

MongoDB Document – Structure and Sample Documents

MongoDB Document

MongoDB Document is an entity in which zero or more ordered field-value pairs are stored.

In comparison to Relational Databases, it is analogous to a record or row in table.

Document in MongoDB follows [BSON](http://bsonspec.org/) Specifications. BSON is binary encoded serialization of JSON-like documents. With BSON, MongoDB Documents can be traversed easily. As BSON uses C data types, encoding data to BSON or decoding from BSON is easier in most of the programming languages.

A document can have documents nested in them. MongoDB Documents are the building blocks of a [MongoDB Collection](#).

We shall learn following topics in this tutorial :

- **Operations** that could be performed on Documents in MongoDB
- **Structure** of a MongoDB Document
- **Sample** MongoDB Documents

MongoDB Document Operations

Following operations could be performed on MongoDB Documents.

- **MongoDB Insert Document** to a Collection
- **MongoDB Query Documents** of a Collection
- **MongoDB Update Document** of a Collection
- **MongoDB Delete Document** from a Collection

Structure of MongoDB Document

Following is structure of a Document in MongoDB:

```
{
  field1:value1;
  field2:value2;
  .
  .
  fieldN:valueN;
}
```

Document can contain N number of field-value pairs.

The values can have any datatype that is supported by BSON specification.

Simple MongoDB Document

Following is a simple Document with field-value pairs

```
{
  name: "Midhuna"
}

{
  name: "Midhuna",
  age: 23,
  place: "New York",
  hobbies: ["Singing", "Reading Books"]
}
```

Sample MongoDB Document

Lets see a Document containing other documents nested in.

```
{
  name: "Midhuna"
}

{
  name: "Midhuna",
  age: 23,
  place: "New York",
  hobbies: ["Singing", "Reading Books"]
  spouse: {
    name: "Akash",
    age: 25
  }
}
```

Conclusion :

In this MongoDB Tutorial, we have learnt about MongoDB Documents, its structure with samples and the operations that could be performed on them.

↳ [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](#)