Hola Javier, buenas tardes. Revisé tu proyecto. Está realmente bien. Implementas todas las herramientas que vimos a lo largo de la cursada con criterio. Tanto las dependencias (Passport, multer, mongoose), como patrones (MVC, etc.). El código está prolijo, y estructurado de acuerdo a lo que vimos en clase. No “reventó” el código, al menos durante las pruebas que realicé, por lo que deduzco tenes bien implementado tu errorHandler. El logger, también funciona de manera correcta. Si me tope con un caso, en la ruta de registro de usuarios, donde el error no era claro, y creo que no era del todo adecuado el manejo del mismo. Más adelante lo detallaré.

Routers:

* Sessions: funciona bien. Hay un error, que me surgió debido a que decidiste utilizar cualquier nombre para tus propiedades. Si hubieras respetado los del enunciado, se me hubiese pasado por alto francamente, jajaja. Si no completas las propiedades nombre y lasName, se produce un error medio raro. El resto funciona.
* Products: funciona bien. Todo según lo esperado
* Carts: ídem products. Funciona.

Algo no menor, los tres routers también se integran a la perfección, permitiendo concretar operaciones de compra sin problemas.

Test, funciona bien. Raro, porque muchos compañeros tuvieron algún que otro inconveniente en esta parte. No sé si le habrán dado poca importancia a ese punto, pero en general les fallaba.

Algunos detalles para corregir. Lo que señalaba antes en relación al router de sessions. Cuando haces uso del UserDtoFirstLetterUpperCase, si no envias nombre o lastName entre las propiedades, fijate que el código falla. No llega a reventar, pero tanto logger como errorHandler fallan.

// Turns fist letter of name and last name to Capital letter

export class UserDTOfirstLettertoUpperCase {

  constructor(user) {

    console.log(user)

    if (user.name && user.lastName) {

Creo que viene por acá. El error que aparece es “error is not defined”. Al no ingresar en el if, devuelve un objeto vacío, que luego hace fallar el código. Para ver el detalle, agrega un console.log(error) al principio de tu errorHandler.

Parte documentación, si bien mostrar un manejo adecuado de Swagger, te faltan algunos detalles, como ser alguna mención a la parte de autenticación. Es habitual. En general la parte de documentación suele tomarse un poco a la ligera. Es tediosa la configuración de cada ruta para que represente fielmente la realidad. Imagino que viene por ese lado.

No mucho más, Javier. Lo dicho al comienzo: has realizado un buen laburo. Según el checklist, tu nota es de 8. Felicitaciones…!!!

Insisto con lo que dije la última clase: fue un placer haber compartido con ustedes esta cursada. Trabajaron muy bien. Siendo un curso extenso, y con un proyecto complejo y de envergadura (no sé si notaste la cantidad de líneas de código que generaste), le da un plus al desempeño de todos ustedes, que han llegado hasta esta instancia.

Me permito hacerte algunas sugerencias: procura seguir capacitándote siempre que puedas. TypeScript, algún framework (Nest podría ser… previo aprender typescript), git, también muy útil. Inglés. Prompts de IA. No aflojes. Hay mucho por aprender…!!! Y recuerda siempre practicar. Sentarse a codificar es la única forma, te podría decir, de afianzar los conocimientos.

Te mando un saludo enorme, y reitero mis sinceras felicitaciones.

Creo que el problema esta en que Passport.config no devuelve error. La solución que encontré es:

Reemplazar static error en el sessionsController por

static error = (req, res) => {

    res.setHeader("Content-Type", "application/json");

    return res.status(500).json({

      error: `Wrong credentials`,

    });

  };