

MongoDB Tutorial

MongoDB Tutorial

In this **MongoDB Tutorial**, we shall learn the basics of MongoDB, different CRUD Operations available for MongoDB Documents, Collections and Databases, and integrating MongoDB to applications developed using programming languages like Java, Python, Kotlin, Java Script, etc. We shall also take you through different MongoDB examples for better understanding the syntax.

Prerequisites

There is nothing in particular that you should know before you get started with this series of MongoDB Tutorials. But it could help, if you know a little bit about the concepts of Relational Databases and NoSQL Databases.

What is MongoDB

MongoDB is a database program that is

- **Free** – You don't have to pay a price to use it in your application.
- **Open Source** – The source is open and open source community contributes to the development of MongoDB database program.
- **Cross Platform** – MongoDB build on a platform (say Linux) works on other platforms(Windows, MacOS, etc.) as well without any modification to the code.
- **Document Oriented** – MongoDB stores records in form of JSON like documents. There are many advantages in doing so. A notable advantage is that adding new features(columns in relational database) to the database requires no hassle.
- **NoSQL (Not only SQL)** – MongoDB uses document based databases where each document is a key with complex value. The document value can contain multiple key-value pairs.

MongoDB is written in C++ language.

How MongoDB is better than Relational Databases

MongoDB packs the advantages of NoSQL. NoSQL database concepts were developed to make the applications more modern and fast. NoSQL databases overcome the disadvantages of SQL.

- **Structured Data** – MongoDB can handle structured data(like a spread-sheet or table), semi structured data and unstructured data (data with no rules, examples are multimedia content).
- **Agile Development** – All the NoSQL features of MongoDB makes it perfect fit for Agile methodology for developing applications. Modern applications are designed to have fewer development time and quick updates to the ever ending requirements.
- **Scalability** – MongoDB is horizontally scalable and thus can respond to the increasing application requirements with less cost.

Some other features of MongoDB

- **Replication** – You can setup replication of data between MongoDB instances. When an instance goes down, other instance can handle the requests with same data available.
- **MapReduce** – You can apply MapReduce algorithm on the MongoDB Collections to pull out some useful reports.

When to use MongoDB in your application ?

You might be wondering, with such a wide variety of databases available out there, why should you select MongoDB or what project requirements demand the use of MongoDB.

Try to answer the following questions, and if you get an YES for most of them, choose MongoDB.

1. Is your data structured or not ? Structured !
2. Does your application need scalability? horizontally in particular !
3. Do you want your schema design to be flexible ?
4. Does you data objects move around the application as JSON ?
5. You do not require JOINS on different collections ?

Applications of MongoDB

Following are some of the Application of MongoDB that are known in production :

1. **Analyzing Application Logs** MongoDB's schema-less feature makes it a best fit for analyzing application logs where record schema may change from time to time.
2. **Automatic Purging** Capped Collection of MongoDB helps to purge old records automatically.

Index – MongoDB Tutorial

With a brief introduction tour to MongoDB, let us dive into working with MongoDB.

- **Getting started with MongoDB** In the following MongoDB tutorials, we shall install MongoDB and get started with Mongo Shell. We shall also learn to write code in JavaScript files.
 1. [Install MongoDB on Ubuntu](#)
 2. [Mongo Shell](#)
 3. [MongoDB Script](#)
- **MongoDB Database** Following MongoDB tutorials provides the idea of what a MongoDB is, how to create one and delete if not required further.
 1. [MongoDB Database](#)
 2. [MongoDB Create Database](#)
 3. [MongoDB Delete Database](#)
- **MongoDB Collection** MongoDB Collection is analogous to a table in SQL. It holds many documents (records). Following MongoDB Tutorials will provide better understanding about MongoDB Collection, how to create a MongoDB collection and delete it.
 1. [MongoDB Collection](#)
 2. [MongoDB Create Collection](#)
 3. [MongoDB Delete Collection](#)
- **MongoDB Document** MongoDB Document contains key-value pairs like a property-value in a table. Following MongoDB Tutorials help you with the CRUD operations you can do on a MongoDB document and more advanced concepts like

replica sets, etc.

1. [MongoDB Document](#)
2. [MongoDB Insert Document](#)
3. [MongoDB Query Documents](#)
4. [MongoDB Update Document](#)
5. [MongoDB Delete Document](#)
6. [MongoDB Limit Documents](#)
7. [MongoDB Skip Documents](#)
8. [MongoDB Sort Documents](#)
9. [MongoDB Setup Replica Set](#)

- **MongoDB Concepts** MongoDB is truly modern. It employs functionalities like Text Search, MapReduce, Backup, etc., at its core.

1. [MongoDB Text Search](#)
2. [MongoDB MapReduce](#)
3. [MongoDB Backup](#)

- **MongoDB Integration with Other Programming Languages** If you want to integrate your MongoDB application with applications written using programming languages like Java, Kotlin, Python, etc., following MongoDB tutorials help you with the integration process.

1. [Connect to MongoDB from Java](#)
2. [Connect to MongoDB from Python](#)
3. [Connect to MongoDB from Kotlin](#)

- **MongoDB Others** Following are miscellaneous MongoDB Tutorials.

1. [MongoDB Interview Questions](#)
2. [Uninstall MongoDB from Ubuntu](#)

This series of MongoDB Tutorials are provided with lucid explanation and well formed examples.

MongoDB Tutorial

↳ [MongoDB Tutorial](#)

↳ [Install MongoDB on Ubuntu](#)

↳ [Start MongoDB Server](#)

↳ [MongoDB Shell](#)

↳ [Check MongoDB Version](#)

↳ [MongoDB Server Port Number - Default Value and How to Change it](#)

↳ [MongoDB Script](#)

Database

↳ [MongoDB Database](#)

↳ [MongoDB Create Database](#)

↳ [MongoDB Delete Database](#)

Collections

↳ [MongoDB Collection](#)

↳ [MongoDB Create Collection](#)

↳ [MongoDB Delete Collection](#)

Documents

↳ [MongoDB Document](#)

↳ [MongoDB Insert Document](#)

↳ [MongoDB Query Documents](#)

↳ [MongoDB Project Fields in Result](#)

↳ [MongoDB Update Document](#)

↳ [MongoDB Delete Document](#)

↳ [MongoDB Limit Documents](#)

↳ [MongoDB Skip Documents](#)

↳ [MongoDB Sort Documents](#)

↳ [MongoDB Setup Replica Set](#)

↳ [MongoDB Locks](#)

MongoDB Concepts

↳ [MongoDB Text Search](#)

↳ [MongoDB MapReduce](#)

↳ [MongoDB Backup - mongodump](#)

MongoDB Queries

↳ [MongoDB Date](#)

MongoDB Queries

↳ [MongoDB Date\(\)](#)

MongoDB Integration

MongoDB Java

↳ [Connect to MongoDB from Java](#)

MongoDB Python

↳ [Connect to MongoDB from Python](#)

MongoDB Kotlin

↳ [Connect to MongoDB from Kotlin](#)

MongoDB Node.js

↳ [Node.js MongoDB](#)

↳ [Node.js MongoDB Connection](#)

‡ [Node.js MongoDB Create Database](#)

‡ [Node.js MongoDB Drop Database](#)

‡ [Node.js MongoDB Create Collection](#)

‡ [Node.js MongoDB Delete Collection](#)

‡ [Node.js MongoDB Insert Documents](#)

‡ [MongoError: failed to connect to server](#)

MongoDB Others

‡ [MongoDB Interview Questions](#)

‡ [Uninstall MongoDB from Ubuntu](#)

Useful Resources

‡ [How to Learn Programming](#)