

lab4

The screenshot shows the MySQL Workbench interface. In the top-left pane, there's a tree view of database management, performance monitoring, and instance status. The main area contains a query editor with the following SQL code:

```
1 -- CREATE DATABASE moviedb;
2 USE moviedb;
3 
4 -- CREATE TABLE movies(title VARCHAR(50) NOT NULL,genre VARCHAR(30) NOT NULL,director VARCHAR(60) NOT NULL,release_year INT NOT NULL,PRIMARY KEY(title));
5 
6 -- INSERT INTO movies VALUE ("angrybird","psychological thriller","Todd Phillips", 2019);
7 
8 -- SELECT * FROM movies;
9 
10
11
12
13
```

Below the query editor is a results grid displaying the data from the movies table:

title	genre	director	release_year
angrybird	psychological thriller	Todd Phillips	2019
luser1	psychological thriller	Todd Phillips	2019
luser2	psychological thriller	Todd Phillips	2019
luser3	psychological thriller	Todd Phillips	2019
luser4	psychological thriller	Todd Phillips	2019
luser5	psychological thriller	Todd Phillips	2019
luser6	psychological thriller	Todd Phillips	2019
luser7	psychological thriller	Todd Phillips	2019
luser8	psychological thriller	Todd Phillips	2019
luser9	psychological thriller	Todd Phillips	2019
luser10	psychological thriller	Todd Phillips	2019
luser11	psychological thriller	Todd Phillips	2019
luser12	psychological thriller	Todd Phillips	2019
luser13	psychological thriller	Todd Phillips	2019
luser14	psychological thriller	Todd Phillips	2019
luser15	psychological thriller	Todd Phillips	2019
luser16	psychological thriller	Todd Phillips	2019
luser17	psychological thriller	Todd Phillips	2019
luser18	psychological thriller	Todd Phillips	2019
luser19	psychological thriller	Todd Phillips	2019
luser20	psychological thriller	Todd Phillips	2019
luser21	psychological thriller	Todd Phillips	2019
luser22	psychological thriller	Todd Phillips	2019
luser23	psychological thriller	Todd Phillips	2019
luser24	psychological thriller	Todd Phillips	2019
luser25	psychological thriller	Todd Phillips	2019
luser26	psychological thriller	Todd Phillips	2019
luser27	psychological thriller	Todd Phillips	2019
luser28	psychological thriller	Todd Phillips	2019
luser29	psychological thriller	Todd Phillips	2019
luser30	psychological thriller	Todd Phillips	2019
luser31	psychological thriller	Todd Phillips	2019
luser32	psychological thriller	Todd Phillips	2019
luser33	psychological thriller	Todd Phillips	2019
luser34	psychological thriller	Todd Phillips	2019
luser35	psychological thriller	Todd Phillips	2019
luser36	psychological thriller	Todd Phillips	2019
luser37	psychological thriller	Todd Phillips	2019
luser38	psychological thriller	Todd Phillips	2019
luser39	psychological thriller	Todd Phillips	2019
luser40	psychological thriller	Todd Phillips	2019
luser41	psychological thriller	Todd Phillips	2019
luser42	psychological thriller	Todd Phillips	2019
luser43	psychological thriller	Todd Phillips	2019
luser44	psychological thriller	Todd Phillips	2019
luser45	psychological thriller	Todd Phillips	2019
luser46	psychological thriller	Todd Phillips	2019
luser47	psychological thriller	Todd Phillips	2019
luser48	psychological thriller	Todd Phillips	2019
luser49	psychological thriller	Todd Phillips	2019
luser50	psychological thriller	Todd Phillips	2019
luser51	psychological thriller	Todd Phillips	2019
luser52	psychological thriller	Todd Phillips	2019
luser53	psychological thriller	Todd Phillips	2019
luser54	psychological thriller	Todd Phillips	2019
luser55	psychological thriller	Todd Phillips	2019
luser56	psychological thriller	Todd Phillips	2019
luser57	psychological thriller	Todd Phillips	2019
luser58	psychological thriller	Todd Phillips	2019
luser59	psychological thriller	Todd Phillips	2019
luser60	psychological thriller	Todd Phillips	2019
luser61	psychological thriller	Todd Phillips	2019
luser62	psychological thriller	Todd Phillips	2019
luser63	psychological thriller	Todd Phillips	2019
luser64	psychological thriller	Todd Phillips	2019
luser65	psychological thriller	Todd Phillips	2019
luser66	psychological thriller	Todd Phillips	2019
luser67	psychological thriller	Todd Phillips	2019
luser68	psychological thriller	Todd Phillips	2019
luser69	psychological thriller	Todd Phillips	2019
luser70	psychological thriller	Todd Phillips	2019
luser71	psychological thriller	Todd Phillips	2019
luser72	psychological thriller	Todd Phillips	2019
luser73	psychological thriller	Todd Phillips	2019
luser74	psychological thriller	Todd Phillips	2019
luser75	psychological thriller	Todd Phillips	2019
luser76	psychological thriller	Todd Phillips	2019
luser77	psychological thriller	Todd Phillips	2019
luser78	psychological thriller	Todd Phillips	2019
luser79	psychological thriller	Todd Phillips	2019
luser80	psychological thriller	Todd Phillips	2019
luser81	psychological thriller	Todd Phillips	2019
luser82	psychological thriller	Todd Phillips	2019
luser83	psychological thriller	Todd Phillips	2019
luser84	psychological thriller	Todd Phillips	2019
luser85	psychological thriller	Todd Phillips	2019
luser86	psychological thriller	Todd Phillips	2019
luser87	psychological thriller	Todd Phillips	2019
luser88	psychological thriller	Todd Phillips	2019
luser89	psychological thriller	Todd Phillips	2019
luser90	psychological thriller	Todd Phillips	2019
luser91	psychological thriller	Todd Phillips	2019
luser92	psychological thriller	Todd Phillips	2019
luser93	psychological thriller	Todd Phillips	2019
luser94	psychological thriller	Todd Phillips	2019
luser95	psychological thriller	Todd Phillips	2019
luser96	psychological thriller	Todd Phillips	2019
luser97	psychological thriller	Todd Phillips	2019
luser98	psychological thriller	Todd Phillips	2019
luser99	psychological thriller	Todd Phillips	2019
luser100	psychological thriller	Todd Phillips	2019

At the bottom, the status bar shows "Query Completed" and the query details: "Time 16-04-13 Action SELECT * FROM movies LIMIT 0,1000".

The screenshot shows the Postman application interface. On the left, the "My Workspace" sidebar lists collections, environments, and history. The main workspace shows a "New Request" configuration for a GET request to "http://localhost:4000". The "Params" tab is selected, showing a table with one row and columns for "Key" and "Value". The "Body" tab is also visible, containing the response body: "

```
<h3> Welcome to API Back-end Ver. 1.0.0 </h3>
```

". At the bottom, there are various toolbars and status indicators.

The screenshot shows the Postman application interface. In the left sidebar, under 'My Workspace', there is a collection named 'ENGSE203' which contains a folder 'LAB4' with a 'getmovie' request. The main workspace displays a 'getmovie' request with the URL `http://localhost:4000/api/movie/search?search_text=jok`. The 'Params' tab is selected, showing two query parameters: 'search_text' with value 'jok'. The 'Body' tab is active, showing the response body in 'Pretty' format:

```
1  [{"statusCode":200,"returnCode":1,"data":[{"title":"Joker","genre":"psychological thriller","director":"Todd Phillips","release_year":2019},{"title":"Joker1","genre":"psychological thriller","director":"Todd Phillips","release_year":2019},{"title":"Joker2","genre":"psychological thriller","director":"Todd Phillips","release_year":2019},{"title":"Joker3","genre":"psychological thriller","director":"Todd Phillips","release_year":2019},{"title":"Joker4","genre":"psychological thriller","director":"Todd Phillips","release_year":2019},{"title":"Joker5","genre":"psychological thriller","director":"Todd Phillips","release_year":2019},{"title":"Joker6","genre":"psychological thriller","director":"Todd Phillips","release_year":2019},{"title":"Joker7","genre":"psychological thriller","director":"Todd Phillips","release_year":2019}]}]
```

The status bar at the bottom indicates a 200 OK response with 12 ms and 1.21 KB.

Home Workspaces API Network

Search Postman

Invite Upgrade

My Workspace

New Import

LAB4 / New Request

POST http://localhost:4000/api/movie/insert

Params Authorization Headers (9) Body **JSON** Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "title": "Joker71",  
3   "genze": "psychological-thriller",  
4   "director": "Todd Phillips",  
5   "release_year": 2019  
6 }  
7
```

Response

Click Send to get a response

Online Find and replace Console

Postbot Runner Start Proxy Cookies Trash

Teams and Channels | x | localhost:4000 | x | React App | x | YouTube | x | lab4-engse203 - Google | +

localhost:3001

Lab4 Movie

Create Movies

title

genre

director

release_year

[Insert](#)

Movies Created

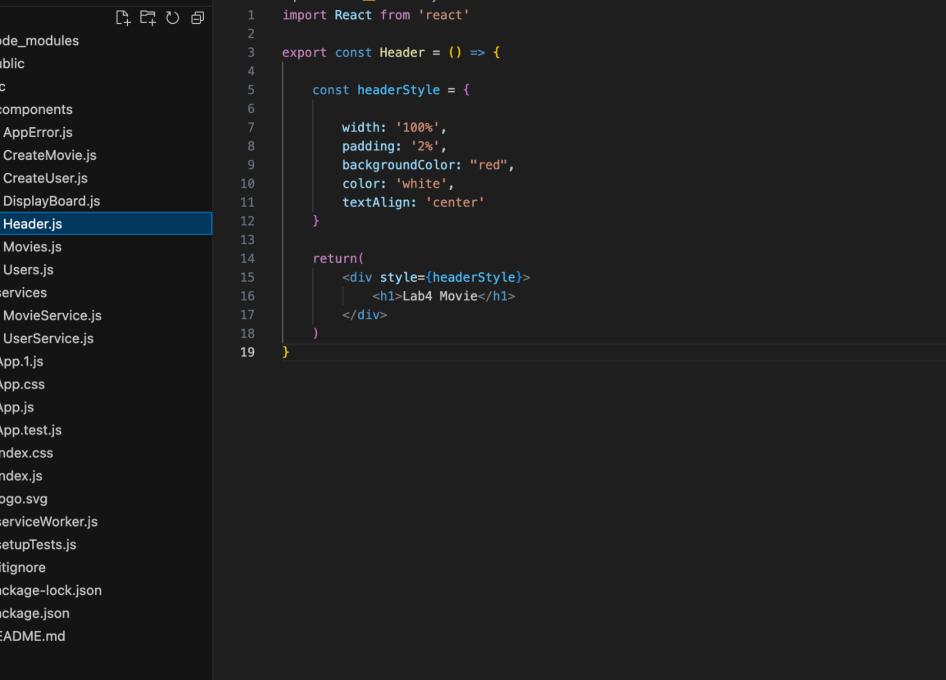
10

[Get all Movies](#)

Movies

Movie Id	Title	Genre	Director	Release
1	angrybird	psychological thriller	Todd Phillips	2019
2	Joker	psychological thriller	Todd Phillips	2019
3	Joker1	psychological thriller	Todd Phillips	2019
4	Joker2	psychological thriller	Todd Phillips	2019
5	Joker3	psychological thriller	Todd Phillips	2019
6	Joker4	psychological thriller	Todd Phillips	2019
7	Joker5	psychological thriller	Todd Phillips	2019
8	Joker6	psychological thriller	Todd Phillips	2019
9	Joker7	psychological thriller	Todd Phillips	2019
10	Joker71	psychological thriller	Todd Phillips	2019

code my-app



The screenshot shows the VS Code interface with the following details:

- EXPLORER** view on the left showing the project structure under **MY-APP**, including files like **App.js**, **CreateUser.js**, **CreateMovie.js**, **DisplayBoard.js**, **Movies.js**, **Header.js** (which is selected), **AppError.js**, **CreateMovie.js**, **CreateUser.js**, **DisplayBoard.js**, **Movies.js**, **Header.js**, **App.1.js**, **App.css**, **App.js**, **App.test.js**, **index.css**, **index.js**, **logo.svg**, **serviceWorker.js**, **setupTests.js**, **.gitignore**, **package-lock.json**, **package.json**, and **README.md**.
- EDITOR** tab showing the code for **Header.js**:

```
src > components > Header.js > Header
1  import React from 'react'
2
3  export const Header = () => {
4
5      const headerStyle = {
6
7          width: '100%',
8          padding: '2%',
9          backgroundColor: "red",
10         color: 'white',
11         textAlign: 'center'
12     }
13
14     return(
15         <div style={headerStyle}>
16             <h1>Lab4 Movie</h1>
17         </div>
18     )
19 }
```

The screenshot shows a code editor interface with the following details:

- EXPLORER** sidebar on the left showing the project structure:
 - MY-APP
 - node_modules
 - public
 - src
 - components
 - AppError.js
 - CreateMovie.js
 - CreateUser.js
 - DisplayBoard.js
 - Header.js
 - Movies.js**
 - services
 - MovieService.js
 - UserService.js
 - App.1.js
 - App.css
 - App.js
 - App.test.js
 - index.css
 - index.js
 - logo.svg
 - serviceWorker.js
 - setupTests.js
 - .gitignore
 - package-lock.json
 - package.json
 - README.md
- OPEN EDITORS** tab bar at the top with tabs for App.js, CreateUser.js, CreateMovie.js, DisplayBoard.js, Movies.js (highlighted), Header.js, and Users.js.
- Movies.js** code editor tab showing the following code:

```
src > components > Movies.js > ...
1  import React from 'react'
2
3  export const Movies = ({movies}) => {
4
5    console.log('movies length:::', movies.length)
6    if (movies.length === 0) return null
7
8    const MovieRow = (movie, index) => {
9
10      return(
11        <tr key={index} className={index%2 === 0?'odd':'even'}>
12          <td>{index + 1}</td>
13          <td>{movie.title}</td>
14          <td>{movie.genre}</td>
15          <td>{movie.director}</td>
16          <td>{movie.release_year}</td>
17        </tr>
18      )
19    }
20
21    const movieTable = movies.map((movie, index) => MovieRow(movie, index))
22
23    return(
24      <div className="container">
25        <h2>Movies</h2>
26        <table className="table table-bordered">
27          <thead>
28            <tr>
29              <th>Movie Id</th>
30              <th>Title</th>
31              <th>Genre</th>
32              <th>Director</th>
33              <th>Release</th>
34            </tr>
35          </thead>
36          <tbody>
37            {movieTable}
38          </tbody>
39        </table>
40      </div>
41    )
42  }
43
```

The screenshot shows a code editor interface with the following details:

- EXPLORER** sidebar on the left displaying the project structure:
 - MY-APP
 - node_modules
 - src
 - components
 - services
 - MovieService.js (selected)
 - UserService.js
 - App.1.js
 - App.css
 - App.js
 - App.test.js
 - index.css
 - index.js
 - logo.svg
 - serviceWorker.js
 - setupTests.js
 - .gitignore
 - package-lock.json
 - package.json
 - README.md
- CODE** tab on the top right.
- MOVIE SERVICE.JS** tab in the tab bar.
- Content**:

```
1  export async function getAllMovies() {
2
3    try{
4      //const response = await fetch('/api/users');
5      //const response = await fetch('http://localhost:4000/api/movie/all');
6      //const response = await fetch('/api/movie/all');
7      return await response.json();
8    }catch(error) {
9      return [];
10   }
11
12 }
13 //http://localhost:3001/api/movie/insert
14 //---- ດັບໄຟລື ----
15 export async function createMovie(data) {
16   const response = await fetch('http://localhost:4000/api/movie/insert', {
17     method: 'POST',
18     headers: {'Content-Type': 'application/json'},
19     body: JSON.stringify({user: data})
20   })
21   return await response.json();
22 }
```

```
src > App.js > ...
1 import React, { useState, useEffect } from 'react';
2 import 'bootstrap/dist/css/bootstrap.min.css';
3 import './App.css';
4 import { Header } from './components/Header'
5 import { Users } from './components/Users'
6 import { DisplayBoard } from './components/DisplayBoard'
7 import CreateUser from './components/CreateUser'
8 import { getAllUsers, createUser } from './services/UserService'
9 /**
10 import { Movies } from './components/Movies'
11 import CreateMovie from './components/CreateMovie'
12 import { getAllMovies, createMovie } from './services/MovieService'
13
14 function App() {
15
16     const [user, setUser] = useState({})
17     const [users, setUsers] = useState([])
18     const [numberOfUsers, setNumberOfUsers] = useState(0)
19
20     /**
21     const [movie, setMovie] = useState({})
22     const [movies, setMovies] = useState([])
23     const [numberOfMovies, setNumberOfMovies] = useState(0)
24
25     const userCreate = (e) => {
26
27         createUser(user)
28             .then(response => {
29                 console.log(response);
30                 setNumberOfUsers(numberOfUsers+1)
31             });
32     }
33
34     const movieCreate = (e) => {
35
36         createMovie(movie)
37             .then(response => {
38                 console.log(response);
39                 setNumberOfMovies(numberOfMovies+1)
40             });
41     }
42
43
44     /*
45     const fetchAllUsers = () => {
46         getAllUsers()
47             .then(users => {
48                 console.log(users)
49                 setUsers(users);
50                 setNumberOfUsers(users.length)
51             });
52     }
53 */
54 */


```

The screenshot shows a code editor interface with the following details:

- EXPLORER** sidebar on the left displaying the project structure:

 - MY-APP folder containing node_modules, public, and src.
 - src folder containing components (AppError.js, CreateUser.js, CreateMovie.js, DisplayBoard.js, Header.js, Movies.js, Users.js), services (MovieService.js, UserService.js), and files (App.1.js, App.css, App.js, App.test.js, index.css, index.js, logo.svg, serviceWorker.js, setupTests.js).
 - Global files: .gitignore, package-lock.json, package.json, and README.md.

- OPEN EDITORS** tab bar at the top showing multiple tabs: App.js, CreateUser.js, CreateMovie.js, DisplayBoard.js, Movies.js, and UserService.js.
- App.js** code editor tab showing the following code:

```
53 }
54 */
55 const fetchAllMovies = () => {
56   getAllMovies()
57     .then(movies => {
58       console.log(movies)
59       setMovies(movies);
60       setNumberOfMovies(movies.length)
61     });
62 }
63
64 useEffect(() => {
65   /*
66    getAllUsers()
67    .then(users => {
68      | console.log(users)
69      | setUsers(users);
70      | setNumberOfUsers(users.length)
71    });
72   */
73   getAllMovies()
74     .then(movies => {
75       | console.log(movies)
76       | setMovies(movies);
77       | setNumberOfMovies(movies.length)
78     });
79 }, []);
80
81 /*
82 const onChangeForm = (e) => {
83   if (e.target.name === 'firstname') {
84     | | user.firstName = e.target.value;
85   } else if (e.target.name === 'lastname') {
86     | | user.lastName = e.target.value;
87   } else if (e.target.name === 'email') {
88     | | user.email = e.target.value;
89   }
90   setUser(user)
91 }
92 */
93
94 const onChangeForm = (e) => {
95   if (e.target.name === 'title') {
96     | | movie.title = e.target.value;
97   } else if (e.target.name === 'genre') {
98     | | movie.genre = e.target.value;
99   } else if (e.target.name === 'director') {
100    | | movie.director = e.target.value;
101 } else if (e.target.name === 'release_yr') {
102    | | movie.release_yr = e.target.value;
103 }
104 setMovie(movie)
105 }
106 }
107 }
```

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar, which lists the project structure. The main area displays the `App.js` file, which is currently open in the editor.

EXPLORER

- > OPEN EDITORS
- ✓ MY-APP
 - > node_modules
 - > public
 - ✓ src
 - components
 - AppError.js
 - CreateMovie.js
 - CreateUser.js
 - DisplayBoard.js
 - Header.js
 - Movies.js
 - Users.js
 - services
 - MovieService.js
 - UserService.js
 - App.1.js
 - App.css
 - App.js
 - App.test.js
 - index.css
 - index.js
 - logo.svg
 - serviceWorker.js
 - setupTests.js
 - .gitignore
 - package-lock.json
 - package.json
 - README.md

App.js

```
src > App.js > ...
105     |     setMovie(movie)
106 }
107
108
109     return (
110         <div className="App">
111             <Header></Header>
112             <div className="container mrgnbtm">
113                 <div className="row">
114                     /* <div className="col-md-8">
115                         <CreateUser
116                             user={user}
117                             onChangeForm={onChangeForm}
118                             createUser={userCreate}
119                         >
120                         </CreateUser>
121                     </div> */
122                     <div className="col-md-8">
123                         <CreateMovie
124                             user={user}
125                             onChangeForm={onChangeForm}
126                             createMovie={movieCreate}
127                         >
128                         </CreateMovie>
129                     </div>
130                     <div className="col-md-4">
131                         <DisplayBoard
132                             numberOfMovies={numberOfMovies}
133                             getAllMovies={fetchAllMovies}
134                         >
135                         </DisplayBoard>
136                     </div>
137                     /* <div className="row mrgnbtm">
138                         <Users users={users}></Users>
139                     </div> */
140                     <div className="row mrgnbtm">
141                         <Movies movies={movies}></Movies>
142                     </div>
143                 </div>
144             </div>
145         );
146     }
147
148     export default App;
149 }
```

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER** sidebar on the left showing the project structure:

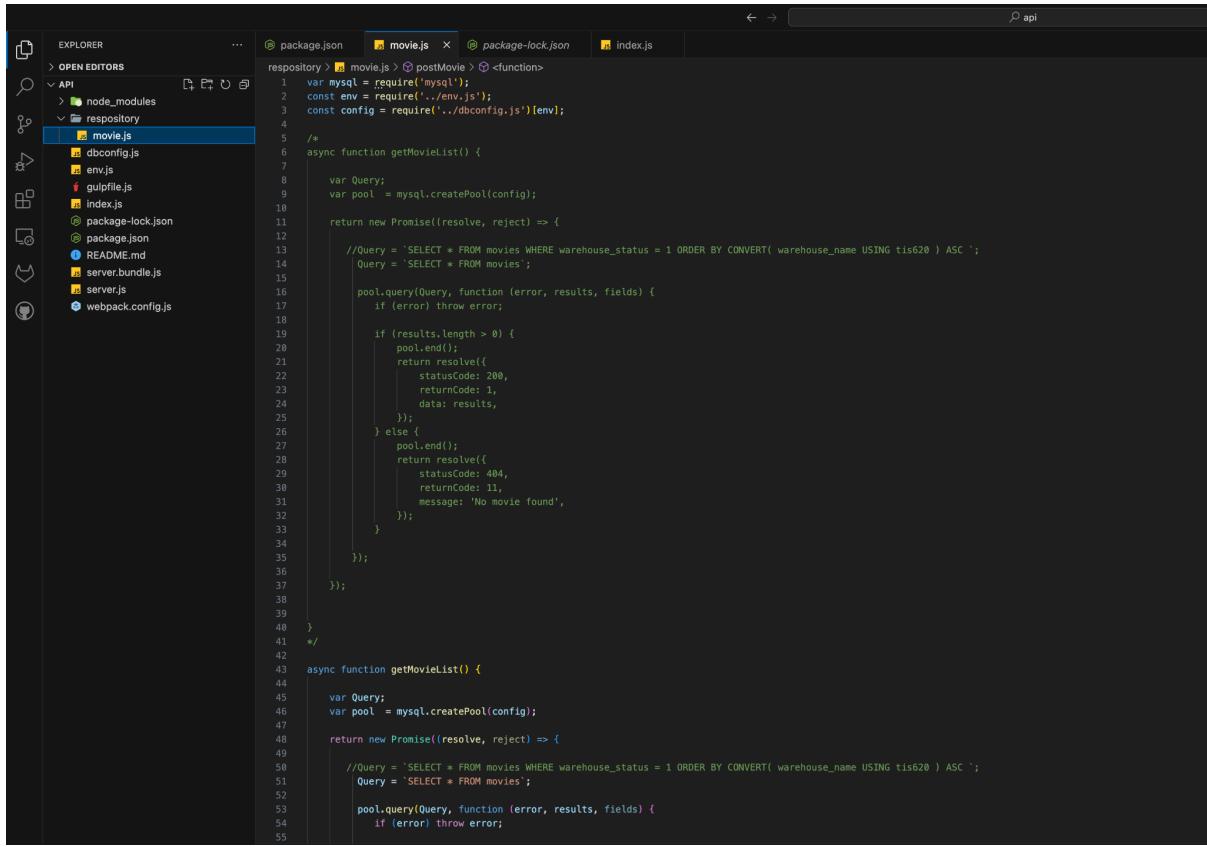
 - MY-APP folder containing:
 - node_modules
 - public
 - src folder containing:
 - components folder with AppError.js, CreateUser.js, DisplayBoard.js, Header.js, Movies.js, and Users.js
 - services folder with MovieService.js, UserService.js, App.1.js, App.css, App.js, App.test.js, and index.css
 - index.js (highlighted with a blue selection bar)
 - logo.svg
 - serviceWorker.js
 - setupTests.js
 - .gitignore
 - package-lock.json
 - package.json
 - README.md

- EDITOR** tab bar at the top with tabs for App.js, index.js (active), CreateUser.js, CreateMovie.js, and DisplayBoard.js.
- CODE** editor area showing the content of index.js:

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import App from './App';
5 import * as serviceWorker from './serviceWorker';
6 import { AppError } from './components/AppError';
7
8 ReactDOM.render(
9   <React.StrictMode>
10    <AppError>
11      <App />
12    </AppError>
13  </React.StrictMode>,
14  document.getElementById('root')
15 );
16
17 // If you want your app to work offline and load faster, you can change
18 // unregister() to register() below. Note this comes with some pitfalls.
19 // Learn more about service workers: https://bit.ly/CRA-PWA
20 serviceWorker.unregister();
21
```

```
1  {
2    "name": "my-app",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@testing-library/jest-dom": "^4.2.4",
7      "@testing-library/react": "^9.5.0",
8      "@testing-library/user-event": "^7.2.1",
9      "bootstrap": "^4.5.0",
10     "react": "^16.13.1",
11     "react-bootstrap": "^1.0.1",
12     "react-dom": "^16.13.1",
13     "react-scripts": "3.4.1"
14   },
15   "scripts": {
16     "start": "PORT=3001 react-scripts start",
17     "build": "react-scripts build",
18     "test": "react-scripts test",
19     "eject": "react-scripts eject"
20   },
21   "proxy": "http://localhost:4000",
22   "eslintConfig": {
23     "extends": "react-app"
24   },
25   "browserslist": {
26     "production": [
27       ">0.2%",
28       "not dead",
29       "not op_mini all"
30     ],
31     "development": [
32       "last 1 chrome version",
33       "last 1 firefox version",
34       "last 1 safari version"
35     ]
36   }
37 }
38 }
```

code api



The screenshot shows a code editor interface with the following details:

- EXPLORER:** On the left, there is a tree view of the project structure. The "movie.js" file is selected.
- EDITOR:** The main area displays the content of the "movie.js" file. The code is written in JavaScript and uses MySQL queries to interact with a database.

```
repository > movie.js > postMovie > <function>
1 var mysql = require('mysql');
2 const env = require('../env.js');
3 const config = require('../dbconfig.js').env;
4
5 /*
6  * async function getMovieList() {
7
8     var Query;
9     var pool = mysql.createPool(config);
10
11    return new Promise((resolve, reject) => {
12
13        //Query = 'SELECT * FROM movies WHERE warehouse_status = 1 ORDER BY CONVERT(warehouse_name USING tis620 ) ASC ';
14        Query = 'SELECT * FROM movies';
15
16        pool.query(Query, function (error, results, fields) {
17            if (error) throw error;
18
19            if (results.length > 0) {
20                pool.end();
21                return resolve({
22                    statusCode: 200,
23                    returnCode: 1,
24                    data: results,
25                });
26            } else {
27                pool.end();
28                return resolve({
29                    statusCode: 404,
30                    returnCode: 11,
31                    message: 'No movie found',
32                });
33            }
34        });
35    });
36
37 });
38
39
40 */
41 */
42
43 async function getMovieList() {
44
45     var Query;
46     var pool = mysql.createPool(config);
47
48    return new Promise((resolve, reject) => {
49
50        //Query = 'SELECT * FROM movies WHERE warehouse_status = 1 ORDER BY CONVERT( warehouse_name USING tis620 ) ASC ';
51        Query = 'SELECT * FROM movies';
52
53        pool.query(Query, function (error, results, fields) {
54            if (error) throw error;
55
56        });
57    });
58
59
60
61 */
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
179
180
181
182
183
184
185
186
187
188
189
189
190
191
192
193
194
195
196
197
198
199
199
200
201
202
203
204
205
206
207
208
209
209
210
211
212
213
214
215
216
217
217
218
219
219
220
221
222
223
224
225
226
226
227
228
228
229
229
230
231
231
232
232
233
233
234
234
235
235
236
236
237
237
238
238
239
239
240
240
241
241
242
242
243
243
244
244
245
245
246
246
247
247
248
248
249
249
250
250
251
251
252
252
253
253
254
254
255
255
256
256
257
257
258
258
259
259
260
260
261
261
262
262
263
263
264
264
265
265
266
266
267
267
268
268
269
269
270
270
271
271
272
272
273
273
274
274
275
275
276
276
277
277
278
278
279
279
280
280
281
281
282
282
283
283
284
284
285
285
286
286
287
287
288
288
289
289
290
290
291
291
292
292
293
293
294
294
295
295
296
296
297
297
298
298
299
299
300
300
301
301
302
302
303
303
304
304
305
305
306
306
307
307
308
308
309
309
310
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
```

```
repository > movie.js > postMovie > <function>
  56     if (results.length > 0) {
  57         pool.end();
  58         return resolve(results);
  59     } else {
  60         pool.end();
  61         return resolve({
  62             statusCode: 404,
  63             returnCode: 11,
  64             message: 'No movie found',
  65         });
  66     });
  67 });
  68 );
  69 );
  70 );
  71 );
  72 }
  73 }

  74
  75

  76 async function getMovieSearch(search_text) {
  77
  78     var Query;
  79     var pool = mysql.createPool(config);
  80
  81     return new Promise((resolve, reject) => {
  82
  83         Query = `SELECT * FROM movies WHERE title LIKE '%${search_text}%'`;
  84
  85         pool.query(Query, function (error, results, fields) {
  86             if (error) throw error;
  87
  88             if (results.length > 0) {
  89                 pool.end();
  90                 return resolve({
  91                     statusCode: 200,
  92                     returnCode: 1,
  93                     data: results,
  94                 });
  95             } else {
  96                 pool.end();
  97                 return resolve({
  98                     statusCode: 404,
  99                     returnCode: 11,
 100                     message: 'No movie found',
 101                 });
 102             }
 103         });
 104     });
 105
 106 );
 107
 108
 109 }
```

The screenshot shows the VS Code interface with the movie.js file open in the central editor area. The code is a Node.js script for a movie database. It includes functions for searching movies by title and inserting new movies into the 'movies' table. The code uses MySQL and promises.

```
repository > movie.js > postMovie > function>
109 }
110
111     async function postMovie(p_title,p_genre,p_director,p_release_year) {
112
113         var Query;
114         var pool = mysql.createPool(config);
115
116         return new Promise((resolve, reject) => {
117
118             //Query = 'SELECT * FROM movies WHERE title LIKE "%${search_text}%"';
119
120             var post = {
121                 title: p_title,
122                 genre: p_genre,
123                 director: p_director,
124                 release_year: p_release_year
125             };
126
127             console.log('post is: ', post);
128
129             Query = 'INSERT INTO movies SET ?';
130             pool.query(Query, post, function (error, results, fields) {
131                 //pool.query(Query, function (error, results, fields) {
132
133
134                 //if (error) throw error;
135                 if(error){
136                     console.log('error_code_msg: ', error.code+':'+error.sqlMessage);
137                     pool.end();
138                     return resolve({
139                         statusCode: 405,
140                         returnCode: 9,
141                         message: error.code+': '+error.sqlMessage
142                     });
143                 }
144                 else{
145                     console.log('results: ', results);
146                     if (results.affectedRows > 0) {
147                         pool.end();
148                         return resolve({
149                             statusCode: 200,
150                             returnCode: 1,
151                             message: 'Movie list was inserted',
152                         });
153                     }
154                 }
155             });
156
157
158
159
160         });
161
162
163     });
};
```

The screenshot shows the VS Code interface with the dbconfig.js file open in the central editor area. The code defines a dbconfig object with development and production environments. Both environments share the same host ('localhost'), port ('3306'), user ('root'), and password (''). The database is set to 'moviedb'.

```
package.json movie.js dbconfig.js index.js
dbconfig.js > dbconfig > production
1 var dbconfig = {
2     development: {
3         //connectionLimit : 10,
4         host      : 'localhost',
5         port      : '3306',
6         user      : 'root',
7         password  : '',
8         database  : 'moviedb'
9     },
10    production: {
11        //connectionLimit : 10,
12        host      : 'localhost',
13        port      : '3306',
14        user      : 'root',
15        password  : '',
16        database  : 'moviedb'
17    }
18 };
19 module.exports = dbconfig;
```

The screenshot shows the Visual Studio Code interface. The left sidebar is the Explorer view, showing a project structure under 'API'. The 'env.js' file is selected and highlighted with a blue background. The top bar shows tabs for 'package.json', 'movie.js', 'env.js', and 'index.js'. The main editor area displays the content of 'env.js':

```
1 var env = process.env.NODE_ENV || 'development';
2 //var env = process.env.NODE_ENV || 'production';
3 module.exports = env;
4
```

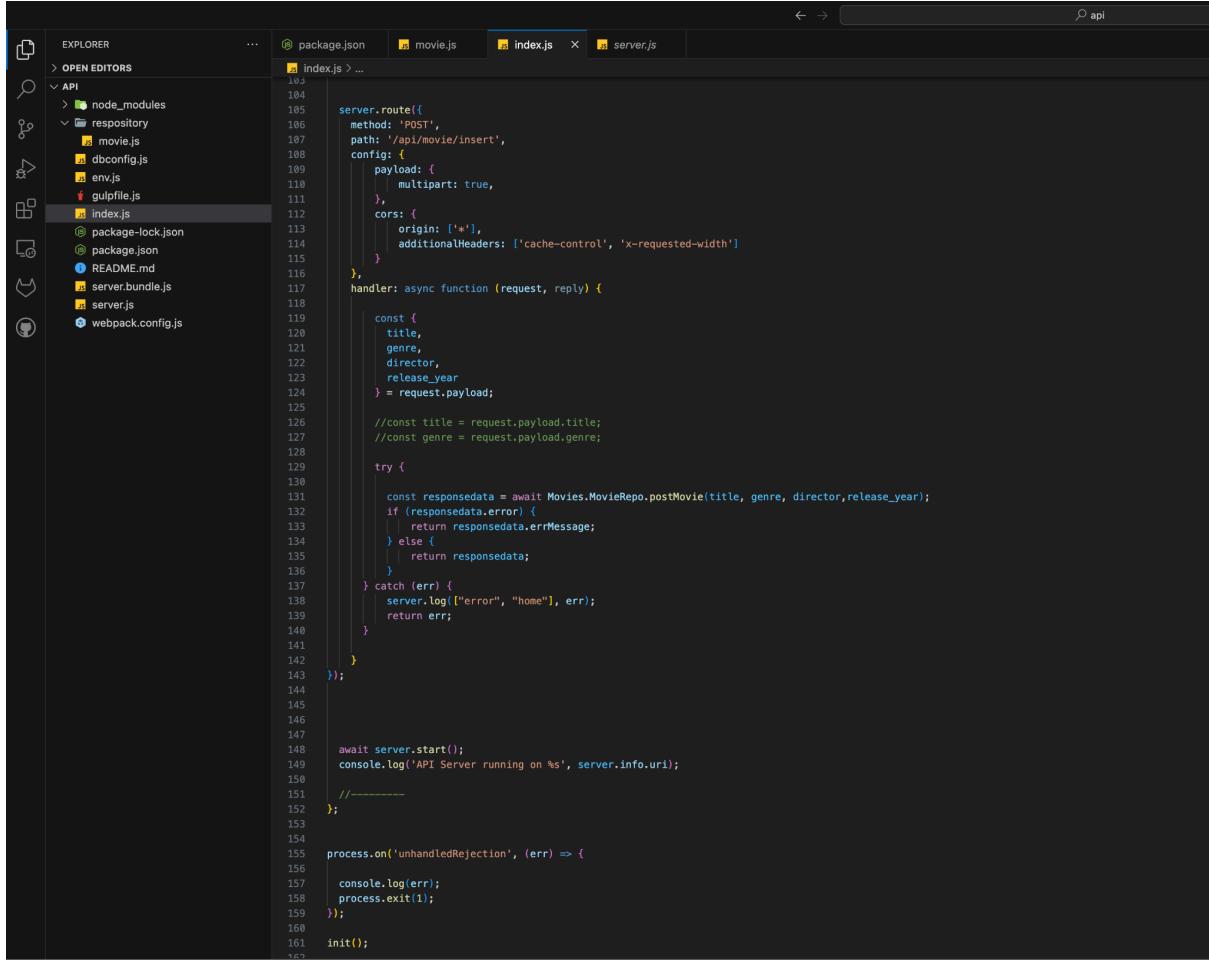
The screenshot shows the Visual Studio Code interface. The left sidebar is the Explorer view, showing a project structure under 'API'. The 'index.js' file is selected and highlighted with a blue background. The top bar shows tabs for 'package.json', 'movie.js', 'index.js', and 'server.js'. The main editor area displays the content of 'index.js':

```
1 const hapi = require('@hapi/hapi');
2 const env = require('./env.js');
3 const Movies = require('../repository/movie');
4
5 const express = require('express');
6 const app = express();
7
8 const path = require('path');
9   |   bodyParser = require("body-parser");
10
11 //-----
12 const api_port = 4000;
13 const web_port = 4001;
14
15 //-----
16 //----- hapi -----
17
18 console.log(`Running Environment: ${env}`);
19
20
21 const init = async () => {
22
23   const server = hapi.Server({
24     port: api_port,
25     host: '0.0.0.0',
26     routes: {
27       cors: true
28     }
29   });
30
31 //-----
32
33 await server.register(require('@hapi/inert'));
34
35 server.route({
36   method: "GET",
37   path: "/",
38   handler: () => {
39     return `Welcome to API Back-end Ver. 1.0.0`;
40   }
41 });
42
43
44 //API: http://localhost:3001/api/movie/all
45 server.route({
46   method: 'GET',
47   path: '/api/movie/all',
48   config: {
49     cors: {
50       origin: ['*'],
51       additionalHeaders: ['cache-control', 'x-requested-width']
52     },
53   },
54 });
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1190
1191
1192
1193
1194
1195
1195
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1290
1291
1292
1293
1294
1295
1296
1296
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1390
1391
1392
1393
1394
1395
1396
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1490
1491
1492
1493
1494
1495
1496
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1590
1591
1592
1593
1594
1595
1596
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1690
1691
1692
1693
1694
1695
1696
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1790
1791
1792
1793
1794
1795
1796
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1890
1891
1892
1893
1894
1895
1896
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
21
```

The screenshot shows a code editor interface with the following details:

- EXPLORER** sidebar on the left containing project files:
 - API folder with node_modules
 - repository folder with movie.js, dbconfig.js, env.js, gulpfile.js, index.js (highlighted), package-lock.json, package.json, README.md, server.bundle.js, server.js, and webpack.config.js
- Top navigation bar showing package.json, movie.js, index.js (active tab), and server.js.
- Code editor area displaying the content of index.js:

```
54     handler: async function (request, reply) {
55         //var param = request.query;
56         //const category_code = param.category_code;
57
58         try {
59
60             const respondedata = await Movies.MovieRepo.getMovieList();
61             if (respondedata.error) {
62                 return respondedata.errorMessage;
63             } else {
64                 return respondedata;
65             }
66         } catch (err) {
67             server.log(["error", "home"], err);
68             return err;
69         }
70     });
71
72 });
73
74 server.route({
75     method: 'GET',
76     path: '/api/movie/search',
77     config: {
78         cors: {
79             origin: ['*'],
80             additionalHeaders: ['cache-control', 'x-requested-width']
81         }
82     },
83     handler: async function (request, reply) {
84         var param = request.query;
85         const search_text = param.search_text;
86         //const title = param.title;
87
88         try {
89
90             const respondedata = await Movies.MovieRepo.getMovieSearch(search_text);
91             if (respondedata.error) {
92                 return respondedata.errorMessage;
93             } else {
94                 return respondedata;
95             }
96         } catch (err) {
97             server.log(["error", "home"], err);
98             return err;
99         }
100    });
101 });
102 });
103 }
```



```
index.js > ...
server.route({
  method: 'POST',
  path: '/api/movie/insert',
  config: {
    payload: {
      multipart: true,
    },
    cors: {
      origin: ['*'],
      additionalHeaders: ['cache-control', 'x-requested-width']
    }
  },
  handler: async function (request, reply) {
    const {
      title,
      genre,
      director,
      release_year
    } = request.payload;
    //const title = request.payload.title;
    //const genre = request.payload.genre;
    try {
      const responddata = await Movies.MovieRepo.postMovie(title, genre, director, release_year);
      if (responddata.error) {
        return responddata.errorMessage;
      } else {
        return responddata;
      }
    } catch (err) {
      server.log(['error', 'home'], err);
      return err;
    }
  }
};

await server.start();
console.log('API Server running on %s', server.info.uri);
//-----
};

process.on('unhandledRejection', (err) => {
  console.log(err);
  process.exit(1);
});
init();
```

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER** sidebar on the left showing project files: API, node_modules, repository, package.json, README.md, server.bundle.js, server.js, and webpack.config.js.
- EDITOR** tab bar at the top with package.json, movie.js, index.js, and server.js.
- package.json** content:

```
1  {
2    "name": "react-nodejs-example",
3    "version": "1.0.0",
4    "description": "example project react with nodejs",
5    "main": "server.js",
6    "scripts": {
7      "start": "node server.bundle.js",
8      "build": "webpack",
9      "dev": "nodemon ./index.js localhost 4000"
10    },
11    "repository": {
12      "type": "git",
13      "url": "git+https://github.com/bbachi/react-nodejs-example.git"
14    },
15    "author": "Bhargav Bachina",
16    "license": "ISC",
17    "bugs": {
18      "url": "https://github.com/bbachi/react-nodejs-example/issues"
19    },
20    "homepage": "https://github.com/bbachi/react-nodejs-example#readme",
21    "dependencies": {
22      "@hapi/hapi": "^21.3.2",
23      "@hapi/inert": "^7.1.0",
24      "express": "^4.17.1",
25      "http-proxy-middleware": "^1.0.4",
26      "mysql": "^2.18.1"
27    },
28    "devDependencies": {
29      "del": "^5.1.0",
30      "fancy-log": "^1.3.3",
31      "gulp": "^4.0.2",
32      "gulp-zip": "^5.0.1",
33      "nodemon": "^2.0.4",
34      "webpack": "^4.43.0",
35      "webpack-cli": "^3.3.11",
36      "webpack-stream": "5.2.1"
37    }
38  }
39
```

The screenshot shows a code editor interface with the following details:

- EXPLORER** sidebar on the left containing the project structure:
 - API folder
 - node_modules
 - repository
 - movie.js
 - dbconfig.js
 - env.js
 - gulpfile.js
 - index.js
 - package-lock.json
 - package.json
 - README.md
 - server.bundle.js
 - server.js** (highlighted with a blue selection bar)
 - webpack.config.js- OPEN EDITORS** tab bar at the top right with four tabs: package.json, movie.js, index.js, and **server.js**.
- server.js** code editor tab showing the following JavaScript code:

```
const express = require('express');
const path = require('path');
const app = express(),
      bodyParser = require("body-parser");
const port = 4000;

// place holder for the data
const users = [
  {
    firstName: "first1",
    lastName: "last1",
    email: "abc@gmail.com"
  },
  {
    firstName: "first2",
    lastName: "last2",
    email: "abc@gmail.com"
  },
  {
    firstName: "first3",
    lastName: "last3",
    email: "abc@gmail.com"
  }
];

app.use(bodyParser.json());
app.use(express.static(path.join(__dirname, '../my-app/build')));

app.get('/api/users', (req, res) => {
  console.log('api/users called!')
  res.json(users);
});

app.post('/api/user', (req, res) => {
  const user = req.body.user;
  console.log('Adding user:::::', user);
  users.push(user);
  res.json("user added");
});

app.get('/', (req,res) => {
  res.sendFile(path.join(__dirname, '../my-app/build/index.html'));
});

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