

### Table of Contents

1. What is Node.js?
  2. Node.js Basics
  3. What is Express.js?
  4. Express.js Basics
  5. What is MongoDB?
  6. MongoDB Basics
  7. Connecting Node.js & MongoDB
  8. Building a Simple API Example
  9. Best Practices
  10. Helpful Resources
- 

### What is Node.js?

Node.js is a JavaScript runtime built on Chrome's V8 engine.

It allows you to run JavaScript on the server side to build fast, scalable network applications.

---

### Node.js Basics

#### Initialize a project:

bash

CopyEdit

npm init

#### Install packages:

bash

CopyEdit

npm install

#### Create a simple HTTP server:

javascript

CopyEdit

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

```
http.createServer((req, res) => {  
  res.write('Hello World');  
  res.end();  
}).listen(3000);
```

---

## 3 What is Express.js?

Express.js is a minimal, flexible Node.js web application framework.  
It provides robust tools for building web and mobile applications and APIs.

---

## 4 Express.js Basics

### ☒ Install Express:

bash

CopyEdit

```
npm install express
```

### ☒ Basic Express server:

javascript

CopyEdit

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

```
const app = express();
```

```
app.get('/', (req, res) => {
```

```
  res.send('Hello Express');
```

```
});
```

```
app.listen(3000, () => console.log('Server running on port 3000'));
```

---

## 5 What is MongoDB?

MongoDB is a NoSQL, document-oriented database.  
It stores data in flexible, JSON-like documents, making it great for modern applications.

---

## 6 MongoDB Basics

✓ Install MongoDB locally or use MongoDB Atlas (cloud).

✓ Basic MongoDB commands:

bash

CopyEdit

mongo

bash

CopyEdit

use myDatabase

```
db.myCollection.insertOne({ name: 'John', age: 30 })
```

---

## 7 Connecting Node.js & MongoDB

✓ Use Mongoose (popular ODM for MongoDB):

bash

CopyEdit

```
npm install mongoose
```

✓ Example connection:

javascript

CopyEdit

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

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

```
const User = mongoose.model('User', { name: String, age: Number });
```

```
const user = new User({ name: 'Alice', age: 25 });
```

```
user.save().then(() => console.log('User saved'));
```

---

## 8 Building a Simple API Example

### ✓ Example CRUD API:

javascript

CopyEdit

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

```
const app = express();
```

```
app.use(express.json());
```

```
app.get('/users', (req, res) => { /* GET all users */ });
```

```
app.post('/users', (req, res) => { /* CREATE new user */ });
```

```
app.put('/users/:id', (req, res) => { /* UPDATE user */ });
```

```
app.delete('/users/:id', (req, res) => { /* DELETE user */ });
```

```
app.listen(3000, () => console.log('API running on port 3000'));
```

---

## 9 Best Practices

- ✓ Use environment variables (dotenv).
- ✓ Validate inputs & sanitize data.
- ✓ Implement proper error handling.
- ✓ Secure your app (helmet, cors).
- ✓ Use async/await for promises.

---

## 10 Helpful Resources

- Node.js Docs: <https://nodejs.org/en/docs/>
- Express.js Docs: <https://expressjs.com/>
- MongoDB Docs: <https://www.mongodb.com/docs/>
- Mongoose Docs: <https://mongoosejs.com/>