Mongoose Tutorial

Last Updated : 14 Dec, 2024

Mongoose is a popular ODM (Object Data Modeling) library for MongoDB and Node.js that simplifies database interactions by providing a schema-based solution to model application data. It is widely used to build scalable, structured, and efficient database-driven applications.

- Built on MongoDB for seamless integration with Node.js applications.
- Provides schema-based modeling to define document structure.
- Includes built-in validation to ensure data consistency.
- Enables easy querying and data relationships with chain query methods.

To Start with Mongoose, you need to install and import it into your project. Follow these articles to install depending on your system:

- How to Use MongoDB and Mongoose with Node.js
- npm mongoose

Let us now take a look at our first code example.

```
//import mongoose in the application
const mongoose = require('mongoose');

// Connect to MongoDB
mongoose.connect('mongodb://localhost:27017/myDatabase', {
    useNewUrlParser: true,
    useUnifiedTopology: true,
});

// Define a schema
const userSchema = new mongoose.Schema({
    name: String,
    age: Number,
    email: String,
});

// Create a model
const User = mongoose.model('User', userSchema);

// Insert a document
const user = new User({
    name: 'Mohit Kumar',
    age: 25,
    email: 'mohit@example.com',
});
```

```
user.save()
    .then(() => console.log('User saved'))
    .catch((err) => console.error('Error:', err));
```

It will insert a document with the provided details into the User collection in MongoDB.

In this example

- Mongoose connects to a local MongoDB database using mongoose.connect().
- A schema is defined (userSchema) to enforce the structure of documents in the User collection.
- A model (User) is created based on the schema, acting as an interface for database operations.
- A new User instance is created and saved to the database using the .save() method.

Mongoose Tutorial -GeeksforGeeks

Why Learn Mongoose?

- Simplifies MongoDB operations with built-in schema validation.
- Reduces boilerplate code for database interactions.
- Supports middleware for pre/post operations.
- Well-suited for Node.js applications.

Mongoose Tutorial Prerequisites: JavaScript, Node.js, and MongoDB basics

Mongoose Basics

- Connect Node to database using Mongoose
- Mongoose Module Introduction
- Mongoose Connections

- Mongoose Schematype
- Mongoose SchemaType Options
- Mongoose Schema API
- Mongoose Schemas Virtuals
- Mongoose Schemas Indexes
- Mongoose Document API
- Mongoose Documents
- Mongoose Plugins
- Mongoose Populate
- Mongoose Queries
- Mongoose Virtuals
- CRUD Operations Using Mongoose
- Mongoose Documents vs Models
- Mongoose Schemas Creating a model
- Mongoose Validation
- Aggregation in MongoDB
- Transactions in Mongoose

Mongoose Functions

- Mongoose Populate() method
- Mongoose find() Function
- Mongoose where() Function
- Mongoose remove() Function
- Mongoose exists() Function
- Mongoose update() Function
- Mongoose insertMany() Function
- Mongoose findById() Function
- Mongoose findByIdAndDelete() Function
- Mongoose findByIdAndUpdate() Function
- Mongoose findByIdAndRemove() Function
- Mongoose findOneAndDelete() Function
- Mongoose findOneAndUpdate() Function
- Mongoose findOneAndReplace() Function
- Mongoose findOneAndRemove() Function
- Mongoose replaceOne() Function
- Mongoose updateOne() Function
- Mongoose updateMany() Function

- Mongoose insertMany() Function
- Mongoose findOne() Function
- Mongoose deleteOne() Function

Mongoose Projects

- Login form using Node JS and MongoDB
- Upload and Retrieve Image on MongoDB using Mongoose
- Pagination on an API
- Nodejs – Connect MongoDB with Node app using MongooseJS
- Signup Form Using Nodejs and MongoDB
- Login form using Node.js and MongoDB
- Connect Django Project to MongoDB using Django

MongoDB Introduction

- How do Document Databases Work?
- How MongoDB works?
- MongoDB: An introduction
- What is MongoDB – Working and Features
- Difference between RDBMS and MongoDB
- MongoDB vs MySQL

MongoDB Installation

- How to Install and Configure MongoDB in Ubuntu?
- How to install MongoDB on MacOS?
- How to install MongoDB on Windows?

Basics of MongoDB

- MongoDB – Database, Collection, and Document
- MongoDB Cursor
- DataTypes in MongoDB
- What is ObjectId in MongoDB
- What is a MongoDB Query?
- MongoDB | Create a Database using MongoShell
- MongoDB | Delete Database using MongoShell
- MongoDB CRUD operations

MongoDB Methods

- MongoDB – Insert() Method
- MongoDB – insertOne() Method
- MongoDB – insertMany() Method
- MongoDB – Bulk.insert() Method
- MongoDB – bulkWrite() Method
- MongoDB – Update() Method
- MongoDB – updateOne() Method
- MongoDB – updateMany() Method
- MongoDB – Find() Method
- MongoDB – FindAndModify() Method
- MongoDB – sort() Method
- MongoDB – copyTo() Method
- MongoDB – count() Method
- MongoDB – countDocuments() Method
- MongoDB – drop() Method
- MongoDB – Remove() Method
- MongoDB – deleteOne() Method
- MongoDB – getIndexes() Method
- MongoDB – dropIndex() Method
- MongoDB – dropIndexes() Method

MongoDB Operators

Comparison Operators

- MongoDB – Comparison Query Operators
- MongoDB $cmp Operator
- MongoDB – Greater than Operator $gt
- MongoDB – Less than Operator $lt
- MongoDB – Equality Operator $eq
- MongoDB – Less than equals to Operator $lte
- MongoDB – Greater than equals to Operator $gte
- MongoDB – Inequality Operator $ne
- MongoDB $in Operator
- MongoDB – $nin Operator

Logical Operators

- MongoDB – Logical Query Operators
- MongoDB AND operator ($and)
- MongoDB OR operator ($or)
- MongoDB NOT operator ($not)
- MongoDB NOR operator ($nor)

Arithmetic Operators

- MongoDB $add Operator
- MongoDB $subtract Operator
- MongoDB $multiply Operator
- MongoDB $divide Operator
- MongoDB $abs Operator
- MongoDB $floor Operator
- MongoDB $ceil Operator
- MongoDB $mod Operator
- MongoDB $sqrt Operator
- MongoDB $pow Operator
- MongoDB $exp Operator
- MongoDB $log Operator
- MongoDB $log10 Operator
- MongoDB $ln Operator

Field Update Operators

- MongoDB – Field Update Operators
- MongoDB – Maximum operator ($max)
- MongoDB – Minimum operator ($min)
- MongoDB – Increment Operator ($inc)
- MongoDB – Multiply Operator ($mul)
- MongoDB – Rename Operator ($rename)
- MongoDB – Current Date Operator ($currentDate)
- MongoDB – SetOnInsert Operator ($setOnInsert)
- MongoDB Bitwise Update Operator

Array Expression Operators

- MongoDB $isArray Operator

- MongoDB $size Operator
- MongoDB $arrayElemAt Operator
- MongoDB $concatArrays Operator
- MongoDB $reverseArray Operator

Array Update Operators

- MongoDB – $pull Operator
- MongoDB – $pop Operator
- MongoDB – $pullAll Operator
- MongoDB – $push Operator
- MongoDB – Positional Operator ($)
- MongoDB – All Positional Operator ($[])
- MongoDB – $position Modifier
- MongoDB – $addToSet Operator
- MongoDB – $each Modifier
- MongoDB – $sort Modifier
- MongoDB – $slice Modifier

String Expression Operators

- MongoDB $concat Operator
- MongoDB $strcasecmp Operator
- MongoDB $toUpper Operator
- MongoDB $toLower Operator
- $substrCP (aggregation) operator in MongoDB

Working with Documents and Collections

- Defining, Creating, and Dropping a MongoDB collection
- Adding and Querying the data in MongoDB
- How to Create Database & Collection in MongoDB
- MongoDB – Query Documents using Mongo Shell
- MongoDB – Insert Single Document Using MongoShell
- MongoDB – Insert Multiple Document Using MongoShell
- MongoDB – Update Single Document Using MongoShell
- MongoDB – Update Multiple Documents Using MongoShell
- MongoDB – Replace Documents Using MongoShell
- MongoDB – Delete Single Document Using MongoShell

- MongoDB – Delete Multiple Documents Using MongoShell
- MongoDB – Check the existence of the fields in the specified collection
- Sorting Documents in MongoDB
- Capped Collections in MongoDB

Indexing in MongoDB

- Indexing in MongoDB
- MongoDB Index Types
- MongoDB – Compound Indexes
- MongoDB – Text Indexes
- MongoDB – Multikey Indexes

MongoDB Advance

- Export data from MongoDB
- Import data to MongoDB
- MongoDB – Regex
- MongoDB Projection
- MongoDB – Embedded Documents
- MongoDB – Query Embedded Documents Using Mongo Shell
- Aggregation in MongoDB
- How to Enable Authentication on MongoDB?
- Create a user and add a role in MongoDB
- MongoDB – Replication and Sharding
- MongoDB – Backup and Restoration

MongoDB For Interview

- Top 50 MongoDB Interview Questions with Answers for 2024
- MongoDB Interview Experience for Backend developer
- MongoDB Exercises
- MongoDB Cheat Sheet (Basic to Advanced)
- 30 Days of Mongo DB: A Complete Beginners Guide

Advantages of Mongoose

- **Schema Definition:** Mongoose allows you to define structured schema model for MongoDB collections, because of that you get a clear understanding about the structure of model data.

- **Data Validation:** It can be used for data validation, which ensures that only valid and properly formatted data is stored in the database, which helps to manage data integrity.
- **Middleware Support:** Mongoose has the support of middleware which helps in the execution of custom logic before or after database operations, that offers flexibility in handling data interactions.
- **Query Building:** We don't have to write those complex queries which we were writing in MongoDB because Mongoose simplifies the process by providing a high-level API that makes it easier to interact with the database.
- **Modeling Relationships and Population:** You can define relationships between different data models and it also supports population, due to this you can work with related data without disturbing database normalization.

Mongoose vs MongoDB Native Driver

Feature	Mongoose	MongoDB Native Driver
Abstraction Level	High (uses models and schemas)	Low (raw queries)
Data Validation	Built-in	Manual
Middleware Support	Yes	No
Learning Curve	Moderate	Steeper
Use Case	Complex Applications	Lightweight Apps or Scripts

More on Mongoose:

- For more article, you can read recently published article's on MongoDB: Recent Article on MongoDB and Mongoose: Recent articles on Mongoose

Comment More info Advertise with us

Next Article

Mongoose Schemas Creating a Model

Similar Reads

Mongoose Tutorial

Mongoose is a popular ODM (Object Data Modeling) library for MongoDB and Node.js that simplifies database interactions by providing a schema-based solution to model application data. It is widely used t...

6 min read

Mongoose Schemas

Mongoose SchemaTypes

Mongoose Documents

Mongoose Queries

Mongoose Populate

Mongoose Schema API

Mongoose Connection API

Mongoose Document API

Mongoose Model API

Company
About Us
Legal
Privacy Policy
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

Languages
Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

DSA
Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML
Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies
HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial
Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science
Operating Systems
Computer Network

DevOps
Git
Linux

Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Inteview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development
Data Science
CS Subjects