<code>child_changed</code>	Listen for changes to the items in a list. Use with <code>child_added</code> and <code>child_removed</code> to monitor changes to lists.
<code>child_removed</code>	Listen for items being removed from a list. Use with <code>child_added</code> and <code>child_changed</code> to monitor changes to lists.
<code>child_moved</code>	Listen for changes to the order of items in an ordered list. <code>child_moved</code> events always follow the <code>child_changed</code> event that caused the item's order to change (based on your current order-by method).

To add an event listener, use the `on()` method.

Examples

Handle a new value:

```
ref.on('value', function(dataSnapshot) {  
  ...  
});
```

Handle a new child:

```
ref.on('child_added', function(childSnapshot, prevChildKey) {  
  ...  
});
```

Handle child removal:

```
ref.on('child_removed', function(oldChildSnapshot) {  
  ...  
});
```

Handle child data changes:

```
ref.on('child_changed', function(childSnapshot, prevChildKey) {  
  ...  
});
```

Handle child ordering changes:

```
ref.on('child_moved', function(childSnapshot, prevChildKey) {  
  ...  
});
```

데이터 관리 메소드

NoSQL은 RDBMS와는 달리 SQL 쿼리를 사용하지 않기에

정렬(Sort) 관련 메소드

Sorting and filtering data

You can use the Realtime Database `Query` class to retrieve data sorted by key, by value, or by value of a child. You can also filter the sorted result to a specific number of results or a range of keys or values.

★ Note: Filtering and sorting can be expensive, especially when done on the client. If your app uses queries, define the `.indexOn` rule to index those keys on the server and improve query performance as described in [Indexing Your Data](#).

Sort data

To retrieve sorted data, start by specifying one of the order-by methods to determine how results are ordered:

Method	Usage
<code>orderByChild()</code>	Order results by the value of a specified child key.
<code>orderByKey()</code>	Order results by child keys.
<code>orderByValue()</code>	Order results by child values.

You can only use **one** order-by method at a time. Calling an order-by method multiple times in the same query throws an error.

필터링 관련 메소드

Filtering data

To filter data, you can combine any of the limit or range methods with an order-by method when constructing a query.

Method	Usage
<code>limitToFirst()</code>	Sets the maximum number of items to return from the beginning of the ordered list of results.
<code>limitToLast()</code>	Sets the maximum number of items to return from the end of the ordered list of results.
<code>startAt()</code>	Return items greater than or equal to the specified key or value, depending on the order-by method chosen.
<code>endAt()</code>	Return items less than or equal to the specified key or value, depending on the order-by method chosen.
<code>equalTo()</code>	Return items equal to the specified key or value, depending on the order-by method chosen.

Unlike the order-by methods, you can combine multiple limit or range functions. For example, you can combine the `startAt()` and `endAt()` methods to limit the results to a specified range of values.

그 외 기능

- 타임 스탬프 (서버의 현재 시각)
- 사용자 접속 끊김 감지

`firebase.database.Reference.onDisconnect()`

- <https://firebase.google.com/docs/reference/js/firebase.database.Reference#onDisconnect>
- Sample Code

```
var userLastOnlineRef = firebase.database().ref("users/joe/lastOnline");
userLastOnlineRef.onDisconnect().set(firebase.database.ServerValue.TIMESTAMP);
```

[Tip] SQL 변환 참고 자료

- 일반적으로 자주 쓰이는 형태의 쿼리 변환

(주의할 점 : 애초에 설계를 NoSQL의 정규화 방법으로
하는 편이 좋음)

- <https://firebase.googleblog.com/2013/10/queries-part-1-common-sql-queries.html>

So let's jump in. Here's what we're going to cover today:

- Select a user by ID (WHERE id = x)
- Find a user by email address (WHERE email = x)
- Get messages posted yesterday (WHERE timestamp BETWEEN x AND y)
- Paginate through widgets (LIMIT 10 OFFSET 10)
- Join records using an id (FROM table1 JOIN table2 USING id)
- More tools to come

실시간 접속자 만들기

실시간 접속자의 DB 구조



- 사용자들 정보 (/onUsers)
- 사용자 정보 (/onUsers/[userUID])
 - 사용자 UID를 Key로 사용
- 사용자의 닉네임 (/onUsers/[userUID]/username)
- 사용자의 접속 여부 (/onUsers/[userUID]/siteOn)
 - 0 : 사이트에 접속하지 않음
 - 1 : 사이트에 접속함

다음 파일을 편집기로 열어주세요

[project folder]/public/database_siteon.html

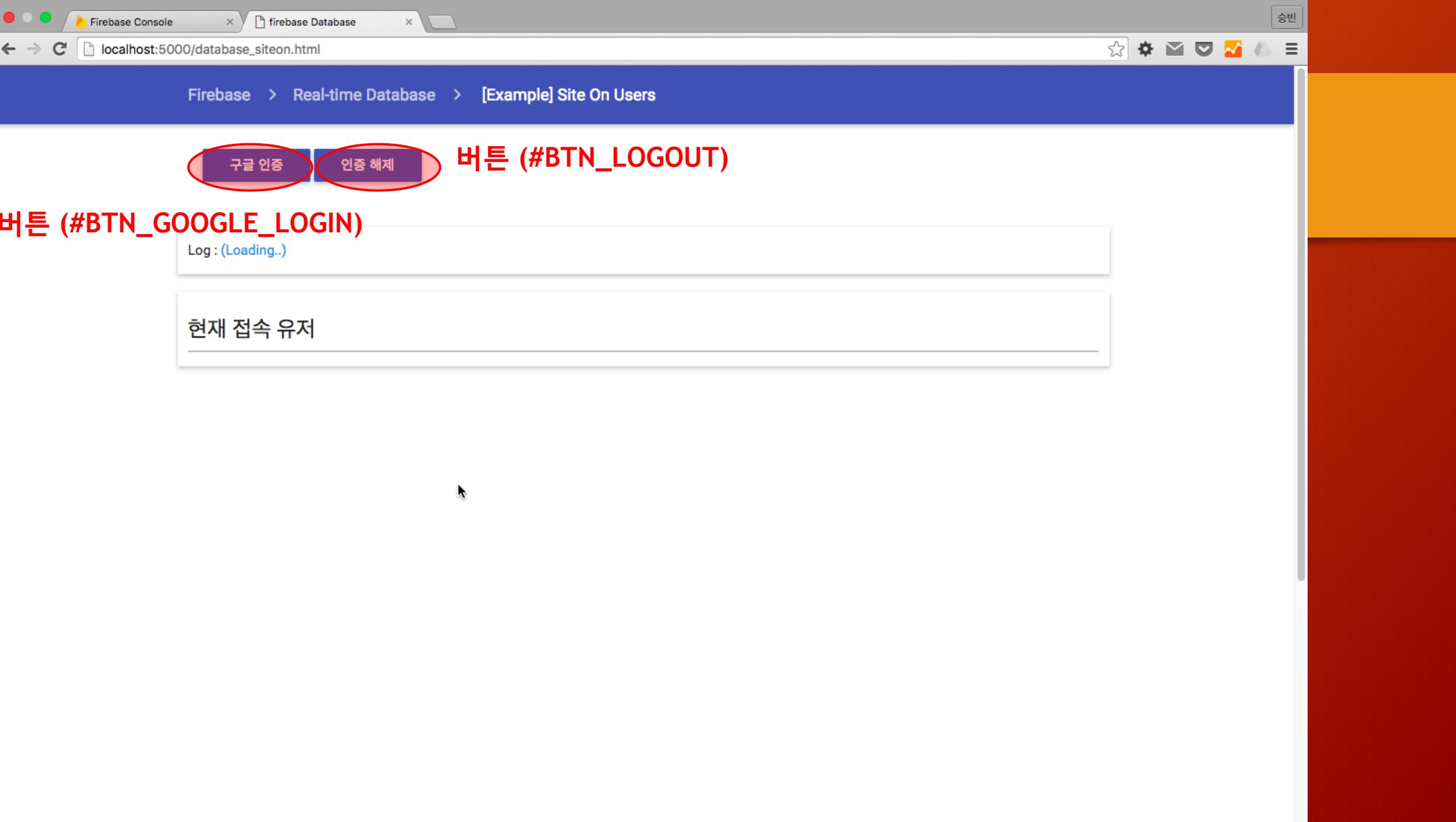
1. firebase SDK 링크 + 구글 인증
2. 인증상태 감지하기
3. 인증하면 접속 상태 DB에 기록하기
4. 접속상태 DB 감지 리스너 작성
5. 접속자를 화면에 보여주기
6. 페이지를 떠날 시 상태 변경
7. 접속이 끊길 시 상태 변경

```
database_basic_step2.html
database_basic_step3.html
database_basic_step4.html
database_basic_step5.html
database_basic_step6.html
database_siteon_step1.html
database_siteon_step2.html
database_siteon_step3.html
database_siteon_step4.html
database_siteon_step5.html
database_siteon_step6.html
database_siteon_step7.html
font
fonts
images
js
404.html
auth_firebase_ui.html
auth_google.html
auth_mail.html
auth_term.html
auth.html
database_basic.html
database_siteon.html
database.html
fireinit.js
fireinit.js.sample
hosting.html
index.html
index2.html
mail_auth.html
.DS_Store
.firebaseio
database.rules.json
firebase.json

database_siteon.html x

21         <a href="/database.html" class="breadcrumb">Real-time Database</a>
22             <a href="#" class="breadcrumb"> [Example] Site On Users </a>
23         </div>
24     </div>
25 </nav>
26
27 <!-- 내용 부분 -->
28 <div class="container" id="CONTENTS">
29     <br>
30     <!-- TODO 1. 인증 버튼 확인 -->
31     <div class="row" style="margin: 0; padding: 5px;">
32         <div class="col l12 m12 s12">
33             <button id="BTN_GOOGLE_LOGIN" type="button" class="btn indigo">구글 인증</button>
34             <button id="BTN_LOGOUT" type="button" class="btn indigo">인증 해제</button>
35         </div>
36     </div>
37     <br><br>
38
39     <div class="row z-depth-1">
40         <div class="col l12 m12 s12">
41             <p>
42                 Log :
43                 <!-- TODO 2. 인증 상태 보기용 span 확인-->
44                 <span id="LOG_MESSAGE" class="blue-text">(Loading..)</span>
45             </p>
46         </div>
47     </div>
48
49     <!-- TODO 5. 접속 유저 ol박스 확인 -->
50     ..
```

구글 인증과 해제 버튼 확인



```
// <!-- ////////////////////////////// javascript 스크립트 영역 //////////////////////////////-->
78
79    <!--Import jQuery before materialize.js-->
80    <script type="text/javascript" src="https://code.jquery.com/jquery-2.1.1.min.js"></script>
81    <script type="text/javascript" src="/js/materialize.min.js"></script>
82
83    <!-- TODO 1. firebase SDK 링크 -->
84    <script type="text/javascript">
85        /* TODO 1. 버튼을 눌렀을 경우 구글 인증 동작하게 */
86
87        /* TODO 1. 인증 해제 */
88
89        /* TODO 2. 접속한 경우 Database의 상태 변경 */
90        firebase.auth().onAuthStateChanged(function(user) {
91            if (user) {
92                // TODO 3. 디비에 접속 하면 상태 변경 (등록이 안된 상태면 새로 쓰기)
93                // TODO 4. 상태가 바뀔 때를 감지할 리스너(JS옵저버) 장착
94                // TODO 7. 디비에 접속 끊길때 이벤트 감지하하면 특정값 변경 (siteOn 상태를 0으로 변경)
95            } else {
96                // TODO 2. 인증이 해제 되거나 인증되지 않은 경우
97            }
98        });
99
100       // TODO 6. 페이지 떠나게 되면
101
102       </script>
103
104   </body>
105
106   </html>
```

SDK 링크 + 인증까지 코드 작성합니다

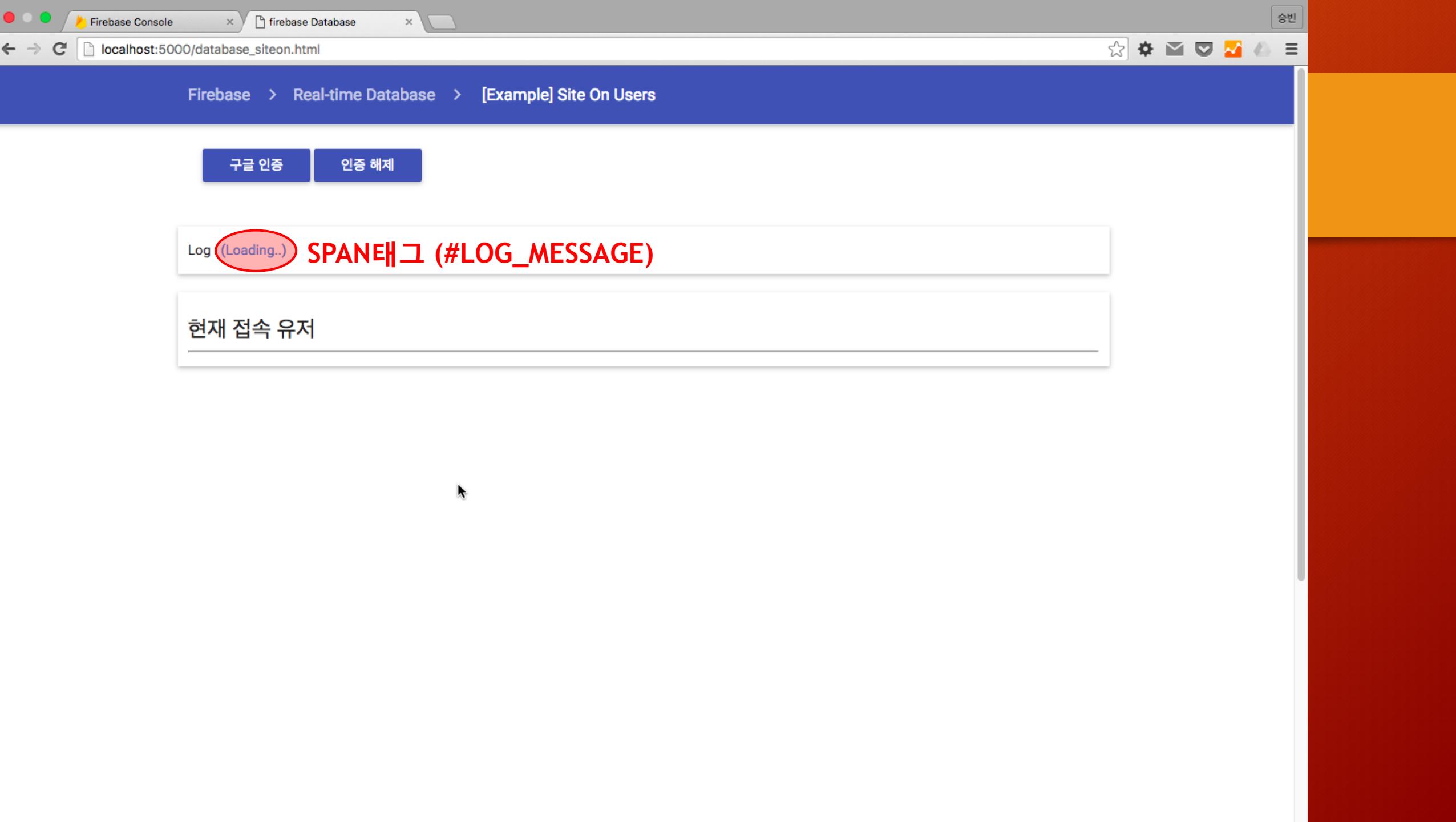
- > font
- > fonts
- > images
- > js
 - 404.html
 - auth_firebase_ui.html
 - auth_google.html
 - auth_mail.html
 - auth_term.html
 - auth.html
 - database_basic.html
 - database_siteon.html
 - database.html
 - fireinit.js
 - fireinit.js.sample
 - hosting.html
 - index.html
 - index2.html
 - mail_auth.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase.json

```
firebase database_siteon.html * database_siteon_step1.html x
83     <!-- 1. firebase SDK 링크 -->
84     <script src="https://www.gstatic.com/firebasejs/3.1.0.firebaseio.js"></script>
85     <script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-app.js"></script>
86     <script src="https://www.gstatic.com/firebasejs/3.1.0/firebase-auth.js"></script>
87     <script src="https://www.gstatic.com/firebasejs/3.1.0.firebaseio-database.js"></script>
88     <script src="/fireinit.js"></script>
89     <script type="text/javascript">
90         /* 1. 버튼을 눌렀을 경우 구글 인증 동작하게 */
91         $('#BTN_GOOGLE_LOGIN').click(function(){
92             var provider = new firebase.auth.GoogleAuthProvider();
93             firebase.auth().signInWithPopup(provider).then(function(result) {
94             }).catch(function(error) {
95                 alert(error.message)
96             });
97         });
98
99         /* 1. 인증 해제 */
100        $('#BTN_LOGOUT').click(function(){
101            // TODO 4. DB의 변화를 더이상 감지 하지 않음
102            // TODO 3. 접속상태값을 변경
103            // 인증 해제
104            firebase.auth().signOut().then(function() {
105                alert("인증이 해제되었습니다.");
106            }, function(error) {
107                alert(error.message);
108            });
109        });
110
111        /* TODO 2. 접속한 경우 Database의 상태 변경 */
112        var uid = null; // 유저의 uid를 저장해둔 저연 변수
```

firebase
public
auth
css
database
database_basic_step1.html
database_basic_step2.html
database_basic_step3.html
database_basic_step4.html
database_basic_step5.html
database_basic_step6.html
database_siteon_step1.html
database_siteon_step2.html
database_siteon_step3.html
database_siteon_step4.html
database_siteon_step5.html
database_siteon_step6.html
database_siteon_step7.html
font
fonts
images
js
404.html
auth_firebase_ui.html
auth_google.html
auth_mail.html
auth_term.html
auth.html
database_basic.html
database_siteon.html
database.html
fireinit.js
fireinit.js.sample
hosting.html
index.html

```
35      </div>
36    </div>
37    <br><br>
38
39    <div class="row z-depth-1">
40      <div class="col l12 m12 s12">
41        <p>
42          Log :
43          <!-- 2. 인증 상태 보기용 span 확인-->
44          <span id="LOG_MESSAGE" class="blue-text">(Loading..)</span>
45        </p>
46      </div>
47    </div>
48
49    <!-- TODO 5. 접속 유저 ol박스 확인 -->
50    <div class="row z-depth-1">
51      <div class="col l12 m12 s12 input-field">
52        <h5>현재 접속 유저</h5>
53        <hr>
54        <ol id="SITE_ON_USERS"><!-- 접속한 USERS 목록을 띄울 박스 --></ol>
55      </div>
56    </div>
57
58  </div>
59
60  <!-- 바닥글 영역 -->
61  <footer class="page-footer indigo">
62    <div class="container">
63      <div class="row center">
64        <a href="/database.html" class="btn white indigo-text">돌아가기</a>
```

인증 상태를 표시할 SPAN태그 확인



firebase

public

auth

css

database

- database_basic_step1.html
- database_basic_step2.html
- database_basic_step3.html
- database_basic_step4.html
- database_basic_step5.html
- database_basic_step6.html
- database_siteon_step1.html
- database_siteon_step2.html
- database_siteon_step3.html
- database_siteon_step4.html
- database_siteon_step5.html
- database_siteon_step6.html
- database_siteon_step7.html

font

fonts

images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database_siteon.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting.html
- index.html

database_siteon_step2.html x

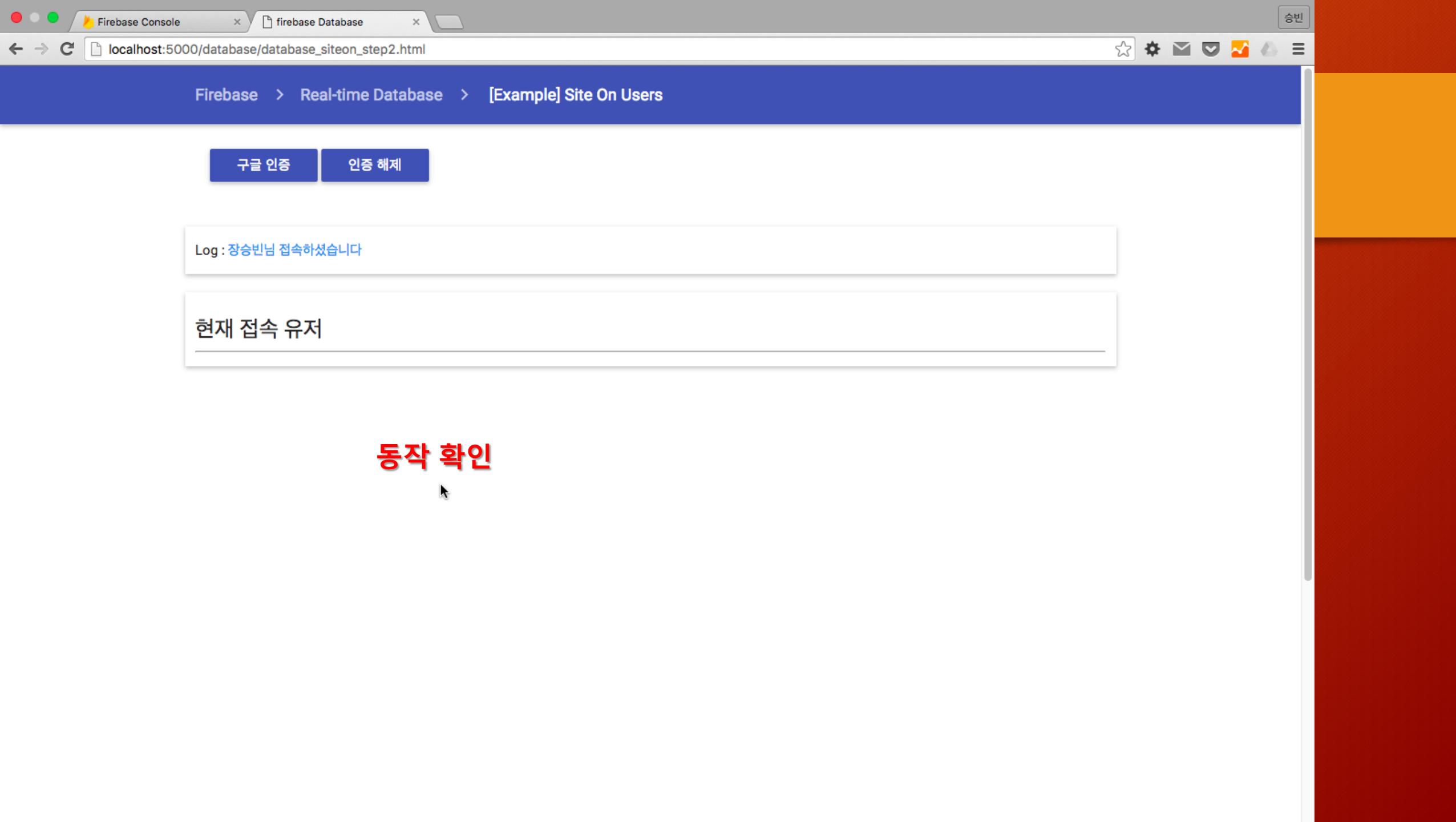
```
104     firebase.auth().signOut().then(function() {
105         alert("인증이 해제되었습니다.");
106     }, function(error) {
107         alert(error.message);
108     });
109 });
110
111 /* 2. 접속한 경우 Database의 상태 변경 */
112 var _uid = null; // 유저의 uid를 저장해둘 전역 변수
113 firebase.auth().onAuthStateChanged(function(user) {
114     if (user) {
115         _uid = user.uid;
116         $("#LOG_MESSAGE").text(user.displayName+"님 접속하셨습니다");
117         // TODO 3. 디비에 접속 하면 상태 변경 (등록이 안된 상태면 새로 쓰기)
118         // TODO 4. 상태가 바뀔 때를 감지할 리스너(JS옵저버) 장착
119         // TODO 7. 디비에 접속 끊길때 이벤트 감지하하면 특정값 변경 (siteOn 상태를 0으로 변경)
120     } else { // 2. 인증이 해제 되거나 인증되지 않은 경우
121         $("#LOG_MESSAGE").text("(인증하셔야 볼 수 있습니다)");
122         _uid = null;
123     }
124 });
125
126 // TODO 6. 페이지 떠나게 되면
127
128 </script>
129
130 </body>
131 </html>
132
```

인증 상태 변경 내용 코드 작성

firebase
public
auth
css
database
database_basic_step1.html
database_basic_step2.html
database_basic_step3.html
database_basic_step4.html
database_basic_step5.html
database_basic_step6.html
database_siteon_step1.html
database_siteon_step2.html
database_siteon_step3.html
database_siteon_step4.html
database_siteon_step5.html
database_siteon_step6.html
database_siteon_step7.html
font
fonts
images
js
404.html
auth_firebase_ui.html
auth_google.html
auth_mail.html
auth_term.html
auth.html
database_basic.html
database_siteon.html
database.html
fireinit.js
fireinit.js.sample
hosting.html
index.html

```
104     firebase.auth().signOut().then(function() {
105       alert("인증이 해제되었습니다.");
106     }, function(error) {
107       alert(error.message);
108     });
109   });
110
111 /* 2. 접속한 경우 Database의 상태 변경 */
112 var _uid = null; // 유저의 uid를 저장해둘 전역 변수
113 firebase.auth().onAuthStateChanged(function(user) {
114   if (user) {
115     _uid = user.uid;
116     $("#LOG_MESSAGE").text(user.displayName+"님 접속하셨습니다");
117     // TODO 3. 디비에 접속 하면 상태 변경 (등록이 안된 상태면 새로 쓰기)
118     // TODO 4. 상태가 바뀔 때를 감지할 리스너(JS옵저버) 장착
119     // TODO 7. 디비에 접속 끊길때 이벤트 감지하하면 특정값 변경 (siteOn 상태를 0으로 변경)
120   } else { // 2. 인증이 해제 되거나 인증되지 않은 경우
121     $("#LOG_MESSAGE").text("(인증하셔야 볼 수 있습니다)");
122     _uid = null;
123   }
124 });
125
126 // TODO 6. 페이지 떠나게 되면
127
128 </script>
129
130 </body>
131 </html>
132
```

인증 완료시 uid를 저장할 임시 전역 변수를 추가



Firebase > Real-time Database > [Example] Site On Users

구글 인증

인증 해제

Log : 장승빈님 접속하셨습니다

현재 접속 유저

동작 확인

firebase

public

auth

css

database

- database_basic_step1.html
- database_basic_step2.html
- database_basic_step3.html
- database_basic_step4.html
- database_basic_step5.html
- database_basic_step6.html
- database_siteon_step1.html
- database_siteon_step2.html
- database_siteon_step3.html
- database_siteon_step4.html
- database_siteon_step5.html
- database_siteon_step6.html
- database_siteon_step7.html

font

fonts

images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database_siteon.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting.html
- index.html

database_siteon_step2.html x

```
104     firebase.auth().signOut().then(function() {
105         alert("인증이 해제되었습니다.");
106     }, function(error) {
107         alert(error.message);
108     });
109 });
110
111 /* 2. 접속한 경우 Database의 상태 변경 */
112 var _uid = null; // 유저의 uid를 저장해둘 전역 변수
113 firebase.auth().onAuthStateChanged(function(user) {
114     if (user) {
115         _uid = user.uid;
116         $("#LOG_MESSAGE").text(user.displayName+"님 접속하셨습니다");
117         // TODO 3. 디비에 접속 하면 상태 변경 (등록이 안된 상태면 새로 쓰기)
118         // TODO 4. 상태가 바뀔 때를 감지할 리스너(JS옵저버) 장착
119         // TODO 7. 디비에 접속 끊길때 이벤트 감지하하면 특정값 변경 (siteOn 상태를 0으로 변경)
120     } else { // 2. 인증이 해제 되거나 인증되지 않은 경우
121         $("#LOG_MESSAGE").text("(인증하셔야 볼 수 있습니다)");
122         _uid = null;
123     }
124 });
125
126 // TODO 6. 페이지 떠나게 되면
127
128 </script>
129
130 </body>
131 </html>
132 }
```

DB에 접속 상태를 알리는 코드 작성

firebase

public

auth

css

database

- database_basic_step1.html
- database_basic_step2.html
- database_basic_step3.html
- database_basic_step4.html
- database_basic_step5.html
- database_basic_step6.html
- database_siteon_step1.html
- database_siteon_step2.html
- database_siteon_step3.html**
- database_siteon_step4.html
- database_siteon_step5.html
- database_siteon_step6.html
- database_siteon_step7.html

font

fonts

images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database_siteon.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting.html
- index.html

database_siteon_step3.html x

```
106         alert("인증이 해제되었습니다.");
107     }, function(error) {
108         alert(error.message);
109     });
110 });
111
112 /* 2. 접속한 경우 Database의 상태 변경 */
113 var _uid = null; // 유저의 uid를 저장해둘 전역 변수
114 firebase.auth().onAuthStateChanged(function(user) {
115     if (user) {
116         _uid = user.uid;
117         $("#LOG_MESSAGE").text(user.displayName+"님 접속하셨습니다");
118         // 3. 디비에 접속 하면 상태 변경 (등록이 안된 상태면 새로 쓰기)
119         firebase.database().ref('/onUsers/' + user.uid).set({
120             "username": user.displayName,
121             "siteOn": 1
122         });
123         // TODO 4. 상태가 바뀔 때를 감지할 리스너(JS옵저버) 장착
124         // TODO 7. 디비에 접속 끊길때 이벤트 감지하하면 특정값 변경 (siteOn 상태를 0으로 변경)
125     } else { // 2. 인증이 해제 되거나 인증되지 않은 경우
126         $("#LOG_MESSAGE").text("(인증하셔야 볼 수 있습니다)");
127         _uid = null;
128     }
129 });
130 </script>
131
132 </body>
133 </html>
134 }
```

```
95         alert(error.message)
96     });
97 });
98
99     /* 1. 인증 해제 */
100    $('#BTN_LOGOUT').click(function(){
101        // TODO 4. DB의 변화를 더이상 감지 하지 않음
102        // 3. 접속상태값을 변경
103        firebase.database().ref("/onUsers/"+_uid+"/siteOn").set(0);
104        // 인증 해제
105        firebase.auth().signOut().then(function() {
106            alert("인증이 해제되었습니다.");
107        }, function(error) {
108            alert(error.message);
109        });
110    });
111
112     /* 2. 접속한 경우 Database의 상태 변경 */
113    var _uid = null; // 유저의 uid를 저장해둘 전역 변수
114    firebase.auth().onAuthStateChanged(function(user) {
115        if (user) {
116            _uid = user.uid;
117            $("#LOG_MESSAGE").text(user.displayName+"님 접속하셨습니다");
118            // 3. 디비에 접속 하면 상태 변경 (등록이 안된 상태면 새로 쓰기)
119            firebase.database().ref('/onUsers/' + user.uid).set({
120                "username": user.displayName,
121                "siteOn": 1
122            });
123            // TODO 4. 상태가 바뀔 때를 감지할 리스너(JS옵저버) 장착
124            // TODO 7. 디비에 접속 끊길때 이벤트 감지하하면 특정값 변경 (siteOn 상태를 0으로 변경)
125        } else { // 2. 인증이 해제 되거나 인증되지 않은 경우
126            $("#LOG_MESSAGE").text("(인증하셔야 볼 수 있습니다)");
127            _uid = null;
128        }
129    });
130
131    // 접속이 끊기면 firebase.auth().onAuthStateChanged()에서
132    // 데이터베이스에서 해당 유저의 정보가 삭제되는 것을 막기 위해
133    // firebase.auth().signOut()을 실행하는 코드를 넣어야 합니다.
134
```

Firebase Console firebase Database 승인

<https://console.firebaseio.google.com/project/hellofirebase4web/database/data>

Firebase HelloFirebase4Web 문서로 이동

Realtime Database

데이터 RULES USAGE

Analytics

DEVELOP

Auth

Database

Storage

Hosting

Remote Config

Test Lab

Crash

GROW

Notifications

Dynamic Links

EARN

AdMob

Spark 무료 업그레이드

Realtime Database

데이터 RULES USAGE

https://hellofirebase4web.firebaseio.com/

★ 기본 권한 규칙을 사용하면 사용자가 인증을 받아야 합니다.

자세히 알아보기 닫기

hellofirebase4web

onUsers

mbLfZ20MsyP2qM0e90214yCVC8F3

siteOn: 1

username: "장승빈"

메세지: "hello firebase database"

인증시 변화 확인

The screenshot shows the Firebase Realtime Database interface. On the left, there's a sidebar with various project settings and services like Analytics, Storage, and Hosting. The main area is titled 'Realtime Database' and shows a tree structure of database references. A specific node under 'onUsers' is highlighted with a green border, containing child nodes 'siteOn' and 'username'. The value for 'username' is set to '장승빈'. Below this node, a message in Korean is displayed: '메세지: "hello firebase database"'. At the top of the main area, there's a warning message in Korean: '★ 기본 권한 규칙을 사용하면 사용자가 인증을 받아야 합니다.' (★ If you use basic permission rules, users must be authenticated). To the right of this message, there are links for '자세히 알아보기' (Learn more) and '닫기' (Close). A large red text overlay '인증시 변화 확인' (Check for changes when authenticating) is positioned on the right side of the database preview window.

Firebase Console firebase Database 승인

https://console.firebaseio.google.com/project/hellofirebase4web/database/data

Firebase HelloFirebase4Web 문서로 이동

Realtime Database

데이터 RULES USAGE

Analytics

DEVELOP

Auth

Database

Storage

Hosting

Remote Config

Test Lab

Crash

GROW

Notifications

Dynamic Links

EARN

AdMob

Spark 무료 업그레이드

Realtime Database

https://hellofirebase4web.firebaseio.com/

★ 기본 권한 규칙을 사용하면 사용자가 인증을 받아야 합니다.

자세히 알아보기 닫기

hellofirebase4web + X

onUsers

mbLfZ20MsyP2qM0e90214yCVC8F3

siteOn: 0

username: "장승빈"

메세지: "hello firebase database"

해제시 변화 확인

firebase

public

auth

css

database

- database_basic_step1.html
- database_basic_step2.html
- database_basic_step3.html
- database_basic_step4.html
- database_basic_step5.html
- database_basic_step6.html
- database_siteon_step1.html
- database_siteon_step2.html
- database_siteon_step3.html**
- database_siteon_step4.html
- database_siteon_step5.html
- database_siteon_step6.html
- database_siteon_step7.html

font

fonts

images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database_siteon.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting.html
- index.html

database_siteon_step3.html x

```
106     alert("인증이 해제되었습니다.");
107 }, function(error) {
108     alert(error.message);
109 });
110 });

112 /* 2. 접속한 경우 Database의 상태 변경 */
113 var _uid = null; // 유저의 uid를 저장해둘 전역 변수
114 firebase.auth().onAuthStateChanged(function(user) {
115     if (user) {
116         _uid = user.uid;
117         $("#LOG_MESSAGE").text(user.displayName+"님 접속하셨습니다");
118         // 3. 디비에 접속 하면 상태 변경 (등록이 안된 상태면 새로 쓰기)
119         firebase.database().ref('/onUsers/' + user.uid).set({"username": user.displayName, "siteOn": true});
120         // TODO 4. 상태가 바뀔 때를 감지할 리스너(JS옵저버) 장착
121         // TODO 7. 디비에 접속 끊길때 이벤트 감지하하면 특정값 변경 (siteOn 상태를 0으로 변경)
122     } else { // 2. 인증이 해제 되거나 인증되지 않은 경우
123         $("#LOG_MESSAGE").text("(인증하셔야 볼 수 있습니다)");
124         _uid = null;
125     }
126 });
127
128 // TODO 6. 페이지 떠나게 되면
129
130 </script>
131
132 </body>
133 </html>
134
```

누군가 접속 상태가 변할때마다 화면 갱신을
하기 위한 리스너 코드를 작성

```
firebase
  public
    auth
    css
  database
    database_basic_step1.html
    database_basic_step2.html
    database_basic_step3.html
    database_basic_step4.html
    database_basic_step5.html
    database_basic_step6.html
    database_siteon_step1.html
    database_siteon_step2.html
    database_siteon_step3.html
  database_siteon_step4.html
    database_siteon_step5.html
    database_siteon_step6.html
    database_siteon_step7.html
  font
  fonts
  images
  js
    404.html
    auth_firebase_ui.html
    auth_google.html
    auth_mail.html
    auth_term.html
    auth.html
    database_basic.html
    database_siteon.html
    database.html
    fireinit.js
    fireinit.js.sample
    hosting.html
  index.html

database_siteon_step3.html x database_siteon_step4.html x

111      });
112  });
113
114  /* 2. 접속한 경우 Database의 상태 변경 */
115  var _uid = null; // 유저의 uid를 저장해둘 전역 변수
116  firebase.auth().onAuthStateChanged(function(user) {
117    if (user) {
118      _uid = user.uid;
119      $("#LOG_MESSAGE").text(user.displayName+"님 접속하셨습니다");
120      // 3. 디비에 접속 하면 상태 변경 (등록이 안된 상태면 새로 쓰기)
121      firebase.database().ref('/onUsers/' + user.uid).set({"username": user.displayName, "siteOn": true});
122      // 4. 상태가 바뀔 때를 감지할 리스너(JS옵저버) 장착
123      firebase.database().ref('/onUsers/').on('value', function(snapshot_users) {
124        // TODO 5. 유저를 하나씩 읽어오기
125      });
126      // TODO 7. 디비에 접속 끊길때 이벤트 감지하하면 특정값 변경 (siteOn 상태를 0으로 변경)
127    } else { // 2. 인증이 해제 되거나 인증되지 않은 경우
128      $("#LOG_MESSAGE").text("(인증하셔야 볼 수 있습니다)");
129      _uid = null;
130    }
131  });
132
133  // TODO 6. 페이지 떠나게 되면
134
135  </script>
136
137  </body>
138  </html>
139
```

firebase
public
auth
css
database
database_basic_step1.html
database_basic_step2.html
database_basic_step3.html
database_basic_step4.html
database_basic_step5.html
database_basic_step6.html
database_siteon_step1.html
database_siteon_step2.html
database_siteon_step3.html
database_siteon_step4.html
database_siteon_step5.html
database_siteon_step6.html
database_siteon_step7.html
font
fonts
images
js
404.html
auth_firebase_ui.html
auth_google.html
auth_mail.html
auth_term.html
auth.html
database_basic.html
database_siteon.html
database.html
fireinit.js
fireinit.js.sample
hosting.html
index.html

```
97  });
98
99  /* 1. 인증 해제 */
100 $( '#BTN_LOGOUT' ).click(function(){
101     // TODO 4. DB의 변화를 더이상 감지 하지 않음
102     // 3. 접속상태값을 변경
103     firebase.database().ref("/onUsers/"+_uid+"/siteOn").set(0);
104     // 인증 해제
105     firebase.auth().signOut().then(function() {
106         alert("인증이 해제되었습니다.");
107     }, function(error) {
108         alert(error.message);
109     });
110 });
111
112 /* 2. 접속한 경우 Database의 상태 변경 */
113 var _uid = null; // 유저의 uid를 저장해둘 전역 변수
114 firebase.auth().onAuthStateChanged(function(user) {
115     if (user) {
116         _uid = user.uid;
117         $("#LOG_MESSAGE").text(user.displayName+"님 접속하셨습니다");
118         // 3. 디비에 접속 하면 상태 변경 (등록이 안된 상태면 새로 쓰기)
119         firebase.database().ref('/onUsers/' + user.uid).set({"username": user.displayName, "siteOn": 1});
120         // TODO 4. 상태가 바뀔 때를 감지할 리스너(JS옵저버) 장착
121         // TODO 7. 디비에 접속 끊길때 이벤트 감지하면 특정값 변경 (siteOn 상태를 0으로 변경)
122     } else { // 2. 인증이 해제 되거나 인증되지 않은 경우
123         $("#LOG_MESSAGE").text("(인증하셔야 볼 수 있습니다)");
124         _uid = null;
125     }
126 })
```

인증 해제 시 리스트너도
해제하는 코드도 작성합니다

firebase

public

auth

css

database

- database_basic_step1.html
- database_basic_step2.html
- database_basic_step3.html
- database_basic_step4.html
- database_basic_step5.html
- database_basic_step6.html
- database_siteon_step1.html
- database_siteon_step2.html
- database_siteon_step3.html
- database_siteon_step4.html**
- database_siteon_step5.html
- database_siteon_step6.html
- database_siteon_step7.html

font

fonts

images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database_siteon.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting.html
- index.html

database_siteon_step3.html x database_siteon_step4.html x

```
97    });
98
99    /* 1. 인증 해제 */
100   $('#BTN_LOGOUT').click(function(){
101     // 4. DB의 변화를 더이상 감지 하지 않음
102     firebase.database().ref('/onUsers/').off();
103     // 3. 접속상태값을 변경
104     firebase.database().ref("/onUsers/"+_uid+"/siteOn").set(0);
105     $("#SITE_ON_USERS").html(''); // 접속 유저 목록 초기화
106     // 인증 해제
107     firebase.auth().signOut().then(function() {
108       alert("인증이 해제되었습니다.");
109     }, function(error) {
110       alert(error.message);
111     });
112   });
113
114   /* 2. 접속한 경우 Database의 상태 변경 */
115   var _uid = null; // 유저의 uid를 저장해둘 전역 변수
116   firebase.auth().onAuthStateChanged(function(user) {
117     if (user) {
118       _uid = user.uid;
119       $("#LOG_MESSAGE").text(user.displayName+"님 접속하셨습니다");
120       // 3. 디비에 접속 하면 상태 변경 (등록이 안된 상태면 새로 쓰기)
121       firebase.database().ref('/onUsers/' + user.uid).set({"username": user.displayName, "siteOn": 1});
122       // 4. 상태가 바뀔 때를 감지할 리스너(JS옵저버) 장착
123       firebase.database().ref('/onUsers/').on('value', function(snapshot_users) {
124         // TODO 5. 유저를 하나씩 읽어오기
125       });
126       // TODO 7. 다른페이지로 이동할 때마다 firebase.onAuthStateChanged()를 통해 유저 상태를 확인하고 변경
```

`snapshot_user.val()`의 값을 로그로 찍어보면
child json object의 집합이긴 합니다

그런데 object 목록을 어떻게 구분하죠?

`child_added` 이벤트의 또 다른 용도

firebase

public

auth

css

database

- database_basic_step1.html
- database_basic_step2.html
- database_basic_step3.html
- database_basic_step4.html
- database_basic_step5.html
- database_basic_step6.html
- database_siteon_step1.html
- database_siteon_step2.html
- database_siteon_step3.html
- database_siteon_step4.html
- database_siteon_step5.html
- database_siteon_step6.html
- database_siteon_step7.html

font

fonts

images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database_siteon.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting.html
- index.html

database_siteon_step4.html x database_siteon_step5.html x

```
46      </div>
47    </div>
48
49    <!-- 5. 접속 유저 ol박스 확인 -->
50    <div class="row z-depth-1">
51      <div class="col l12 m12 s12 input-field">
52        <h5>현재 접속 유저</h5>
53        <hr>
54        <ol id="SITE_ON_USERS"><!-- 접속한 USERS 목록을 띄울 박스 --></ol>
55      </div>
56    </div>
57
58  </div> }
59
60  <!-- 바닥글 영역 -->
61  <footer class="page-footer indigo">
62    <div class="container">
63      <div class="row center">
64        <a href="/database.html" class="btn white indigo-text">돌아가기</a>
65      </div>
66    </div>
67    <div class="footer-copyright">
68      <div class="container">
69        &copy; 2016 Copyright <a href="http://sbsoft.kr" target="_blank" class="white-text">SBJang</a>
70        <a href="http://abcds.kr/" target="_blank" class="white-text">ABCD</a> (
71          <a href="https://www.facebook.com/groups/aboutCoding/" target="_blank" class="white-text">Fa
72            <a class="white-text right" href="http://materializecss.com/" target="_blank">design by Mate
73          </div>
74        </div>
75      </div>
```

접속한 사용자의 목록을 띄울 OL태그 확인

public/database/database_siteon_step5.html 55:1 (2, 66) CRLF UTF-8 HTML 2 updates

firebase

public

auth

css

database

- database_basic_step1.html
- database_basic_step2.html
- database_basic_step3.html
- database_basic_step4.html
- database_basic_step5.html
- database_basic_step6.html
- database_siteon_step1.html
- database_siteon_step2.html
- database_siteon_step3.html
- database_siteon_step4.html
- database_siteon_step5.html
- database_siteon_step6.html
- database_siteon_step7.html

font

fonts

images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database_siteon.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting.html
- index.html

database_siteon_step4.html x database_siteon_step5.html x

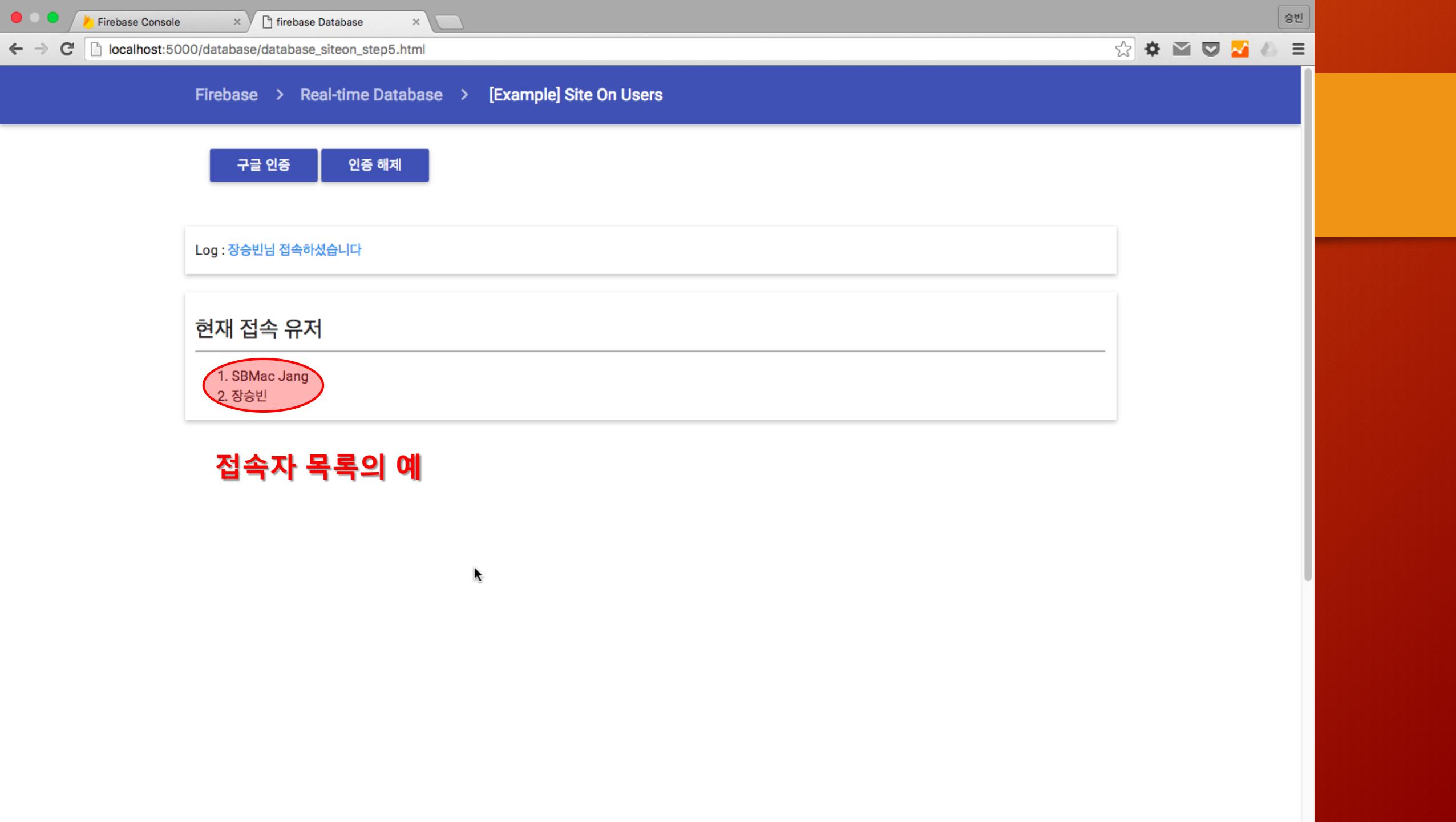
```
        document.getElementById("LOG_MESSAGE"),  
111     });  
112   };  
113  
114   /* 2. 접속한 경우 Database의 상태 변경 */  
115   var _uid = null; // 유저의 uid를 저장해둘 전역 변수  
116   firebase.auth().onAuthStateChanged(function(user) {  
117     if (user) {  
118       _uid = user.uid;  
119       $("#LOG_MESSAGE").text(user.displayName+"님 접속하셨습니다");  
120       // 3. 디비에 접속 하면 상태 변경 (등록이 안된 상태면 새로 쓰기)  
121       firebase.database().ref('/onUsers/' + user.uid).set({ "username": user.displayName, "siteOn": true});  
122       // 4. 상태가 바뀔 때를 감지할 리스너(JS옵저버) 장착  
123       firebase.database().ref('/onUsers/').on('value', function(snapshot_users) {  
124         // TODO 5. 유저를 하나씩 읽어오기  
125         });  
126         // TODO 7. 디비에 접속 끊길때 이벤트 감지하하면 특정값 변경 (siteOn 상태를 0으로 변경)  
127       } else { // 2. 인증이 해제 되거나 인증되지 않은 경우  
128         $("#LOG_MESSAGE").text("(인증하셔야 볼 수 있습니다)");  
129         _uid = null;  
130       }  
131     };  
132  
133     // TODO 6. 페이지 떠나게 되면  
134  
135   </script>  
136  
137   </body>  
138   </html>  
139
```

child_added 이벤트를 이용하는 코드를 작성

```
firebase
  public
    auth
    css
  database
    database_basic_step1.html
    database_basic_step2.html
    database_basic_step3.html
    database_basic_step4.html
    database_basic_step5.html
    database_basic_step6.html
    database_siteon_step1.html
    database_siteon_step2.html
    database_siteon_step3.html
    database_siteon_step4.html
    database_siteon_step5.html
    database_siteon_step6.html
    database_siteon_step7.html
  font
  fonts
  images
  js
    404.html
    auth_firebase_ui.html
    auth_google.html
    auth_mail.html
    auth_term.html
    auth.html
    database_basic.html
    database_siteon.html
    database.html
    fireinit.js
    fireinit.js.sample
    hosting.html
  index.html

database_siteon_step4.html x database_siteon_step5.html x

118     _uid = user.uid;
119     $("#LOG_MESSAGE").text(user.displayName+"님 접속하셨습니다");
120     // 3. 디비에 접속 하면 상태 변경 (등록이 안된 상태면 새로 쓰기)
121     firebase.database().ref('/onUsers/' + user.uid).set({"username": user.displayName, "siteOn": 1});
122     // 4. 상태가 바뀔 때를 감지할 리스너(JS옵저버) 장착
123     firebase.database().ref('/onUsers/').on('value', function(snapshot_users) {
124         $("#SITE_ON_USERS").html('');
125         // 5. 유저를 하나씩 읽어오기
126         firebase.database().ref('/onUsers/').on("child_added", function(snapshot) {
127             if (snapshot.val().siteOn == 1) {
128                 var li_tag = '<li>' + snapshot.val().username + '</li>';
129                 $("#SITE_ON_USERS").append(li_tag);
130             }
131         });
132     });
133     // TODO 7. 디비에 접속 끊길때 이벤트 감지하하면 특정값 변경 (siteOn 상태를 0으로 변경)
134 } else { // 2. 인증이 해제 되거나 인증되지 않은 경우
135     $("#LOG_MESSAGE").text("(인증하셔야 볼 수 있습니다)");
136     _uid = null;
137 }
138 });
139
140 // TODO 6. 페이지 떠나게 되면
141
142 </script>
143
144 </body>
145 </html>
146
```



구글 인증

인증 해제

Log : 장승빈님 접속하셨습니다

현재 접속 유저

1. SBMac Jang
2. 장승빈

접속자 목록의 예

firebase

public

auth

css

database

- database_basic_step1.html
- database_basic_step2.html
- database_basic_step3.html
- database_basic_step4.html
- database_basic_step5.html
- database_basic_step6.html
- database_siteon_step1.html
- database_siteon_step2.html
- database_siteon_step3.html
- database_siteon_step4.html
- database_siteon_step5.html
- database_siteon_step6.html
- database_siteon_step7.html

font

fonts

images

js

- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database_siteon.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting.html
- index.html

database_siteon_step5.html x database_siteon_step6.html x

```
118     _uid = user.uid;
119     $("#LOG_MESSAGE").text(user.displayName+"님 접속하셨습니다");
120     // 3. 디비에 접속 하면 상태 변경 (등록이 안된 상태면 새로 쓰기)
121     firebase.database().ref('/onUsers/' + user.uid).set({"username": user.displayName, "siteOn": 1});
122     // 4. 상태가 바뀔 때를 감지할 리스너(JS옵저버) 장착
123     firebase.database().ref('/onUsers/').on('value', function(snapshot_users) {
124         $("#SITE_ON_USERS").html('');
125         // 5. 접속 유저 목록 초기화
126         firebase.database().ref('/onUsers/').on("child_added", function(snapshot) {
127             if (snapshot.val().siteOn == 1) {
128                 var li_tag = '<li>' + snapshot.val().username + '</li>';
129                 $("#SITE_ON_USERS").append(li_tag);
130             }
131         });
132     });
133     // TODO 7. 디비에 접속 끊길때 이벤트 감지하하면 특정값 변경 (siteOn 상태를 0으로 변경)
134 } else { // 2. 인증이 해제 되거나 인증되지 않은 경우
135     $("#LOG_MESSAGE").text("(인증하셔야 볼 수 있습니다)");
136     _uid = null;
137 }
138 });
139
140 // TODO 6. 페이지 떠나게 되면
141
142 </script>
143
144 </body>
145 </html>
146 }
```

정식적인 동작을 위하여 페이지를 떠날 시에도 상태를 변경하는 코드를 작성합니다

- firebase
- public
 - auth
 - css
- database
 - database_basic_step1.html
 - database_basic_step2.html
 - database_basic_step3.html
 - database_basic_step4.html
 - database_basic_step5.html
 - database_basic_step6.html
 - database_siteon_step1.html
 - database_siteon_step2.html
 - database_siteon_step3.html
 - database_siteon_step4.html
 - database_siteon_step5.html
 - database_siteon_step6.html
 - database_siteon_step7.html
- font
- fonts
- images
- js
 - 404.html
 - auth_firebase_ui.html
 - auth_google.html
 - auth_mail.html
 - auth_term.html
 - auth.html
 - database_basic.html
 - database_siteon.html
 - database.html
 - fireinit.js
 - fireinit.js.sample
 - hosting.html
 - index.html

```
// 5. 접속 유저 목록 초기화
123     firebase.database().ref('/onUsers/').on('value', function(snapshot_users) {
124         $("#SITE_ON_USERS").html(''); // 5. 접속 유저 목록 초기화
125         // 5. 유저를 하나씩 읽어오기
126         firebase.database().ref('/onUsers/').on("child_added", function(snapshot) {
127             if (snapshot.val().siteOn == 1) {
128                 var li_tag = '<li>' + snapshot.val().username + '</li>';
129                 $("#SITE_ON_USERS").append(li_tag);
130             }
131         });
132     });
133     // TODO 7. 디비에 접속 끊길때 이벤트 감지하하면 특정값 변경 (siteOn 상태를 0으로 변경)
134 } else { // 2. 인증이 해제 되거나 인증되지 않은 경우
135     $("#LOG_MESSAGE").text("(인증하셔야 볼 수 있습니다)");
136     _uid = null;
137 }
138 });
139
140 // 6. 페이지 떠나게 되면
141 $(window).on("beforeunload", function(){
142     if(_uid != null){
143         firebase.database().ref("/onUsers/"+_uid+"/siteOn").set(0); // 상태 변경
144     }
145 });
146
147 </script>
148
149 </body>
150 </html>
151
```

firebase
public
auth
css
database
database_basic_step1.html
database_basic_step2.html
database_basic_step3.html
database_basic_step4.html
database_basic_step5.html
database_basic_step6.html
database_siteon_step1.html
database_siteon_step2.html
database_siteon_step3.html
database_siteon_step4.html
database_siteon_step5.html
database_siteon_step6.html
database_siteon_step7.html
font
fonts
images
js
404.html
auth_firebase_ui.html
auth_google.html
auth_mail.html
auth_term.html
auth.html
database_basic.html
database_siteon.html
database.html
fireinit.js
fireinit.js.sample
hosting.html
index.html

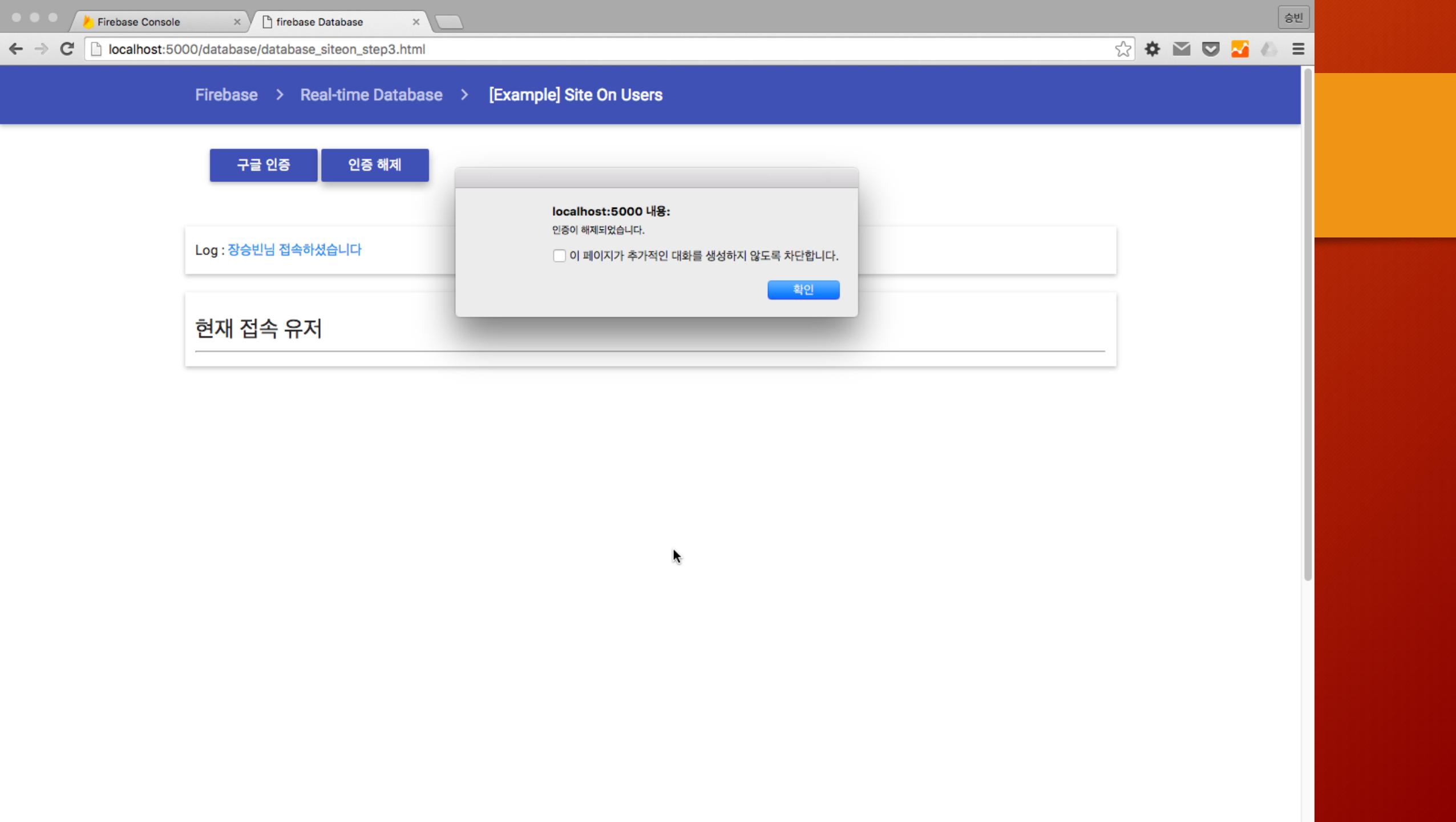
```
// 5. 접속 유저 목록 초기화
123   firebase.database().ref('/onUsers/').on('value', function(snapshot_users) {
124     $("#SITE_ON_USERS").html(''); // 5. 접속 유저 목록 초기화
125     // 5. 유저를 하나씩 읽어오기
126     firebase.database().ref('/onUsers/').on("child_added", function(snapshot) {
127       if (snapshot.val().siteOn == 1) {
128         var li_tag = '<li>' + snapshot.val().username + '</li>';
129         $("#SITE_ON_USERS").append(li_tag);
130       }
131     });
132   });
133   // TODO 7. 디비에 접속 끊길때 이벤트 감지하하면 특정값 변경 (siteOn 상태를 0으로 변경)
134 } else { // 2. 인증이 해제 되거나 인증되지 않은 경우
135   $("#LOG_MESSAGE").text("(인증하셔야 볼 수 있습니다)");
136   _uid = null;
137 }
138 });
139
140 // 6. 페이지 떠나게 되면
141 $(window).on("beforeunload", function(){
142   if(_uid != null){
143     firebase.database().ref("/onUsers/" + _uid + "/siteOn").set(0); // 상태 변경
144   }
145 });
146
147 </script>
148
149 </body>
150 </html>
151
```

접속이 끊겼을 시를 감지하여 서버에서
상태가 변하는 코드를 작성합니다

```
firebase
public
  auth
  css
database
  database_basic_step1.html
  database_basic_step2.html
  database_basic_step3.html
  database_basic_step4.html
  database_basic_step5.html
  database_basic_step6.html
  database_siteon_step1.html
  database_siteon_step2.html
  database_siteon_step3.html
  database_siteon_step4.html
  database_siteon_step5.html
  database_siteon_step6.html
  database_siteon_step7.html
font
fonts
images
js
  404.html
  auth_firebase_ui.html
  auth_google.html
  auth_mail.html
  auth_term.html
  auth.html
  database_basic.html
  database_siteon.html
  database.html
  fireinit.js
  fireinit.js.sample
  hosting.html
  index.html

database_siteon_step6.html x database_siteon_step7.html x

124     $("#SITE_ON_USERS").html(''); // 5. 접속 유저 목록 초기화
125     // 5. 유저를 하나씩 읽어오기
126     firebase.database().ref('/onUsers/').on("child_added", function(snapshot) {
127         if (snapshot.val().siteOn == 1) {
128             var li_tag = '<li>' + snapshot.val().username + '</li>';
129             $("#SITE_ON_USERS").append(li_tag);
130         }
131     });
132 });
133 // 7. 디비에 접속 끊길때 이벤트 감지하하면 특정값 변경 (siteOn 상태를 0으로 변경)
134 firebase.database().ref("/onUsers/"+user.uid+"/siteOn").onDisconnect().set(0);
135 } else { // 2. 인증이 해제 되거나 인증되지 않은 경우
136     $("#LOG_MESSAGE").text("(인증하셔야 볼 수 있습니다)");
137     _uid = null;
138 }
139 });
140
141 // 6. 페이지 떠나게 되면
142 $(window).on("beforeunload", function(){
143     if(_uid != null){
144         firebase.database().ref("/onUsers/"+_uid+"/siteOn").set(0); // 상태 변경
145     }
146 });
147
148 </script>
149
150 </body>
151 </html>
152
```



질문 있으신가요?



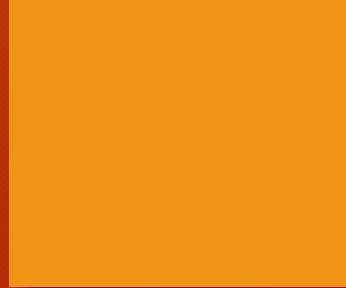
firebase 궁금증이 생기면
[Facebook ABCD Group](#)에
언제든 등록해주세요

모두의 코딩
by ABCD

매주 수요일, 모비프렌

장르, 시간 불문

혹은 일찍 오셔도 됩니다



수고하셨습니다