

## **Module 7 Activity 2: Implementing HTTP Status Codes and Error Handling in**

**Node.jsNode.js on Windows**

**Brandon Trinkle**

**IFT 458: Middleware Programming**

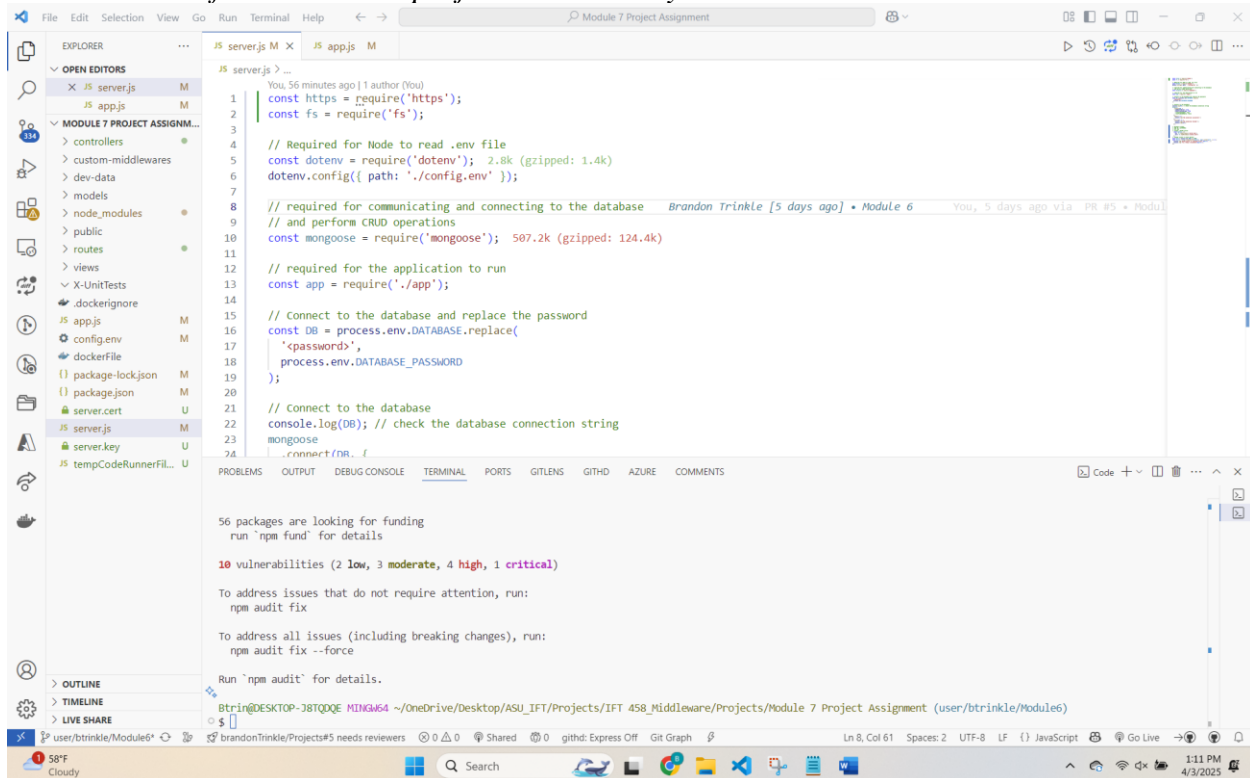
**Professor Dinesh Sthapit**

**April 3, 2025**

## Module 7 Activity 2: Implementing HTTP Status Codes and Error Handling in Node.js

### Node.js on Windows

*Screenshot 1: Of the initialized project structure in your IDE.*



*Screenshot 2: Showing the app.js file with implemented routes for different HTTP status codes.*

The screenshot shows a Visual Studio Code editor with a file explorer on the left and a code editor in the center. The file explorer shows a project structure with folders like controllers, custom-middleware, dev-data, models, node\_modules, public, routes, views, X-UnitTests, and files like .dockerignore, app.js, config.env, dockerfile, package-lock.json, package.json, server.cert, server.js, server.key, and tempCodeRunnerFil... The code editor shows the app.js file with the following code:

```

// View routes
const viewRouter = require('./routes/viewRoutes');
app.use(`${process.env.API_VERSION}/views`, middlewareLogger('View Router'), viewRouter);

// User routes
const userRouter = require('./routes/userRoutes');
app.use('/users', middlewareLogger('User Router'), userRouter);

// Brandon Trinkle
// ASU ID: 1217455031
// IFT 458
// 201 + 400 Handler
// Middleware Routes
app.get('/201', (req, res) => {
  // Not committed yet
  res.status(201).send('Resource created successfully (201)');
});

app.get('/400', (req, res) => {
  res.status(400).send('Bad request (400)');
});

```

The terminal window at the bottom shows the output of an npm audit command:

```

56 packages are looking for funding
  run `npm fund` for details

10 vulnerabilities (2 low, 3 moderate, 4 high, 1 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

```

*Screenshot 3: Displaying the implemented middleware in app.js for request logging and error handling.*

```

// Brandon Trinkle
// ASU ID: 1217459031
// IFT 458 - Not committed yet
// Home Route (200 OK)
app.get('/', middlewareLogger('Home Route'), (req, res) => {
  res.render('home', {
    title: 'Dashboard',
    user: req.user,
    books: [],
    api_version: process.env.API_VERSION
  });
});

// HTTP Status Code
app.get('/200', (req, res) => {
  res.status(200).send('OK: The request has succeeded (200)');
});

app.get('/201', (req, res) => {
  res.status(201).send('Resource created successfully (201)');
});

app.get('/400', (req, res) => {
  res.status(400).send('Bad request (400)');
});

// Async Route (returns 200 on success)
// Complexity is 5 Everything is cool!
app.get('/async', async (req, res, next) => {
  try {
    await new Promise(resolve, reject) => {
      setTimeout(() => resolve('Done'), 1000);
    };
    res.status(200).send('Async operation completed');
  } catch (err) {
    next(err); // Triggers the 500 error handler
  }
});

```

*Note: 200, 201 and 400 routes*

```

// 404 Not Found Handler - Not committed yet
app.use((req, res) => {
  const viewPath = path.join(__dirname, 'views', '404.ejs');
  if (fs.existsSync(viewPath)) {
    res.status(404).render('404', { url: req.originalUrl });
  } else {
    console.error('[ErrorHandling] Missing 404.ejs view file. ');
    res.status(404).send('404 - Page not found. ');
  }
  console.log('[ErrorHandling] 404 - Route not found: ${req.originalUrl}');
});

// Global Error Handler (500 Internal Server Error)
// Complexity is 4 Everything is cool!
app.use((err, req, res, next) => {
  const isDev = process.env.NODE_ENV === 'development';
  const viewPath = path.join(__dirname, 'views', '500.ejs');

  if (isDev) {
    console.error('[ErrorHandling] 500 - Internal Server Error: ', err.stack);
  } else {
    console.error('[ErrorHandling] 500 - Internal Server Error: ', err.message);
  }

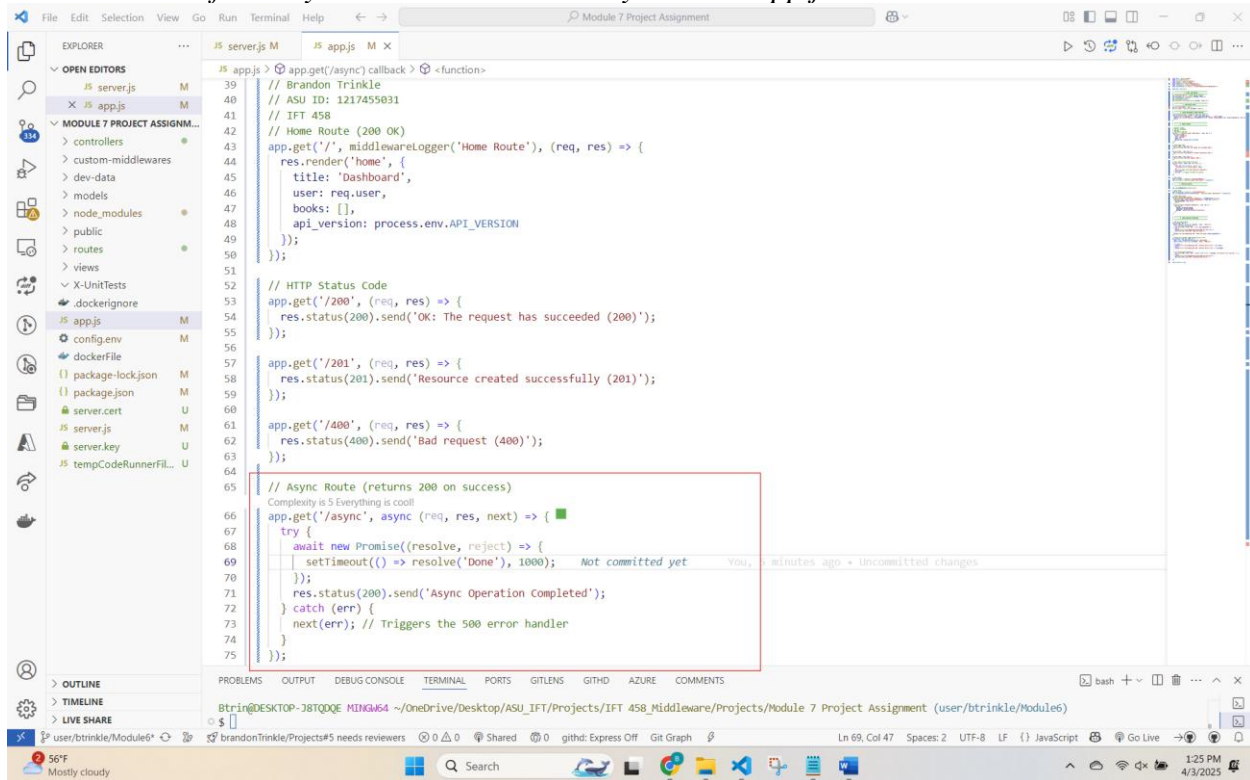
  if (fs.existsSync(viewPath)) {
    res.status(500).render('500', { error: isDev ? err : { message: 'An internal error occurred.' } });
  } else {
    console.error('[ErrorHandling] Missing 500.ejs view file. ');
    res.status(500).send('500 - Internal Server Error. ');
  }
});

module.exports = app;

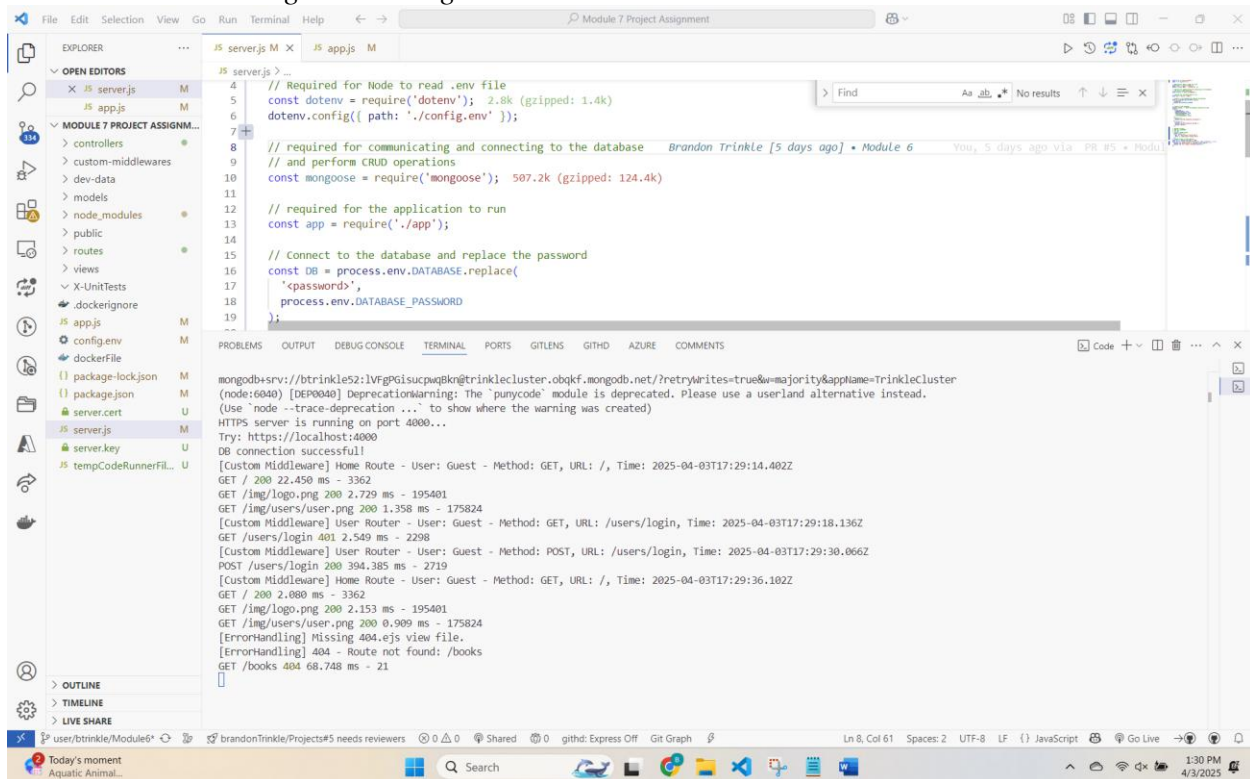
```

*Note: 404 & 500*

Screenshot 4: Of the asynchronous route with try/catch in app.js.



Screenshot 5: Showing the running server in the terminal.



Note: Error is with .env file specifying api/v1

*Screenshot 6: Displaying responses from the server for different routes in Postman or a web browser.*

The screenshot shows a VS Code editor with a project named 'Module 7 Project Assignment'. The Explorer sidebar on the left shows the file structure, including 'server.js', 'app.js', 'controllers', 'custom-middleware', 'dev-data', 'models', 'node\_modules', 'public', 'routes', 'bookRoutes.js', 'userRoutes.js', 'viewRoutes.js', 'views', 'X-UnitTests', '.dockerignore', 'app.js', 'config.env', 'dockerfile', 'package-lock.json', 'package.json', 'server.cert', 'server.js', 'server.key', and 'tempCodeRunnerFile...'. The main editor window displays the code for 'server.js', which includes comments and code for connecting to a database and handling routes. The terminal window at the bottom shows the output of running the application, including the command 'node .', the server starting on port 4000, and various HTTP requests and responses with their respective status codes and times.

```

// Required for Node to read .env file
const dotenv = require('dotenv'); 2.8k (gzipped: 1.4k)
dotenv.config({ path: './config.env' });

// required for communicating and connecting to the database
// and perform CRUD operations
const mongoose = require('mongoose'); 507.2k (gzipped: 124.4k)

// required for the application to run
const app = require('./app');

// Connect to the database and replace the password
const DB = process.env.DATABASE.replace(
  '<password>',
  process.env.DATABASE_PASSWORD
);

```

```

$ node "c:\Users\btrinkle\OneDrive\Desktop\ASU_IPT\Projects\IFT_458_Middleware\Projects\Module 7 Project Assignment\server.js"
mongodb+srv://btrinkle52:lvfgp6isucpwbkbn@trinklecluster.obqkf.mongodb.net/?retryWrites=true&w=majority&appName=TrinkleCluster
(node:8568) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(Use 'node --trace-deprecation ...' to show where the warning was created)
HTTPS server is running on port 4000...
Try: https://localhost:4000
DB connection successful!

[Custom Middleware] Home Route - User: Guest - Method: GET, URL: /, Time: 2025-04-03T17:33:47.534Z
GET / 200 17.379 ms - 3362
GET /img/logo.png 200 3.185 ms - 195401
GET /img/users/user.png 200 1.800 ms - 175824
[Custom Middleware] User Router - User: Guest - Method: GET, URL: /users/login, Time: 2025-04-03T17:33:53.367Z
GET /users/login 401 3.014 ms - 2298
[Custom Middleware] User Router - User: Guest - Method: POST, URL: /users/login, Time: 2025-04-03T17:34:03.379Z
POST /users/login 200 401.415 ms - 2719
[Custom Middleware] Home Route - User: Guest - Method: GET, URL: /, Time: 2025-04-03T17:34:04.982Z
GET / 200 2.022 ms - 3362
GET /img/logo.png 200 2.286 ms - 195401
GET /img/users/user.png 200 1.036 ms - 175824
[Custom Middleware] Book Router - User: btrinkle1@asu.edu - Method: GET, URL: /books, Time: 2025-04-03T17:34:06.633Z
GET /books 200 160.778 ms - 2386

```

*Note: Fixed by removing the API version in .env file.*