Day 50

“Web Development + Security”

Mongoose:

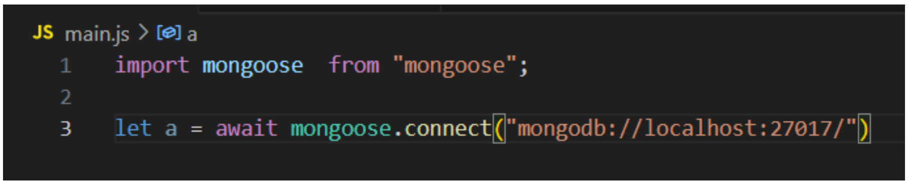
What is Mongoose?

Mongoose is an ODM (Object Data Modeling) library for MongoDB and Node.js. It helps you define the structure (schema) of your data and gives you powerful tools to interact with MongoDB easily.

Why Use Mongoose Instead of Raw MongoDB?

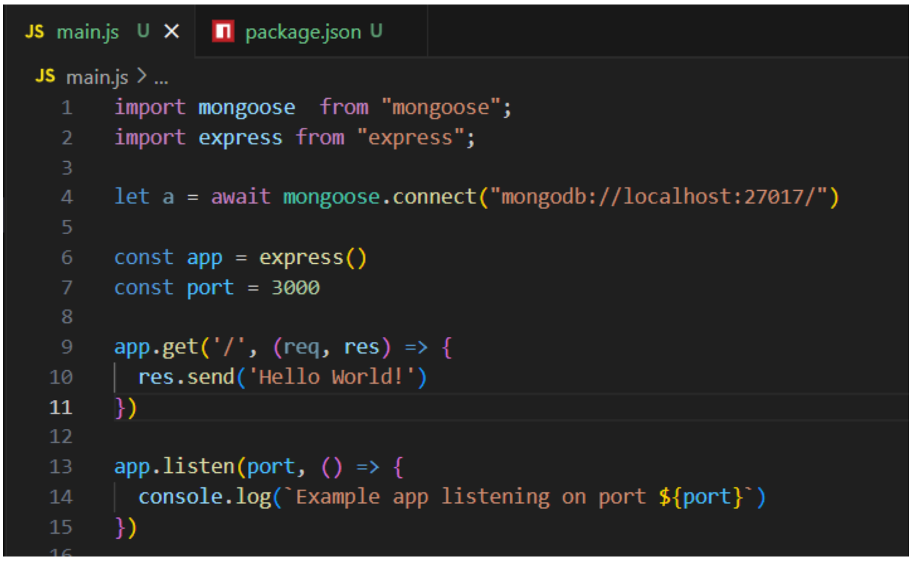
Feature	**Raw MongoDB**	**Mongoose**
Schema validation	No schema	Schema-based
Relationships	Manual	Built-in population
Validation	Manual	Built-in validators
Middleware (hooks)	No	Yes (pre/post save, update, etc.)
Query helpers	Limited	Advanced chainable methods

Now, in order to bring it to the code we will write the following code:



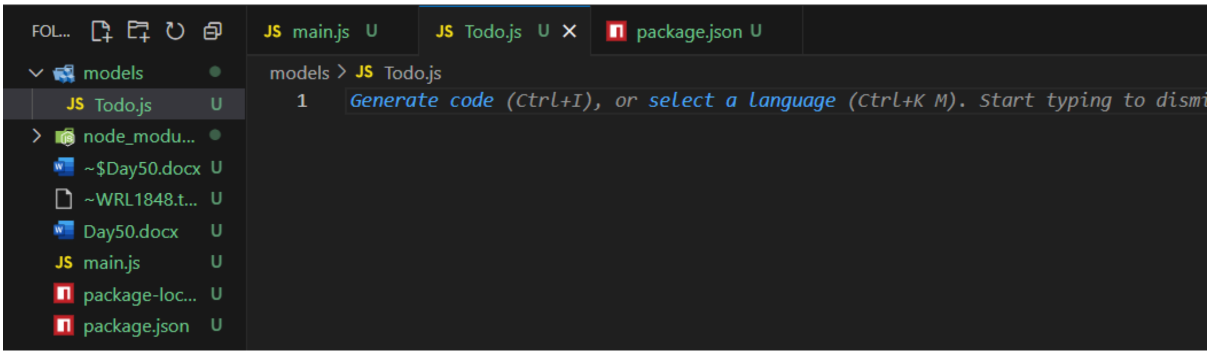
```
import mongoose  from "mongoose";

let a = await mongoose.connect("mongodb://localhost:27017/")
```

Now, importing express and connecting it with mongoose:

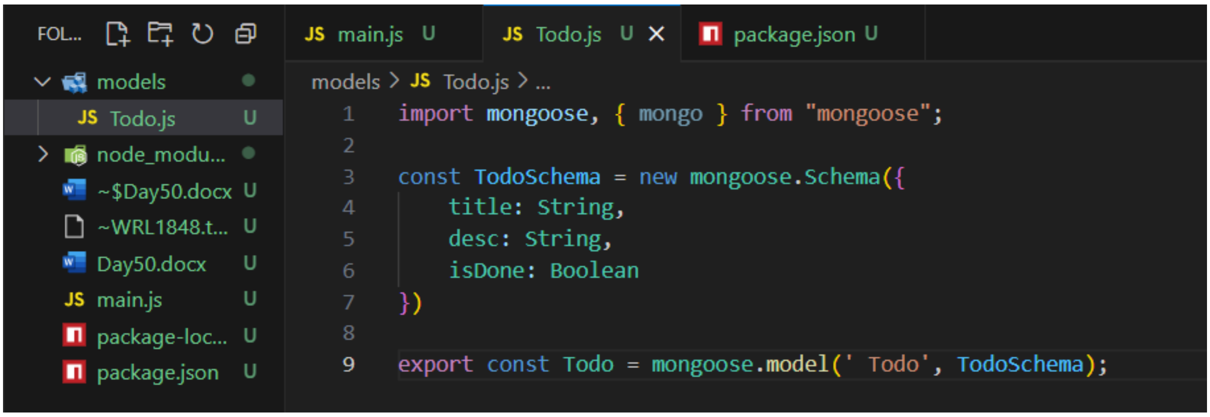


```
import mongoose  from "mongoose";
import express from "express";

let a = await mongoose.connect("mongodb://localhost:27017/")

const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

Now, creating a schema under the model folder:

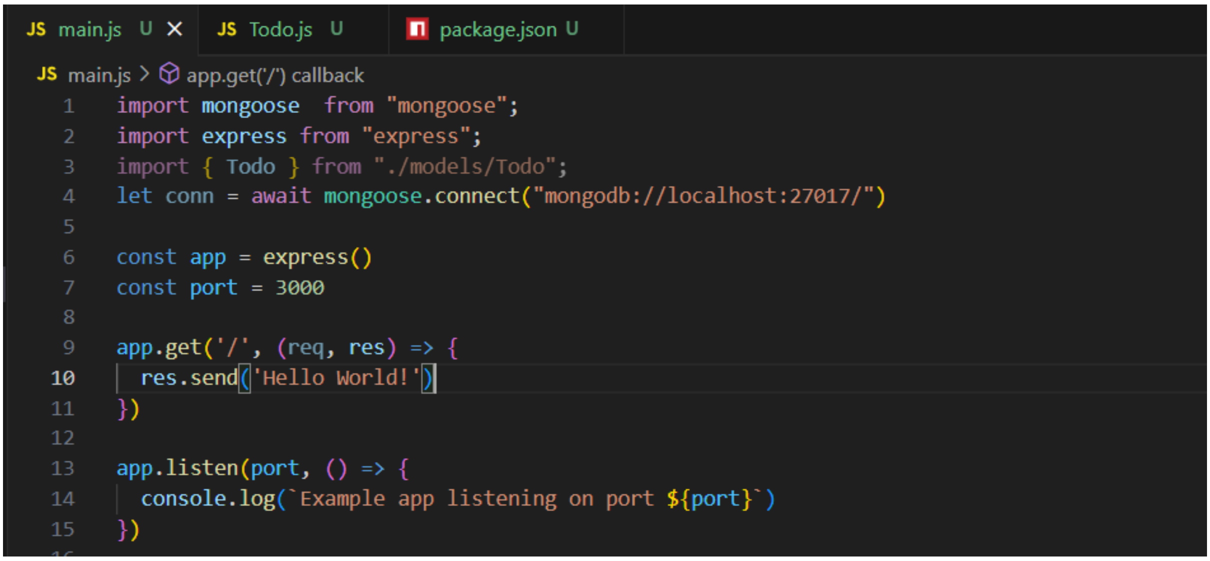


Now, we will create the schema, using the following code:



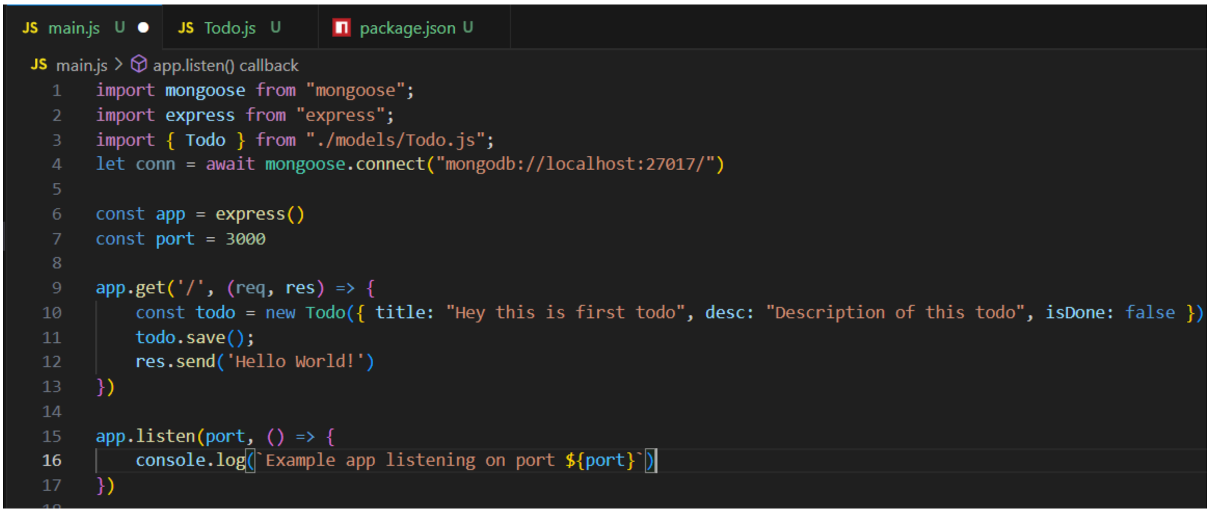
```
import mongoose, { mongo } from "mongoose";

const TodoSchema = new mongoose.Schema({
    title: String,
    desc: String,
    isDone: Boolean
})

export const Todo = mongoose.model(' Todo', TodoSchema);
```

Now, we will import this to the main.js using the following code:



```
import mongoose  from "mongoose";
import express from "express";
import { Todo } from "./models/Todo";
let conn = await mongoose.connect("mongodb://localhost:27017/")

const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

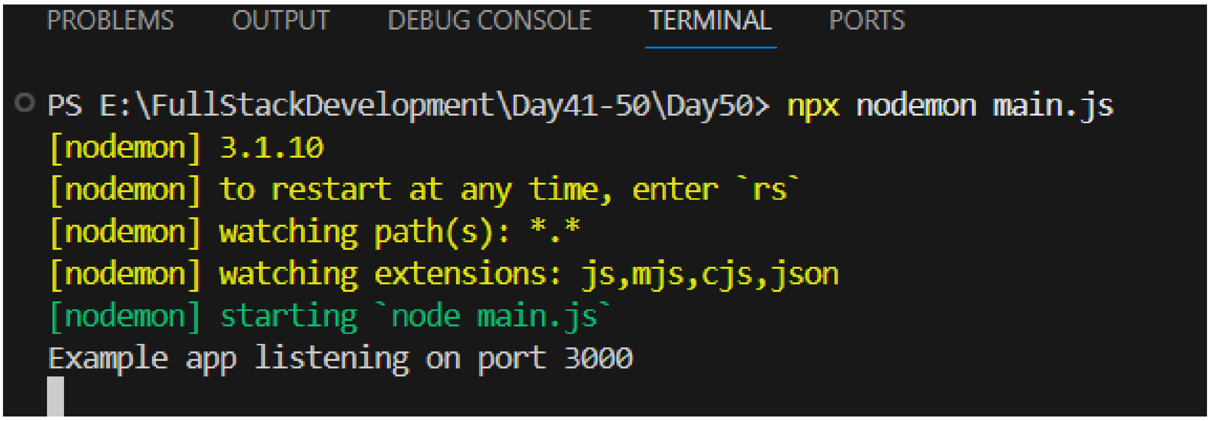
Now, import is modified and added the content:



```javascript
import mongoose from "mongoose";
import express from "express";
import { Todo } from "./models/Todo.js";
let conn = await mongoose.connect("mongodb://localhost:27017/")

const app = express()
const port = 3000

app.get('/', (req, res) => {
    const todo = new Todo({ title: "Hey this is first todo", desc: "Description of this todo", isDone: false })
    todo.save();
    res.send('Hello World!')
})

app.listen(port, () => {
    console.log(`Example app listening on port ${port}`)
})
```

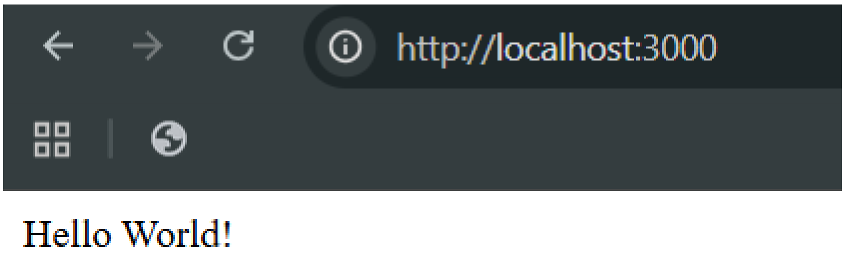
Now, start the server:

Terminal:



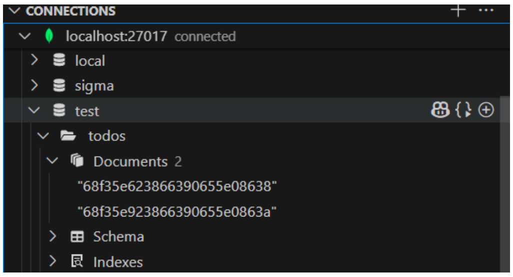
```
PS E:\FullStackDevelopment\Day41-50\Day50> npx nodemon main.js
[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node main.js`
Example app listening on port 3000
```

Browser:



Hello World!

Now, check if the Db is created or not: clearly it is created and inserted the data we had wished.

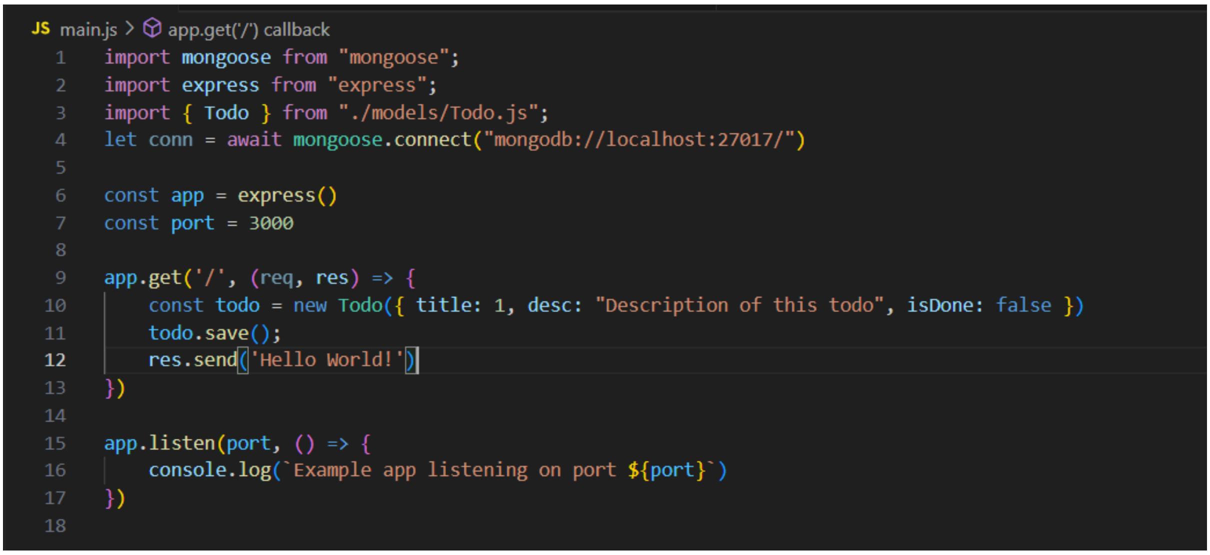


```
localhost:27017 connected
  local
  sigma
  test
    todos
      Documents 2
        "68f35e623866390655e08638"
        "68f35e923866390655e0863a"
      Schema
      Indexes
```

As we had refreshed the page 2 times, so it inserted that data two time.

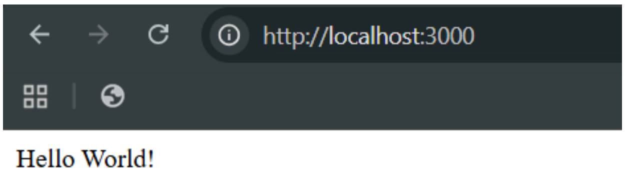
Now, in case we changed the value to be inserted from string to the number, then it will change it from number to string and then only insert:

Main.js:



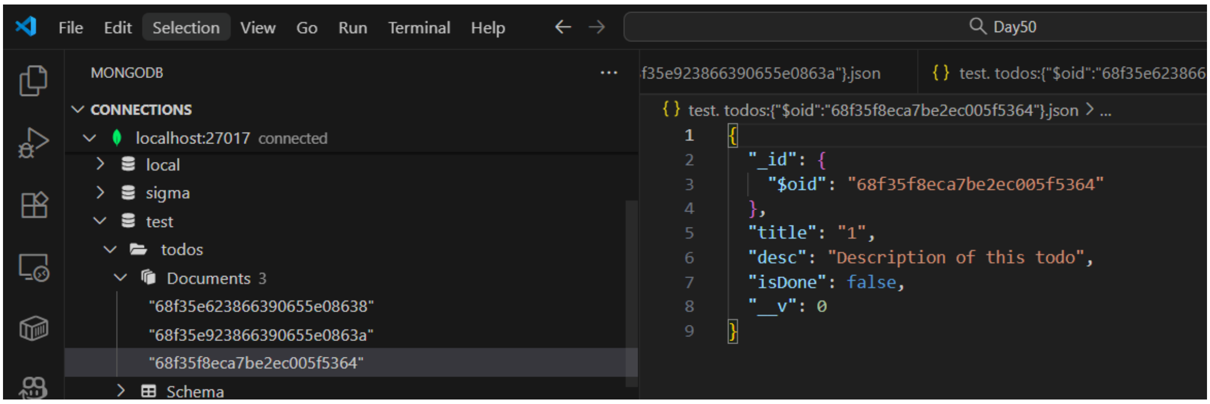
```
import mongoose from "mongoose";
import express from "express";
import { Todo } from "./models/Todo.js";
let conn = await mongoose.connect("mongodb://localhost:27017/")

const app = express()
const port = 3000

app.get('/', (req, res) => {
    const todo = new Todo({ title: 1, desc: "Description of this todo", isDone: false })
    todo.save();
    res.send('Hello World!')
})

app.listen(port, () => {
    console.log(`Example app listening on port ${port}`)
})

```

Browser:



Hello World!

DB: clearly it inserted but converted that to string.


File
Edit
Selection
View
Go
Run
Terminal
Help
Day50
MONGODB
CONNECTIONS
localhost:27017 connected
local
sigma
test
todos
Documents 3
"68f35e623866390655e08638"
"68f35e923866390655e0863a"
"68f35f8eca7be2ec005f5364"
Schema
f35e923866390655e0863a"}.json
{ } test. todos:{"$oid":"68f35e623866
{ } test. todos:{"$oid":"68f35f8eca7be2ec005f5364"}.json > ...

```
{
  "_id": {
    "$oid": "68f35f8eca7be2ec005f5364"
  },
  "title": "1",
  "desc": "Description of this todo",
  "isDone": false,
  "__v": 0
}
```

--The End--