

接口文档-user

Certainly! Below is a sample documentation for the API endpoints defined in your `UserController` class, formatted for use in Postman. This documentation includes details like the request type, URL, request body format, and a brief description of what each endpoint does.

1. User Login

POST `/user/login`

- **Description:** Authenticates a user based on the provided credentials.
- **Request Body (JSON):**

```
{
  "name": "username",
  "password": "password"
}
```

- **Response:** Returns the details of the authenticated user.
- **Response Example:**

```
{
  "id": 1,
  "name": "username",
  "password": "password",
  "phone": "1234567890",
  "address": "123 Baker Street"
}
```

- **Status Codes:**
 - `200 OK`: If the login is successful.
 - `401 Unauthorized`: If the login credentials are invalid.

2. Insert User

POST `/user/insertUser`

- **Description:** Adds a new user to the system.
- **Request Body (JSON):**

```
{
  "name": "newUser",
  "password": "newPassword",
  "phone": "0987654321",
  "address": "456 Elm Street"
}
```

- **Response:** Returns a boolean value indicating success or failure.

- **Response Example:**

- `true`: User creation was successful.
- `false`: User creation failed.

- **Status Codes:**

- `200 OK`: If the request was processed successfully.
- `500 Internal Server Error`: If there was an error processing the request.

order-service

Based on your `OrderController`, here's the API documentation for testing in Postman:

API Documentation for Order Controller

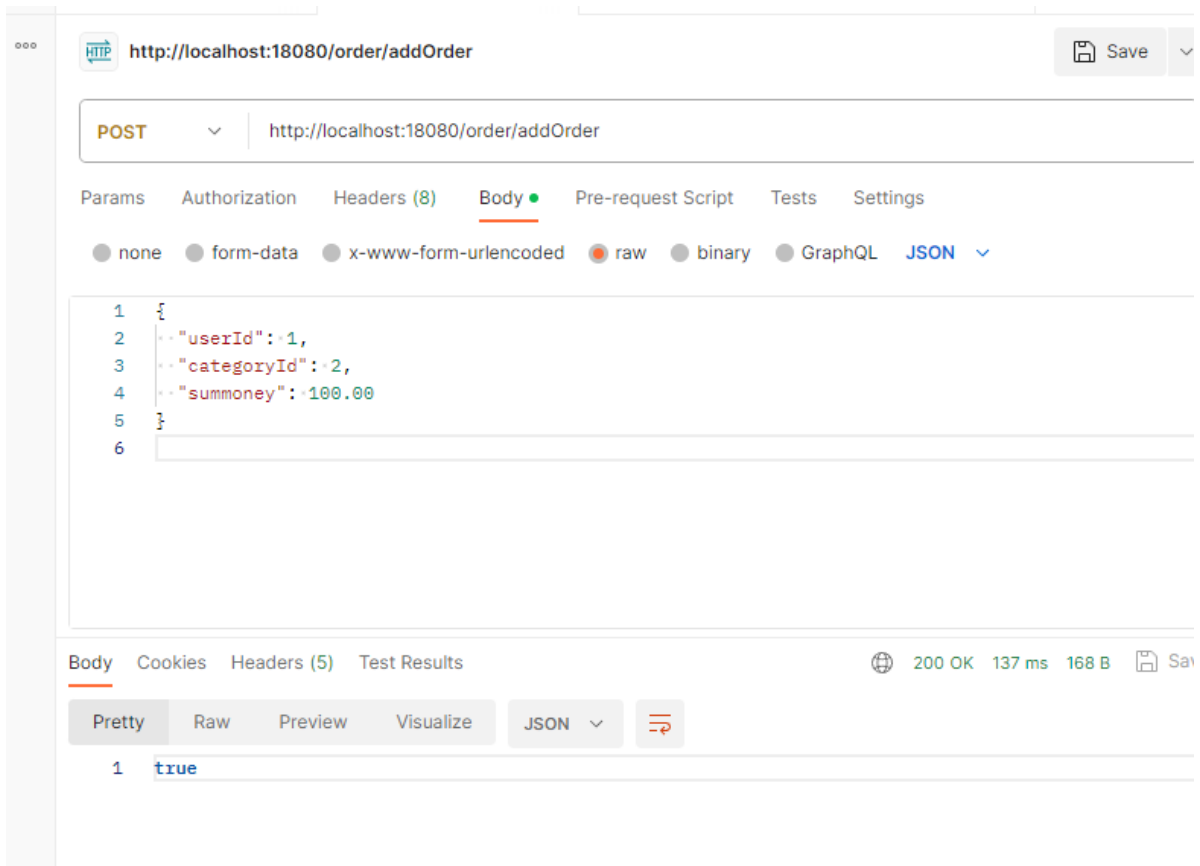
1. Add Order

Endpoint: `/order/addOrder`

- **Method:** `POST`
- **Description:** Adds a new order to the system.
- **Request Body (JSON):**

```
{
  "userId": 1,
  "categoryId": 2,
  "summoney": 100.00
}
```

- `userId`: ID of the user placing the order.
- `categoryId`: ID of the category of the order.
- `summoney`: Total sum of money for the order.
- **Response:**
 - Returns a boolean value indicating the success (`true`) or failure (`false`) of the operation.
- **Status Codes:**
 - `200 OK`: Request processed successfully.
 - `500 Internal Server Error`: Server error or failure in processing the request.



2. Get Order By ID

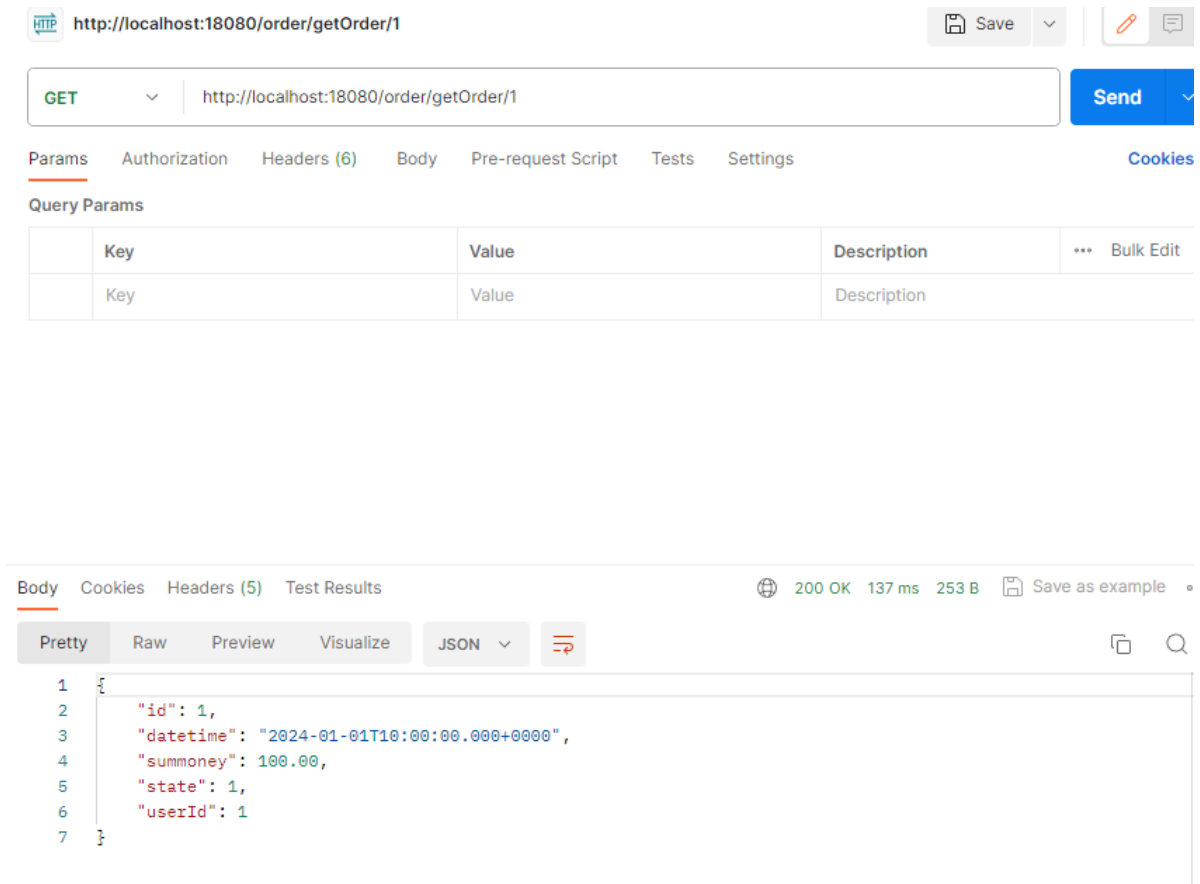
Endpoint: `/order/getOrder/{id}`

- **Method:** GET
- **Description:** Retrieves an order by its ID.
- **URL Parameters:**
 - `id`: The unique identifier of the order.
- **Response:**
 - **Success:** Returns the details of the order.

```
{  "id": 1,  "datetime": "2024-01-01T10:00:00",  "summoney": 100.00,  "state": 1,  "userId": 1}
```

- **Failure:** Returns an HTTP 404 Not Found status if the order is not found.
- **Status Codes:**

- 200 OK : Order found and returned successfully.
- 404 Not Found : No order found with the given ID.



The screenshot displays a Postman interface for a REST client. At the top, the URL bar shows `http://localhost:18080/order/getOrder/1`. The method is set to `GET`. Below the URL bar, the 'Params' tab is active, showing a table with columns 'Key', 'Value', and 'Description'. The table is currently empty. The 'Body' tab is also visible. The response section at the bottom shows a status of `200 OK` with a response time of `137 ms` and a size of `253 B`. The response body is displayed in JSON format:

```
{
  "id": 1,
  "datetime": "2024-01-01T10:00:00.000+0000",
  "summoney": 100.00,
  "state": 1,
  "userId": 1
}
```

category-service

Based on the provided `CategoryController`, I will create API documentation for testing in Postman. This controller seems to handle operations related to categories, including retrieving a category by ID, saving a new category, and listing categories based on certain criteria.

API Documentation for Category Controller

1. Get Category by ID

Endpoint: `/category/{id}`

- **Method:** `GET`
- **Description:** Retrieves details of a category by its ID.
- **URL Parameters:**
 - `id`: The unique identifier of the category.
- **Response:**
 - Returns the details of the category with the specified ID.
- **Status Codes:**
 - `200 OK`: Category found and returned successfully.
 - `404 Not Found`: No category found with the given ID.

The screenshot shows a REST client interface with a GET request to `http://localhost:18082/category/1` successfully executed. The response status is `200 OK` with a response time of `156 ms` and a body size of `981 B`. The response body is displayed in JSON format, showing details for 'Android Phones'.

```
37  {
38    "parentid": 0,
39    "orderNumber": 1,
40    "childs": [],
41    "name": "Android Phones",
42    "price": 300.00,
43    "img": "android_phones.jpg",
44    "detail": "Variety of Android smartphones"
45  },
46  ],
47  {
48    "name": "Smartphones",
49    "price": 500.00,
50    "img": "smartphones.jpg"
51  }
52 }
```

At the bottom of the interface, there are several utility buttons: Postbot, Runner, Start Proxy, Cookies, Trash, and a window management icon.

2. Create or Update Category

Endpoint: `/category`

- **Method:** `POST`
- **Description:** Saves a new category or updates an existing one.
- **Request Body (JSON):**

```
{
  "id": 1, // optional for new category
  "name": "categoryName",
  "level": 1,
  // other category fields
}
```

- **Response:**
 - Returns the details of the saved category along with the location URI in the response headers.
- **Response Headers:**
 - `Location`: URI of the newly saved category.
- **Status Codes:**
 - `201 Created`: Category created or updated successfully.

The screenshot displays a REST client interface with the following details:

- URL:** `http://localhost:18082/category`
- Method:** `POST`
- Body (JSON):**

```
1 {
2   "id": 1, // Optional for new category
3   "name": "categoryName",
4   "level": 1
5 }
6 |
```
- Response:** `201 Created` (67 ms, 341 B)
- Response Body (JSON):**

```
1 {
2   "id": 1,
3   "level": 1,
4   "parentId": null,
5   "orderNumber": null,
6   "childs": null,
7   "name": "categoryName",
8   "price": null,
9   "img": null,
10  "detail": null
11 }
```

3. List Categories

Endpoint: `/category`

- **Method:** `GET`
- **Description:** Lists categories based on the specified level and name.
- **Query Parameters:**
 - `level`: The level of categories to filter by. Default is `0`.
 - `name`: The name of categories to filter by. Default is `"top"`.
- **Response:**
 - Returns a list of categories that match the given criteria.
- **Status Codes:**
 - `200 OK`: Successfully retrieved the list of categories.

The screenshot shows a REST client interface with a GET request to `http://localhost:18082/category?level=1&name=categoryName`. The request is saved and ready to be sent. The response is a JSON array with one object, displayed in the 'Body' tab. The response status is `200 OK` with a response time of `14 ms` and a size of `291 B`.

```
1 [
2   {
3     "id": 1,
4     "level": 1,
5     "parentId": null,
6     "orderNumber": null,
7     "childs": [],
8     "name": "categoryName",
9     "price": null,
10    "img": null,
11    "detail": null
12  }
13 ]
```