Firebase API

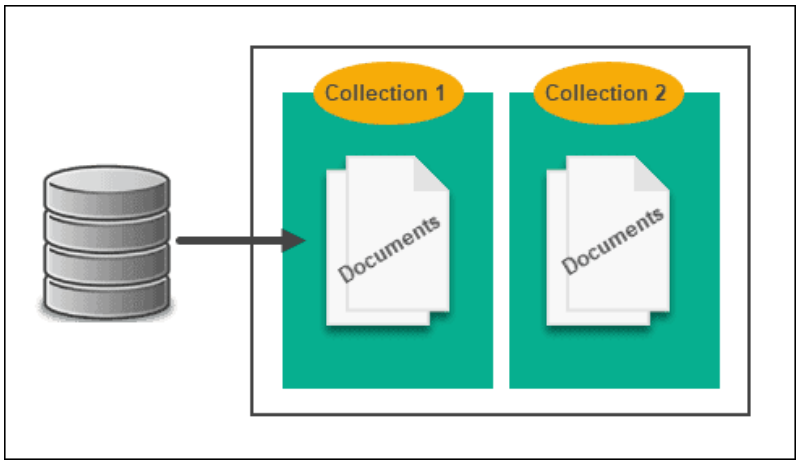
Firebase is used for storing JS object-like documents

CRUD

Create Read Update Delete
Official docs here

Database, Collection, Document, Data

Firebase uses a non-relational and document-oriented database



Getting `db`

```
import app from "./path/to/firebase.ts"
const db = getFirestore(app);
```

Create

Create data with specified ID

Set *must* be called with an ID

```
import { doc, setDoc } from "firebase/firestore"

// "tasks" - collection
// "wake up" - document ID
await setDoc(doc(db, "tasks", "wake up"), {
  taskName: "5am Wake Up",
  category: "routine",
  type: "daily",
  points: 5
});
```

Creating data with automatic id assignment

(try catch is optional but useful for debugging)

```
import { doc, addDoc } from "firebase/firestore"
try {                                    // third arg
is left out
  const docRef = await addDoc(collection(db, "tasks"),
{
    taskName: "5am wake up",
    category: "routine",
    type: "daily",
    points: 5
  })
  console.log("Document written with ID: ", docRef.id)
} catch (e) {
  console.error("Error adding document", e)
}
```

Read

Get all documents in a collection

```
import { doc, getDoc } from "firebase/firestore"

const tasksQuery = query(collection(db, "tasks"))
const tasksDocs = await getDocs(tasksQuery)
tasksDocs.forEach((doc) => {
  // doc.data() is never undefined for query doc
snapshots
  console.log(doc.id, " => ", doc.data());
});
```

Get a document by ID

```
import { doc, getDoc } from "firebase/firestore"

const wakeUpTaskRef = doc(db, "tasks", "wake up")
const wakeUpTaskDoc = await getDoc(wakeUpTaskRef)
```

```
if (wakeUpTaskDoc.exists()) {
  console.log("Document data:", wakeUpTaskDoc.data());
} else {
  // docSnap.data() will be undefined in this case
  console.log("No such document!");
}
```

Update

Update with specified ID

Overwriting

When using `setDoc` on an existing document, it will be **completely** overwritten

```
import { doc, setDoc } from "firebase/firestore"

await setDoc(doc(db, "tasks", "wake up"), {
  taskName: "5am Wake Up",
  category: "routine",
// category isn't included in the request -> the updated document will not include it in the database entry
  points: 5
});
```

Partial update

```
import { doc, updateDoc } from "firebase/firestore"

await updateDoc(doc(db, "tasks", "wake up"), {
  taskName: "5am Wake Up",
  category: "routine",
// category isn't included in the request -> the updated document in the database will keep its old
```

```
value
  points: 5
});
```

Delete

Delete a document

```
import { doc, deleteDoc } from "firebase/firestore";

await deleteDoc(doc(db, "tasks", "wake up"));
```

Delete a field in a document

```
import { doc, updateDoc, deleteField } from
"firebase/firestore";

const wakeUpDocRef = doc(db, 'tasks', 'wake up');

// Remove the 'capital' field from the document
await updateDoc(wakeUpDocRef, {
    category: deleteField()
});
```