



Firebase for Web

with ABCD

Created at 2016. 08. / 장승빈

실습 자료 다운로드

- 실습 자료를 다운로드 받습니다.
 - https://github.com/denlyou/study_firebase4web_abcd
- Firebase project로 초기화 되어 있는 폴더에 public 폴더의 파일을 복사합니다
 - 첫시간 index.html 파일은 hosting_profile.html로 변경해 주세요



Storage

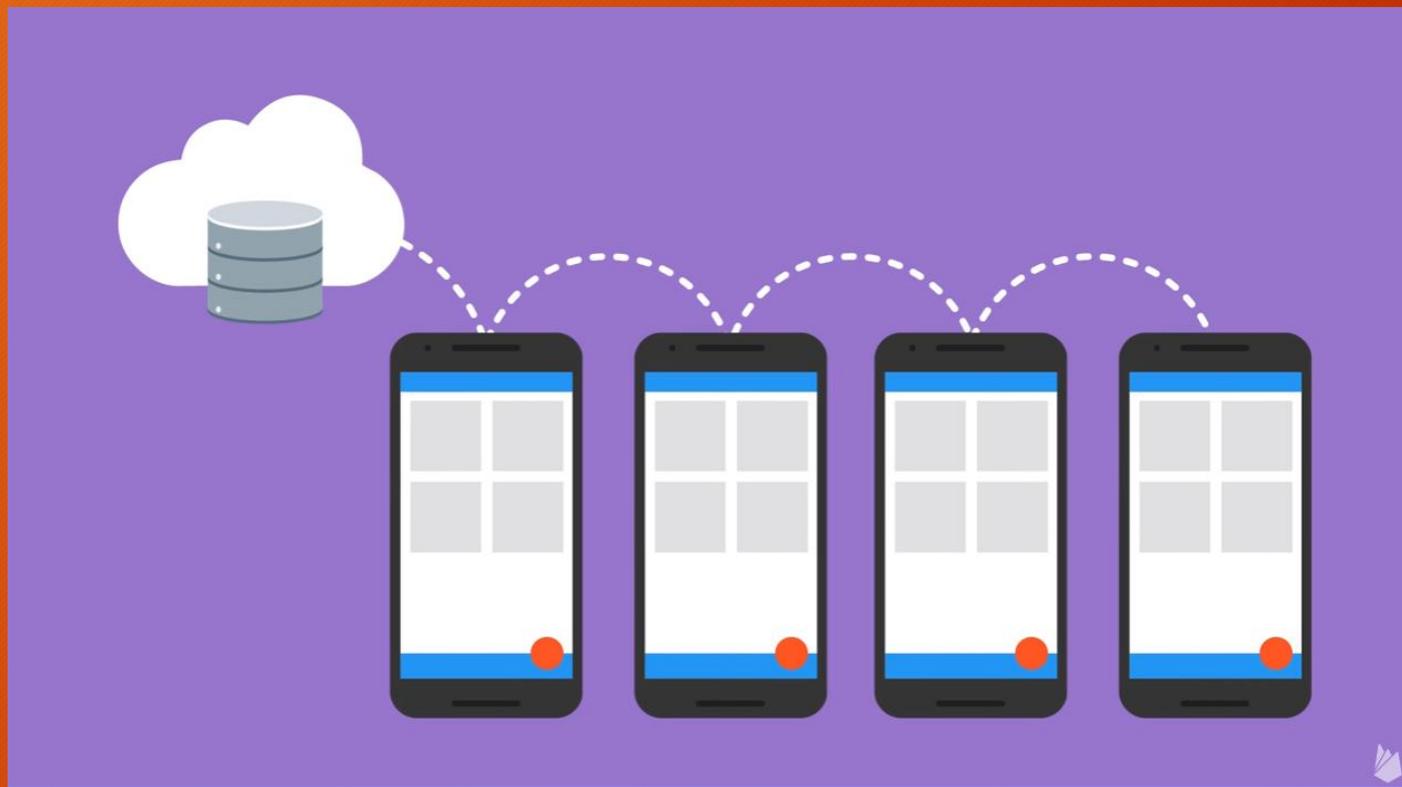
서버측 코드 없이 이미지, 오디오, 동영상과 같은 사용자가 생성한 콘텐츠를 저장하고 가져올 수 있습니다.

[자세히 알아보기](#) [시작하기](#)



Firebase Storage

Firestore Storage?



https://youtu.be/_tyjqozrEPY?list=PLI-K7zZEsYLmOF_07layrTntevxtbUxDL

Firestore Storage 특징

Robust

Firestore Storage performs uploads and downloads regardless of network quality. Uploads and downloads are robust, meaning they restart where they stopped, saving your users time and bandwidth.

Secure

Firestore Storage integrates with Firestore Authentication to provide simple and intuitive authentication for developers. You can use our declarative security model to allow access based on filename, size, content type, and other metadata.

Scalable

Firestore Storage is backed by Google Cloud Storage for petabyte scale when your app goes viral. Effortlessly grow from prototype to production using the same infrastructure that powers Snapchat.

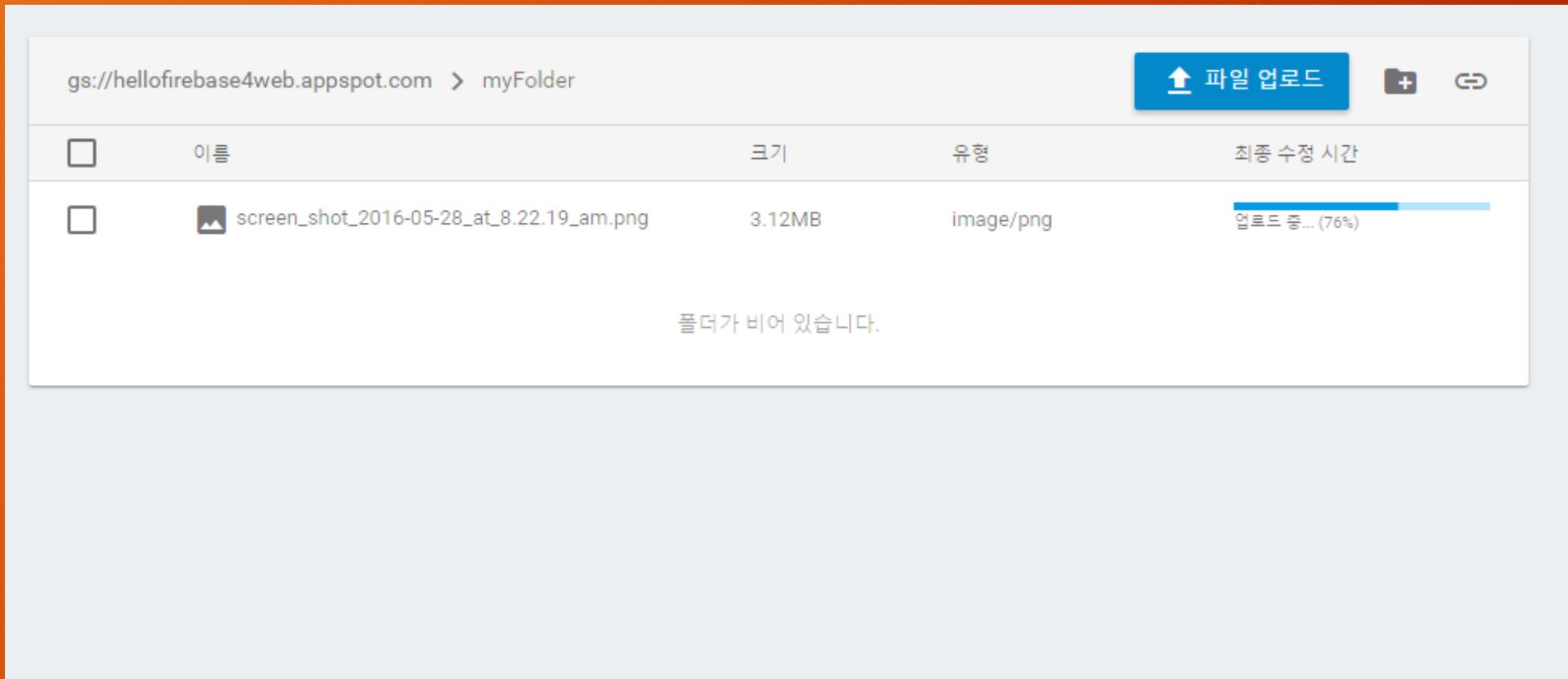
- 요약 : 탄탄하고 안전하고 엄청난(!) 스케일의 저장소 서비스

[잠깐] firebase의 요금 정책

- Storage 무료는 5Gb저장 공간의 30Gb 다운로드 트래픽 (50,000회)

		SPARK Free <small>Generous limits for hobbyists</small>	FLAME \$25/month <small>Predictable pricing for growing apps</small>	BLAZE Pay as you go <small>Commodity pricing for apps at scale</small>
Realtime Database	Simultaneous connections	100	Unlimited ¹	Unlimited ¹
	GB stored	1 GB	2.5 GB	\$5/GB
	GB downloaded	10 GB	20 GB	\$1/GB
	Daily private backups	×	✓	✓
File Storage	GB stored	5 GB	50 GB	\$0.026/GB ²
	GB downloaded	30 GB	50 GB	about \$0.12/GB ²
	Uploads & downloads	50,000 each	100,000 each	about \$0.01/K ²

Firestore 콘솔에서 기본 사용이 가능합니다 OwO



Storage 기본 사용법

파일 접근 (Reference) 방법

- DB에서 사용했던 `.ref()`과 매우 흡사합니다
- Ref 변경이 용이합니다
 - `.child()` : 자식 Ref 찾기
 - `.parent` : 부모 Ref
 - `.root` : 최상위 Ref

```
// Points to the root reference
var storageRef = firebase.storage().ref();

// Points to 'images'
var imagesRef = storageRef.child('images');

// Points to 'images/space.jpg'
// Note that you can use variables to create child values
var fileName = 'space.jpg';
var spaceRef = imagesRef.child(fileName);

// File path is 'images/space.jpg'
var path = spaceRef.fullPath

// File name is 'space.jpg'
var name = spaceRef.name

// Points to 'images'
var imagesRef = spaceRef.parent;
```

Storage : 파일 업로드

- UploadTask 클래스를 이용합니다

firebase.storage.UploadTask.on()

- <https://firebase.google.com/docs/reference/js/firebase.storage.UploadTask#on>

- Sample Code

```
uploadTask.on('state_changed', function(snapshot){
  // Observe state change events such as progress, pause, and resume
  // See below for more detail
}, function(error) {
  // Handle unsuccessful uploads
}, function() {
  // Handle successful uploads on complete
  // For instance, get the download URL: https://firebasestorage.googleapis.com/...
  var downloadURL = uploadTask.snapshot.downloadURL;
});
```

Storage : 파일 다운로드

- `getDownloadURL()` 메소드를 이용하는데 비동기 통신으로 http주소를 얻어옵니다

`firebase.storage.Reference.getDownloadURL()`

- <https://firebase.google.com/docs/reference/js/firebase.storage.Reference#getDownloadURL>

- Sample Code

```
storageRef.child('images/stars.jpg').getDownloadURL().then(function(url) {  
  // Get the download URL for 'images/stars.jpg'  
  // This can be inserted into an <img> tag  
  // This can also be downloaded directly  
}).catch(function(error) {  
  // Handle any errors  
});
```

Storage : 파일 삭제하기

- `firebase.storage.Reference.delete()`
 - <https://firebase.google.com/docs/reference/js/firebase.storage.Reference#delete>
- Sample Code

```
// Create a reference to the file to delete
var desertRef = storageRef.child('images/desert.jpg');

// Delete the file
desertRef.delete().then(function() {
  // File deleted successfully
}).catch(function(error) {
  // Uh-oh, an error occurred!
});
```

실습 : 파일 메모장 웹 앱 만들기

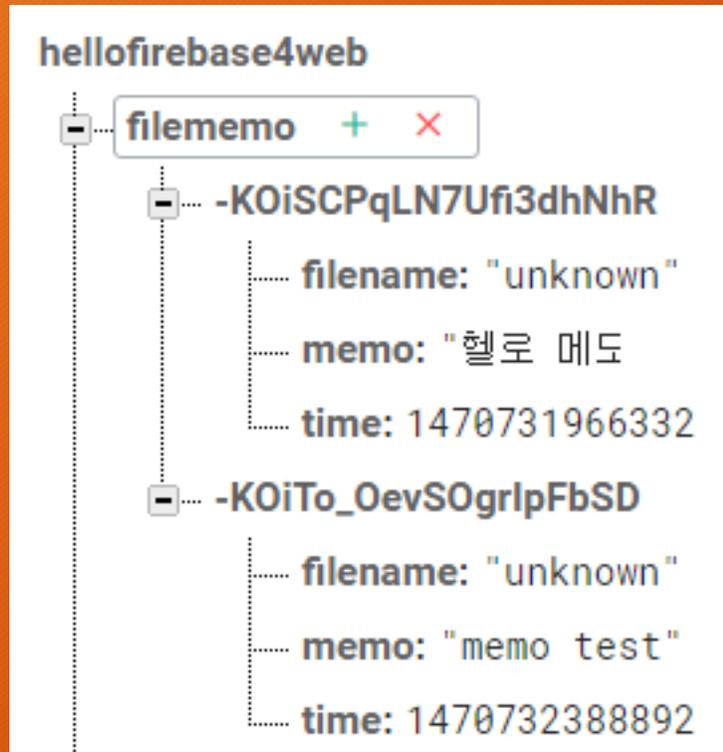
NoSQL은 RDBMS와는 달리 SQL 쿼리를 사용하지 않기에

다음 파일을 편집기로 열어 .sample 지우고 저장하세요

```
[project folder]/public/storage_memo.html.sample
```

1. Firebase SDK 링크 + 구글 인증
2. DB에 데이터 쓰기
3. DB에서 데이터 읽어오기
4. File 객체 구하기
5. 파일 업로드 하기
6. 파일 다운로드 하기
7. 파일 삭제하기

파일 메모장 DB 구조



- 파일 메모들 정보 (/filememo)
- 파일 메모 정보 (/filememo/[pushKey])
 - push()를 이용해 자동 생성되는 UniqueKey값 사용
- 파일 메모 내용 (/filememo/[pushKey]/memo)
- 업로드한 파일명 (/filememo/[pushKey]/filename)
- 메모 등록 시간 (/filememo/[pushKey]/time)
 - 서버에서 자동으로 생성되는 timestamp 이용

- js
- storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
- 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_profile.html
- hosting.html
- index.html
- index2.html
- mail_auth.html
- storage_memo.html
- storage_memo.html.s
- storage.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase-debug.log
- firebase.json

```
47     </div>
48     <div class="col l12 m12 s12 input-field">
49       <!-- TODO 1. 구글 인증 버튼 확인 -->
50       <span class="right"><button id="BTN_GOOGLE_LOGIN" type="button" class="btn indigo">구글
51       • 인증</button></span>
52       <!-- TODO 메모 작성 버튼 확인 -->
53       <button id="BTN_WRITE" type="button" class="btn indigo">메모 작성하기</button>
54     </div>
55     <br><br><br><br><span class="white-text">dummy string</span>
56   </div>
57 </form>
58 <!-- 메모 내용 읽어오기 영역 -->
59 <div class="row">
60   <div class="col s12 center">
61     <!-- TODO 로딩 프로그래스용 프리로더 -->
62     <div class="preloader-wrapper big active" id="PROGRESS">
63       <div class="spinner-layer spinner-blue-only">
64         <div class="circle-clipper left"><div class="circle"></div></div>
65         <div class="gap-patch"><div class="circle"></div></div>
66         <div class="circle-clipper right"><div class="circle"></div></div>
67       </div>
68     </div>
69   </div>
70 </div>
71 <!-- TODO 기록된 메모들을 읽어올 UL 태그 -->
```

구글 인증 버튼 확인

파일 첨부 메모장

파일선택

메모할 내용

메모 작성하기

구글 인증

버튼 (#BTN_GOOGLE_LOGIN)



```

storage_memo.html.sample — /Users/SBJang/Desktop/firebase
storage_memo.html.sample x
92 <!-- ////////////////////////////////////// javascript 스크립트 영역 ////////////////////////////////////// >
93
94 <!--Import jQuery before materialize.js-->
95 <script type="text/javascript" src="https://code.jquery.com/jquery-2.1.1.min.js"></script>
96 <script type="text/javascript" src="/js/materialize.min.js"></script>
97 <!-- TODO 1. firebase SDK 링크 (auth, db, storage 모두) -->
98 <script type="text/javascript">
99     /* TODO 1. 버튼을 눌렀을 경우 구글 인증 동작하게 */
100     $("#BTN_GOOGLE_LOGIN").click(function() {
101
102     });
103
104     /* TODO 2. 메모하기 눌렀을 때 DB에 기록 */
105     $("#BTN_WRITE").click(function(){
106         // TODO 5. 파일 선택이 안됐을 경우 등록 방지
107         $("#PROGRESS").show(); // 프리로더 보이기
108         /* TODO 5. uploadTask를 이용해 선택된 파일을 등록 */
109         /* TODO 2. DB에 내용을 기록 */
110         /* TODO 2. 폼 및 요소 초기화 */
111         $("#PROGRESS").hide(); // 프리로더 감추기
112     });
113
114     /* TODO 4. 파일객체 가져오기 */
115
116     /* TODO 3. DB에서 메모 내용 읽어오기 */
117     firebase.database().ref('/filememo').on('value', function(snapshot) {
118
119

```

SDK 링크와 구글 인증 코드 작성

- js
 - storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - 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_profile.html
 - hosting.html
 - index.html
 - index2.html
 - mail_auth.html
 - storage_memo.html
 - storage_memo.html.s
 - storage.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase-debug.log
- firebase.json

```

storage_memo_step1.html x
92 <!-- ////////////////////////////////////// javascript 스크립트 영역 //////////////////////////////////////
93
94 <!--Import jQuery before materialize.js-->
95 <script type="text/javascript" src="https://code.jquery.com/jquery-2.1.1.min.js"></script>
96 <script type="text/javascript" src="/js/materialize.min.js"></script>
97 <!-- 1. firebase SDK 링크 (auth, db, storage 모두) -->
98 <script src="https://www.gstatic.com/firebasejs/3.2.1/firebase.js"></script>
99 <script src="https://www.gstatic.com/firebasejs/3.2.1/firebase-app.js"></script>
100 <script src="https://www.gstatic.com/firebasejs/3.2.1/firebase-auth.js"></script>
101 <script src="https://www.gstatic.com/firebasejs/3.2.1/firebase-database.js"></script>
102 <script src="https://www.gstatic.com/firebasejs/3.2.1/firebase-storage.js"></script>
103 <script src="/fireinit.js"></script>
104 <script type="text/javascript">
105     /* TODO 1. 버튼을 눌렀을 경우 구글 인증 동작하게 */
106     $("#BTN_GOOGLE_LOGIN").click(function() {
107         var provider = new firebase.auth.GoogleAuthProvider();
108         firebase.auth().signInWithRedirect(provider).catch(function(error) { // 로그인에 실패 했을시
109             console.log(error.message);
110         });
111     });
112
113     /* TODO 2. 메모하기 눌렀을 때 DB에 기록 */
114     $("#BTN_WRITE").click(function(){
115         // TODO 5. 파일 선택이 안됐을 경우 등록 방지
116         $("#PROGRESS").show(); // 프리로더 보이기
117         /* TODO 5. uploadTask를 이용해 선택된 파일을 등록 */
118     });
119

```

- js
- storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
- 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_profile.html
- hosting.html
- index.html
- index2.html
- mail_auth.html
- storage_memo.html
- storage_memo.html.s
- storage.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase-debug.log
- firebase.json

```

42     </div>
43     <div class="col l12 m12 s12 input-field">
44         <!-- TODO 메모 내용 input 태그 확인 -->
45         <input id="INPUT_MESSAGE" type="text" class="validate">
46         <label for="INPUT_MESSAGE">메모할 내용</label>
47     </div>
48     <div class="col l12 m12 s12 input-field">
49         <!-- TODO 1. 구글 인증 버튼 확인 -->
50         <span class="right"><button id="BTN_GOOGLE_LOGIN" type="button" class="btn indigo">구글
51         인증</button></span>
52         <!-- TODO 메모 작성 버튼 확인 -->
53         <button id="BTN_WRITE" type="button" class="btn indigo">메모 작성하기</button>
54     </div>
55     <br><br><br><br><span class="white-text">dummy string</span>
56 </div>
57 </form>
58 <!-- 메모 내용 읽어오기 영역 -->
59 <div class="row">
60     <div class="col s12 center">
61         <!-- TODO 로딩 프로그래스용 프리로더 -->
62         <div class="preloader-wrapper big active" id="PROGRESS">
63             <div class="spinner-layer spinner-blue-only">
64                 <div class="circle-clipper left"><div class="circle"></div></div>
65                 <div class="gap-patch"><div class="circle"></div></div>
66                 <div class="circle-clipper right"><div class="circle"></div></div>

```

메모 INPUT 태그와 버튼 확인

파일 첨부 메모장

파일선택

메모할 내용

메모 작성하기

구글 인증

INPUT태그 (#INPUT_MESSAGE)

버튼 (#BTN_WRITE)



- js
- storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database_sideon.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting_profile.html
- hosting.html
- index.html
- index2.html
- mail_auth.html
- storage_memo.html
- storage_memo.html.s
- storage.html
- .DS_Store
- .firebase
- database.rules.json
- firebase-debug.log
- firebase.json

```

113  /* TODO 2. 메모하기 눌렀을 때 DB에 기록 */
114  $("#BTN_WRITE").click(function(){
115      // TODO 5. 파일 선택이 안됐을 경우 등록 방지
116      $("#PROGRESS").show(); // 프리로더 보이기
117      /* TODO 5. uploadTask를 이용해 선택된 파일을 등록 */
118      /* TODO 2. DB에 내용을 기록 */
119      /* TODO 2. 폼 및 요소 초기화 */
120      $("#PROGRESS").hide(); // 프리로더 감추기
121  });
122
123  /* TODO 4. 파일객체 가져오기 */
124
125  /* TODO 3. DB에서 메모 내용 읽어오기 */
126  firebase.database().ref('/filememo').on('value', function(snapshot) {
127      $("#PROGRESS").show(); // 프리로더 보이기
128      /* TODO 3. 메모 객체 하나씩 li요소로 변경해서 ul(#READ_MEMO)에 추가 */
129      $("#PROGRESS").hide(); // 프리로더 감추기
130
131      /* TODO 3. 삭제 이벤트 연결 */
132      /* TODO 6. 다운로드 작성 */
133  });
134
135  </script>
136
137  </body>
138  </html>

```

DB에 내용을 기록하고 폼 태그를 초기화합니다

- js
- storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database_sideon.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting_profile.html
- hosting.html
- index.html
- index2.html
- mail_auth.html
- storage_memo.html
- storage_memo.html.si
- storage.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase-debug.log
- firebase.json

```

115 // TODO 5. 파일 선택이 안됐을 경우 등록 방지
116 $("#PROGRESS").show(); // 프리로더 보이기
117 /* TODO 5. uploadTask를 이용해 선택된 파일을 등록 */
118 /* 2. DB에 내용을 기록 */
119 var new_memo = $("#INPUT_MESSAGE").val();
120 var _filename = "unknown"; // 4. TODO 파일명으로 수정
121 firebase.database().ref('/filememo').push({
122     "memo" : new_memo ,
123     "time" : firebase.database.ServerValue.TIMESTAMP ,
124     "filename" : _filename
125 });
126
127 /* 2. 폼 및 요소 초기화 */
128 $("#form").each(function() {
129     this.reset();
130 });
131 // TODO 4. 전역 파일 객체 초기화
132 $("#PROGRESS").hide(); // 프리로더 감추기
133 });
134
135 /* TODO 4. 파일객체 가져오기 */
136
137 /* TODO 3. DB에서 메모 내용 읽어오기 */
138 firebase.database().ref('/filememo').on('value', function(snapshot) {
139     $("#PROGRESS").show(); // 프리로더 보이기
140     /* TODO 3. 메모 객체 하나씩 li요소로 변경해서 ul(#READ_MEMO)에 추가 */

```

파일 첨부 메모장

파일선택

메모할 내용
헬로 메모

메모 작성하기

메모를 작성해 봅니다

구글 인증

- Home HelloFirebase4Web
- 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

filememo

- KOiSCPqLN7Ufi3dhNhR
 - filename: "unknown"
 - memo: "헬로 메모"
 - time: 1470731966332

onUsers

- 698QxDGQWHYqmQA7NuTMdClqJV01
 - siteOn: 0
 - username: "SBMac Jang"
- mbLfZ20MsyP2qM0e90214yCVC8F3
 - siteOn: 0
 - username: "장승빈"

메세지: "hello"

Firestore console에서 확인해 봅니다

- js
- storage
 - storage_memo_step1
 - storage_memo_step2
 - storage_memo_step3
 - storage_memo_step4
 - storage_memo_step5
 - storage_memo_step6
 - storage_memo_step7
- 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_profile.html
- hosting.html
- index.html
- index2.html
- mail_auth.html
- storage_memo.html
- storage_memo.html.sample
- storage.html
- .DS_Store
- .firebase
- database.rules.json
- firebase-debug.log
- firebase.json

```

66         <div class="circle-clipper right"><div class="circle"></div></div>
67     </div>
68 </div>
69 </div>
70 </div>
71 <!-- TODO 기록된 메모들을 읽어올 UL 태그 -->
72 <ul class="collapsible" data-collapsible="expandable" id="READ_MEMO"></ul>
73 </div>
74
75 <!-- 바닥글 영역 -->
76 <footer class="page-footer indigo">
77     <div class="container">
78         <div class="row center">
79             <a href="/storage.html" class="btn white indigo-text">돌아가기</a>
80         </div>
81     </div>
82     <div class="footer-copyright">
83         <div class="container">
84             &copy; 2016 Copyright <a href="http://sbsoft.kr" target="_blank" class="white-text">SBJang</a>
85             <a href="http://abcds.kr/" target="_blank" class="white-text">ABCD</a> (
86             <a href="https://www.facebook.com/groups/aboutCoding/" target="_blank" class="white-text">
87             <a class="white-text right" href="http://materializecss.com/" target="_blank">design by Ma
88         </div>
89     </div>
90 </footer>
91

```

DB에서 내용을 읽어와 보여줄 UL태그 확인

파일 첨부 메모장

파일선택

메모할 내용

메모 작성하기

구글 인증



UL태그 (#READ_MEMO)

- js
 - storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - 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_profile.html
 - hosting.html
 - index.html
 - index2.html
 - mail_auth.html
 - storage_memo.html
 - storage_memo.html.s
 - storage.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase-debug.log
- firebase.json

```

126
127     /* 2. 폼 및 요소 초기화 */
128     $("form").each(function() {
129         this.reset();
130     });
131     // TODO 4. 전역 파일 객체 초기화
132     $("#PROGRESS").hide(); // 프리로더 감추기
133 });
134
135     /* TODO 4. 파일객체 가져오기 */
136
137     /* TODO 3. DB에서 메모 내용 읽어오기 */
138     firebase.database().ref('/filememo').on('value', function(snapshot) {
139         $("#PROGRESS").show(); // 프리로더 보이기
140     /* TODO 3. 메모 객체 하나씩 li요소로 변경해서 ul(#READ_MEMO)에 추가 */
141     $("#PROGRESS").hide(); // 프리로더 감추기
142
143     /* TODO 3. 삭제 이벤트 연결 */
144     /* TODO 6. 다운로드 작성 */
145     });
146
147     </script>
148
149 </body>
150 </html>
151

```

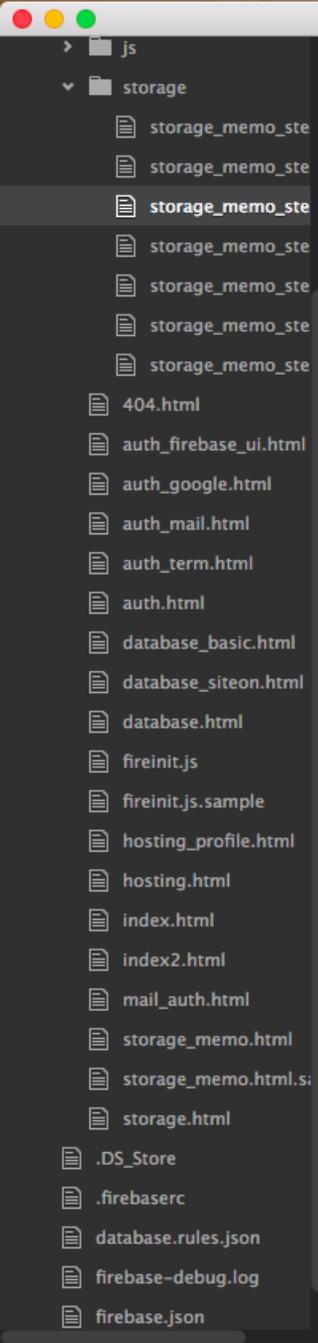
Object.keys()를 이용해 HTML 코드를 APPEND하는 코드를 작성해 봅시다

- js
- storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
- 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_profile.html
- hosting.html
- index.html
- index2.html
- mail_auth.html
- storage_memo.html
- storage_memo.html.s
- storage.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase-debug.log
- firebase.json

```

144     return false;
145 }
146 /* 3. 메모 객체 하나씩 li요소로 변경해서 ul(#READ_MEMO)에 추가 */
147 var _keys = Object.keys(snapshot.val());
148 for (var i=_keys.length; 0<i; i--){
149     var child = snapshot.child(_keys[(i-1)]).val()
150     var uploadDate = new Date(child.time*1000);
151     var appendHTML =
152     '
153     + '         <div class="collapsible-header"><i
154     + '     • class="material-icons">subject</i>'+child.memo+'</div>'
155     + '         <div class="collapsible-body">'
156     + '             <p><span class="right"><button data-f="'+child.filename+'
157     + '             • data-k="'+_keys[(i-1)]+'" class="btn red BTN_DELETE">삭제</button>'
158     + '             <button data-f="'+child.filename+' class="btn BTN_DOWNLOAD">다운로드</button>'
159     + '             </span>파일명 : '+child.filename+'<br>등록일 : '+uploadDate+'</p>'
160     + '         </div>'
161     + '     </li>';
162     $("#READ_MEMO").append(appendHTML);
163 }
164 $("#PROGRESS").hide(); // 프리로더 감추기
165
166 /* 3. 삭제 이벤트 연결 */
167 $(".BTN_DELETE").off("click");
168 $(".BTN_DELETE").click(function(){
169     // TODO 7. 파일도 삭제 하기

```



```

storage_memo_step2.html x storage_memo_step3.html x
134
135 /* TODO 4. 파일객체 가져오기 */
136
137 /* 3. DB에서 메모 내용 읽어오기 */
138 firebase.database().ref('/filememo').on('value', function(snapshot) {
139     $("#PROGRESS").show(); // 프리로더 보이기
140     $("#READ_MEMO").html("");
141     if (snapshot.val() == null ) {
142         $("#READ_MEMO").append('<div class="collapsible-header">아직 등록된 파일이 없습니다.</div>');
143         $("#PROGRESS").hide();
144         return false;
145     }
146     /* 3. 메모 객체 하나씩 li요소로 변경해서 ul(#READ_MEMO)에 추가 */
147     var _keys = Object.keys(snapshot.val());
148     for (var i=_keys.length; 0<i; i--){
149         var child = snapshot.child(_keys[(i-1)]).val()
150         var uploadDate = new Date(child.time*1000);
151         var appendHTML =
152         '
153         + ' <div class="collapsible-header"><i
154         + ' class="material-icons">subject</i>'+child.memo+'</div>'
155         + ' <div class="collapsible-body">'
156         + ' <p><span class="right"><button data-f="'+child.filename+'
157         + ' data-k="'+_keys[(i-1)]+'"' class="btn red BTN_DELETE">삭제</button>'
158         + ' <button data-f="'+child.filename+'"' class="btn BTN_DOWNLOAD">다운로드</button>'
159         + ' </span>파일명 : '+child.filename+'<br>등록일 : '+uploadDate+'</p>'

```

다시 읽어올 경우엔 초기화

요소가 없을 땐 별도 메시지

- js
- storage
 - storage_memo_ste
 - 404.html
 - auth_firebase_ui.html
 - auth_google.html
 - auth_mail.html
 - auth_term.html
 - auth.html
 - database_basic.html
 - database_sideon.html
 - database.html
 - fireinit.js
 - fireinit.js.sample
 - hosting_profile.html
 - hosting.html
 - index.html
 - index2.html
 - mail_auth.html
 - storage_memo.html
 - storage_memo.html.s
 - storage.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase-debug.log
- firebase.json

```

154 + '      <div class="collapsible-body">'
155 + '          <p><span class="right"><button data-f="'+child.filename+'
      • data-k="'+_keys[(i-1)]+'" class="btn red BTN_DELETE">삭제</button>'
156 + '          <button data-f="'+child.filename+'" class="btn BTN_DOWNLOAD">다운로드</button>'
157 + '          </span>파일명 : '+child.filename+'<br>등록일 : '+uploadDate+'</p>'
158 + '      </div>'
159 + '  </li>';
160      $("#READ_MEMO").append(appendHTML);
161  }
162      $("#PROGRESS").hide(); // 프리로더 감추기
163
164      /* 3. 삭제 이벤트 연결 */
165      $(".BTN_DELETE").off("click");
166      $(".BTN_DELETE").click(function(){
167          // TODO 7. 파일도 삭제 하기
168          var _key = $(this).data("k");
169          firebase.database().ref('/filememo').child(_key).remove();
170      });
171      /* TODO 6. 다운로드 작성 */
172      });
173
174  </script>
175
176  </body>
177  </html>
178

```

remove()를 이용해 삭제하는 코드도 작성해 줍니다

파일 첨부 메모장

파일선택

메모할 내용

메모 작성하기

구글 인증

☰ memo test

☰ 헬로 메모

파일명 : unknown
등록일 : Wed Aug 23 48575 02:18:52 GMT+0900 (KST)

삭제

다운로드

등록한 메모의 목록이 잘 뜨는지 확인해 봅니다!

- js
- storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - 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_profile.html
 - hosting.html
 - index.html
 - index2.html
 - mail_auth.html
 - storage_memo.html
 - storage_memo.html.s
 - storage.html
- .DS_Store
- .firebase
- database.rules.json
- firebase-debug.log
- firebase.json

```
26
27 <!-- 내용 부분 -->
28 <div class="container" id="CONTENTS">
29   <form id="FRM">
30     <div class="row z-depth-1">
31       <div class="col s12"><h3>파일 첨부 메모장</h3></div>
32       <div class="col l12 m12 s12 input-field">
33         <div class="file-field input-field">
34           <div class="btn">
35             <!-- TODO 첨부파일 input 태그 확인 -->
36             <span>파일선택</span><input type="file" id="INPUT_FILE">
37           </div>
38           <div class="file-path-wrapper">
39             <input class="file-path validate" type="text">
40           </div>
41         </div>
42       </div>
43       <div class="col l12 m12 s12 input-field">
44         <!-- TODO 메모 내용 input 태그 확인 -->
45         <input id="INPUT_MESSAGE" type="text" class="validate">
46         <label for="INPUT_MESSAGE">메모할 내용</label>
47       </div>
48       <div class="col l12 m12 s12 input-field">
49         <!-- TODO 1. 구글 인증 버튼 확인 -->
50         <span class="right"><button id="BTN_GOOGLE_LOGIN" type="button" class="btn indigo">구글
인증</button></span>
```

파일 type인 INPUT 태그 확인

파일 첨부 메모장

파일선택

메모할 내용

메모 작성하기

구글 인증

INPUT태그 (#INPUT_FILE)



- js
- storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
- 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_profile.html
- hosting.html
- index.html
- index2.html
- mail_auth.html
- storage_memo.html
- storage_memo.html.s
- storage.html
- .DS_Store
- .firebaseerc
- database.rules.json
- firebase-debug.log
- firebase.json

```

129     this.reset();
130   });
131   // TODO 4. 전역 파일 객체 초기화
132   $("#PROGRESS").hide(); // 프리로더 감추기
133 });
134
135 /* TODO 4. 파일객체 가져오기 */
136
137 /* 3. DB에서 메모 내용 읽어오기 */
138 firebase.database().ref('/filememo').on('value', function(snapshot) {
139   $("#PROGRESS").show(); // 프리로더 보이기
140   $("#READ_MEMO").html("");
141   if (snapshot.val() == null ) {
142     $("#READ_MEMO").append('<div class="collapsible-header">아직 등록된 파일이 없습니다.</div>');
143     $("#PROGRESS").hide();
144     return false;
145   }
146   /* 3. 메모 객체 하나씩 li요소로 변경해서 ul(#READ_MEMO)에 추가 */
147   var _keys = Object.keys(snapshot.val());
148   for (var i=_keys.length; 0<i; i--){
149     var child = snapshot.child(_keys[(i-1)]).val()
150     var uploadDate = new Date(child.time*1000);
151     var appendHTML =
152     '       <li>
153 +       <div class="collapsible-header"><i
154 +       class="material-icons">subject</i>'+child.memo+'</div>'
155 +       <div class="collansible-body">'

```

JavaScript 코드를 이용하여 File 객체를 구합니다

- js
- storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
- 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_profile.html
- hosting.html
- index.html
- index2.html
- mail_auth.html
- storage_memo.html
- storage_memo.html.s
- storage.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase-debug.log
- firebase.json

```

storage.html x storage_memo_step3.html x storage_memo_step4.html x
130     });
131     _file = null; // 4. 전역 파일 객체 초기화
132     $("#PROGRESS").hide(); // 프리로더 감추기
133     });
134
135     /* 4. 파일객체 가져오기 */
136     var _file = null;
137     document.getElementById("INPUT_FILE").addEventListener("change", handleFiles, false);
138     function handleFiles() {
139         _file = (this.files)[0];
140     }
141
142     /* 3. DB에서 메모 내용 읽어오기 */
143     firebase.database().ref('/filememo').on('value', function(snapshot) {
144         $("#PROGRESS").show(); // 프리로더 보이기
145         $("#READ_MEMO").html("");
146         if (snapshot.val() == null ) {
147             $("#READ_MEMO").append('<div class="collapsible-header">아직 등록된 파일이 없습니다.</div>');
148             $("#PROGRESS").hide();
149             return false;
150         }
151         /* 3. 메모 객체 하나씩 li요소로 변경해서 ul(#READ_MEMO)에 추가 */
152         var _keys = Object.keys(snapshot.val());
153         for (var i=_keys.length; 0<i; i--){
154             var child = snapshot.child(_keys[(i-1)]).val()
155             var uploadDate = new Date(child.time*1000);

```

- js
 - storage
 - storage_memo_ste
 - 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_profile.html
 - hosting.html
 - index.html
 - index2.html
 - mail_auth.html
 - storage_memo.html
 - storage_memo.html.s
 - storage.html
 - .DS_Store
 - .firebase
 - database.rules.json
 - firebase-debug.log
 - firebase.json

```

115 // TODO 5. 파일 선택이 안됐을 경우 등록 방지
116 $("#PROGRESS").show(); // 프리로더 보이기
117 /* TODO 5. uploadTask를 이용해 선택된 파일을 등록 */
118 /* 2. DB에 내용을 기록 */
119 var new_memo = $("#INPUT_MESSAGE").val();
120 var _filename = "unknown"; // 4. TODO 파일명으로 수정
121 firebase.database().ref('/filememo').push({
122     "memo" : new_memo ,
123     "time" : firebase.database.ServerValue.TIMESTAMP ,
124     "filename" : _filename
125 });
126
127 /* 2. 폼 및 요소 초기화 */
128 $("#form").each(function() {
129     this.reset();
130 });
131 // TODO 4. 전역 파일 객체 초기화
132 $("#PROGRESS").hide(); // 프리로더 감추기
133 });
134
135 /* TODO 4. 파일객체 가져오기 */
136
137 /* 3. DB에서 메모 내용 읽어오기 */
138 firebase.database().ref('/filememo').on('value', function(snapshot) {
139     $("#PROGRESS").show(); // 프리로더 보이기
140     $("#READ_MEMO").html("");

```

DB에 입력할때 실제 파일 이름으로 들어가게 수정합니다

- js
 - storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - 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_profile.html
 - hosting.html
 - index.html
 - index2.html
 - mail_auth.html
 - storage_memo.html
 - storage_memo.html.s
 - storage.html
- .DS_Store
- .firebase
- database.rules.json
- firebase-debug.log
- firebase.json

```

115 // TODO 5. 파일 선택이 인풋을 경우 등록 방식
116 $("#PROGRESS").show(); // 프리로더 보이기
117 /* TODO 5. uploadTask를 이용해 선택된 파일을 등록 */
118 /* 2. DB에 내용을 기록 */
119 var new_memo = $("#INPUT_MESSAGE").val();
120 var _filename = _file.name; // 4. 파일명으로 수정
121 firebase.database().ref('/filememo').push({
122   "memo" : new_memo ,
123   "time" : firebase.database.ServerValue.TIMESTAMP ,
124   "filename" : _filename
125 });
126
127 /* 2. 폼 및 요소 초기화 */
128 $("form").each(function() {
129   this.reset();
130 });
131 _file = null; // 4. 전역 파일 객체 초기화
132 $("#PROGRESS").hide(); // 프리로더 감추기
133 });
134
135 /* 4. 파일객체 가져오기 */
136 var _file = null;
137 document.getElementById("INPUT_FILE").addEventListener("change", handleFiles, false);
138 function handleFiles() {
139   _file = (this.files)[0];
140 }
141

```

- js
- storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
- 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_profile.html
- hosting.html
- index.html
- index2.html
- mail_auth.html
- storage_memo.html
- storage_memo.html.s
- storage.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase-debug.log
- firebase.json

```

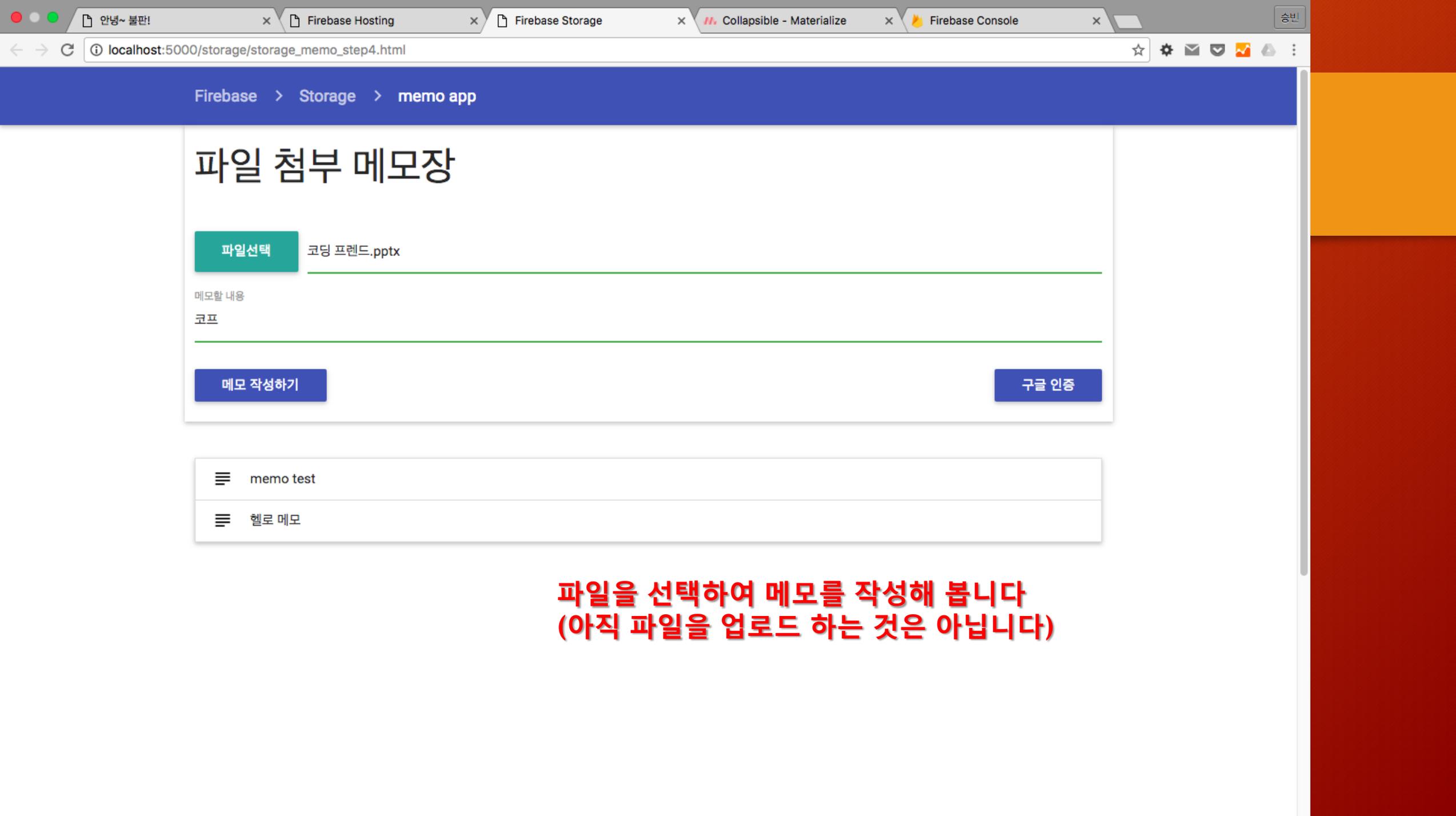
123     "time" : firebase.database.ServerValue.TIMESTAMP ,
124     "filename" : _filename
125   });
126
127   /* 2. 폼 및 요소 초기화 */
128   $("form").each(function() {
129     this.reset();
130   });
131   // TODO 4. 전역 파일 객체 초기화
132   $("#PROGRESS").hide(); // 프리로더 감추기
133 });
134
135 /* TODO 4. 파일객체 가져오기 */
136
137 /* 3. DB에서 메모 내용 읽어오기 */
138 firebase.database().ref('/filememo').on('value', function(snapshot) {
139   $("#PROGRESS").show(); // 프리로더 보이기
140   $("#READ_MEMO").html("");
141   if (snapshot.val() == null ) {
142     $("#READ_MEMO").append('<div class="collapsible-header">아직 등록된 파일이 없습니다.</div>');
143     $("#PROGRESS").hide();
144     return false;
145   }
146   /* 3. 메모 객체 하나씩 li요소로 변경해서 ul(#READ_MEMO)에 추가 */
147   var _keys = Object.keys(snapshot.val());
148   for (var i=_keys.length; 0<i; i--){

```

등록 되면 전역 객체도 초기화합니다

- js
- storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
- 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_profile.html
- hosting.html
- index.html
- index2.html
- mail_auth.html
- storage_memo.html
- storage_memo.html.s
- storage.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase-debug.log
- firebase.json

```
123     "time" : firebase.database.ServerValue.TIMESTAMP ,
124     "filename" : _filename
125   });
126
127   /* 2. 폼 및 요소 초기화 */
128   $("form").each(function() {
129     this.reset();
130   });
131   _file = null; // 4. 전역 파일 객체 초기화
132   $("#PROGRESS").hide(); // 프리로더 감추기
133 });
134
135 /* 4. 파일객체 가져오기 */
136 var _file = null;
137 document.getElementById("INPUT_FILE").addEventListener("change", handleFiles, false);
138 function handleFiles() {
139   _file = (this.files)[0];
140 }
141
142 /* 3. DB에서 메모 내용 읽어오기 */
143 firebase.database().ref('/filememo').on('value', function(snapshot) {
144   $("#PROGRESS").show(); // 프리로더 보이기
145   $("#READ_MEMO").html("");
146   if (snapshot.val() == null ) {
147     $("#READ_MEMO").append('<div class="collapsible-header">아직 등록된 파일이 없습니다.</div>');
148     $("#PROGRESS").hide();
149   }
150   return false;
151 }
```



파일 첨부 메모장

파일선택

코딩 프렌드.pptx

메모할 내용

코프

메모 작성하기

구글 인증

☰ memo test

☰ 헬로 메모

**파일을 선택하여 메모를 작성해 봅니다
(아직 파일을 업로드 하는 것은 아닙니다)**

- Analytics
- DEVELOP
- Auth
- Database
- Storage
- Hosting
- Remote Config
- Test Lab
- Crash
- GROW
- Notifications
- Dynamic Links
- EARN
- AdMob
- Spark 무료 업그레이드

https://hellofirebase4web.firebaseio.com/

기본 권한 규칙을 사용하면 사용자가 인증을 받아야 합니다. 자세히 알아보기 닫기

```
hellofirebase4web
├── filememo
│   ├── -K0iSCPqLN7Ufi3dhNhR
│   │   ├── filename: "unknown"
│   │   ├── memo: "헬로 메모"
│   │   └── time: 1470731966332
│   ├── -K0iTo_OevSOgrlpFbSD
│   │   ├── filename: "unknown"
│   │   ├── memo: "memo test"
│   │   └── time: 1470732388892
│   └── -K0iXfvLvdGeMAe3f-Z1
│       ├── filename: "코딩 프렌드.pptx"
│       ├── memo: "코프"
│       └── time: 1470733402125
└── onUsers
    ├── 698QxDGQWHYqmQA7NuTMdClqJV01
    │   ├── siteOn: 0
    │   └── username: "SBMac Jang"
    └── mbLfZ20MsyP2qM0e90214yCVC8F3
        ├── siteOn: 0
```

파일 이름까지 DB에 같이 잘 들어가는지 확인해 봅시다!

- js
- storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
- 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_profile.html
- hosting.html
- index.html
- index2.html
- mail_auth.html
- storage_memo.html
- storage_memo.html.si
- storage.html
- .DS_Store
- .firebase
- database.rules.json
- firebase-debug.log
- firebase.json

```

115 // TODO 5. 파일 선택이 안됐을 경우 등록 방지
116 $("#PROGRESS").show(); // 프리로더 보이기
117 /* TODO 5. uploadTask를 이용해 선택된 파일을 등록 */
118 /* 2. DB에 내용을 기록 */
119 var new_memo = $("#INPUT_MESSAGE").val();
120 var _filename = _file.name; // 4. 파일명으로 수정
121 firebase.database().ref('/filememo').push({
122   "memo" : new_memo ,
123   "time" : firebase.database.ServerValue.TIMESTAMP ,
124   "filename" : _filename
125 });
126
127 /* 2. 폼 및 요소 초기화 */
128 $("form").each(function() {
129   this.reset();
130 });
131 _file = null; // 4. 전역 파일 객체 초기화
132 $("#PROGRESS").hide(); // 프리로더 감추기
133 });
134
135 /* 4. 파일객체 가져오기 */
136 var _file = null;
137 document.getElementById("INPUT_FILE").addEventListener("change", handleFiles, false);
138 function handleFiles() {
139   _file = (this.files)[0];
140 }

```

파일을 업로드 해 봅니다!

- js
- storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
- 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_profile.html
- hosting.html
- index.html
- index2.html
- mail_auth.html
- storage_memo.html
- storage_memo.html.s
- storage.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase-debug.log
- firebase.json

```
119     }
120     $("#PROGRESS").show(); // 프리로더 보이기
121     /* 5. uploadTask를 이용해 선택된 파일을 등록 */
122     var storageRef = firebase.storage().ref(); // 스토리지의 root
123     var uploadTask = storageRef.child('filememo/' + _file.name).put(_file);
124     uploadTask.on('state_changed', function(snapshot){
125         var progress = (snapshot.bytesTransferred / snapshot.totalBytes) * 100;
126         console.log('Upload is ' + progress + '% done');
127         // 참고 :
128         •
129         https://firebase.google.com/docs/storage/web/upload-files#monitor_upload_progress
130     }, function(error) { // 오류났을때
131         alert(error.message);
132         return false;
133     }, function() {
134         // TODO 5. 업로드 완료 시에
135     });
136
137     /* 2. DB에 내용을 기록 */
138     var new_memo = $("#INPUT_MESSAGE").val();
139     var _filename = _file.name; // 4. 파일명으로 수정
140     firebase.database().ref('/filememo').push({
141         "memo" : new_memo ,
142         "time" : firebase.database.ServerValue.TIMESTAMP ,
143         "filename" : _filename
144     });
```

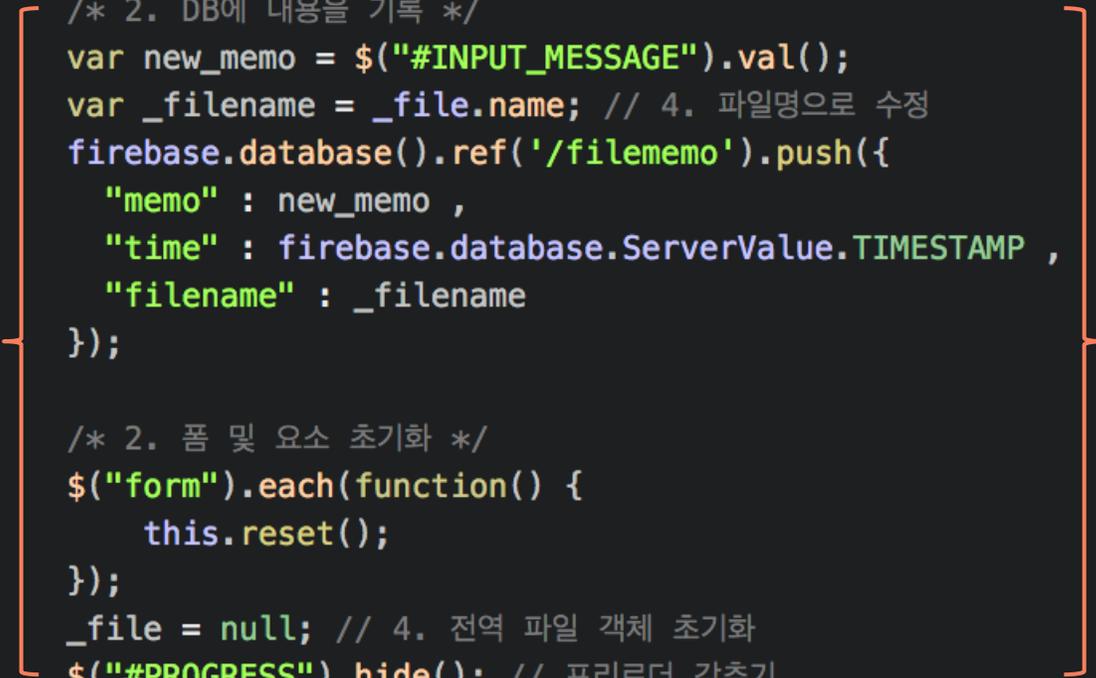
- js
- storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
- 404.html
- auth_firebase_ui.html
- auth_google.html
- auth_mail.html
- auth_term.html
- auth.html
- database_basic.html
- database_siteon.html
- database.html
- fireinit.js
- fireinit.js.sample
- hosting_profile.html
- hosting.html
- index.html
- index2.html
- mail_auth.html
- storage_memo.html
- storage_memo.html.s
- storage.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase-debug.log
- firebase.json

```

127 // 참고 :
128 https://firebase.google.com/docs/storage/web/upload-files#monitor_upload_progress
129 }, function(error) { // 오류났을때
130 alert(error.message);
131 return false;
132 }, function() {
133 // TODO 5. 업로드 완료 시에
134 });
135
136 /* 2. DB에 내용을 기록 */
137 var new_memo = $("#INPUT_MESSAGE").val();
138 var _filename = _file.name; // 4. 파일명으로 수정
139 firebase.database().ref('/filememo').push({
140   "memo" : new_memo ,
141   "time" : firebase.database.ServerValue.TIMESTAMP ,
142   "filename" : _filename
143 });
144
145 /* 2. 폼 및 요소 초기화 */
146 $("#form").each(function() {
147   this.reset();
148 });
149 _file = null; // 4. 전역 파일 객체 초기화
150 $("#PROGRESS").hide(); // 프리로더 감추기
151 });

```

내용 기록과 초기화 코드를
파일 업로드 이후로 옮겨줍니다



- js
 - storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - 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_profile.html
 - hosting.html
 - index.html
 - index2.html
 - mail_auth.html
 - storage_memo.html
 - storage_memo.html.s
 - storage.html
 - .DS_Store
 - .firebaseerc
 - database.rules.json
 - firebase-debug.log
 - firebase.json

```

storage.html x storage_memo_step4.html x storage_memo_step5.html o
https://firebase.google.com/docs/storage/web/upload-files#monitor_upload_progress
128 }, function(error) { // 오류났을때
129     alert(error.message);
130     return false;
131 }, function() {
132     // 5. 업로드 완료 시에
133     /* 2. DB에 내용을 기록 */
134     var new_memo = $("#INPUT_MESSAGE").val();
135     var _filename = _file.name; // 4. 파일명으로 수정
136     firebase.database().ref('/filememo').push({
137         "memo" : new_memo ,
138         "time" : firebase.database.ServerValue.TIMESTAMP ,
139         "filename" : _filename
140     });
141
142     /* 2. 폼 및 요소 초기화 */
143     $("#form").each(function() {
144         this.reset();
145     });
146     _file = null; // 4. 전역 파일 객체 초기화
147     $("#PROGRESS").hide(); // 프리로더 감추기
148 });
149 });
150
151 /* 4. 파일객체 가져오기 */
152 var _file = null;
153 document.getElementById("INPUT_FILE").addEventListener("change" handleFiles, false);

```

- js
- storage
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
 - storage_memo_ste
- 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_profile.html
- hosting.html
- index.html
- index2.html
- mail_auth.html
- storage_memo.html
- storage_memo.html.s
- storage.html
- .DS_Store
- .firebase
- database.rules.json
- firebase-debug.log
- firebase.json

```

112
113  /* 2. 메모하기 눌렀을 때 DB에 기록 */
114  $("#BTN_WRITE").click(function(){
115      // TODO 5. 파일 선택이 안됐을 경우 등록 방지
116      $("#PROGRESS").show(); // 프리로더 보이기
117      /* TODO 5. uploadTask를 이용해 선택된 파일을 등록 */
118      /* 2. DB에 내용을 기록 */
119      var new_memo = $("#INPUT_MESSAGE").val();
120      var _filename = _file.name; // 4. 파일명으로 수정
121      firebase.database().ref('/filememo').push({
122          "memo" : new_memo ,
123          "time" : firebase.database.ServerValue.TIMESTAMP ,
124          "filename" : _filename
125      });
126
127      /* 2. 폼 및 요소 초기화 */
128      $("form").each(function() {
129          this.reset();
130      });
131      _file = null; // 4. 전역 파일 객체 초기화
132      $("#PROGRESS").hide(); // 프리로더 감추기
133  });
134
135      /* 4. 파일객체 가져오기 */
136      var _file = null;
137      document.getElementById("INPUT_FILE").addEventListener("change", handleFiles, false);

```

파일을 선택 안하면 업로드를 불가능하도록 임시로 방지합니다

- js
- storage
 - storage_memo_ste
 - 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_profile.html
 - hosting.html
 - index.html
 - index2.html
 - mail_auth.html
 - storage_memo.html
 - storage_memo.html.s
 - storage.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase-debug.log
- firebase.json

```

storage.html x storage_memo_step4.html x storage_memo_step5.html x
112
113 /* 2. 메모하기 눌렀을 때 DB에 기록 */
114 $("#BTN_WRITE").click(function(){
115     // 5. 파일 선택이 안됐을 경우 등록 방지
116     if( _file == null ){
117         alert("업로드할 파일을 선택해 주세요");
118         return false;
119     }
120     $("#PROGRESS").show(); // 프리로더 보이기
121     /* 5. uploadTask를 이용해 선택된 파일을 등록 */
122     var storageRef = firebase.storage().ref(); // 스토리지의 root
123     var uploadTask = storageRef.child('filememo/' + _file.name).put(_file);
124     uploadTask.on('state_changed', function(snapshot){
125         var progress = (snapshot.bytesTransferred / snapshot.totalBytes) * 100;
126         console.log('Upload is ' + progress + '% done');
127         // 참고 :
128         • https://firebase.google.com/docs/storage/web/upload-files#monitor_upload_progress
129     }, function(error) { // 오류났을때
130         alert(error.message);
131         return false;
132     }, function() {
133         // 5. 업로드 완료 시에
134         /* 2. DB에 내용을 기록 */
135         var new_memo = $("#INPUT_MESSAGE").val();
136         var _filename = _file.name; // 4. 파일명으로 수정
137         firebase.database().ref('/filememo').push({
138             "memo" : new_memo

```

파일 첨부 메모장

파일선택

StartupAlarm.zip

메모할 내용

테스트 업로드

메모 작성하기

구글 인증

memo test

헬로 메모

파일을 선택하여 메모를 다시 작성해 봅니다

- Home HelloFirestore4Web
- Analytics
- DEVELOP
 - Auth
 - Database
 - Storage
 - Hosting
 - Remote Config
 - Test Lab
 - Crash
- GROW
 - Notifications
 - Dynamic Links
- EARN
 - AdMob

Realtime Database

데이터 RULES USAGE

https://hellofirebase4web.firebaseio.com/

★ 기본 권한 규칙을 사용하면 사용자가 인증을 받아야 합니다. 자세히 알아보기 닫기

```
hellofirebase4web
├── filememo
│   ├── -KOiSCPqLN7Ufi3dhNhR
│   │   ├── filename: "unknown"
│   │   ├── memo: "헬로 메모"
│   │   └── time: 1470731966332
│   ├── -KOiTo_OevS0grlpFbSD
│   │   ├── filename: "unknown"
│   │   ├── memo: "memo test"
│   │   └── time: 1470732388892
│   └── -KOiZGDy97GaGe82oK7o
│       ├── filename: "StartupAlarm.zip"
│       ├── memo: "테스트 업로."
│       └── time: 1470733816967
└── onUsers
    ├── 698QxDGQWHYqmQA7NuTMdClqJV01
    │   ├── siteOn: 0
    │   └── username: "SBMac Jang"
    └── mbLfZ20MsvP2qM0e90214vCVC8F3
```

Firestore console에서 우선 DB확인해 봅니다

- Home HelloFirestore4Web
- Analytics
- DEVELOP
 - Auth
 - Database
 - Storage
 - Hosting
 - Remote Config
 - Test Lab
 - Crash
- GROW
 - Notifications
 - Dynamic Links
- EARN
 - AdMob
- Spark 무료 업그레이드

Storage

FILES 규칙

gs://hellofirebase4web.appspot.com > filememo

파일 업로드 +

<input type="checkbox"/>	이름	크기	유형	최종 수정 시간
<input type="checkbox"/>	StartupAlarm.zip	1.51MB	application/zip	2016. 8. 9.

그리고 이번엔 storage를 확인해 봅니다.

- js
- storage
 - storage_memo_ste
 - 404.html
 - auth_firebase_ui.html
 - auth_google.html
 - auth_mail.html
 - auth_term.html
 - auth.html
 - database_basic.html
 - database_sideon.html
 - database.html
 - fireinit.js
 - fireinit.js.sample
 - hosting_profile.html
 - hosting.html
 - index.html
 - index2.html
 - mail_auth.html
 - storage_memo.html
 - storage_memo.html.s
 - storage.html
- .DS_Store
- .firebase
- database.rules.json
- firebase-debug.log
- firebase.json

```

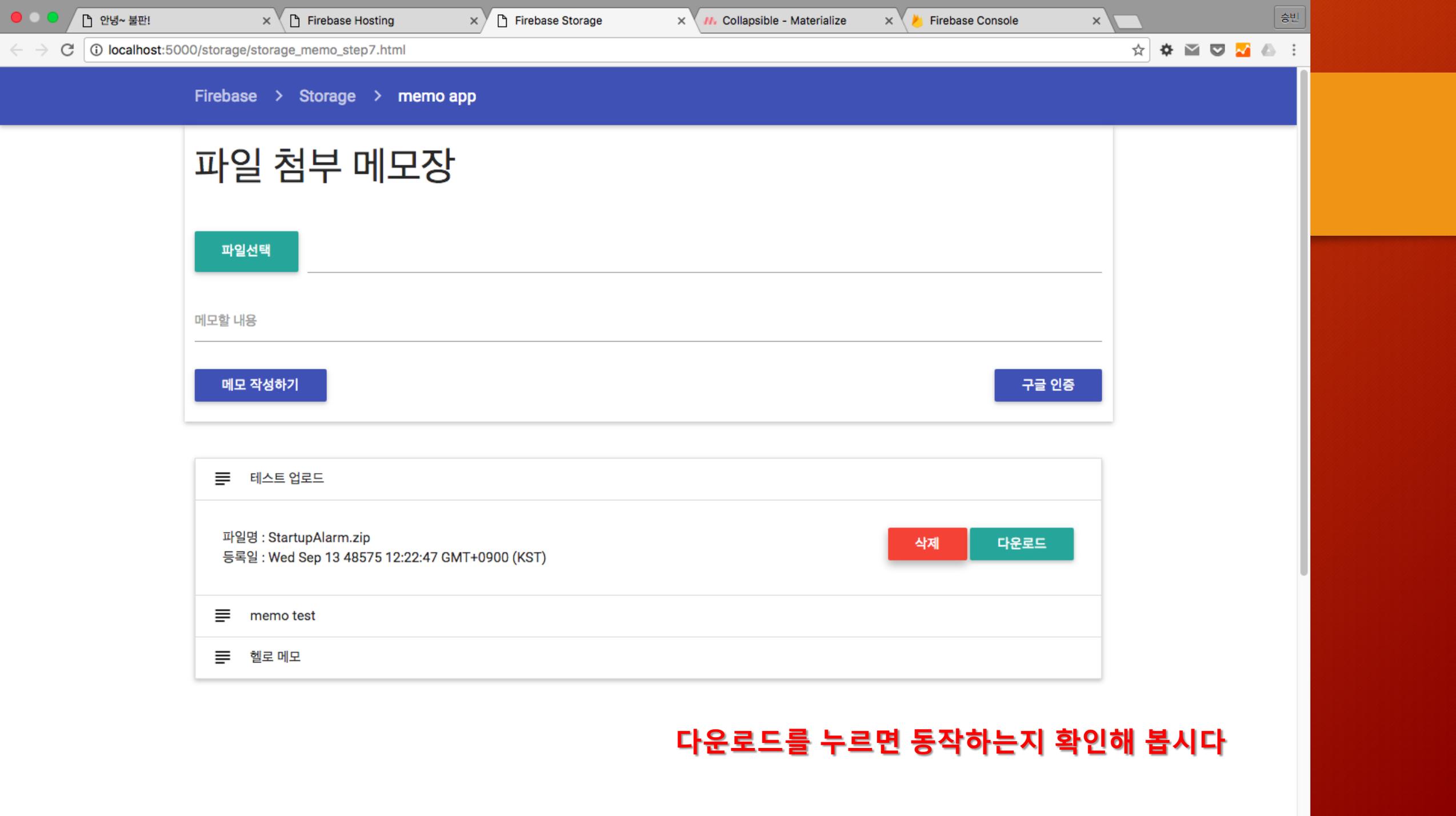
175 + '      <div class="collapsible-body">'
176 + '          <p><span class="right"><button data-f="'+child.filename+'
      • data-k="'+_keys[(i-1)]+'" class="btn red BTN_DELETE">삭제</button>'
177 + '          <button data-f="'+child.filename+'" class="btn BTN_DOWNLOAD">다운로드</button>'
178 + '          </span>파일명 : '+child.filename+'<br>등록일 : '+uploadDate+'</p>'
179 + '      </div>'
180 + '  </li>';
181      $("#READ_MEMO").append(appendHTML);
182  }
183  $("#PROGRESS").hide(); // 프리로더 감추기
184
185  /* 3. 삭제 이벤트 연결 */
186  $(".BTN_DELETE").off("click");
187  $(".BTN_DELETE").click(function(){
188      // TODO 7. 파일도 삭제 하기
189      var _key = $(this).data("k");
190      firebase.database().ref('/filememo').child(_key).remove();
191  });
192  /* TODO 6. 다운로드 작성 */
193  });
194
195  </script>
196
197  </body>
198  </html>
199

```

파일을 웹에서 다운로드 받을 수 있도록 HTTP URL을 구하는 코드를 작성합니다

- js
 - storage
 - storage_memo_ste
 - 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_profile.html
 - hosting.html
 - index.html
 - index2.html
 - mail_auth.html
 - storage_memo.html
 - storage_memo.html.s
 - storage.html
 - .DS_Store
 - .firebaserc
 - database.rules.json
 - firebase-debug.log
 - firebase.json

```
189     firebase.storage().ref('filememo/'+_filename).delete().then(function() {
190         // 파일 삭제 성공
191     }).catch(function(error) {
192         alert(error.message);
193     });
194     var _key = $(this).data("k");
195     firebase.database().ref('/filememo').child(_key).remove();
196 });
197     /* 6. 다운로드 작성 */
198     $(".BTN_DOWNLOAD").off("click");
199     $(".BTN_DOWNLOAD").click(function(){
200         var _filename = $(this).data("f");
201         firebase.storage().ref('filememo/'+_filename).getDownloadURL().then(function(url) {
202             console.log(url);
203             location.href=url;
204         }).catch(function(error) {
205             alert(error.message);
206         });
207     });
208 });
209
210 </script>
211
212 </body>
213 </html>
214
```



파일 첨부 메모장

파일선택

메모할 내용

메모 작성하기

구글 인증

☰ 테스트 업로드

파일명 : StartupAlarm.zip
등록일 : Wed Sep 13 48575 12:22:47 GMT+0900 (KST)

삭제

다운로드

☰ memo test

☰ 헬로 메모

다운로드를 누르면 동작하는지 확인해 봅시다

- js
- storage
 - storage_memo_ste
 - 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_profile.html
 - hosting.html
 - index.html
 - index2.html
 - mail_auth.html
 - storage_memo.html
 - storage_memo.html.s
 - storage.html
- .DS_Store
- .firebaserc
- database.rules.json
- firebase-debug.log
- firebase.json

```
183
184   /* 3. 삭제 이벤트 연결 */
185   $(".BTN_DELETE").off("click");
186   $(".BTN_DELETE").click(function(){
187     // TODO 7. 파일도 삭제 하기
188     var _key = $(this).data("k");
189     firebase.database().ref('/filememo').child(_key).remove();
190   });
191   /* 6. 다운로드 작성 */
192   $(".BTN_DOWNLOAD").off("click");
193   $(".BTN_DOWNLOAD").click(function(){
194     var _filename = $(this).data("f");
195     firebase.storage().ref('filememo/'+_filename).getDownloadURL().then(function(url) {
196       console.log(url);
197       location.href=url;
198     }).catch(function(error) {
199       alert(error.message);
200     });
201   });
202 });
203
204 </script>
205
206 </body>
207 </html>
208
```

마지막으로 파일을 삭제하는 코드를 작성합니다

파일 첨부 메모장

파일선택

메모할 내용

메모 작성하기

구글 인증

☰ 테스트 업로드

파일명 : StartupAlarm.zip
등록일 : Wed Sep 13 48575 12:22:47 GMT+0900 (KST)

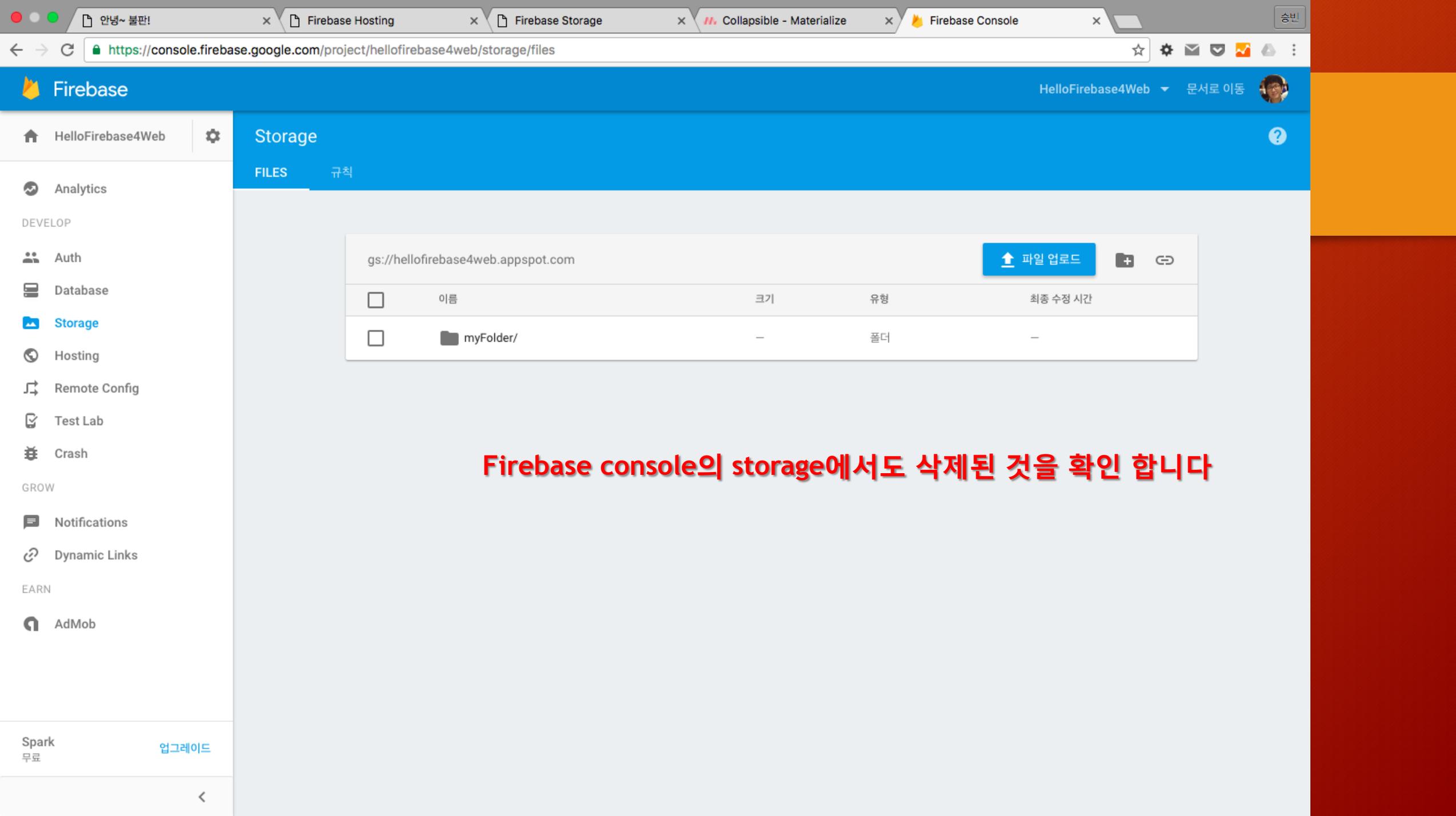
삭제

다운로드

☰ memo test

☰ 헬로 메모

파일이 첨부된 메모를 삭제해 봅니다



gs://hellofirebase4web.appspot.com

[파일 업로드](#) [+](#) [🔗](#)

<input type="checkbox"/>	이름	크기	유형	최종 수정 시간
<input type="checkbox"/>	📁 myFolder/	-	폴더	-

Firebase console의 storage에서도 삭제된 것을 확인 합니다

- Analytics
- DEVELOP
- Auth
- Database
- Storage**
- Hosting
- Remote Config
- Test Lab
- Crash
- GROW
- Notifications
- Dynamic Links
- EARN
- AdMob

Firestore for Web 정리

Web 개발을 위한 Firebase의 서비스



Database

연결된 모든 클라이언트에서 실시간으로 데이터를 저장하고 동기화할 수 있습니다.

[자세히 알아보기](#)

[시작하기](#)



Auth

서버측 코드 없이 다양한 제공업체의 사용자를 인증하고 관리할 수 있습니다.

[자세히 알아보기](#)

[시작하기](#)



Storage

서버측 코드 없이 이미지, 오디오, 동영상과 같은 사용자가 생성한 콘텐츠를 저장하고 가져올 수 있습니다.

[자세히 알아보기](#)

[시작하기](#)



Hosting

보안 글로벌 콘텐츠 전송 네트워크를 사용하면 몇 초 만에 웹 앱과 모바일 앱을 배포할 수 있습니다.

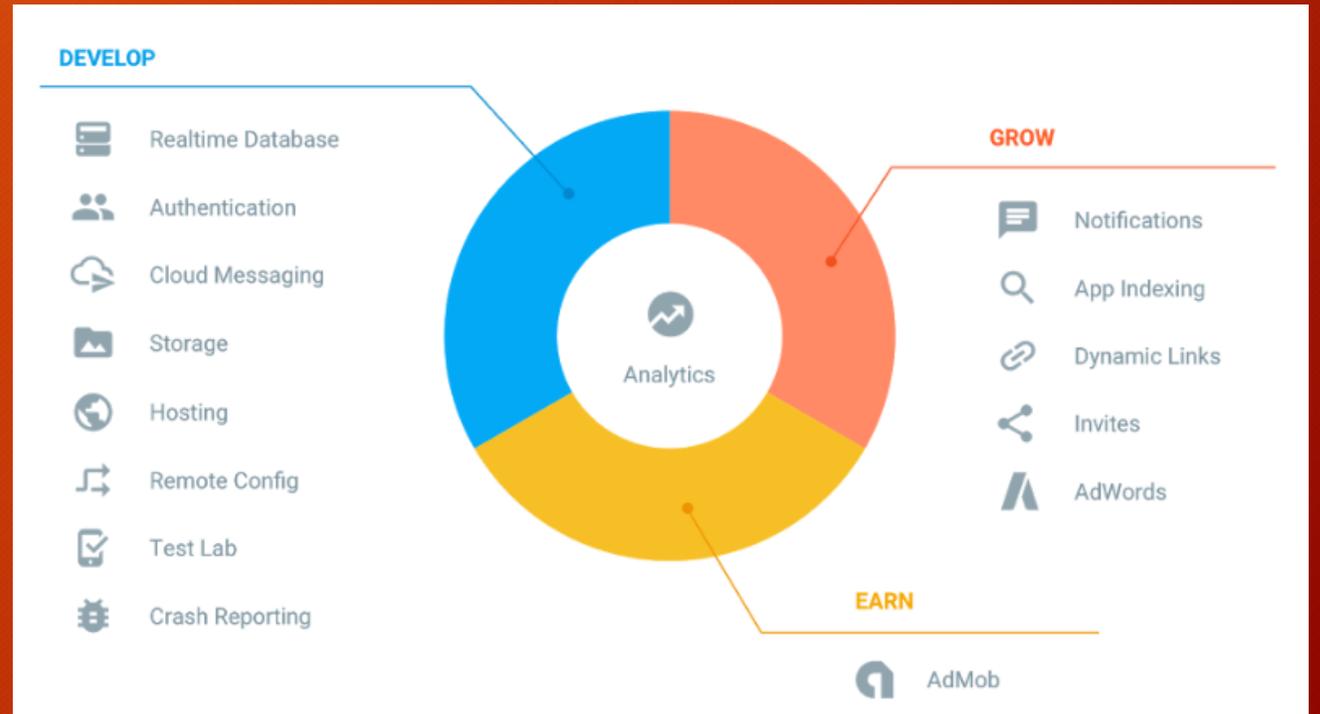
[자세히 알아보기](#)

[시작하기](#)

실시간 DB만으로도 웹 개발 도구로서의 가치가 충분히 높지만

하지만 Firebase는 ios, android, web 통합 플랫폼!

웹과 함께하는 앱 개발을 한다면
Firebase는 이제 시작에 불과합니다

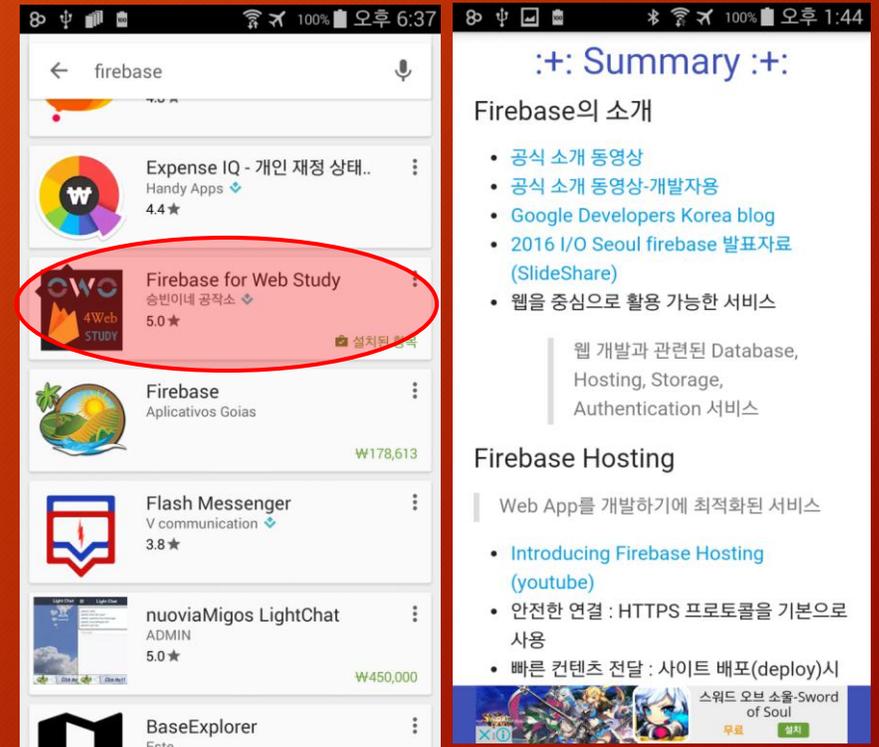


더 나아가고 싶다면 다음을 꼭 참고해 보세요

- Youtube firecasts 채널
 - https://www.youtube.com/channel/UCP4bf6IHJJQehibu6ai__cq
- Firebase 공식 sample들
 - <https://firebase.google.com/docs/samples/>
- 2016 Google I/O Code lab
 - <https://codelabs.developers.google.com/io2016?cat=Firebase>

그리고!

- 우리가 공부한 내용을 APP으로 만들었습니다
 - 안드로이드 play store에서 firebase를 검색
 - <https://play.google.com/store/apps/details?id=kr.sbsoft.firebase4android>
 - (실수로 배너 광고를 눌러 주시면 정말 감사하겠습니다)
 - 아이폰은 조금 더 기다려 주세요
- for Android, for iOS 기대(?!)해 주세요
 - Mobile(Online)에서





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

모두의 코딩

by ABCD

매주 수요일, 모비프렌

장르, 시간 불문

개인 프로젝트 하러 오셔도 됩니다

질문 있으신가요?

오늘의T

Posted by 차정인기자의 T-타임

794 Views

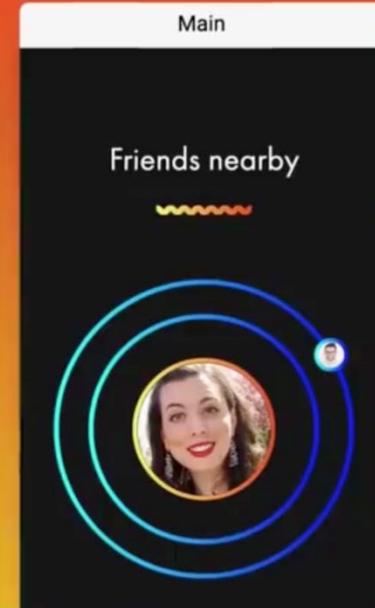
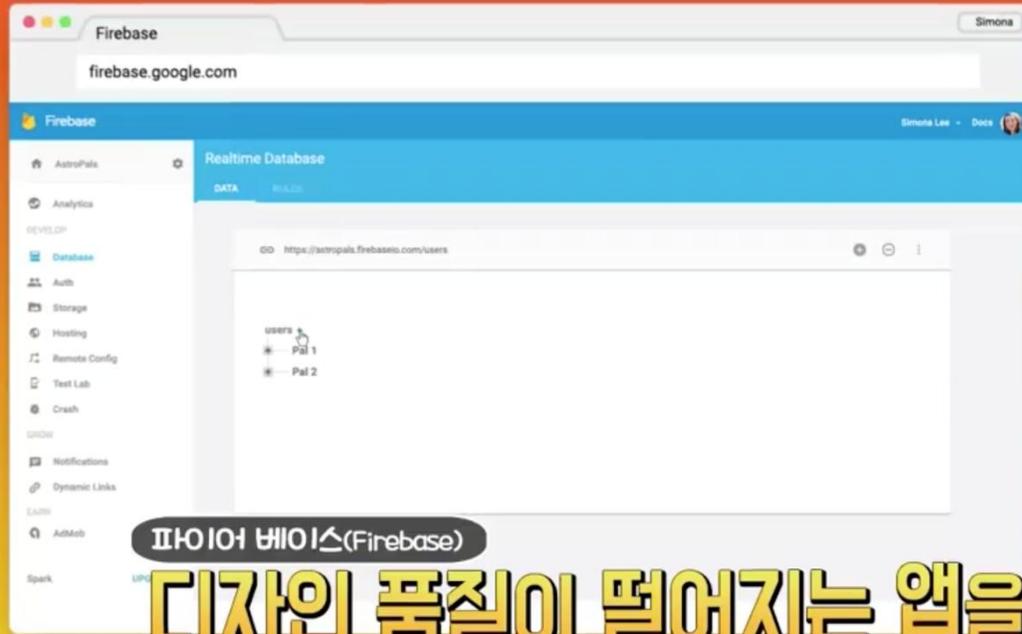


Weds 4:04 PM

Simona Lee

KBS

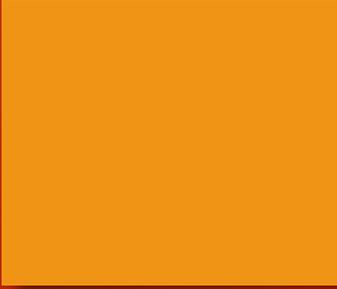
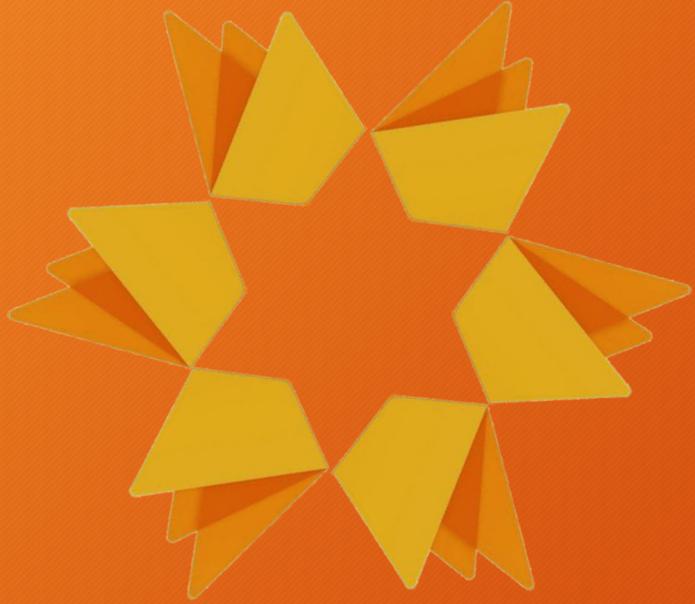
출처: Google Developers



파이이어 베이스(Firebase)

디자인 품질이 떨어지는 앱을 꾸며주는 도구

이제 어디가서 이러면 안됩니다(?!)



수고 많으셨습니다

