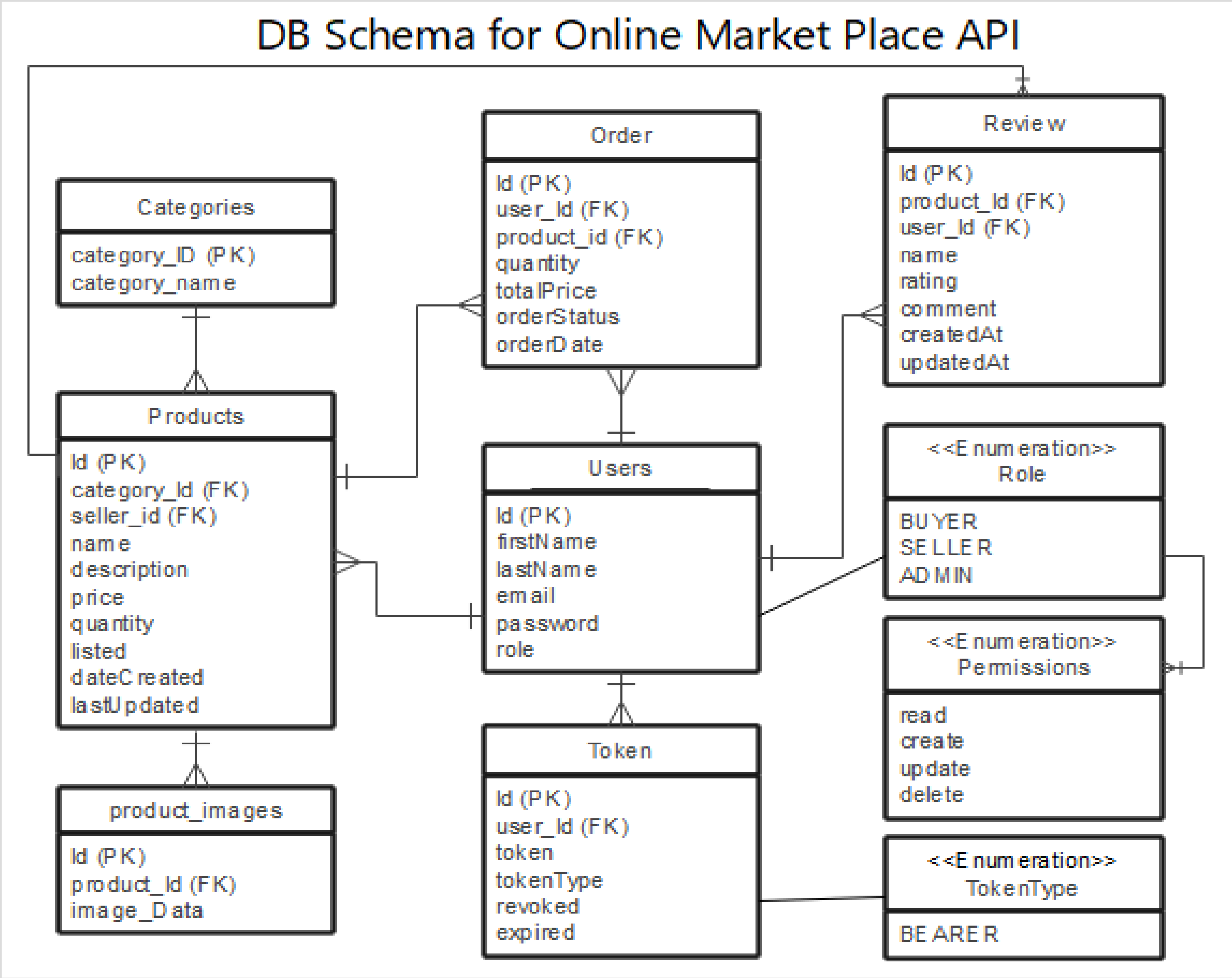
ONLINE MARKETPLACE REST API DOCUMENTATION

Project Description:

The online marketplace API aims to provide a platform for users to buy and sell products. It includes features such as user authentication, product management, order processing, review submission, and user management. The API is designed to support different roles, including buyers, sellers, and administrators, each with specific permissions and access levels.


DB Schema for Online Market Place API

Database Schema:

Users Table:

- Store information about users such as their first name, last name, email, password, and role.
- Each user can have multiple products, orders, and reviews associated with them.

Products Table:

- Represents products available in the marketplace.
- Contains information like product name, description, price, quantity, listed status, and timestamps for creation and update.
- Each product belongs to a category and a seller (user).

Categories Table:

- Stores categories for organizing products.
- Each category can have multiple products associated with it.

Orders Table:

- Represents orders placed by users for purchasing products.
- Contains details such as the product, buyer (user), quantity, total price, order status, and timestamps for order creation.

Reviews Table:

- Stores reviews submitted by users for products.
- Includes information like the product, user, rating, comment, and timestamps for creation and update.

Product Images Table:

- Stores images associated with products.
- Images are stored and retrieved as binary data (base64).
- Each product can have multiple images.

Token Table:

- Stores tokens associated with users.
- One user has many tokens stored but the latest is the valid one.

Core features:

User Authentication:

- Users can register for a new account (/api/v1/auth/register), authenticate themselves (/api/v1/auth/authenticate) using JWT tokens which is valid for one day and logout using (/api/v1/auth/logout).
- Token refresh functionality is provided (/api/v1/auth/refresh-token) and is validated for 7 days.

Product Management:

- Sellers and admins can create, update, delete products ("/api/l3/product/create_product",
  "/api/l3/product/update/{productId}",
  "/api/l3/product/del/{productId}").

- All users can search for products (/api/l1/search/{searchParam}) and view listed products (/api/l1/listed).

Order Processing:

- Buyer can place orders (/api/l2/order/press_order) and Seller review the order made to their product (/api/l2/order/orders_list).

Review and Filter Management:

- Buyer can submit reviews for products (/api/l2/reviews/submitReview).

- Average ratings for a specific product can be retrieved (/api/l2/reviews/product/{productId}/average-rating).
- Reviews for a specific product can be fetched (/api/l2/reviews/product/{productId}).
- Popular product, highRated and sorted by Rating can be found

  ("/api/l3/product/popular", "/api/l3/product/highRated", "/api/l3/product/sortedByRating")

User Management:

- Admins can manage users by updating their details (/api/l2/users/update/{userId}) and roles (/api/l2/users/updateRole/{userIdToUpdate}).
- All users can be listed (/api/l2/users/all_users).

Category Management:

- Admins can create, update, and delete product categories (/api/l3/category/* endpoints).
- All categories can be listed (/api/l3/category/all_list).

Image Management:

- Sellers and admins can upload and delete images for products (/api/l3/images/* endpoints).

Endpoints Access Categorization:

Buyer Controller

/api/l1/listed

- Description: Endpoint to retrieve a list of available products with a quantity greater than zero.
- Method: GET
- Response: List of products

/api/l1/search/{searchParam}

- Description: Endpoint to search for products by name.
- Method: GET
- Parameters: searchParam (String)
- Response: List of products matching the search parameter

Category Controller

/api/l3/category/createCategory

- Description: Endpoint to create a new category.
- Method: POST
- Request Body: CategoryDto
- Response: Created category object

/api/l3/category/update/{categoryId}

- Description: Endpoint to update an existing category.
- Method: PUT
- Parameters: categoryId (Integer)
- Request Body: CategoryDto

- Response: Updated category object

/api/l3/category/del/{categoryId}

- Description: Endpoint to delete an existing category.
- Method: DELETE
- Parameters: categoryId (Integer)
- Response: No content

/api/l3/category/all_list

- Description: Endpoint to retrieve a list of all categories.
- Method: GET
- Response: List of categories

Order Controller

/api/l2/order/press_order

- Description: Endpoint to submit a new order.
- Method: POST
- Request Body: OrderDto
- Response: Created order object

/api/l2/order/orders_list

- Description: Endpoint to retrieve a list of orders for the current user.
- Method: GET
- Response: List of orders

/api/l2/order/update/{orderId}

- Description: Endpoint to update the status of a placed order.
- Method: PUT

- Parameters: orderId (Integer)
- Request Body: Map<String, String>
- Response: Updated order object

Product Controller

/api/l3/product/create_product

- Description: Endpoint to create a new product.
- Method: POST
- Request Body: ProductDto
- Response: Created product object

/api/l3/product/{productId}

- Description: Endpoint to retrieve a product by its ID.
- Method: GET
- Parameters: productId (Integer)
- Response: Product object

/api/l3/product/del/{productId}

- Description: Endpoint to delete a product by its ID.
- Method: DELETE
- Parameters: productId (Integer)
- Response: No content

/api/l3/product/notListed

- Description: Endpoint to retrieve a list of products that are not listed.
- Method: GET
- Response: List of products

Product Image Controller

/api/l3/images/upload/{productId}

- Description: Endpoint to upload an image for a specified product.
- Method: POST
- Parameters: productId (Integer)
- Request Body: MultipartFile
- Response: No content

/api/l3/images/products/{productId}

- Description: Endpoint to retrieve a product with its associated images.
- Method: GET
- Parameters: productId (Integer)
- Response: Product object with images

/api/l3/images/del/{imageId}

- Description: Endpoint to delete a product image by its ID.
- Method: DELETE
- Parameters: imageId (Integer)
- Response: No content

Review Controller

/api/l2/reviews/submitReview

- Description: Endpoint to submit a new review for a product.
- Method: POST
- Request Body: ReviewDto
- Response: Created review object

/api/l2/reviews/product/{productId}

- Description: Endpoint to retrieve reviews for a product.
- Method: GET
- Parameters: productId (Integer)
- Response: List of reviews

/api/l2/reviews/product/{productId}/average-rating

- Description: Endpoint to calculate the average rating for a product.
- Method: GET
- Parameters: productId (Integer)
- Response: Average rating (Double)

User Controller

/api/l2/users/all_users

- Description: Endpoint to retrieve a list of all users.
- Method: GET
- Response: List of users

/api/l2/users/update/{userId}

- Description: Endpoint to update a user's information.
- Method: PUT
- Parameters: userId (Integer)
- Request Body: RegisterRequest
- Response: Updated user object

/api/l2/users/updateRole/{userIdToUpdate}

- Description: Endpoint to update a user's role if the requester is an admin.

- Method: PUT
- Parameters: userIdToUpdate (Integer)
- Request Body: Map<String, String>
- Response: Success message or error message