EXAMEN MAYO 2023

Enunciado - Ordenación de productos por precio

Realice las modificaciones que considere necesarias, tanto en backend como en frontend, para satisfacer los nuevos requisitos que a continuación se describen.

Se desea ofrecer a los propietarios que los productos de sus restaurantes aparezcan ordenados según el campo `order` de la entidad Producto o según el campo `price` del producto, y que puedan determinar cual será el orden predeterminado en cada restaurante, de manera que cuando se listen los productos aparezcan siempre según el orden que haya decidido.

Recuerde que actualmente los productos se muestran en la pantalla de detalle del restaurante y el backend los devuelve siempre ordenados según el campo `order`. Por defecto, cada restaurante ordenará sus productos según el mencionado campo `order`.

Implemente los cambios necesarios en Backend y Frontend para incluir dicha funcionalidad. Se espera que el Frontend siga un diseño como el que se muestra en las siguientes capturas.

Nótese que **no** se pide modificación de las pantallas de creación o edición de restaurante.

RESOLUCION:

BACKEND

Para la resolución de este ejercicio como hay 2 posibilidades de orden que son , order y prices , creamos una nueva propiedad en la tabla Restaurants de tipo Boolean , que si es 0 se organice según el order y si es 1 se organize según el Price

Models RESTAURANT.JS

```
Restaurant.init({
  name: DataTypes.STRING,
  description: DataTypes.TEXT,
  OrderDefault: {
    type: DataTypes.BOOLEAN,
    defaultValue: false
  },
  address: DataTypes.STRING,
  postalCode: DataTypes.STRING,
  url: DataTypes.STRING,
  shippingCosts: DataTypes.DOUBLE,
  averageServiceMinutes: DataTypes.DOUBLE,
  email: DataTypes.STRING,
  phone: DataTypes.STRING,
  logo: DataTypes.STRING,
  heroImage: DataTypes.STRING,
  status: {
    type: DataTypes.ENUM,
```

Llamamos a la propiedad OrderDefault

De tipo Boolean y valor por defecto false

Migrations CREATE-RESTAURANT

```
      type: Sequelize.TEXT
    },
    OrderDefault: {
      type: Sequelize.BOOLEAN,
      defaultValue: false
    },
    address: {
      allowNull: false,
      type: Sequelize.STRING
```

Creamos la propiedad en los migrations que es la que mete la propiedad en la tabla , importante poner el MISMO NOMBRE,OrderDefautl , de tipo boolean y defaultValue false

Controller RESTAURANTCONTROLLER

Añadimos la propiedad OrderDefault añadida en el index de Restaurant

```
exports.index = async function (req, res) {
  try {
    const restaurants = await Restaurant.findAll(
      {
        attributes: ['id', 'name', 'description', 'OrderDefault', 'address', 'postalCode', 'url', 'shippingCosts', 'averageServiceMinutes', 'email', 'phone', 'logo', 'her
        include:
      {
        model: RestaurantCategory,
        as: 'restaurantCategory'
      },
        order: [[{ model: RestaurantCategory, as: 'restaurantCategory' }, 'name', 'ASC']]
      }
    )
    res.json(restaurants)
  } catch (err) {
    res.status(500).send(err)
  }
}
```

Añadimos la propiedad OrderDefault añadida en el indexOwner de Restaurant

```
exports.indexOwner = async function (req, res) {
  try {
    const restaurants = await Restaurant.findAll(
      {
        attributes: ['id', 'name', 'description', 'OrderDefault', 'address', 'postalCode', 'url', 'shippingCosts', 'averageServiceMinutes', 'email', 'phone', 'logo', 'her
        where: { userId: req.user.id }
      })
    res.json(restaurants)
  } catch (err) {
    res.status(500).send(err)
  }
}
```

Se modifica la función show , que muestra información de un restaurante especifico ,

Añadimos la const orderCriteriaRestaurant con el attributes: [‘OrderDefault] es decir , que a la hora de buscar un restaurante especifico para req.params.restaurantId , solo nos interesa el valor de la propiedad OrderDefault, antes definida

Con exclude userId , ignoramos toda información sobre el propietario del restaurante

Luego añadimos lo que esta subrayado en amarillo ,que básicamente

Definimos el order de los productos

Si según el orderCriterianRestaurant. OrderDefault, [[{model: Product, as ‘products’} (se ordenan los productos según)

Si orderDefault es true (1), se ordena según el precio

Si orderDefault es false (0) , se ordena según la propiedad order

Añadimos una nueva función exports llamada ChageSort, que se utiliza para cambiar el order de ordenamiento de los restaurantes , entre su valor actual y su opuesto

Creamos la constante restaurante , que busca un restaurante según su restaurantId

Y mediante la función update , actualizamos el valor de la propiedad OrderDefault -> !OrderDefault , donde el id corresponda con el restaurantId

Se crea la constante updatedRestaurant , que es el restaurante con la propiedad modificada y luego mediante el res.json(updatedRestaurant) se sube a la database

```
exports.changeSort = async function (req, res) {
  try {
    const restaurant = await Restaurant.findByPk(req.params.restaurantId)
    await Restaurant.update({ OrderDefault: !restaurant.OrderDefault }, { where: { id: req.params.restaurantId } })
    const updatedRestaurant = await Restaurant.findByPk(req.params.restaurantId)
    res.json(updatedRestaurant)
  } catch (err) {
    res.status(500).send(err)
  }
}
```

Routes RESTAURANT-ROUTES

Creamos una ruta nueva para la nueva función ChangeSort .

```
app.route('/restaurants/:restaurantId/changeSort')
  .patch(
    middlewares.isLoggedIn,
    middlewares.hasRole('owner'),
    middlewares.checkEntityExists(Restaurant, 'restaurantId'),
    middlewares.checkRestaurantOwnership,
    RestaurantController.changeSort)
}
```

Verfica que :

Esta loggeado

Sea owner

Existe el restaurante

Sea el owner del restaurante

Llama a la función changeSort

FRONTEND

endPoints RESTAURANT-ENDPOINTS

```
function changeSort (id) {
  return patch(`restaurants/${id}/changeSort`)
}

export { getAll, getDetail, getRestaurantCategories, create, update, remove, changeSort }
```

MUY IMPORTANTE -> Poner los mismos nombres , changeSort

Screens RESTAURANT-SCREENS

Importamos de endPoints la función changeSort

```
import { changeSort, getAll, remove } from '../../api/RestaurantEndpoints'
import ImageCard from '../../components/ImageCard'
```

Creamos una función llamada changeSortValue

Se utiliza para cambiar el valor de ordenamiento de un restaurante y luego actualizar la lista de restaurantes
Primero, se llama a la función changeSort pasando el **id** del restaurante como argumento. Esta función se encarga de cambiar el valor de ordenamiento del restaurante.
Luego, se llama a la función **fetchRestaurants**. Esta función se encarga de obtener y actualizar la lista de restaurantes.

```
const changeSortValue = async (restaurant) => {
  try {
    await changeSort(restaurant.id)
    await fetchRestaurants()
  } catch (error) {
    console.log(error)
  }
}
```

Añadimos un botón en renderRestaurant

```
<Pressable
    onPress={() => changeSortValue(item)
    }
    style={({ pressed }) => [
      {
        backgroundColor: pressed
          ? GlobalStyles.brandSuccess
          : GlobalStyles.brandSuccessDisabled
      },
      styles.actionButton
    ]}>
  <View style={[{ flex: 1, flexDirection: 'row', justifyContent: 'center' }]}>
  <MaterialCommunityIcons name='sort' color={'white'} size={20} />
    {!item.OrderDefault && <TextRegular textStyle={styles.text}>Sort by: price</TextRegular>}
    {item.OrderDefault && <TextRegular textStyle={styles.text}>Sort by: default</TextRegular>}
  </View>
</Pressable>
```

onPress={() => changeSortValue → Llama a la función antes definida , es decir que cuando se pulse el botón se va a ejecutar esa función , y por tanto se cambiaran los valor del orden

style={({pressed}) → Define que si el botón esta siendo pulsado tenga un color de fondo brandSuccess (verde fuerte) y que si no esta siendo pulsado tenga un color de fondo brandSuccessDisabled (verde flojito)

<Material CommunityIcons … → Lo que hace es añadir este icono 

Y a la hora de poner el nombre OrderBy: Price o default , nos fijamos en el valor de la propiedad OrderDefault , que si es 0 pone Price y si es 1 pone default

RECORDATORIO , PONER LOS NOMBRE IGUALES Y FIJARSE EN TODOS

SOLUCION