Furniture Store API Guide

Introduction

Welcome to the Furniture Store API. This guide details how to interact with our online furniture store, including browsing inventory, managing shopping carts, processing orders, and administrative functions.

Important Notes for Reviewers

1. **Furniture Constraints:**

    - Each furniture type has specific available colors.
      Each table has specific material.
      Each chair has two boolean attributes - Is_adjustable, has_armrest.
        - **Dining Tables**
          Color → Brown, Gray
          Material → Wood, Metal
        - **Work Desks** → Black, White
          Material → Wood, Glass
        - **Coffee Tables** → Gray, Red
          Material → Glass, Plastic
        - **Work Chairs** → Red, White
        - **Gaming Chairs** → Black, Blue
    - Furniture names **must follow manufacturer conventions** (e.g., "dining_table" instead of "table").

2. **Shopping Cart Behavior:**

    - Adding an item **updates quantity** if it already exist.
    - Removing an item **clears it completely** from the cart.

3. **Tax Policy:**

    - The store operates tax-free for **Eilat residents**.
    - If an order is shipped outside Eilat, a manager must **apply tax manually** via the `/apply_tax_on_cart` endpoint.

4. **The format of all responses is JSON.** The response content is described below.

5. **Coupon Constraints:**
   Available coupon codes are -
   “SAVE10” - 10% discount
   “DISCOUNT20” - 20% discount
   “PROMO30” - 30% discount
   “COUPON40” - 40% discount
   “OFFER50” - 50% discount

---

1. User Management APIs

Register a New User - POST

Creates a new user account.

Endpoint: http://127.0.0.1:5005/user_register

```
{
    "name": "Amit",
    "email": "amit@example.com",
    "password": "Amit1234",
    "address": "Arlozorov 17, Tel Aviv"
}
```

Response - Confirmation of successful registration or failure.

Success example:

```
{
    "message": "User Amit created successfully,the username for connection is:
amit@example.com"
}
```

Notes:

- Email must be unique. Name Address and password is required.

Sign In- GET

Endpoint: http://127.0.0.1:5005/sign_in?email={email}&password={password}

Response:

```
{

    "message": "Signing in was successful!You are logged in to our site, Amit - welcome back!",

    "user_data": {

        "email": "amit@example.com",

        "name": "Amit"

    }

}
```

View User Info - GET

Authenticates a user for account access.

Endpoint: http://127.0.0.1:5005/get_user_info?email={email}

Response:

```
{

    "user_info": "User: Name =Amit, Email=amit@example.com,Address=Arlozorov 17, Tel Aviv, Credit=0.0"

}
```

Notes: User must be logged in.

Edit User Details - PUT

Updates a user's name or address.

Endpoint: http://127.0.0.1:5005/edit_user's_details

Example:

```
{
  "email": "ron@example.com",
  "new_address": "Updated Address",
  "new_name": "Ron Updated"
}
```

Response:

```
{
    "message": "Your details have been successfully updated."
}
```

Notes:

- Only name and address or one of them can be modified. If modifying only name/address, only the desired key should be written.
- User must be logged in.

Update Password - PUT

Changes a user's password.

Endpoint: http://127.0.0.1:5005/update_password

Body example:

```
{
  "email": "amit@example.com",
  "new_password": "newpassword123"
}
```

Response example:

```
{
    "message": "Password updated successfully"
}
```

Notes:

- Strong passwords required (minimum 8 characters).
  Response for not valid input:
  ```
  {
      "error": "new_password length not valid"
  }
  ```
- User must be logged in.

2. Shopping & Inventory APIs

These endpoints facilitate browsing inventory and managing shopping carts.

Browse Furniture by Price Range - GET

Filters furniture items within specified price parameters.

Endpoint:
http://127.0.0.1:5005/get_furniture_info_by_price_range?max_price={X}&min_price={Y}

Response example:

```
[
    {
        "color": "black",
        "depth": 55,
        "f_desc": "A high quality work desk in black.",
        "f_name": "black work desk",
        "furniture_type": 2,
        "has_armrest": null,
        "high": 120,
        "id": 2,
        "is_adjustable": null,
        "material": "glass",
        "price": 199.99,
        "quantity": 7,
        "width": 65
    }, To be continued ...
```

Notes:

- Only returns available inventory (items with quantity > 0).

View Shopping Cart - GET

Displays current cart contents.

Endpoint: http://127.0.0.1:5005/view_shoppingcart?email={email}

Responses Example:

```
{

    "your shopping cart": {

        "brown dining table": 2

    }

}
```

Notes:

- User must be logged in.

Add Item to Cart - PUT

Endpoint: http://127.0.0.1:5005/add_item_to_cart

Body Example for table:

```
{
  "email": "amit@example.com",
  "amount": 1,
  "object_type": "dining_table",
  "item":{
           "color": "brown",
           "table": {"material": "metal"}
          }
}
```

Body Example for chair:

```
{
  "email": "amit@example.com",
  "amount": 2,
  "object_type": "work_chair",
  "item":{
           "color": "brown",
           "chair": {"is_adjustable": true, “has_armrest”: true}
          }
}
```

Response example:

```
{
    "advertisement": "*** SPECIAL OFFER !!! ***    We found a matching table for
your chair! Description: A high quality work desk in black. Material: wood It's the
perfect table for you, and it's in stock!",
    "message": "The addition to the shopping cart was completed successfully!"
}
```

Notes:

- If the item exists, quantity is updated instead of accumulating quantity.
- Color must be valid for the selected furniture type, and special attributes needed to be chosen.
- Object type name must match one of the values of furnitures existing
- User must be logged in.

Remove Item from Cart - DELETE

Removes a specific item from the shopping cart

Endpoint: http://127.0.0.1:5005/remove_item_from_cart

Body example:{

```
"email": "ido2@example.com",
"object_type": "dining_table",
"item": {
   "color": "brown",
   "table": {"material": "wood"}
     }
}
```

Response Example:

```
{
    "message": "Removing the furniture from the shopping cart was successfully
completed."
}
```

Notes:

- Completely removes the item from the cart.
- User must be logged in.

Get Total Price of Cart - GET

Calculates the total cost of cart items, with optional discount code application.

Endpoint: http://127.0.0.1:5005/get_total_price?email={email}&coupon_code={code/None}

Notes:

- If a coupon is applied, the discount is reflected.
- Final price does not include tax unless it is applied manually.
- User must be logged in.

Checkout - POST

Processes payment and finalizes purchase.

Endpoint: http://127.0.0.1:5005/checkout

```
{
  "email": "ron@example.com",
  "credit_card_num": "99999999999",
  "coupon_code": "OFFER50"
}
```

Notes:

- Automatically applies available credit to reduce the total.
- Automatically adds 10% of price paid (by card, not credit) to the user’s credit amount.
- Credit card number must be more than 8 digits.
- Coupon code can be None.
- User must be logged in.

---

3. Order Management APIs

View Order History- GET

Displays past orders for a specific user.

Endpoint: http://127.0.0.1:5005/get_user's_orders_history?email={email}

Notes:

- User must be logged in.

Apply Tax (For Managers Only) - PUT

Endpoint: http://127.0.0.1:5005/apply_tax_on_user

Body Example:

```
{
  "manager_email": "hili@example.com",
  "user_email": "raz@example.com”,
  "tax_rate": 15
}
```

Notes:

- Applies the desired tax on the items in the specified user’s shopping cart.
- Manager must be signed in.

4. Administrative APIs (Manager Access Required)

Update Inventory - PUT

Modifies inventory quantities.

Endpoint: http://127.0.0.1:5005/update_inventory

Body example:

```
{
  "existing_admin_email": "hili@example.com",
  "quantity": 25,
  "sign": 1,
  "object_type": "work_chair",
  "item": {
           "color": "white",
          "chair": {"is_adjustable":false , "has_armrest":false}
        }
}
```

Notes:

- sign = 1 increases stock, sign = -1 decreases stock.
- Manager must be signed in.

View All Orders- GET

Provides a complete view of customer orders.

Endpoint: http://127.0.0.1:5005/get_all_orders_by_manager?email={manager_email}

Notes:

- Manager must be signed in.

Register a New Manager - POST

Creates additional manager accounts.

Endpoint: http://127.0.0.1:5005/secondary_manager_register

```
{
  "existing_admin_email": "tal@example.com",
  "name": "New Manager",
  "email": "newmanager@example.com",
  "password": "securepass"
}
```

Response: Manager account creation confirmation.

Notes:

- Only existing managers can create new manager accounts.
- Manager must be signed in.

Delete a User- DELETE

Removes a user account from the system.

Endpoint: http://127.0.0.1:5005/delete_user

Body Example:

```
{
  "email": "tal@example.com",
  "email_to_delete": "amit@example.com"
}
```

Notes:

- Only managers can delete users.
- Manager must be signed in.

Add Credit to User - PUT

Endpoint: http://127.0.0.1:5005/add_credit_to_user

Body Example:

```
{

"manager_email": "ban@example.com",

"user_email" : "raz@example.com",

"credit" : 1000

}
```

Notes:

- Manager and User must be signed in.

Update Order Status - PUT

Endpoint: http://127.0.0.1:5005/update_order_status

Modifies the status of an existing order.

```
{
  "email": "hili@example.com",
  "order_id": 1
}
```

Response: Order status update confirmation.

Notes:

- Used to mark orders as Shipped, Delivered, etc.
- Manager must be signed in.