

## MERN – MongoDB

- 1) Install mongoDB with Compass
- 2) Start Compass and connect to Database
- 3) Create a directory dbwrite
- 4) Change to dbwrite directory `cd dbwrite`
- 5) Give command `npm init -y` ( this will initialize the project folder )
- 6) Install package axios to interact with database
- 7) `npm install axios`
- 8) create a file in dbwrite folder that will read and write the text in database
- 9) Database with the name of the project is automatically created in mongodb
- 10) Create a file server.js in dbwrite folder that acts as a server and connects to the database
- 11) Start the server by the command window with `node server.js`
- 12) Open another command window and give command `readwrite.js` to read and write a text in the database
- 13) Check the data in mongodb with compass

### Server.js

```
const express = require('express');
const bodyParser = require('body-parser');
const mongoose = require('mongoose');

const app = express();
const port = 3000;

// Middleware
app.use(bodyParser.json());

// MongoDB connection
```

```
mongoose.connect('mongodb://localhost:27017/dbwrite', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
}).then(() => {
  console.log('Connected to MongoDB');
}).catch(err => {
  console.error('MongoDB connection error:', err);
});

// Define a schema and model
const textSchema = new mongoose.Schema({
  content: String
});

const Text = mongoose.model('Text', textSchema);

// Routes
app.post('/texts', async (req, res) => {
  const newText = new Text({
    content: req.body.content
  });

  try {
    const savedText = await newText.save();
    res.status(201).json(savedText);
  } catch (err) {
    res.status(400).json({ message: err.message });
  }
});

app.get('/texts', async (req, res) => {
  try {
    const texts = await Text.find();
    res.json(texts);
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
});

// Start the server
app.listen(port, () => {
  console.log(`Server is running on http://localhost:${port}`);
});
```

## Output

CA. Command Prompt

```
]
C:\Users\Admin\bcatrng\dbwrite>node readwrite.js
Text saved: {
  content: 'Hello, GNIMT ...!',
  _id: '66852ad7fa3549edf2c2ef74',
  __v: 0
}
Texts retrieved: [
  { _id: '6684eebb2a1d1df092b86ba2', content: 'Hello, world!', __v: 0 },
  { _id: '6684fea12a1d1df092b86ba5', content: 'Hello, world!', __v: 0 },
  { _id: '66851668445416f19d71c5c1', content: 'Hello, world!', __v: 0 },
  { _id: '668516a0445416f19d71c5c4', content: 'Hello, world!', __v: 0 },
  { _id: '668517a9445416f19d71c5c8', content: 'Hello, world!', __v: 0 },
  { _id: '66851bd8fa3549edf2c2ef71', content: 'Hello, world!', __v: 0 },
  {
    _id: '66852ad7fa3549edf2c2ef74',
    content: 'Hello, GNIMT ...!',
    __v: 0
  }
]
C:\Users\Admin\bcatrng\dbwrite>node readwrite.js_
```

## Readwrite.js

```
const axios = require('axios');

// Function to write text to the database
const writeTextToDatabase = async (text) => {
  try {
    const response = await axios.post('http://localhost:3000/texts', {
      content: text
    });
    console.log('Text saved:', response.data);
  } catch (error) {
    console.error('Error writing text to database:', error.message);
  }
};

// Function to read text from the database
const readTextFromDatabase = async () => {
  try {
    const response = await axios.get('http://localhost:3000/texts');
    console.log('Texts retrieved:', response.data);
  } catch (error) {
    console.error('Error reading text from database:', error.message);
  }
};

// Test the functions
const testDatabaseOperations = async () => {
  await writeTextToDatabase('Hello, GNIMT ...!');
  await readTextFromDatabase();
};

testDatabaseOperations();
```