1. **Nếu giao dịch được tổng hợp từ các hệ thống khác và định kỳ được tải lên hệ thống tích điểm, thì xử lý như thế nào trong trường hợp cấu hình quy đổi điểm bị thay đổi giữa các lần giao dịch của khách hàng.  
   VD: Khách hàng có 3 giao dịch mua hàng vào buổi sáng, và 2 giao dịch vào buổi chiều cùng ngày. Cấu hình quy đổi điểm được thay đổi vào 12h trưa ngày hôm đó. 22h cuối ngày thì dữ liệu mới được tổng hợp về hệ thống tích điểm.**

Phương án:

Khi thêm cấu hình quy đổi, hệ thống sẽ tạo bản ghi mới trong bảng lưu cấu hình. Khi tổng hợp giao dịch từ các hệ thống khác xuống, cần đối chiếu thời điểm tạo giao dịch với các cấu hình quy đổi, cấu hình áp dụng là cấu hình có thời gian tạo trước và gần nhất so với thời điểm tạo giao dịch.

Ví dụ: Thay đổi cấu hình quy đổi vào 12h, việc tổng hợp dữ liệu được thay đổi vào cuối ngày. Các giao dịch phát sinh trước 12h sẽ có giá trị thời gian tạo trước 12h, cấu hình được sử dụng là cấu hình có thời gian tạo phía trước và gấn nhất với thời điểm tạo giao dịch đó. Các giao dịch từ 12h trở đi sẽ sử dụng cấu hình thiết lập ở thời điểm 12h.

1. **Tăng hiệu năng hệ thống khi có rất nhiều giao dịch của nhiều khách hàng đồng thời thì có giải pháp nào không?**

Để tăng hiệu năng của hệ thống khi có rất nhiều giao dịch đồng thời, tôi xin đưa ra một vài ý kiến giải pháp:

* Nâng cấp hệ thống phần cứng server (tăng số lượng CPU, số lượng server, ổ cứng có tốc độ truy xuất nhanh, ram, …) để giảm thời gian truy vấn dữ liệu.
* Cân bằng tải cho hệ thống (LoadBalancer, CDN,..)
* Xử lý hiện tượng nghẽn cổ chai, Timeout,.. bằng việc xử lý Rate limiting
* Tối ưu hóa truy vấn ( Đánh Index bản ghi, Phân đoạn dữ liệu, …)
* Xử lý Cache dữ liệu để giảm truy vấn Server Database, nâng cao hiệu suất hệ thống.

…

Vấn đề tăng hiệu suất hệ thống, tối ưu tải dữ liệu còn phụ thộc vào rất nhiều yếu tố khác, trên đây là một số ý kiến giải pháp của tôi.

* **Hướng dẫn chạy từng ý của bài tập.**
* Database: SqlServer
* DbSriptFile: Loyalty.sql
* Tạo project và điền thông tin kết nối (servername, username, password) vào ConnectionString LoyaltyDb trong file Web.config
* Chạy project trên visual studio
* Sử dụng Postman để tạo request
* access\_token: Osr1F6FzEHHRkZTR1Tn6BL6jeWH4kN3X.

Trên thực tế việc lấy access\_token sẽ được thực hiện thông qua các phương thức bảo mật khác: Owin, JWT, OAuth..

*Yêu cầu 1: Xử lý chỉnh sửa cấu hình tích điểm lưu trữ vào database:*

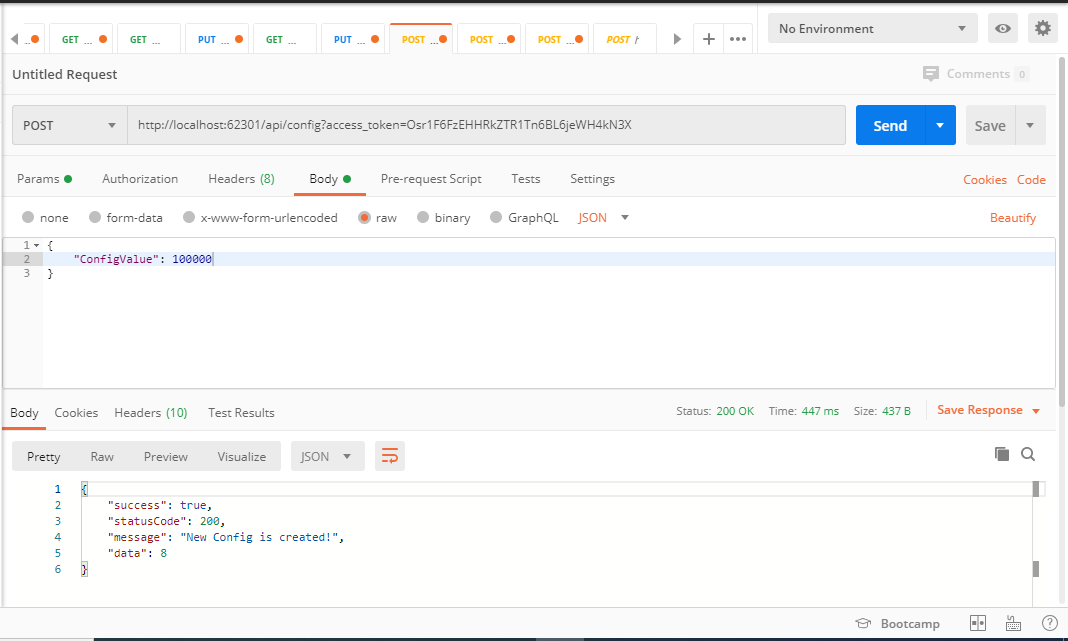
* + Endpoint: <http://localhost:62301/api/config?acess_token> = …
  + Method: POST
  + Data: đối tượng Json chứa giá trị quy đổi.

{

"ConfigValue": 100

}

Request thành công, bản ghi mới sẽ được tạo ra, bản ghi mới nhất tạo ra sẽ được đặt trạng thái Active=true, các cấu hình cũ có trạng thái Active = false.

Hình ảnh minh họa:

*Yêu cầu 2: Ghi nhận giao dịch mới*

Tạo yêu cầu post đến API Transaction theo đường dẫn:

* + Endpoint: <http://localhost:62301/api/transaction?acess_token> = …

Data: đối tượng json chứa thông tin giao dịch theo mấu:

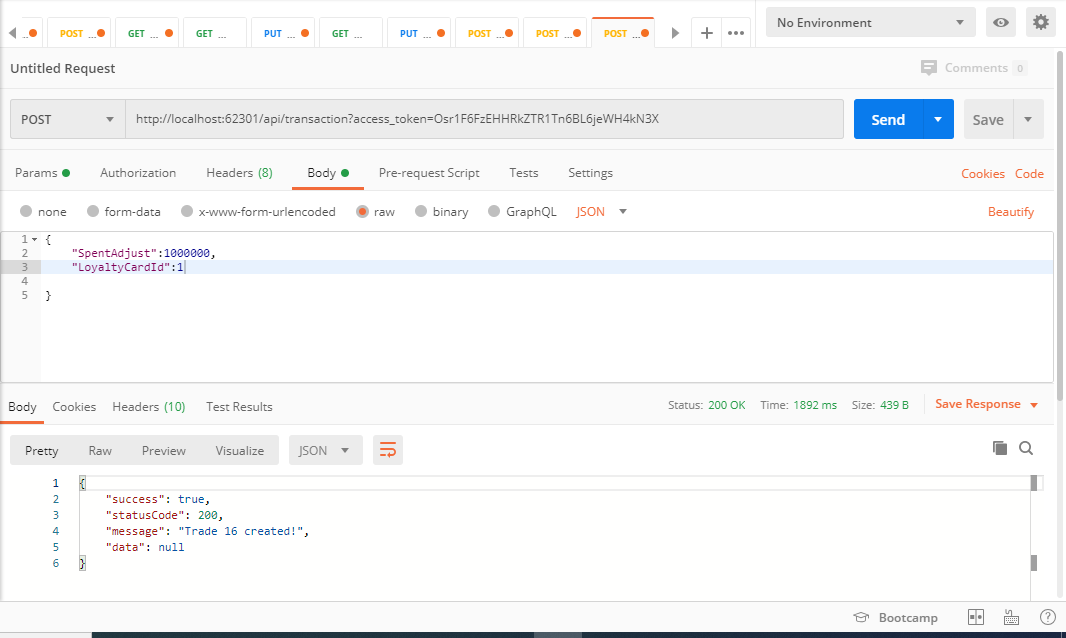
{

"SpentAdjust":1000000,

"LoyaltyCardId":1

}

Hình ảnh mình họa:



Request thành công, giao dịch mới sẽ được tạo ra, điểm từ giao dịch sẽ được quy đổi và cập nhật điểm, hạng, giá trị giao dịch cho thẻ tích điểm và lưu vào hệ database.