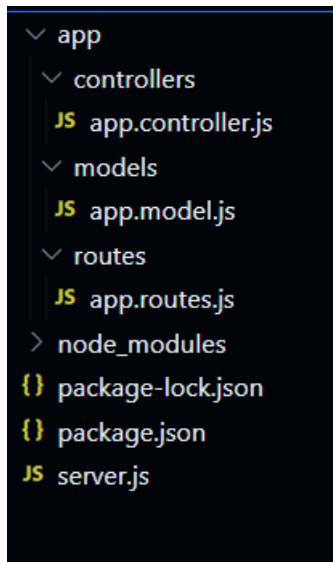


LA8: Building a REST API with Express, Node, and MongoDB

V.S Barath Viknesh
21011102110
BTech CSE(IoT) – B



Server.js

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

mongoose.Promise = global.Promise;
mongoose.connect("mongodb://localhost:27017",
  {
    useNewUrlParser: true,
  }
)
.then(() => {
  console.log("Successfully connected to the database");
})
.catch((err) => {
  console.log("Could not connect to the database. Error...", err);
  process.exit();
});

const app = express();
```

```

app.use(bodyParser.urlencoded({ extended: true }));

app.use(bodyParser.json());

app.get("/", (req, res) => {
  res.json({ message: "Server is running :D" });
});

let PORT = 8080;

app.listen(PORT, () => {
  console.log(`Server is listening on port ${PORT}`);
});

// .....
require('./app/routes/app.routes.js')(app);
// .....

```

App.routes.js

```

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

mongoose.Promise = global.Promise;
mongoose.connect("mongodb://localhost:27017",
  {
    useNewUrlParser: true,
  }
)
.then(() => {
  console.log("Successfully connected to the database");
})
.catch((err) => {
  console.log("Could not connect to the database. Error...", err);
  process.exit();
});

const app = express();

app.use(bodyParser.urlencoded({ extended: true }));

app.use(bodyParser.json());

app.get("/", (req, res) => {
  res.json({ message: "Server is running :D" });
});

```

```

});

let PORT = 8080;

app.listen(PORT, () => {
  console.log(`Server is listening on port ${PORT}`);
});

// .....
require('./app/routes/app.routes.js')(app);
// .....

```

App.model.js

```

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

mongoose.Promise = global.Promise;
mongoose.connect("mongodb://localhost:27017",
  {
    useNewUrlParser: true,
  }
)
.then(() => {
  console.log("Successfully connected to the database");
})
.catch((err) => {
  console.log("Could not connect to the database. Error...", err);
  process.exit();
});

const app = express();

app.use(bodyParser.urlencoded({ extended: true }));

app.use(bodyParser.json());

app.get("/", (req, res) => {
  res.json({ message: "Server is running :D" });
});

let PORT = 8080;

app.listen(PORT, () => {
  console.log(`Server is listening on port ${PORT}`);
});

```

```
// .....  
require('./app/routes/app.routes.js')(app);  
// .....
```

App.controller.js

```
const App = require("../models/app.model.js");  
  
// Create and Save a new Message  
exports.create = (req, res) => {  
  const message = new App({  
    message: req.body.message,  
  });  
  message  
    .save()  
    .then((data) => {  
      res.send(data);  
    })  
    .catch((err) => {  
      res.status(500).send({  
        message:  
          err.message || "Some error occurred while creating the Message.",  
      });  
    });  
};  
  
// Retrieve all messages from the database.  
exports.findAll = (req, res) => {  
  App.find()  
    .then((data) => {  
      res.send(data);  
    })  
    .catch((err) => {  
      res.status(500).send({  
        message:  
          err.message || "Some error occurred while retrieving messages.",  
      });  
    });  
};  
  
// Find a single message with a messageId  
exports.findOne = (req, res) => {  
  App.findById(req.params.messageId)  
    .then((data) => {
```

```

    if (!data) {
      return res.status(404).send({
        message: "Message not found with id " + req.params.messageId,
      });
    }
    res.send(data);
  })
  .catch((err) => {
    if (err.kind === "ObjectId") {
      return res.status(404).send({
        message: "Message not found with id " + req.params.messageId,
      });
    }
    return res.status(500).send({
      message: "Error retrieving message with id " + req.params.messageId,
    });
  });
};

// Update a message identified by the messageId in the request
exports.update = (req, res) => {
  App.findByIdAndUpdate(
    req.params.messageId,
    {
      message: req.body.message,
    },
    { new: true }
  )
  .then((data) => {
    if (!data) {
      return res.status(404).send({
        message: "Message not found with id " + req.params.messageId,
      });
    }
    res.send(data);
  })
  .catch((err) => {
    if (err.kind === "ObjectId") {
      return res.status(404).send({
        message: "Message not found with id " + req.params.messageId,
      });
    }
    return res.status(500).send({
      message: "Error updating message with id " + req.params.messageId,
    });
  });
};

```

```
// Delete a message with the specified messageId in the request
exports.delete = (req, res) => {
  App.findByIdAndRemove(req.params.messageId)
    .then((data) => {
      if (!data) {
        return res.status(404).send({
          message: "Message not found with id " + req.params.messageId,
        });
      }
      res.send({ message: "Message deleted successfully!" });
    })
    .catch((err) => {
      if (err.kind === "ObjectId" || err.name === "NotFound") {
        return res.status(404).send({
          message: "Message not found with id " + req.params.messageId,
        });
      }
      return res.status(500).send({
        message: "Could not delete message with id " + req.params.messageId,
      });
    });
};
```