

## **Module 6 Lab 1: Adding Business Logics, UI, and Protecting Resources**

Kayana L. Wenglarz

Arizona State University

IFT458: Middleware Prog & Database Sec

Prof. Dinesh Sthapit

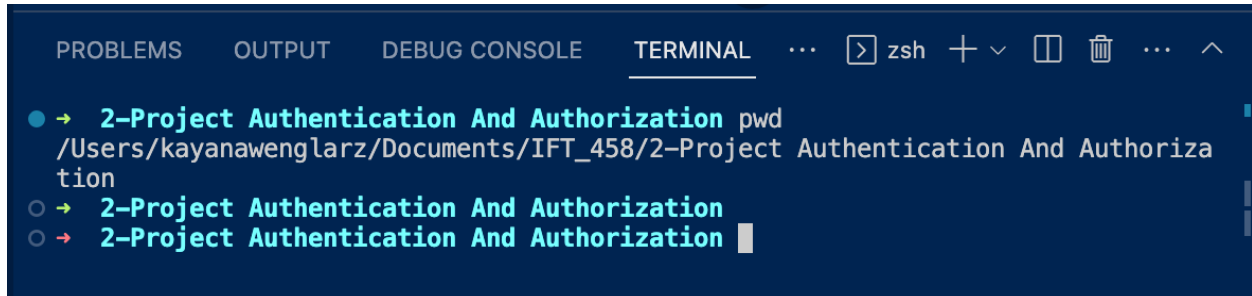
November 5, 2023

## Module 6 Lab 1: Adding Business Logics, UI, and Protecting Resources

### Step 0

Figure 1

*First, I opened the project and ensured it was not running.*

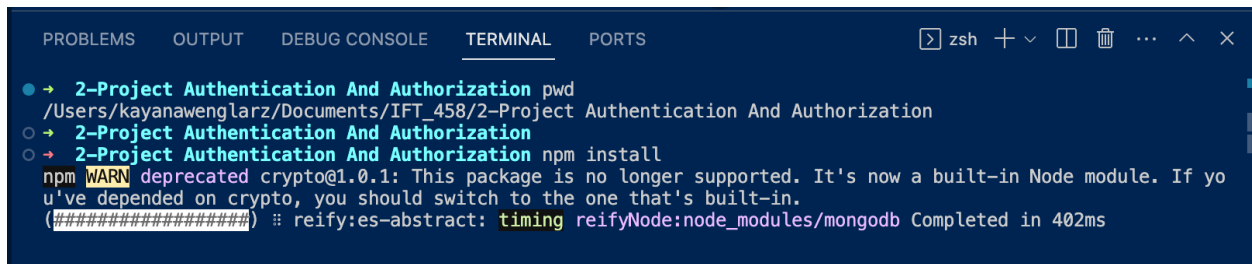


The screenshot shows a terminal window with a dark blue background. The terminal title bar includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, along with icons for zsh, a plus sign, a minus sign, a square, a trash can, and a caret. The terminal content shows the command `pwd` being executed, resulting in the path `/Users/kayanawenglarz/Documents/IFT_458/2-Project Authentication And Authorization`. Below the terminal, a file explorer shows the same project name, `2-Project Authentication And Authorization`, with a plus sign icon next to it.

### Step 1

Figure 2

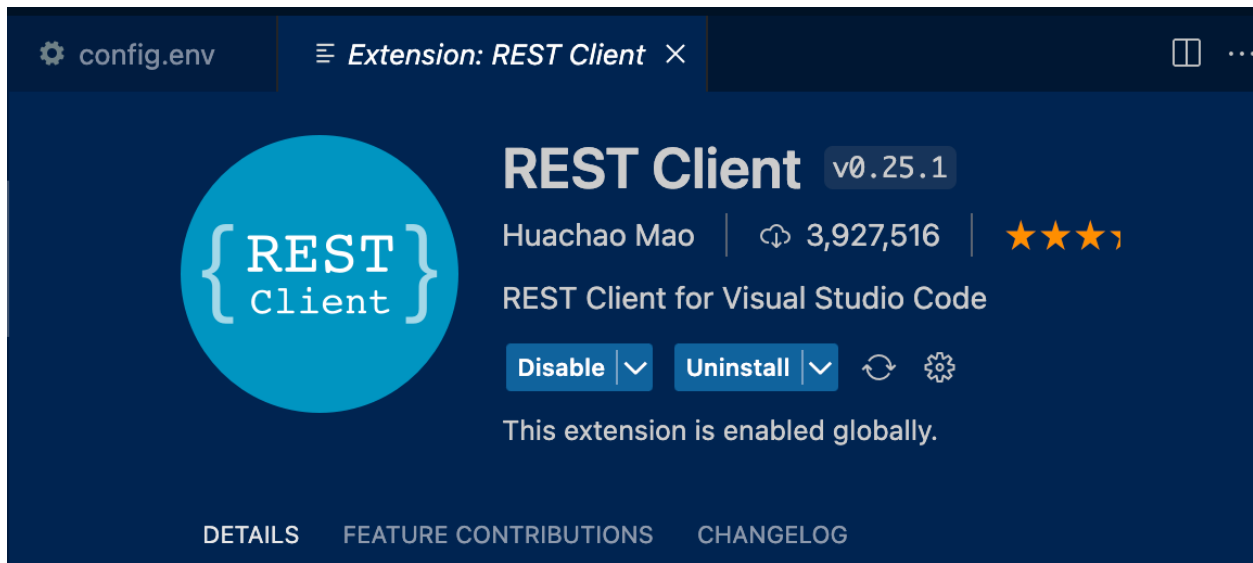
*Next, I restored the project by running `npm install`.*



The screenshot shows a terminal window with a dark blue background. The terminal title bar includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, along with icons for zsh, a plus sign, a minus sign, a square, a trash can, and a caret. The terminal content shows the command `npm install` being executed. The output includes a warning message: `npm WARN deprecated crypto@1.0.1: This package is no longer supported. It's now a built-in Node module. If you've depended on crypto, you should switch to the one that's built-in.` followed by a line indicating the installation of `reify:es-abstract: timing reifyNode:node_modules/mongodb Completed in 402ms`.

Figure 3

*As instructed, I installed the REST client.*



### Steps 2 & 3

**Figure 4**

*I didn't need to create the X-Unit Tests folder since it was already created, but I created the test file and wrote the http requests.*

```

X-UnitTests > book-requests-1215776016.http > ...
Send Request
1 GET http://localhost:4000/api/v1/books
2 ###
Send Request
3 GET http://localhost:4000/api/v1/books/1
4 ###
Send Request
5 POST http://localhost:4000/api/v1/books
6 Content-Type: application/json
7
8 {
9     "title": "Harry Potter and the Sorcerer's Stone",
10    "author": "J.K. Rowling",
11    "genre": "Fantasy",
12    "year": 1997
13 }
14
15 ###

```

## Steps 4 &5

**Figure 5**

*I clicked on Send Request for the get all books route and received a 200 response.*

```

book-requests-1215776016.http  JS bookRoutes.js  ...  Response(7ms)  X
X-UnitTests > book-requests-1215776016.http > ...
Send Request
1 GET http://localhost:4000/api/v1/books
2 ###
Send Request
3 GET http://localhost:4000/api/v1/books/1
4 ###
Send Request
5 POST http://localhost:4000/api/v1/books
6 Content-Type: application/json
7
8 {
9   "title": "Harry Potter and the Sorcerer's Stone",
10  "author": "J.K. Rowling",
11  "genre": "Fantasy",
12  "year": 1997
13 }
14
15 ###

1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: text/html; charset=utf-8
4 Content-Length: 1771
5 ETag: W/"6eb-4NY4XMcIcb/hFigX+LVbUbemtWI"
6 Date: Fri, 03 Nov 2023 17:18:48 GMT
7 Connection: close
8
9 <!DOCTYPE html>
10 <html>
11 <head>
12 <title>Login</title>
13 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous">
14 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hpcxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>
15
16 </head>
17 <body>
18 <div id="currentDateTime"></div>

```

**Figure 6**

*Next, I clicked on Send Request for get book by ID route. I also received a 200 response for this request.*

```

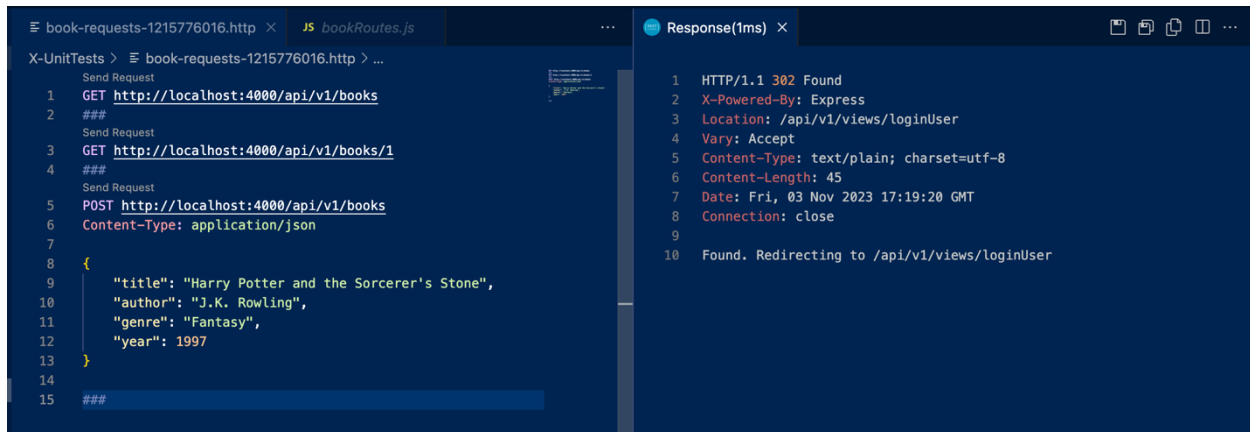
book-requests-1215776016.http  JS bookRoutes.js  ...  Response(2ms)  X
X-UnitTests > book-requests-1215776016.http > ...
Send Request
1 GET http://localhost:4000/api/v1/books
2 ###
Send Request
3 GET http://localhost:4000/api/v1/books/1
4 ###
Send Request
5 POST http://localhost:4000/api/v1/books
6 Content-Type: application/json
7
8 {
9   "title": "Harry Potter and the Sorcerer's Stone",
10  "author": "J.K. Rowling",
11  "genre": "Fantasy",
12  "year": 1997
13 }
14
15 ###

1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: text/html; charset=utf-8
4 Content-Length: 1771
5 ETag: W/"6eb-4NY4XMcIcb/hFigX+LVbUbemtWI"
6 Date: Fri, 03 Nov 2023 17:19:10 GMT
7 Connection: close
8
9 <!DOCTYPE html>
10 <html>
11 <head>
12 <title>Login</title>
13 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous">
14 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hpcxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>
15
16 </head>
17 <body>
18 <div id="currentDateTime"></div>
19
20 <script>

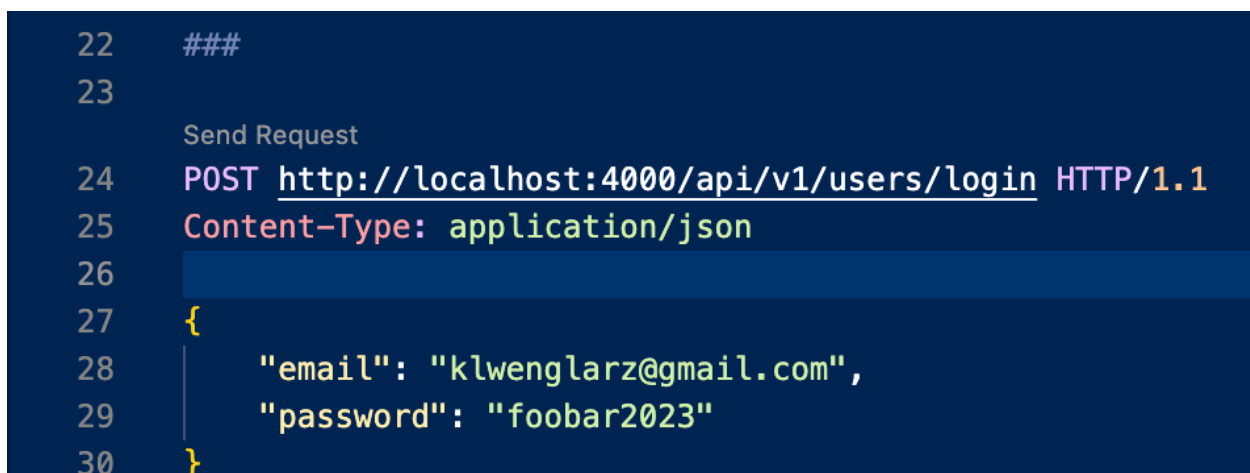
```

**Figure 7**

*I clicked on Send Request for the post route. I received a 302 response for this, but it showed that it redirected to the login page.*

**Figure 8**

*Since all three redirected to a login page, I knew I had missed a step. I added a fourth request to validate the user so we could make these requests and see the desired results. I added the authorization JWT into the request for each of the other requests I had written.*

**Figure 9**

*After that, I sent the requests again and they came back successful and showing information.*

*This is the get all books request.*



```
Response(2813ms) X
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 565
5 ETag: W/"235-qBF1xjxK0wio3lcN4Igge0wB7UE"
6 Date: Fri, 03 Nov 2023 18:26:45 GMT
7 Connection: close
8
9 [
10   {
11     "_id": "654537afb06cbc114ff070e3",
12     "title": "harry",
13     "author": "J.K. Rowling",
14     "description": "foo",
15     "exchangeType": "trade",
16     "owner": "653da5605a5c93f99555ed8f",
17     "status": "available",
18     "createdAt": "2023-11-03T18:10:57.628Z",
19     "updatedAt": "2023-11-03T18:10:57.628Z",
20     "__v": 0
21   },
22   {
23     "_id": "65453a65cecea75420b92525",
24     "title": "Harry Potter and the Sorcerer's Stone",
25     "author": "J.K. Rowling",
26     "description": "children's book",
27     "exchangeType": "trade",
28     "owner": "653da5605a5c93f99555ed8f",
29     "status": "available",
30     "createdAt": "2023-11-03T18:22:29.224Z",
31     "updatedAt": "2023-11-03T18:22:29.224Z",
32     "__v": 0
33   }
34 ]
```

**Figure 10**

*I then ran the request to get a specific book by id.*

```

book-requests-1215776016.http x loginForm.ejs JS bookRoutes.js JS authMiddleware.js ... Response(5969ms) x
X-UnitTests > book-requests-1215776016.http > POST /api/v1/users/login
Send Request
1 GET http://localhost:4000/api/v1/books
2 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY1M2RhNTYwNWE1YzZkZjksNTU1ZWQ4ZiIsImhhdCI6MTY5OTAzNTYzOCwiZXhwIjojNjk5ODk5NjM4fQ.01eZzdXswg9t4EdT_URdAtaJ2lBXC5Do1Qx8XdUa0YY!
3
4 ###
Send Request
5 GET http://localhost:4000/api/v1/books/65453a65cecea75420b92525
6 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY1M2RhNTYwNWE1YzZkZjksNTU1ZWQ4ZiIsImhhdCI6MTY5OTAzNTYzOCwiZXhwIjojNjk5ODk5NjM4fQ.01eZzdXswg9t4EdT_URdAtaJ2lBXC5Do1Qx8XdUa0YY!
7
8 ###
Send Request
9 POST http://localhost:4000/api/v1/books
10 Content-Type: application/json
11 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY1M2RhNTYwNWE1YzZkZjksNTU1ZWQ4ZiIsImhhdCI6MTY5OTAzNTYzOCwiZXhwIjojNjk5ODk5NjM4fQ.01eZzdXswg9t4EdT_URdAtaJ2lBXC5Do1Qx8XdUa0YY!
12
13 {
14   "title": "Harry Potter and the Sorcerer's Stone",
15   "author": "J.K. Rowling",
16   "genre": "Fantasy",
17   "year": 1997,
18   "description": "children's book",
19   "exchangeType": "trade"
20 }
21
22 ###
Send Request
24 POST http://localhost:4000/api/v1/users/login HTTP/1.1
25 Content-Type: application/json
26
27 {
28   "email": "klwenglarz@gmail.com"
29 }
1 HTTP/1.1 201 Created
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 408
5 ETag: W/"198-z8PSAh1FaXTaRTjnPRswv138w9E"
6 Date: Fri, 03 Nov 2023 18:28:05 GMT
7 Connection: close
8
9 {
10   "title": "Harry Potter and the Sorcerer's Stone",
11   "author": "J.K. Rowling",
12   "description": "children's book",
13   "exchangeType": "trade",
14   "owner": {
15     "_id": "653da5605a5c93f99555ed8f",
16     "name": "kayana wenglarz",
17     "email": "klwenglarz@gmail.com",
18     "role": "client",
19     "roles": "client",
20     "__v": 0
21   },
22   "status": "available",
23   "_id": "65453bb3cecea75420b92531",
24   "createdAt": "2023-11-03T18:28:03.572Z",
25   "updatedAt": "2023-11-03T18:28:03.572Z",
26   "__v": 0
27 }

```

Figure 11

*I created another book to update the description and show the status code.*

```

book-requests-1215776016.http x loginForm.ejs JS bookRoutes.js JS authMiddleware.js ... Response(3844ms) x
X-UnitTests > book-requests-1215776016.http > POST /api/v1/books
Send Request
1 GET http://localhost:4000/api/v1/books
2 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY1M2RhNTYwNWE1YzZkZjksNTU1ZWQ4ZiIsImhhdCI6MTY5OTAzNTYzOCwiZXhwIjojNjk5ODk5NjM4fQ.01eZzdXswg9t4EdT_URdAtaJ2lBXC5Do1Qx8XdUa0YY!
3
4 ###
Send Request
5 GET http://localhost:4000/api/v1/books/65453a65cecea75420b92525
6 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY1M2RhNTYwNWE1YzZkZjksNTU1ZWQ4ZiIsImhhdCI6MTY5OTAzNTYzOCwiZXhwIjojNjk5ODk5NjM4fQ.01eZzdXswg9t4EdT_URdAtaJ2lBXC5Do1Qx8XdUa0YY!
7
8 ###
Send Request
9 POST http://localhost:4000/api/v1/books
10 Content-Type: application/json
11 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY1M2RhNTYwNWE1YzZkZjksNTU1ZWQ4ZiIsImhhdCI6MTY5OTAzNTYzOCwiZXhwIjojNjk5ODk5NjM4fQ.01eZzdXswg9t4EdT_URdAtaJ2lBXC5Do1Qx8XdUa0YY!
12
13 {
14   "title": "Harry Potter and the Sorcerer's Stone",
15   "author": "J.K. Rowling",
16   "genre": "Fantasy",
17   "year": 1997,
18   "description": "children's book about a young wizard",
19   "exchangeType": "trade"
20 }
21
22 ###
Send Request
24 POST http://localhost:4000/api/v1/users/login HTTP/1.1
25 Content-Type: application/json
26
27 {
28   "email": "klwenglarz@gmail.com"
29 }
1 HTTP/1.1 201 Created
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 429
5 ETag: W/"1ad-HvyKSGAJhoIwcJKEdu2Nw6qpFo"
6 Date: Fri, 03 Nov 2023 18:30:39 GMT
7 Connection: close
8
9 {
10   "title": "Harry Potter and the Sorcerer's Stone",
11   "author": "J.K. Rowling",
12   "description": "children's book about a young wizard",
13   "exchangeType": "trade",
14   "owner": {
15     "_id": "653da5605a5c93f99555ed8f",
16     "name": "kayana wenglarz",
17     "email": "klwenglarz@gmail.com",
18     "role": "client",
19     "roles": "client",
20     "__v": 0
21   },
22   "status": "available",
23   "_id": "65453c4cecea75420b92534",
24   "createdAt": "2023-11-03T18:30:38.117Z",
25   "updatedAt": "2023-11-03T18:30:38.117Z",
26   "__v": 0
27 }

```

I can see how useful the REST client is when testing an application. It allows the user to send the HTTP request and view the response all in VSCode. It seems pretty similar to Postman in terms of what it does, but it's nice that it is all within VSCode where we are building our application. This makes the testing experience seamless and allows for faster testing.