



Firebase

Firestore lyssnar till referenser till olika objekt

```
firebase.database().ref("posts")
```

```
firebase.database().ref("users")
```

```
firebase.database().ref("comments")
```

Istället för tabeller har vi objekt, samt referenser till dessa objekt

`.on('value')` lyssnar på våra förändringar i
DB

```
firebase.database().ref("messages")  
  .on('value', function(snapshot){  
    console.log(snapshot.val());  
  });
```

Vi måste plocka ut våra värden med `.val()`

```
firebase.database().ref("messages")  
  .on('value', function(snapshot){  
    console.log(snapshot.key);  
  });
```

Vi kan även komma åt nyckeln med

`snapshot.key`

Observera att key inte är en funktion

```
//Add value
firebase.database().ref("todos").push(stuffToPush);
//Remove value
firebase.database().ref("refToRemove").remove();
//Update value
firebase.database().ref("refToChange").set(true);
```

**MORE
LISTENERS**



`.on('value')` hämtar ALLT i referensen
varje gång

Vi kan lyssna på andra **events** än value

"child_added"

```
firebase.database().ref("todos")  
  .on("child_added", (snapshot) => {  
    console.log(snapshot.val());  
  });
```

Returnerar ENBART det som har lagts till

"child_removed"

```
firebase.database().ref("todos")  
  .on("child_removed", (snapshot) => {  
    console.log(snapshot.val());  
  });
```

Returnerar ENBART det som har tagits bort

"child_changed"

```
firebase.database().ref("todos")  
  .on("child_changed", (snapshot) => {  
    console.log(snapshot.val());  
  });
```

Returnerar ENBART det som har ändrats

Vi kan fortfarande använda `.on('value')`

Men det är mer optimerat att använda dessa
funktioner

Vi hämtar mindre data från databasen varje
gång



Snabbare

Minus för detta

Vi måste skriva mer kod

Inget problem, för det är vårt jobb

**FILTRERING /
SORTERING**

Lättast att göra det mesta i JavaScript faktiskt

Men vi kan optimera våra **queries**

```
.orderByChild("childToOrderBy")
```

```
firebase.database()  
  .ref("todos")  
  .orderByChild("date") //before 'on()' and after 'ref'  
  .on("value", (snapshot) =>{  
    console.log(snapshot.val());  
  });
```

Sorterar informationen som kommer från
databasen via en egenskap

Kopplat till detta har vi olika queries

T.ex. `.limitToFirst()`

```
firebase.database()  
  .ref("todos")  
  .orderByChild("date")  
  .limitToFirst(10) //Grab the first 10  
  .on("value", (snapshot) =>{  
    console.log(snapshot.val());  
  }));
```


`.limitToLast()`

```
firebase.database()  
  .ref("todos")  
  .orderByChild("date")  
  .limitToLast(10) //Grab the last 10  
  .on("value", (snapshot) =>{  
    console.log(snapshot.val());  
  });
```

`.startAt()`

```
firebase.database()  
  .ref("todos")  
  .orderByChild("date")  
  .startAt("2017-01-20") // Start at value  
  .on("value", (snapshot) =>{  
    console.log(snapshot.val());  
  }));
```

- `.limitToFirst()`
- `.limitToLast()`
- `.startAt()`
- `.endAt()`
- `.equalTo()`

Alla måste skrivas i kombination med

`.orderByChild()`