Get all from collection

```
this.db.collection(collection)
.valueChanges()
.subscribe();
```

Returns the current state of the collection (observable).
Is great for only displaying data, we don't receive any metadata and id.

Querying snapshotChanges vs stateChanges

```
this.db.collection(collection)
.valueChanges().subscribe(
    snaps => {
       const collection =
snaps.map( snap => {
          return {
             id: snap.payl-
oad.doc.id,
             ...snap.payl-
oad.doc.data()
          }
       }
    }
);
```

Receive document snapshot with type and payload. Returns an observable with all data.

Querying snapshotChanges vs stateChanges

```
this.db.collection(collection)
.stateChanges().subscribe(
    snaps => {
       const collection =
snaps.map( snap => {
          return {
             id: snap.payl-
oad.doc.id,
             ...snap.payl-
oad.doc.data()
          }
       }
    }
```

Receive document snapshot with type and payload. Returns an observable with only updated data (live updated).

Querying with ordering

```
this.db.collection(collection, ref => {
   return ref.orderBy(param).where(param,
"==", what)
})
.snapshotChanges()...}}
...
.where(param, "== , >, <= or any",
what)...
.where(param, "array-contains", what)...
.startAt(number).endAt(number)... (first
element included)
.startAfter(number).endAt(number)...(-
first element is not included)
```

Ordered by params.

Update collection

```
this.db.doc(collection/id).set(changes)...
this.db.doc(collection/id).update(chang-
es)...
```

set - update the collection or create one if not exist
update - only update the collection

Batch collection

```
const fRef =
this.db.doc(collection/id).ref;
const oRef = this.db.doc(collection/i-
d).ref;
const batch = this.db.firestore.batch();
batch.update(fRef, { updates });
...
const batch$ = of(batch.commit());
batch$.subscribe();
```

Updates documents by its ref. Up to 500 data operation in the same batch.

Delete collection

```
const d = this.db
.collection(collection);
return d.doc(id).delete();
```

Delete document by id

Simple file upload

```
private storage:
AngularFireStorage;
...
upload(event) {
 const file: File =
event.target.files[0];
 const path = collecti-
on/id/filename;
 const task = this.stor-
age.upload (path, file);
 task.snapshotChang-
es().subscribe();
}
...
task.percentageChanges()
- get current upload
percentage
```

File upload info and metadata

```
profileUrl:
Observable<string | null>;
  constructor(private
storage: AngularFireS-
torage) {
    const ref = this.s-
torage.ref('users/davide-
ast.jpg');
    this.profileUrl =
ref.getDownloadURL();
  }
...
meta: Observable<any>;
  constructor(private
storage: AngularFireS-
torage) {
    const ref = this.s-
torage.ref('users/davide-
ast.jpg');
    this.meta = ref.ge-
tMetadata();
  }
...
uploadFile(event) {
    const file = event.t-
arget.files[0];
    const filePath =
'name-your-file-path-
here';
    const ref = this.stor-
age.ref(filePath);
    const task = ref.pu-
t(file, { customMetadata:
{ blah: 'blah' } });
  }
```

By **Oleksandr Isaryk** (oisar)
cheatography.com/oisar/
usefur.com