Coder House Trabajo Final BackEnd

Comisión 30950

**Link y Repositorios**

Repositorio GitHub para Heroku

https://github.com/joakozuin/BackEndProyTrabFinalJoaquinZuinH.git

Link raiz Heroku

[https://joakowebbackendtrabfh.herokuapp.com/](https://joakowebbackendtrabfh.herokuapp.com/" \t "_blank)

Repositorio GitHub

<https://github.com/joakozuin/BackEndProyTrabFinalJoaquinZuin.git>

Se incluyen dos repositorios de GitHub uno, con el Front con los link de Heroku y el otro para ejecutar el servidor y Front de forma local.

Breve reseña de uso del FrontEnd que se ha incluido

En la raíz del servidor:

[ <https://joakowebbackendtrabfh.herokuapp.com/> ]

o en

[ http://localhost:8080/ ]

Se activa un FrontEnd básico, para hacer algunos test de las rutas del BackEnd, para cumplir con las consignas que lo solicitan.

**login:** se adjunta tabla con usuarios de pruebas, según el tipo de usuario, ingresa a distintas prestaciones, si es administrador puede cargar nuevos productos, editarlos y borrarlos. En esta opción tiene incluida la opción del chat con Websoket para cumplir con la consigna.

Si es cliente, muestra su carrito si esta creado en la BD con los productos, la opción de comprar otro producto o generar una orden de compra, no están implementadas desde el fontend, si tiene implementada la opción de chat con Websoket para cumplir con la consigna. Para que funcione el chat los usuarios deben ingresar desde equipos diferentes.

Todo el tema de acceso esta echo usando Passport JWT.

**logout:** Permite deslogear al administrador o al cliente.

**registro de un nuevo usuario:** En el registro de un nuevo usuario el front envia todos los datos al back para guardarlo en la BD, pero el administrador debe cambiar en la BD el tipo de usuario, no tiene desde el front implementado esta opción.

Usuarios para hacer Pruebas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | Nombre | passw | tipo | email | carrito |
| 63324f180b3bde7ba6256a64 | Joaquin | 12345 | Administrador | joako@gmail.com | no |
| 6336cfd05945d65fbde5c847 | Gaby | 12345 | Cliente | gaby@gmail.com | no |
|  |  |  |  |  |  |
| 6332fcdb9dec0c62a05db3e3 | Juan | 12345 | Cliente | juan@gmail.com | si |
| 6332f87bc37839e82caadc09 | Alfredo | 12345 | Cliente | alfre@gmail.com | si |

Rutas del BackEnd

this.apiCaminos = {

      carritos:'/api/carritos',

      productos:'/api/productos',

      ordenes:'/api/ordenes',

      chats:'/api/chats',

      login:'/api/login',

      info:"/api/infoServidor"

    };

- rutas carritos

```

router.get('/',                        getCarritos)

router.get('/:id',                     getCarrito)

router.get('/:id/productos',           getProdsCarrito)

router.get('/usuario/:idU',            getUsCarrito)     //Envia el carrito del usuario

router.post('/:idU',                   postCarrito)      //crea carrito con id usuario

router.post('/:id/productos/:idP',     postProdCarrito)  //Agrega un producto al carrito

router.delete('/:id/productos/:idP',   deleteProdCarrito)//Borra un producto del carrito

router.delete('/:id',                   deleteCarrito)

- rutas productos

router.get('/',              getProductos)

router.get('/:id',           getProducto)

router.get('/categoria/:cat',getCatProducto)

router.post('/',             postProducto)

router.post('/form',         postFormProducto)

router.put('/:id',           putProducto)

router.delete('/:id',        deleteProducto)

- rutas ordenes

router.get('/',              getOrdenes)

router.get('/:id',           getOrden)

router.get('/:id/productos', getProdsOrden)

router.post('/:idC',          postOrden) //idC carrito, crea la orden el carrito tiene datos del cliente

router.delete('/:id',         deleteOrden)

- rutas chats

router.get('/',              getChats)

router.get('/email/:email',  getChatsEmail)

router.get('/:id',           getChat)

router.post('/',             postChat)

router.delete('/:id',        deleteChat)

- rutas login

router.post("/registro",     loginPostRegistro);

router.get("/registrar",     loginGetRegistrar);

router.get("/errorRegistro", loginGetErrorRegistro);

router.post("/",             postLogin);

router.get("/errorLogin",    loginGetErrorLogin);

router.get("/datos",         loginGetDatos);

router.get("/logout",        loginGetLogout);

router.get("/",              getLogin);

- rutas info

router.get('/', getConfigServer)