# Firestore

## Setting up Firebase

```html
<!-- import firebase library -->
<script src="https://www.gstatic.com/firebasejs/7.14.5/firebase-app.js"></script>
<!-- use firestore  -->
<script src="https://www.gstatic.com/firebasejs/7.14.5/firebase-firestore.js">
</script>
```

```javascript
//script
var firebaseConfig = {
  // ...
};
// Initialize Firebase
firebase.initializeApp(firebaseConfig);

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

## Getting documents

```javascript
// grab data as a snapshot, get data from each doc
db.collection('collection').get().then(snapshot => {
  snapshot.docs.forEach(doc => {
    // do stuff
  });
})
```

`elem.setAttribute('data-id', doc.id)`

**`elem.textContent = doc.data().field`**

- Sets an HTML's text content to the corresponding `field` of a firebase document

# Saving data

**`collection().add()`**

```
db.collection('collection').add({
  field: value,
});
```

- Adds a new document with the fields given to the Firestore collection

# Deleting data

**`collection().doc().delete()`**

```
let id = elem.getAttribute('data-id');
db.collection('collection').doc(id).delete();
```

- Deletes the document in the Firestore collection with the `id` given

# Queries

**`collection().where()`**

- Takes 3 parameters: field, evaluation, equal

```
db.collection('collection').where('name', '==', 'danny').get().then(snapshot => {
```

```
db.collection('collection').where('name', '==', 'danny').get().then(snapshot => {
  snapshot.docs.forEach(doc => {
    // do stuff

  });
})
```

- Retrieves the documents where field `name` is `danny`
- Can chain `.where().where().where()`

**Error: The query requires an index**

- Firebase requires an index for some queries, just click on the error link to fix it

# Ordering data

**`collection().orderBy()`**

- Order documents by fields

```
db.collection('collection').orderBy('name').get().then(snapshot => {
  snapshot.docs.forEach(doc => {
    // do stuff
  });
})
```

- Documents retrieved are ordered by the `name` field

# Real-Time Data

- fire a function to react on snapshot

```
db.collection('collection').onSnapshot(snapshot => {
  let changes = snapshot.docChanges();
  let changes = snapshot.docChanges();
  changes.forEach(change => {
    if(change.type == 'added'){
      // ...
    } else if(change.type == 'removed'){
      // ...
    }
  });
});
```

- Upon load, initial documents in collection are added changes
- DocumentChange types: `added`, `modified`, `removed`

# Updating Data

**`doc().update()`**

```
db.collection('collection').doc(id).update({
  name: 'new name',
});
```

- Updates the selected document's `name` field to be `'new name'`

**`doc().set()`**

```
db.collection('collection').doc(id).set({
  // name is now undefined
  city: 'my city',
});
```

- Similar to `doc().update()`, however completely overwrites document properties

# Firestore Documentation

**Danny Wu**