

API Documentation

1. Create (POST) a New User

- **Endpoint:** /user
- **Method:** POST
- **Description:** Creates a new user with the provided details.
- **Request:**
 - **Headers:**
 - Content-Type: application/json
 - **Body:**

```
{
  "name": "string",
  "email": "string",
  "password": "string",
  "type": "string"
}
```
 - **Body Parameters:**
 - name: (string) Name of the user.
 - email: (string) Email of the user.
 - password: (string) Password of the user.
 - type: (string) Type of the user (e.g., admin, regular).
 - active: (string) Always set to "true" (default is active).
- **Response:**
 - **Status Code:** 200 OK
 - **Body:**

```
{
  "id": "number",
  "name": "string",
  "email": "string",
  "password": "string",
  "type": "string",
  "active": "true"
}
```

2. Update (PUT) an Existing User

- **Endpoint:** /user/:id
- **Method:** PUT
- **Description:** Updates the user data based on the provided id.
- **Request:**
 - **Headers:**
 - Content-Type: application/json
 - **Body:**

```
{
  "name": "string",
```

```
"email": "string",
"password": "string",
"type": "string"
}
```

- **URL Parameters:**
 - id: (number) ID of the user to be updated.
- **Body Parameters:**
 - name: (string) Name of the user.
 - email: (string) Email of the user.
 - password: (string) Password of the user.
 - type: (string) Type of the user.
 - active: (string) Always set to "true" (default is active).
- **Response:**
 - **Status Code:** 200 OK
 - **Body:**

```
json
Copy code
{
  "id": "number",
  "name": "string",
  "email": "string",
  "password": "string",
  "type": "string",
  "active": "true"
}
```

3. Read (GET) All Users

- **Endpoint:** /users
- **Method:** GET
- **Description:** Retrieves a list of all active users.
- **Request:**
 - **Headers:** None
 - **Query Parameters:** None
 - **Body:** None
- **Response:**
 - **Status Code:** 200 OK
 - **Body:**

```
[
  {
    "id": "number",
    "name": "string",
    "email": "string",
    "password": "string",
    "type": "string",
    "active": "true"
  },
  ...
]
```

4. Read (GET) a Specific User by ID

- **Endpoint:** /user/:id
- **Method:** GET
- **Description:** Retrieves the details of a specific user by their id.
- **Request:**
 - **Headers:** None
 - **URL Parameters:**
 - id: (number) ID of the user to retrieve.
 - **Body:** None
- **Response:**
 - **Status Code:** 200 OK
 - **Body:**

```
[
  {
    "id": "number",
    "name": "string",
    "email": "string",
    "password": "string",
    "type": "string",
    "active": "true"
  }
]
```

5. Delete (DELETE) a User

- **Endpoint:** /user/:id
- **Method:** DELETE
- **Description:** Deletes a user based on their id.
- **Request:**
 - **Headers:** None
 - **URL Parameters:**
 - id: (number) ID of the user to delete.
 - **Body:** None
- **Response:**
 - **Status Code:** 200 OK
 - **Body:**

```
{
  "id": "number",
  "name": "string",
  "email": "string",
  "password": "string",
  "type": "string",
  "active": "true"
}
```

6. Create a New Order

Endpoint: /order

Method: POST

Request:

- **Body Parameters:**
 - name (string, required) - Name of the order item.
 - state (string, required) - State or status of the order.
 - deliverytime (string, required) - Expected delivery time for the order.

Example Request Body:

```
{
  "name": "Item Name",
  "state": "Pending",
  "deliverytime": "2023-09-01T10:00:00Z"
}
```

Response:

- **Status:** 200 OK
- **Body:** JSON object representing the newly created order.

Example Response:

```
{
  "id": 1,
  "name": "Item Name",
  "state": "Pending",
  "deliverytime": "2023-09-01T10:00:00Z"
}
```

7. Update an Existing Order

Endpoint: /order/:id

Method: PATCH

Request:

- **Path Parameters:**
 - id (integer, required) - The unique ID of the order to update.
- **Body Parameters:** JSON object with fields to update, such as name, state, or deliverytime.

Example Request:

```
{
  "state": "Completed"
}
```

Response:

- **Status:** 200 OK
- **Body:** JSON object representing the updated order.

Example Response:

```
{
  "id": 1,
  "name": "Item Name",
  "state": "Completed",
  "deliverytime": "2023-09-01T10:00:00Z"
}
```

8. Get All Orders

Endpoint: /orders

Method: GET

Request: None

Response:

- **Status:** 200 OK
- **Body:** Array of JSON objects, each representing an order.

Example Response:

```
[
  {
    "id": 1,
    "name": "Item Name",
    "state": "Completed",
    "deliverytime": "2023-09-01T10:00:00Z"
  },
  {
    "id": 2,
    "name": "Another Item",
    "state": "Pending",
    "deliverytime": "2023-09-05T12:00:00Z"
  }
]
```

9. Get Order by ID

Endpoint: /order/:id

Method: GET

Request:

- **Path Parameters:**
 - `id` (integer, required) - The unique ID of the order to retrieve.

Response:

- **Status:** 200 OK
- **Body:** JSON object representing the order with the specified ID.

Example Response:

```
{
  "id": 1,
  "name": "Item Name",
  "state": "Completed",
  "deliverytime": "2023-09-01T10:00:00Z"
}
```

10. Delete an Order by ID

Endpoint: /order/:id

Method: DELETE

Request:

- **Path Parameters:**
 - **id** (integer, required) - The unique ID of the order to delete.

Response:

- **Status:** 200 OK
- **Body:** JSON object representing the deleted order.

Example Response:

```
{
  "id": 1,
  "name": "Item Name",
  "state": "Completed",
  "deliverytime": "2023-09-01T10:00:00Z"
}
```

11. Get Average Delivery Time

- **Endpoint:** GET /averagedeliverytime
- **Description:** Calculates the average delivery time for all orders.

Request

- **Method:** GET
- **URL Params:** None
- **Request Body:** None

Response

- **Content Type:** application/json
- **Status Code:** 200 OK
- **Response Body:** Returns the average delivery time (as a numeric value).

```
{
  "average_delivery_time": 5.25
}
```

12. Get Order Trends

- **Endpoint:** GET /ordertrends
- **Description:** Retrieves a list of orders sorted by creation date, with the most recent orders first.

Request

- **Method:** GET
- **URL Params:** None
- **Request Body:** None

Response

- **Content Type:** application/json
- **Status Code:** 200 OK
- **Response Body:** Returns a list of order objects sorted by the `created_at` field in descending order.

```
[
  {
    "id": 1,
    "created_at": "2024-01-01T12:34:56.000Z",
    "delivery_time": 5,
    ...
  },
  {
    "id": 2,
    "created_at": "2023-12-30T15:21:22.000Z",
    "delivery_time": 3,
    ...
  }
]
```

13. Get Active Users

- **Endpoint:** GET /activeusers
- **Description:** Retrieves a list of all active users.

Request

- **Method:** GET

- **URL Params:** None
- **Request Body:** None

Response

- **Content Type:** application/json
- **Status Code:** 200 OK
- **Response Body:** Returns a list of active user objects.

```
[
  {
    "id": 1,
    "name": "John Doe",
    "email": "john.doe@example.com",
    "active": "true",
    ...
  },
  {
    "id": 2,
    "name": "Jane Smith",
    "email": "jane.smith@example.com",
    "active": "true",
    ...
  }
]
```