



BASIC CRUD OPERATIONS

MongoDB

Lecture # 01

Outline

- 01 Setup
- 02 node connecting code
- 03 get all admin data
- 04 adding a document
- 05 adding many documents
- 06 Reading one document
- 07 update One Document
- 08 upsert
- 09 update many documents
- 10 delete one document
- 11 delete many document

Setup

WHAT WE'LL COVER IN THIS SESSION

- Open mongodb altlas or cloud
- Create mongodb Cluster
- create user and connect your IP address to the cluster
- press connect button and select application option
- copy code, patse it in Backend file.

node connection code

```
const { MongoClient } = require("mongodb");
async function main(){
  const uri = "mongodb+srv://javi:javi@cluster0.146wlbr.mongodb.net/?retryWrites=true&w=majority"
  const client = new MongoClient(uri)
  try{
    await client.connect()
    console.log("connected")
  }catch(err){
    console.log(err)
  }finally{
    await client.close()
  }
}
main().catch(console.error)
```



get Admin data

```
async function listDatabases(client){  
    databasesList = await client.db().admin().listDatabases();  
    console.log("Databases:");  
    databasesList.databases.forEach(db => console.log(` - ${db.name}`));  
};  
  
const uri = "mongodb+srv://javi:javi@cluster0.146wlbr.mongodb.net/?retryWrites=true&w=majority"  
  
const client = new MongoClient(uri)  
  
await client.connect()  
  
await listDatabases(client);
```

ADDING A DOCUMENT

```
async function createListing(client, newList){  
    const result = await client.db("Cluster0").collection("listing and Reviews").insertOne(newList)  
    console.log(` new listing is created with following id:${result.insertedId}` )  
}  
  
const uri = "mongodb+srv://javi:javi@cluster0.146wlbr.mongodb.net/?retryWrites=true&w=majority"  
const client = new MongoClient(uri)  
await client.connect()  
await createListing(client,{name:"Mynae",call:"jdkd"})
```

ADDING MANY DOCUMENTS

```
async function createMultipleListing(client,docs){  
    const result = await client.db("Cluster0").collection("users").insertMany(docs)  
}  
  
const uri = "mongodb+srv://javi:javi@cluster0.146wlbr.mongodb.net/?retryWrites=true&w=majority"  
const client = new MongoClient(uri)  
await client.connect()  
await createMultipleListing(client,data)
```

Reading one doc

```
async function findOneListingByName(client,nameOfListing){  
  const result = await client.db("Cluster0").collection("users").findOne({name:nameOfListing})  
  if(result){  
    console.log(result)  
  }else{  
    console.log("No result matches!")  
  }  
}  
  
const uri = "mongodb+srv://javi:javi@cluster0.146wlbr.mongodb.net/?retryWrites=true&w=majority"  
const client = new MongoClient(uri)  
await client.connect()  
await findOneListingByName(client,"Sybil Russell")
```

Reading all docs

```
async function findListingForMinListNumberAndCurrency(client,{ListNum,Currency}){  
    const cursor = await client.db("Cluster0").collection("users")  
        .find({list: {$gte: ListNum}})  
        .sort({last:-1}).limit(5)  
  
    const result = await cursor.toArray()  
    if(result){ result.forEach(i=>console.log(i.name))}  
    else{console.log("No result matches!")}  
}  
  
const uri = "mongodb+srv://javi:javi@cluster0.146wlbr.mongodb.net/?retryWrites=true&w=majority"  
const client = new MongoClient(uri)  
await client.connect()  
await findListingForMinListNumberAndCurrency(client,0)
```

UpdateOne

```
async function updateByName(client,searchValue, setValues){  
  const result = await client.db("Cluster0").collection("users").updateOne({"name":searchValue},{$set:  
    setValues})  
  console.log(result)  
}  
const uri = "mongodb+srv://javi:javi@cluster0.146wlbr.mongodb.net/?retryWrites=true&w=majority"  
const client = new MongoClient(uri)  
await client.connect()  
await updateByName(client,"Quinn Clay",{name: "Javeria",  
  phone :"90-39028e67"  
})
```

upsert

- add one if not exist and update one if exist
- **Parameters are same as in updateOne**

```
await client.db("Cluster0").collection("users").updateOne({  
  "name": docIdentification},  
  {$set: updateOrInsertDocument},  
  {upsert: true})
```

update all

```
async function updateAll(client){  
  const result = await client.db("Cluster0").collection("users").updateMany(  
    {student:{$exists: false}}  
    ,{$set:{student:"Working"}})  
  }  
  
const uri = "mongodb+srv://javi:javi@cluster0.l46wlbr.mongodb.net/?  
retryWrites=true&w=majority"  
const client = new MongoClient(uri)  
await client.connect()  
await updateAll(client)
```

Delete One

```
async function deleteOneRecord(client,query){  
    const result = await client.db("Cluster0")  
        .collection("users")  
        .deleteOne({name:query})  
    }  
  
const uri = "mongodb+srv://javi:javi@cluster0.146wlbr.mongodb.net/?  
retryWrites=true&w=majority"  
  
const client = new MongoClient(uri)  
await client.connect()  
await deleteOneRecord(client,"Mahnoor")
```

Delete Many

```
async function deleteMany(client,q){  
    const result = await client.db("Cluster0")  
        .collection("users")  
        .deleteMany({"numberrange":{$lt:q}})  
}  
  
const uri = "mongodb+srv://javi:javi@cluster0.146wlbr.mongodb.net/?  
retryWrites=true&w=majority"  
  
const client = new MongoClient(uri)  
await client.connect()  
deleteMany(client,4)
```