

# **E-COMMERCE**

## **WEBSITE**

BY: EBTEHAL ALFAIFI

# UPDATE ROLE

## -Current User Permission:

Only an Admin can change user roles.

## -Role Validation:

The new role must be either "Admin" or "Customer."

## -User Lookup:

If the user is found, their role is updated; otherwise, a "User not found" message is returned.

```
// ميثود لتغيير دور المستخدم
public String updateUserRole(String userId, String newRole, 1 usage
                             String currentUserRole) {
    if (!currentUserRole.equalsIgnoreCase( anotherString: "Admin")) {
        return "You do not have permission to change roles";
    }

    // التحقق من صحة الدور الجديد
    if (newRole.equalsIgnoreCase( anotherString: "Admin")
        || newRole.equalsIgnoreCase( anotherString: "Customer")) {
        UserModel user = getUserById(userId);
        if (user != null) {
            user.setRole(newRole); // تحديث الدور
            return "Role updated successfully";
        }
        return "User not found";
    }
    return "Invalid role";
}
```

```
//1 of 5 extra end point
//update role
@PutMapping(🌐"/update-role/{userId}/{newRole}/{currentUserRole}")
public ResponseEntity updateRole(@PathVariable String userId, @PathVariable String newRole
                                ,@PathVariable String currentUserRole) {
    String result = userService.updateUserRole(userId, newRole,currentUserRole);
    if ("Role updated successfully".equals(result)) {
        return ResponseEntity.status(200).body(new Api("Role updated successfully"));
    } else {
        return ResponseEntity.status(400).body(new Api(result));
    }
}
```

# update balance

This code processes a request to add balance to a user's account, checks if the operation is successful, and then returns either a success or error message based on the result.

```
//2 of 5 extra end point
//حدث رصيد بعدما اضيف مبلغ
public String addBalance(String userId, int amount) { 1 usage
    UserModel user = getUserById(userId);
    if (user != null) {
        user.setBalance(user.getBalance()+amount);
        return "Balance updated successfully";
    }
    return "User not found";
}
```

```
//2 of 5 extra end point
//update balance
@PutMapping("/add-balance/{userId}/{amount}")
public ResponseEntity addBalance(@PathVariable String userId,
                                @PathVariable int amount) {
    String result = userService.addBalance(userId, amount);

    if ("Balance updated successfully".equals(result)) {
        return ResponseEntity.status(200).body(new Api(result));
    } else {
        return ResponseEntity.status(400).body(new Api(result));
    }
}
```

```
// 3 of 5 extra endpoint
@PostMapping("/apply-discount/{productId}/{discountPercentage}")
public ResponseEntity applyDiscount(@PathVariable String productId,
                                   @PathVariable double discountPercentage) {
    double discountedPrice = productService.applyDiscount(productId, discountPercentage);

    if (discountedPrice != -1) {
        return ResponseEntity.status(200).body(new Api("Discount applied successfully," +
            " new price: " + discountedPrice));
    }
    return ResponseEntity.status(400).body(new Api("Product not found"));
}
```

```
//3 of 5 extra endpoint
// ميثود لحساب الخصم على المنتج
public double applyDiscount( String prodectid ,double discountPercentage){ 1 usage
    ProductModel product = getProductById(prodectid);
    if (product != null) {
        double discountedPrice = product.getPrice()
            - (product.getPrice() * discountPercentage / 100);
        product.setPrice(discountedPrice);
        return discountedPrice;
    }
    return -1;
}
```

## Applay discount

### -Retrieve the Product:

- If the product is found, the process continues.
- If not, the method returns -1, indicating the product was not found.

### -Calculate the Discounted Price:

- The discount amount is calculated based on the given discount percentage

### -Update the Product Price:

- The product's price is updated with the new discounted price using
- 

### -Return the New Price:

- The method returns the updated discounted price

```
// 4 of 5 extra end point
// ميثود لتحديث المخزون لجميع المنتجات
@PutMapping("/update-all-stocks/{Increase}")
public ResponseEntity updateAllStocks(@PathVariable int Increase) {
    String result = stockService.updateStockForAllProducts(Increase);

    if (result.equals("Stock updated successfully for all products.")) {
        return ResponseEntity.status(200).body(new Api(result));
    } else {
        return ResponseEntity.status(400).body(new Api(result));
    }
}
```

## update all stock

This code increases the stock for all products in the list by a specified amount, ensuring the input is positive and that there are products to update. It provides feedback messages based on different conditions.

```
public String updateStockForAllProducts(int stockIncrease) { 1 usage
    if (stockIncrease <= 0) {
        return "Stock increase must be a positive ";
    }

    if (list.isEmpty()) {
        return "No products available to update.";
    }

    for (StockModel stock : list) {
        stock.setStock(stock.getStock() + stockIncrease);
    }

    return "Stock updated successfully for all products.";
}
```

```
//5 of 5 extra endpoint
//حدث السعر
@PutMapping("/update-price/{productId}")
public ResponseEntity updateProductPrice(@PathVariable String productId,
                                         @RequestParam double newPrice) {
    boolean updated = productService.updateProductPrice(productId, newPrice);
    if (updated) {
        return ResponseEntity.status(200).body(new Api("Price updated successfully"));
    }
    return ResponseEntity.status(400).body(new Api("Product not found"));
}
```

```
// 5 of 5 extra endpoint
//حدث السعر
public boolean updateProductPrice(String productId, double newPrice) {
    for (ProductModel product : list) {
        if (product.getId().equals(productId)) {
            product.setPrice(newPrice);
            return true;
        }
    }
    return false;
}
```

# UPDATE PRICE

This method searches for a product by its ID and updates its price if found. It returns true if the update is successful and false if the product is not found.



**THANK YOU**

