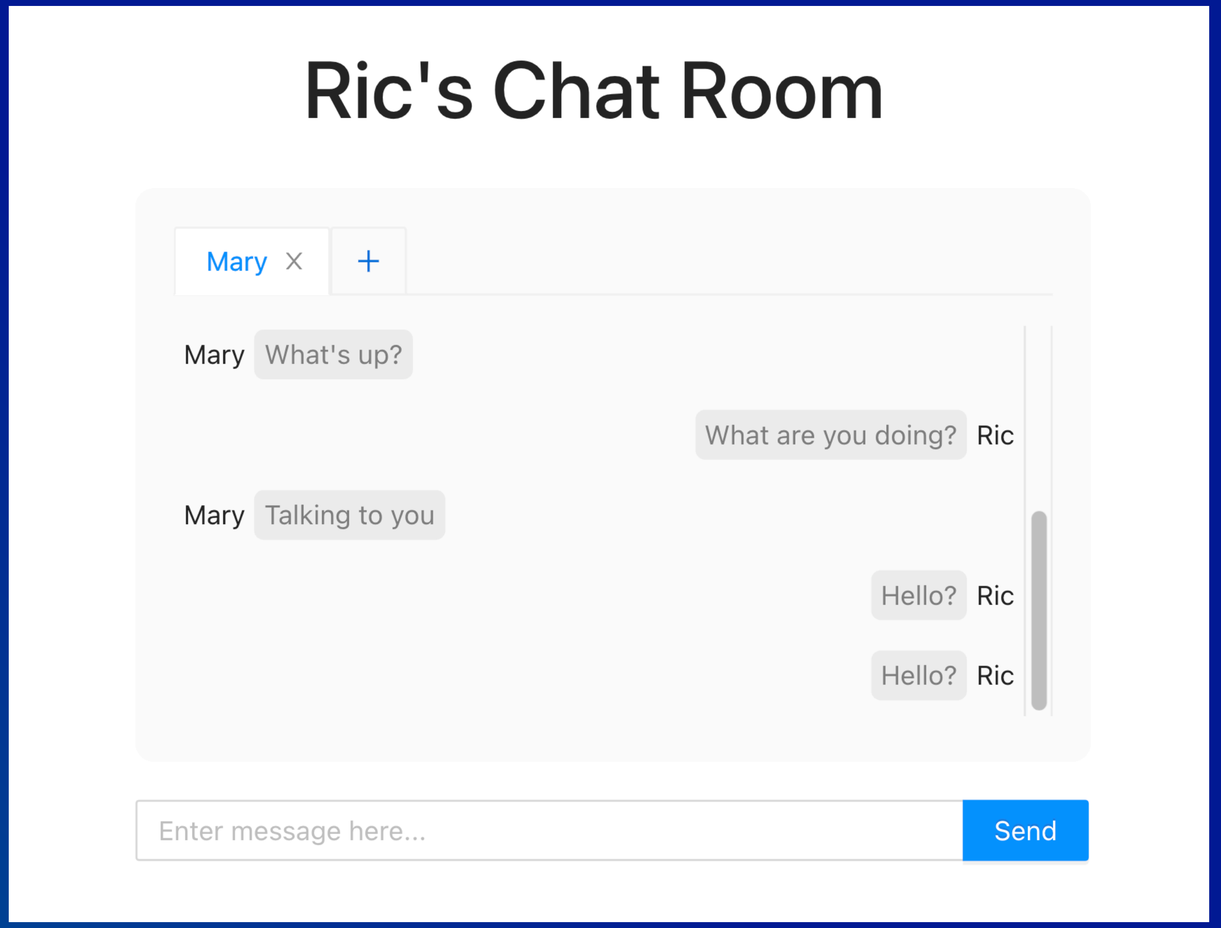
HW#7 Backend



Ric Huang / NTUEE

(EE 3035) Web Programming

基本上 code 都已經給大家了，為了節省時間，這個說明經不再把 code 貼上來，請大家先把 code 準備好，並且 "yarn; yarn dev", 打開 localhoost:8080, 確定可以執行

- 原來的 code 是用 "localhost:27017/testmongo" 作為 MONGO_URL, 所以請大家自行改成 mongoose
- 承上，請從別的 project 把 .env 以及 .env.defaults, copy 過來，並且 yarn add dotenv-defaults
- On UI, 先按 "chat" 建立兩個人的 ChatBox, 然後才能 send message

如何開始讀 code?

先看 File Structure

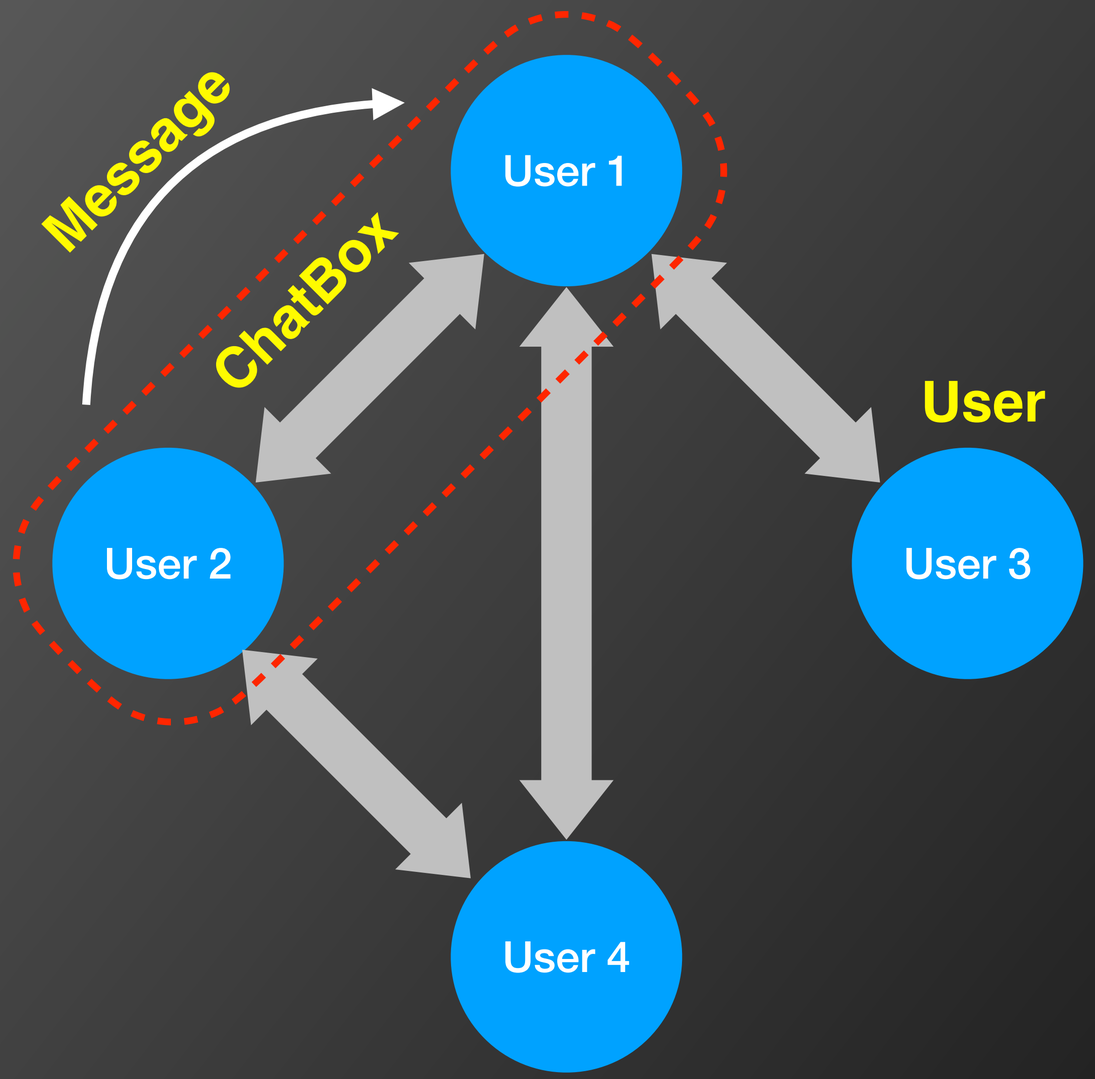
- public/
  - index.html

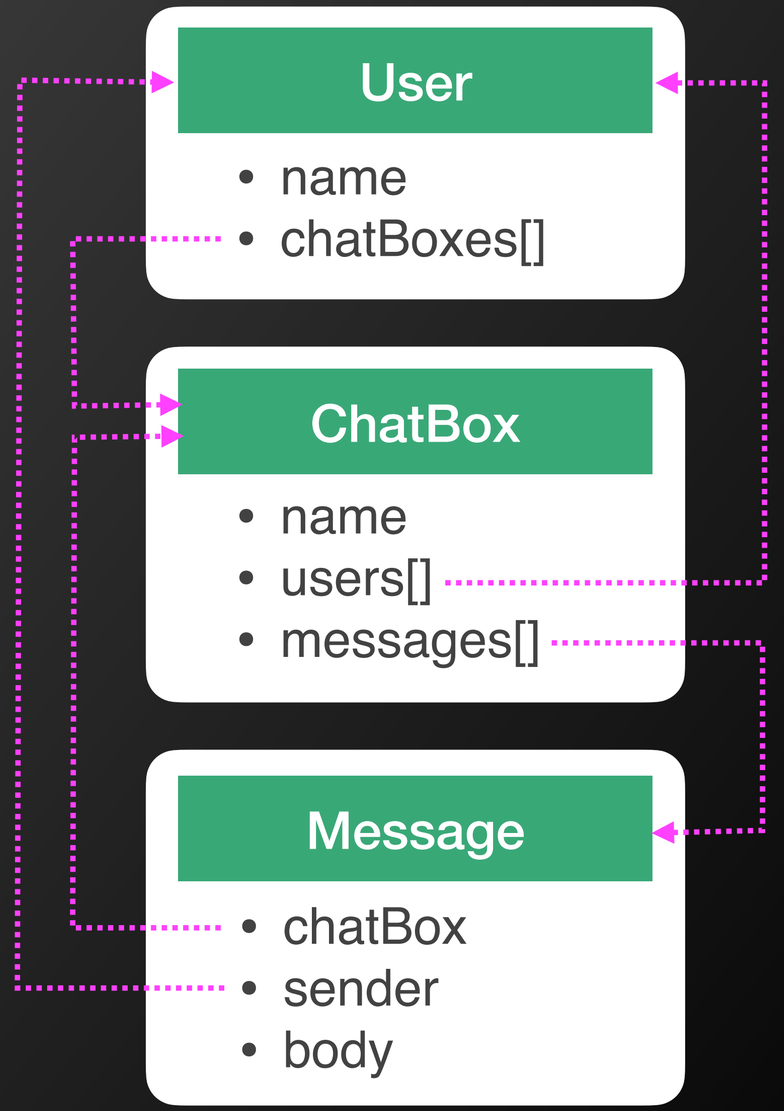
    Front-end UI, web socket client (JS code embedded)
- index.js

  MongoDB schema, Server main, web socket server
- mongo.js

  MongoDB connection

再思考一下 DB 的設計


Message
ChatBox
User 1
User 2
User 3
User 4
User


User
• name
• chatBoxes[]
ChatBox
• name
• users[]
• messages[]
Message
• chatBox
• sender
• body

請看 index.js (Schema Design)

- For foreign key:
  - type: mongoose.Types.ObjectId
  - ref: foreign key referenced schema
- const UserModel = mongoose.model("User", userSchema);
  - Ref: In "ScoreCard.js",

```
1  import { model, Schema } from 'mongoose';
2
3  const scoreCardSchema = new Schema({
4    name: { type: String, required: true },
5    subject: { type: String, required: true },
6    score: { type: Number, required: true },
7  });
8
9  export default model('ScoreCard', scoreCardSchema);
```

再看 mongo.js (connecting to Mongoose)

- function connectMongo
  => 基本的 mongoose connection, 以後照著用就好

- In mongo.js —

```
const mongo = {
  connect: connectMongo,
};
module.exports = mongo;
```

Why not:
module.exports = connectMongo; ?

- In index.js —

```
const mongo = require(“./mongo");
mongo.connect();
```

再回來看 index.js

```
app.use(express.static(path.join(__dirname, "public")));
```

- 這個是在做什麼？
- Recall: app.use([routingPath], callback)
- "__dirname" 是 node 內建的參數
- "path" 是一個 node 的套件 (用 terminal 玩玩看)
- 有定義 [routingPath] 嗎？ 沒有，所以所有 '/' 的 request 都是交給 express.static() 處理
- express.static() 是在做什麼？ 到 terminal 試試看
- 所以 express.static() return 一個 function

進一步了解 mongoDB

- 再看一次 mongoose.model() 的 document

  The first argument is the singular name of the collection your model is for. Mongoose automatically looks for the plural, lowercased version of your model name. Thus, for the example above, the model Tank is for the tanks collection in the database.

- 產生一個新的 user 並且存起來

```
return new UserModel({ name }).save();
```

- thisDoc.populate(fieldName); (ref)
  - box.populate("users");
  - Since user in "users" is a foreign key, when box is created, it stores the objectID only. After populate, it links to the actual object. // 用 debug code 試試看！

WebSocket Connection

- 與之前的 simple chat 比較
- In "server.js"

```
mongoose.connect(...);
const db = mongoose.connection;

db.on('error', (error) => {...});

const wssConnect = ws => { ... }

db.once('open', () => {
  console.log('MongoDB connected!')
  // Server WebSocket 的主邏輯
  wss.on('connection', wssConnect)
  const PORT = process.env.port || 4000
  server.listen(PORT, () => {
    console.log(`Listening on
        http://localhost:${PORT}`)
  })
})
```

- This backend...
- In "index.js"

```
// Server WebSocket 的主邏輯
wss.on("connection", function...);

monogo.connect(); // connectMongo()
server.listen(8080, () => {
  console.log("Server listening at
            http://localhost:8080");
});
```

- In "mongo.js"

```
function connectMongo() {
  mongoose.connect(...);
  const db = mongoose.connection;
  db.on('error',...);
  db.once('open',() => {
    console.log('MongoDB connected!');
  });
}
```

wss.on("connection", function connection(client)...);

- In function connection(client) —

  Note: message as { type, data }

```
client.id = uuid.v4();
client.box = ""; // keep track of client's CURRENT chat box
client.sendEvent = (e) => client.send(JSON.stringify(e));
client.on("message", async (message) => {
  message = JSON.parse(message);
  const { type } = message;
  switch (type) {
    case "CHAT": { ... } // on open chatBox
    case "MESSAGE": { ... } // receive and broadcast message
  }
  client.once("close", () => {
    chatBoxes[client.box].delete(client);
  });
);
```

事實上，
UserModel({chatBoxes}) 與
messageModel({chatBox})
並沒有用到 (Why?)
所以其實可以直接拿掉

最後看一下 public/index.html

感謝聆聽！

Ric Huang / NTUEE

(EE 3035) Web Programming