

1. Project Structure

```
bash
Copier le code
/message-board
  /routes
    index.js
    new.js
  /views
    index.ejs
    form.ejs
    message.ejs
  /public
    app.js
```

2. app.js

In `app.js`, we will import the two route files and use them in the app:

```
javascript
Copier le code
const express = require("express");
const app = express();
const port = 3000;

// Import routes
const indexRouter = require("./routes/index");
const newRouter = require("./routes/new");

// Set EJS as the template engine
app.set("view engine", "ejs");

// Middleware to parse form data
app.use(express.urlencoded({ extended: true }));

// Use the routes
app.use("/", indexRouter);
app.use("/new", newRouter);

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

3. routes/index.js

In `routes/index.js`, we define the route for the index page, including handling for showing messages and individual message details:

```
javascript
Copier le code
const express = require("express");
const router = express.Router();

const messages = [
  { text: "Hi there!", user: "Amando", added: new Date() },
  { text: "Hello World!", user: "Charles", added: new Date() }
];

// Index route - Display all messages
router.get("/", (req, res) => {
```

```

    res.render("index", { title: "Mini Messageboard", messages: messages });
  });

// Message details route - Show a single message
router.get("/message/:id", (req, res) => {
  const message = messages[req.params.id];
  res.render("message", { message });
});

module.exports = router;

```

4. routes/new.js

In `routes/new.js`, we define the routes for showing the form and handling the form submission:

```

javascript
Copier le code
const express = require("express");
const router = express.Router();
const indexRouter = require("../index"); // We need access to the messages array
from indexRouter

// Route to display the form for adding a new message
router.get("/", (req, res) => {
  res.render("form");
});

// Route to handle form submission and add a new message
router.post("/", (req, res) => {
  const messageText = req.body.messageText;
  const messageUser = req.body.messageUser;
  indexRouter.messages.push({ text: messageText, user: messageUser, added: new
Date() });
  res.redirect("/");
});

module.exports = router;

```

5. Update routes/index.js to export messages

To allow `new.js` to access the `messages` array, we need to export it from `index.js`. Update the code in `routes/index.js` as follows:

```

javascript
Copier le code
const express = require("express");
const router = express.Router();

const messages = [
  { text: "Hi there!", user: "Amando", added: new Date() },
  { text: "Hello World!", user: "Charles", added: new Date() }
];

// Index route - Display all messages
router.get("/", (req, res) => {
  res.render("index", { title: "Mini Messageboard", messages: messages });
});

// Message details route - Show a single message
router.get("/message/:id", (req, res) => {

```

```

    const message = messages[req.params.id];
    res.render("message", { message });
  });

module.exports = {
  router,
  messages
};

```

Now, we are exporting both the router and the messages array. In `new.js`, you can access the messages array by requiring `index.js`.

6. Views

No changes are needed in the views, but here is a quick reminder of what they should look like:

`index.ejs`

```

html
Copier le code
<!DOCTYPE html>
<html>
<head>
  <title><%= title %></title>
</head>
<body>
  <h1><%= title %></h1>

  <ul>
    <% messages.forEach((message, index) => { %>
      <li>
        <strong><%= message.user %>:</strong> <%= message.text %>
        <em>(<%= message.added.toLocaleString() %>)</em>
        <a href="/message/<%= index %>">Open</a>
      </li>
    <% }) %>
  </ul>

  <a href="/new">Add a new message</a>
</body>
</html>

```

`form.ejs`

```

html
Copier le code
<!DOCTYPE html>
<html>
<head>
  <title>New Message</title>
</head>
<body>
  <h1>New Message</h1>

  <form method="POST" action="/new">
    <div>
      <label for="messageUser">Your name:</label>
      <input type="text" name="messageUser" id="messageUser" required>
    </div>
    <div>
      <label for="messageText">Message:</label>

```

```

        <input type="text" name="messageText" id="messageText" required>
    </div>
    <button type="submit">Submit</button>
</form>

    <a href="/">Back to the messages</a>
</body>
</html>

```

message.ejs

```

html
Copier le code
<!DOCTYPE html>
<html>
<head>
    <title>Message Details</title>
</head>
<body>
    <h1>Message from <%= message.user %></h1>

    <p><%= message.text %></p>
    <em>Added on <%= message.added.toLocaleString() %></em>

    <a href="/">Back to messages</a>
</body>
</html>

```

7. Run the App

Start the server by running:

```

bash
Copier le code
node app.js

```

Visit `http://localhost:3000/` to see the message board. You can visit `http://localhost:3000/new` to add a new message. After submitting, you'll be redirected back to the index page with the updated messages.