Database Programming with Node JS and MongoDB

Prof. Chintan Makwana
Assistant Professor
Information Technology Department
Shantilal Shah Engineering College, Bhavnagar
www.sites.google.com/view/chintanmakwana

Outline

- Basics of MongoDB
- Data types
- Connect Node JS with MongoDB
- Operations on data (Insert, Find, Query, Sort, Delete, Update) using Node JS

Basics of MongoDB

- MongoDB is cross-platform, open-source, **document-oriented, no sequel (NoSQL) database.**

- NoSQL, also referred to as "not only SQL", "non-SQL", is **an approach to database design that enables the storage and querying of data outside the traditional structures found in relational databases.**
- Instead of storing your data in tables and rows as you would with a relational database, in MongoDB you store JSON-like documents.

- mongoDB = "Hu**mongo**us **DB**"
  - Open-source
  - Document-oriented
  - "High performance, high availability"
  - Automatic scaling

Basics of MongoDB

Relational Database

rollno	name
6001	Umang
6002	Dhaval
6003	Ajay
6004	Rahul
6005	Keval

MongoDB Document

```
[
    { rollno:6001 , name: "Umang" },
    { rollno:6002 , name: "Dhaval" },
    { rollno:6003 , name: "Ajay" },
    { rollno:6004 , name: "Rahul" },
    { rollno:6005 , name: "Keval" }
]
```

Basics of MongoDB

- Documents store data with the help of key-value pair.
- A Collection is a group of documents.
- These collections are stored in the MongoDB database.

SQL	mongoDB
Row / Tuple	Document (JSON, BSON)
Table	Collection
Database	Database

Basics of MongoDB

Relational

Person:

Pers_ID	Surname	First_Name	City
0	Miller	Paul	London
1	Ortega	Alvaro	Valencia
2	Huber	Urs	Zurich
3	Blanc	Gaston	Paris
4	Bertolini	Fabrizio	Rom

no relation

Car:

Car_ID	Model	Year	Value	Pers_ID
101	Bentley	1973	100000	0
102	Rolls Royce	1965	330000	0
103	Peugeot	1993	500	3
104	Ferrari	2005	150000	4
105	Renault	1998	2000	3
106	Renault	2001	7000	3
107	Smart	1999	2000	2

MongoDB Document

```
{
  first_name: 'Paul',
  surname: 'Miller'
  city: 'London',
  location: [45.123,47.232],
  cars: [
   { model: 'Bentley',
     year: 1973,
     value: 100000, … },
   { model: 'Rolls Royce',
     year: 1965,
     value: 330000, … }
  ]
}
```

Basics of MongoDB

- **JSON (JavaScript Object Notation)**
  - Easy for humans to write/read, easy for computers to parse/generate
  - Objects can be nested
  - Built on
    - name/value pairs

- **BSON (Binary JavaScript Object Notation)**
  - Binary-encoded serialization of JSON-like docs
  - Goals
    - Lightweight
    - Traversable
    - Efficient (decoding and encoding)

Basics of MongoDB

- **JSON / BSON Example (Key : Value)**

```
{
        “rollno" : “6001”,
        “name" : “Umang"
}
```

Data types of MongoDB

- Int32 / Int64
- String
- Boolean
- Binary
- Double
- Decimal128
- MinKey / MaxKey
- Array
- Object / ObjectId
- Symbol
- Null
- Date
- TimeStamp
- BSONRegExp / BSONSymbol
- Undefined

Applications of MongoDB

- Internet of Things
- Mobile Application
- Real time analysis
- Personalization
- Catalog management
- Content management

Features of MongoDB

- Ad-hoc queries for optimized, real-time analytics
- Indexing appropriately for better query executions
- Replication for better data availability and stability
- Sharding
- Load balancing

- Detail Reference : https://www.mongodb.com/what-is-mongodb/features

Download MongoDB for Windows

- **Download Link-1 : https://www.mongodb.com/try/download/community**
- **Download Link-2 : http://www.mongodb.org/downloads.**

MongoDB Atlas
MongoDB Enterprise Advanced
MongoDB Community Edition
MongoDB Community Server
MongoDB Community Kubernetes Operator
Tools
Mobile & Edge

The database is also offered as a fully-managed service with MongoDB Atlas. Get access to advanced functionality such as auto-scaling, serverless instances (in preview), full-text search, and data distribution across regions and clouds. Deploy in minutes on AWS, Google Cloud, and/or Azure, with no downloads necessary.

Give it a try with a free, highly-available 512 MB cluster.

Version
6.0.4 (current)

Platform
Windows

Package
msi

Download | Copy link | More Options

Install MongoDB on Windows

- **Installation Steps :**

https://www.youtube.com/watch?v=n3onrnxcfio&list=PLFaWDe_XIA4pMorGET0zpP9qJHVO4aC6E&index=4

Download MongoDB Shell on Windows

- **Download Link : https://www.mongodb.com/try/download/shell**

MongoDB Atlas
MongoDB Enterprise Advanced
MongoDB Community Edition
Tools
MongoDB Shell
MongoDB Compass
Atlas CLI
Atlas Kubernetes Operator
MongoDB CLI for Cloud Manager and Ops Manager
MongoDB Cluster-to-Cluster Sync
MongoDB Database Tools
MongoDB Connector for BI
Mobile & Edge

TOOLS

MongoDB Shell Download

MongoDB Shell is the quickest way to connect to (and work with) MongoDB. Easily query data, configure settings, and execute other actions with this modern, extensible command-line interface – replete with syntax highlighting, intelligent autocomplete, contextual help, and error messages.

Note: MongoDB Shell is an open source (Apache 2.0), standalone product developed separately from the MongoDB Server.

Learn more

Version
1.7.1

Platform
Windows 64-bit (8.1+)

Package
zip

Download | Copy link | More Options

Install MongoDB Shell on Windows

- **Installation Steps**:

https://www.youtube.com/watch?v=oC6sKlhz0OE&t=234s

MongoDB Shell Commands

(1) Database related Commands

(2) Collection related Commands

(3) Document related Commands

MongoDB Shell Database Commands

(1) Create Database

Syntax: use Database_name

Example: use PracticeDB

(2) Display Database

Syntax: show dbs

(3) Drop Database

Syntax: db.dropDatabase()

MongoDB Shell Collection Commands

(1) Create Collections

Syntax: db.createCollection(name)

Example: db.createCollection("students")

(2) Display Collection

Syntax: show collections

(3) Drop Collection

Syntax: db.collection_name.drop()

Example: db.students.drop()

MongoDB Shell Document Commands

- **List of CRUD Operations for Documents**
  1. **Create**
  2. **Read**
  3. **Update**
  4. **Delete**

MongoDB Create Operations

- Create or insert operations add new documents to a collection. If the collection does not currently exist, insert operations will create the collection.

- Syntax:
  - db.collection.insertOne()
  - db.collection.insertMany()

- Example:
  - **db.students.insertOne ({ rollno:6001 , name: "Umang" })**
  - **db.students.insertMany ([**

```
{ rollno:6002 , name: "Dhaval" } ,
{ rollno:6003 , name: "Ajay" }
])
```

MongoDB Read Operations

- Read operations retrieve documents from a collection; i.e. query a collection for documents. MongoDB provides the following methods to read documents from a collection:

- Syntax:
  - db.collection.find()
- Example:
  - **db.students.find()**
  - **db.students.find ({ rollno : 6001 })**
  - **db.students.find ({ rollno : { $in: [6001 , 6003] } })**

MongoDB Update Operations

• Update operations modify existing documents in a collection. MongoDB provides the following methods to update documents of a collection:

- Syntax:
  - db.collection.updateOne()
  - db.collection.updateMany()

- Example:
  - **db.students.updateOne({ rollno : 6002 } , {$set: {name : "Ajay" } })**
  - **db.students.updateMany({ rollno : { $gt : 6160 } } ,**
    { $set {name : "Undefine" } })

MongoDB Delete Operations

- Delete operations remove documents from a collection. MongoDB provides the following methods to delete documents of a collection:
- Syntax:
  - db.collection.deleteOne()
  - db.collection.deleteMany()

- Example:
  - **db.students.deleteOne ({ rollno : 6160 })**
  - **db.students.deleteMany ({ rollno : 6160 })**

Node.js

- Node.js is a cross-platform, JavaScript environment on the server
- It is powered by Google's V8 JavaScript engine
- It provides modules for interacting with local system resources such as processes, file system, networking, etc…

Download Node.js on Windows

- **Download Link : https://nodejs.org/en/**

Gujarat Technological University | How to Install NPM and Node.js | Node.js

nodejs.org/en/

Gmail YouTube Maps

HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | CERTIFICATION | NEWS

Node.js® is an open-source, cross-platform JavaScript runtime environment.

Security releases now available

Download for Windows (x64)

18.14.2 LTS
Recommended For Most Users

Other Downloads | Changelog | API Docs

19.7.0 Current
Latest Features

Other Downloads | Changelog | API Docs

For information about supported releases, see the release schedule.

node-v18.14.2-x64.msi

Show all

Type here to search

19°C Clear ENG 06:43 AM 28-02-2023

Install npm package of mongodb

project_path > npm install mongodb

Package Details: https://www.npmjs.com/package/mongodb

Connect.js

```
const { MongoClient } = require('mongodb');

// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);

// Database Name
const dbName = 'practiceDB';
```

Connect.js

```
async function main() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  const collection = db.collection('students');

 // CRUD Operation Code

  return 'done.';
}

main()
  .then(console.log)
  .catch(console.error)
  .finally(() => client.close());
```

MongoDB-NodeJS Insert Operations

- Insert operations add new documents to a collection. MongoDB provides the following methods to insert documents into a collection:
- Syntax:
  - collection.insertOne()
  - collection.insertMany()

- Example:
  - **collection.insertOne ({ rollno:6001 , name: "Umang" })**
  - **collection.insertMany ([**

```
{ rollno:6002 , "name": "Dhaval" } ,
{ rollno:6003 , "name": "Ajay" }
])
```

MongoDB-NodeJS Insert Operations

```
// insertOne() Operation Code
const insertResult = await collection.insertOne(
   { rollno: 6001 , name:“Umang" });

console.log('Inserted documents =>', insertResult);

// insertMany() Operation Code
const insertResult = await collection.insertMany([
     { rollno: 6002 , name:"Dhaval" },
     { rollno: 6003 , name:"Ajay" } ]);

console.log('Inserted documents =>', insertResult);
```

MongoDB-NodeJS Find Operations

- Find operations retrieve documents from a collection; i.e. query a collection for documents. MongoDB provides the following methods to retrieve documents from a collection:

- Syntax:
  - collection.find({}).toArray()
- Example:
  - **db.students.find({}).toArray()**
  - **db.students.find ({"rollno":6001}) .toArray()**
  - **db.students.find ({ "rollno" : { $in: [6001 , 6003] } }) .toArray()**

MongoDB-NodeJS Find Operations

```
// Read: find() Operation...
const findResult = await collection.find({}).toArray();
console.log('Found documents =>', findResult);
```

MongoDB-NodeJS Query Operations

```
// Read: find() Operation...
const queryResult = await collection.find({rollno:6001}).toArray();
console.log('Found documents =>', queryResult);

 // Read: find() Operation...
 const queryResult = await collection.find(
       {rollno:{$in:[6001,6003]}}).toArray();
 console.log('Found documents =>', queryResult);
```

MongoDB-NodeJS Sort Operation

```
// Sort() Operation...
  const sortResult = await collection.find({}).sort( {rollno:1}
).toArray();

  console.log('Sorted documents =>', sortResult);
```

MongoDB-NodeJS Update Operations

- Update operations modify existing documents in a collection. MongoDB provides the following methods to update documents of a collection:

- Syntax:
  - collection.updateOne()
  - collection.updateMany()

- Example:
  - **collection.updateOne({ rollno : 6001 } , {$set: {name : “ajay” } })**
  - **collection.updateMany({ rollno : 6002 } , {$set: {name : “dipak” } })**

MongoDB-NodeJS Update Operations

```
// updateOne() Operation...
const updateResult = await collection.updateOne(
     { rollno: 6001 }, { $set: { name: "ajay" } });

console.log('Updated documents =>', updateResult);

// updateMany() Operation...
 const updateResult = await collection.updateMany(
     { rollno: 6002 }, { $set: { name: “dipak" } });

console.log('Updated documents =>', updateResult);
```

MongoDB-NodeJS Delete Operations

- Delete operations remove documents from a collection. MongoDB provides the following methods to delete documents of a collection:
- Syntax:
  - collection.deleteOne()
  - collection.deleteMany()

- Example:
  - **collection.deleteOne ({ rollno : 6001})**
  - **collection.deleteMany ({ rollno : 6002})**

MongoDB-NodeJS Delete Operations

```
// deleteOne() Operation...
 const deleteResult = await collection.deleteOne (
    { rollno: 6001 });
 console.log('Deleted documents =>', deleteResult );

// deleteMany() Operation...
 const deleteResult = await collection.deleteMany (
    { rollno: 6002 });
 console.log('Deleted documents =>', deleteResult );
```