**Trade Lens**

User Manual

**Introduction**

Trade Lens offers a robust REST API designed to empower traders to programmatically monitor and manage their trading performance through HTTP requests. With Trade Lens, users can seamlessly integrate trading data and analytics into their applications, enabling efficient tracking and analysis of trading activities.

**API Endpoints**

Below is a comprehensive list of available API endpoints provided by Trade Lens:

1. **POST BASE\_URL/auth/register**
   * **Description:** Public endpoint for registering a user account.
   * **HTTP Method:** **POST**
   * **Endpoint:** BASE\_URL/**auth/register**
   * **Response:** Returns a bearer token if the registration process was successful.
   * **Parameters:** email, username, password
2. **POST BASE\_URL/auth/authenticate**
   * **Description:** Allows users to log into the application, granting permissions depending on their authority level.
   * **HTTP Method:** **POST**
   * **Endpoint:** BASE\_URL/auth/authenticate
   * **Parameters: username, password**
3. **GET BASE\_URL/user**
   * **Description:** Retrieves information about all users registered in the Trade Lens system.
   * **HTTP Method:** GET
   * **Endpoint:** BASE\_URL/user
   * **Response:** Returns a list of user objects containing details such as user ID, username, email, etc.
4. **GET BASE\_URL/user/id**
   * **Description:** Retrieves detailed information about a specific user based on the provided user ID.
   * **HTTP Method:** GET
   * **Endpoint:** BASE\_URL/user/id
   * **Parameters:**
     + id: The unique identifier of the user.
   * **Response:** Returns a user object containing detailed information about the specified user.
5. **POST BASE\_URL/user**
   * **Description:** Creates a new user in the Trade Lens system.
   * **HTTP Method:** POST
   * **Endpoint:** BASE\_URL/user
   * **Request Body:** JSON object containing user details such as username, email, password, etc.
   * **Response:** Returns the newly created user object with generated user ID.
6. **PUT BASE\_URL/user**
   * **Description:** Updates an existing user's information in the Trade Lens system.
   * **HTTP Method:** PUT
   * **Endpoint:** BASE\_URL/user
   * **Request Body:** JSON object containing updated user details along with the user ID.
   * **Response:** Returns the updated user object with modified information.
7. **DELETE BASE\_URL/user/id**
   * **Description:** Deletes a user from the Trade Lens system based on the provided user ID.
   * **HTTP Method:** DELETE
   * **Endpoint:** BASE\_URL/user/id
   * **Parameters:**
     + id: The unique identifier of the user to be deleted.
   * **Response:** Returns a success message confirming the deletion of the specified user.
8. **GET BASE\_URL/trade**
   * **Description:** Retrieves information about all trades recorded in the Trade Lens system.
   * **HTTP Method:** GET
   * **Endpoint:** BASE\_URL/trade
   * **Response:** Returns a list of trade objects containing details such as trade ID, trade date, trade type, etc.
9. **GET BASE\_URL/trade/id**
   * **Description:** Retrieves detailed information about a specific trade based on the provided trade ID.
   * **HTTP Method:** GET
   * **Endpoint:** BASE\_URL/trade/id
   * **Parameters:**
     + id: The unique identifier of the trade.
   * **Response:** Returns a trade object containing detailed information about the specified trade.
10. **POST BASE\_URL/trade**
    * **Description:** Records a new trade in the Trade Lens system.
    * **HTTP Method:** POST
    * **Endpoint:** BASE\_URL/trade
    * **Request Body:** JSON object containing trade details such as trade date, trade type, asset traded, quantity, price, etc.
    * **Response:** Returns the newly recorded trade object with generated trade ID.
11. **PUT BASE\_URL/trade**
    * **Description:** Updates an existing trade's information in the Trade Lens system.
    * **HTTP Method:** PUT
    * **Endpoint:** BASE\_URL/trade
    * **Request Body:** JSON object containing updated trade details along with the trade ID.
    * **Response:** Returns the updated trade object with modified information.
12. **DELETE BASE\_URL/trade/id**
    * **Description:** Deletes a trade from the Trade Lens system based on the provided trade ID.
    * **HTTP Method:** DELETE
    * **Endpoint:** BASE\_URL/trade/id
    * **Parameters:**
      + id: The unique identifier of the trade to be deleted.
    * **Response:** Returns a success message confirming the deletion of the specified trade.

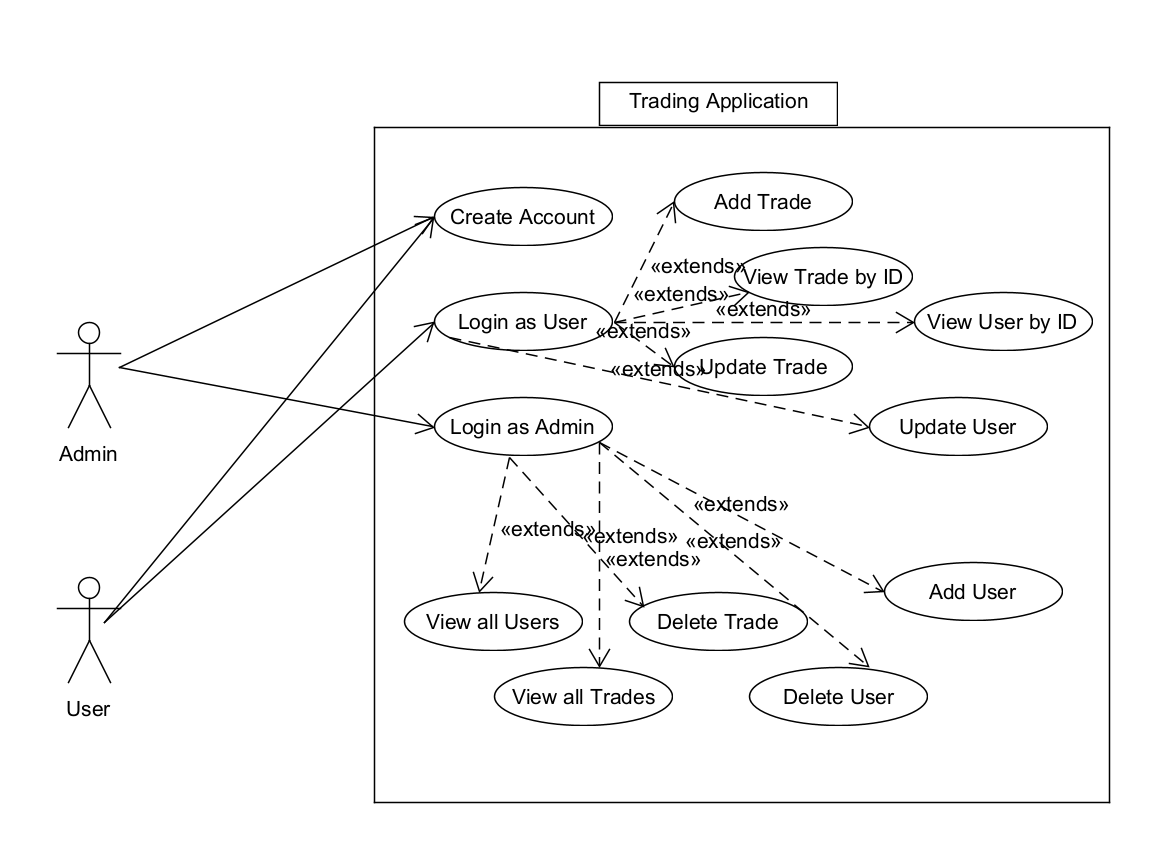
**Note:** Replace BASE\_URL with the actual base URL of the Trade Lens API.

Class diagram:

A screenshot of a computer program

Description automatically generated

Use case diagram:



Database diagram:

