

# Rewards Calculation System API Documentation

The Rewards Calculation System provides the following API endpoints for managing transactions and retrieving reward points.

## Create Transaction

Create a new transaction and store it in the database.

- Method: POST
- Endpoint: /api/v1/transactions
- Request Body: JSON representation of the transaction details (customerId, amount, transactionDate)
- Response: 201 Created if the transaction is created successfully
- Description: Creates a new transaction and stores it in the database.

Example:

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

- URL:** `http://localhost:8080/api/v1/transactions`
- Method:** `POST`
- Body (Request):**

```
{  1  "customerId": 4,  2  "amount": 100.0,  3  "transactionDate": "2023-05-23"  4  }
```
- Response:** `201 Created` (Status), `79 ms` (Time), `239 B` (Size). The response body is shown in JSON format:

```
{  1  "id": 12,  2  "customerId": 4,  3  "amount": 100.0,  4  "transactionDate": "2023-05-23"  5  }
```

## Create Customer

Create a new customer and store it in the database.

- Method: POST
- Endpoint: /api/v1/customer
- Request Body: JSON representation of the customer details (name)
- Response: 201 Created if the customer is created successfully
- Description: Creates a new customer and stores it in the database.

Example:

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

- URL:** `http://localhost:8080/api/v1/customer`
- Method:** POST
- Request Body (JSON):**

```
{
  "name": "Mao Li"
}
```
- Response (JSON):**

```
{
  "id": 4,
  "name": "Mao Li"
}
```
- Status:** 201 Created
- Duration:** 166 ms
- Size:** 193 B

## Calculate Reward Points

Retrieve the reward points for a customer in a specific month and the total reward points.

- Method: GET
- Endpoint: /api/v1/transactions/{customerId}/points
- Path Variables:
  - customerId (Long): Identifier of the customer
- Query Parameters:
  - yearMonth (String): Year and month in "yyyy-MM" format
- Response: JSON representation of the reward points for the customer in the specified month and the total reward points.

- Description: Retrieves the reward points for the customer in the specified month and the total reward points earned.

Example:

For this case, we are trying to get the last three months' transaction reward points data for user with ID 3 with a date as 2023.02. So The points of Feb, 2023, Jan, 2023, and Dec, 2022 should be returned. As shown in the postman screenshot. The data has been successfully returned.

The screenshot displays a Postman interface for a GET request. The URL is `http://localhost:8080/api/v1/transactions/3/points?yearMonth=2023-02`. The 'Params' tab is active, showing a query parameter `yearMonth` with the value `2023-02`. The 'Body' tab is selected, showing a successful JSON response with a status of 200 OK, 116 ms, and 281 B. The response body is formatted as JSON and contains the following data:

```
1 {
2   "customerId": "3",
3   "name": "LeBron James",
4   "monthlyPoints": {
5     "2023-01": 50,
6     "2022-12": 450,
7     "2023-02": 402
8   },
9   "totalPoints": 902
10 }
```

For this second case, we are trying to get the last three months' transaction reward points data for user with ID 2 with a date as 2023.01. So The points of Jan, 2023, Dec, 2022, and Nov, 2022 should be returned. As shown in the postman screenshot. The data has been successfully returned. But since the user only spent \$50 in Dec, 2022, the reward points should be zero. And the user didn't spend any money in Jan, 2023, the reward points for Jan, 2023 is also zero.

The screenshot shows a Postman interface for a GET request to `http://localhost:8080/api/v1/transactions/2/points?yearMonth=2023-01`. The 'Query Params' section shows a single parameter: `yearMonth=2023-01`. The response body is displayed in JSON format, showing a successful 200 OK status with a 35 ms response time and 275 B of data.

```
{
  "customerId": "2",
  "name": "Mike Jordan",
  "monthlyPoints": {
    "2022-11": 20,
    "2023-01": 0,
    "2022-12": 0
  },
  "totalPoints": 20
}
```

## Error Handling

The Rewards Calculation System handles the following error scenarios:

- Invalid or missing input data when creating a transaction or customer, resulting in a Bad Request (400) response.
- Customer not found for the given customer id, resulting in a Not Found (404) response.
- Internal server errors, resulting in an Internal Server Error (500) response.

Please refer to the API documentation for detailed information on request/response formats and examples.