

Web application technologies final project

MongoDB, Express, React, Node.js chat
web application

Karlo Slovjak

Krakow, June 2017.

1. Dependencies used

- Axios: lets us use HTTP methods to communicate with our database.
- Express: provides a framework to set up our Node.js server.
- Body Parser: parses the incoming requests bodies for the info we are sending -> req.body
- Foreman: allows us to boot up our API and webpack-dev-server simultaneously.
- Nodemon: watches our server.js file for changes, then restarts it.
- Marked: to convert markdown syntax to HTML
- Mongoose: abstracts away the MongoDB boilerplate for a simple Schema solution.

2. Description of components

- model/messages.js:

Creates new instance of mongoose schema. Schema has shape of database entries.

- public/index.html:

Template html with one root div where other HTML elements are added.

- src/Message.js:

Handling adding, updating and deleting messages.

- src/MessageBox.js:

Message box where messages

- src/MessageForm.js:

Form for posting message.

- `src/MessageList.js`:

Messages list.

- `src/style.js`

File with styles for application.

- `server.js`

Handling user requests and mapping HTTP requests to corresponding URI. Connecting to database and making operations on it.

3. Conclusion

App has basic functionalities for local mode of operation. Messages are preserved within sessions and main goals were achieved.

Further development will be focused on making app available outside localhost and improving functionalities.