**Technical Documentation for RewardsServiceImpl Service Class**

**Overview**

The RewardsServiceImpl class implements the RewardsService interface and provides methods for managing reward points and creating reward responses based on customer transactions.

**Dependencies**

* **RewardsRepository:** Repository for managing reward entities.
* **TransactionRepository:** Repository for managing transaction entities.

**Methods**

**1. createRewardPoints**

**Description:** Calculates and sets reward points for all transactions.

**Method Signature:**

java

public void createRewardPoints()

**Implementation Details:**

* Fetches all transactions from the TransactionRepository.
* Iterates through each transaction and creates a Reward object based on the transaction details.
* Calculates reward points for each transaction using the computeRewardPoint method.
* Saves all the reward objects to the RewardsRepository.

**2. getRewardResponse**

**Description:** Creates a response containing the reward details for all customers.

**Method Signature:**

java

public ResponseEntity<List<RewardResponse>> getRewardResponse()

**Implementation Details:**

* Fetches all customer IDs from the TransactionRepository.
* For each customer ID:
  + Retrieves all transactions and rewards associated with the customer ID.
  + Creates a RewardResponse object for the customer using the createRewardResponse method.
* Returns a ResponseEntity containing the list of RewardResponse objects with an HTTP status of OK.

**Response Sample:**

json

[

{

"custId": "1",

"name": "Alex",

"total\_points": 250,

"rewardDetails": [

{

"month": "January",

"points": 150,

"amountSpent": 200,

"transactionAndPoints": [

{

"tranDate": "01/01/2023",

"amount": 200,

"points": 150

}

]

}

],

"totalAmountSpent": 200

},

{

"custId": "2",

"name": "Leo",

"total\_points": 300,

"rewardDetails": [

{

"month": "February",

"points": 300,

"amountSpent": 300,

"transactionAndPoints": [

{

"tranDate": "02/01/2023",

"amount": 300,

"points": 300

}

]

}

],

"totalAmountSpent": 300

}

]

**3. getRewardResponse (by customerId)**

**Description:** Creates a response containing the reward details for a specific customer by their ID.

**Method Signature:**

java

public ResponseEntity<RewardResponse> getRewardResponse(String custId)

**Implementation Details:**

* Retrieves all transactions and rewards associated with the given customer ID.
* Throws CustomerIdNotFoundException if no transactions or rewards are found for the customer ID.
* Creates a RewardResponse object for the customer using the createRewardResponse method.
* Returns a ResponseEntity containing the RewardResponse object with an HTTP status of OK.

**Response Sample:**

json

{

"custId": "1",

"name": "Alex",

"total\_points": 250,

"rewardDetails": [

{

"month": "January",

"points": 150,

"amountSpent": 200,

"transactionAndPoints": [

{

"tranDate": "01/01/2023",

"amount": 200,

"points": 150

}

]

}

],

"totalAmountSpent": 200

}

**Error Response Sample (404 Not Found):**

json

{

"message": "Customer Id non-existent not in DB",

"details": "Customer ID does not exist"

}

**4. createRewardResponse (Helper Method)**

**Description:** Creates a RewardResponse object for a specific customer based on their transactions and rewards.

**Method Signature:**

java

private RewardResponse createRewardResponse(String custId, List<Transaction> transactionsByCustId, List<Reward> rewardsByCustId)

**Implementation Details:**

* Initializes a RewardResponse object with the customer ID and name.
* Iterates through the transactions and rewards to create RewardDetail objects grouped by month.
* Aggregates the total points and total amount spent.
* Sorts the reward details by month.
* Sets the reward details, total points, and total amount spent in the RewardResponse object.
* Returns the RewardResponse object.

**5. monthSort (Helper Method)**

**Description:** Returns the sort order for a given month name.

**Method Signature:**

java

private int monthSort(String month)

**Implementation Details:**

* Uses a HashMap to map month names to their corresponding order.
* Returns the order for the given month name.

**6. computeRewardPoint**

**Description:** Computes the reward points based on the transaction amount.

**Method Signature:**

java

public int computeRewardPoint(int amount)

**Implementation Details:**

* Returns 0 if the amount is less than or equal to 50.
* Returns amount - 50 if the amount is between 51 and 100.
* Returns 2 \* (amount - 100) + 50 if the amount is greater than 100.

**Exception Handling**

* **CustomerIdNotFoundException:** Thrown when a customer ID is not found in the database.

**Summary**

The RewardsServiceImpl class provides methods for setting reward points based on transactions and generating reward responses for customers. It interacts with the RewardsRepository and TransactionRepository to manage and retrieve data. The class includes helper methods for creating reward responses, sorting by month, and computing reward points. The database is normalized to ensure data integrity and efficient data management.