

Coding Challenge:

Problem:

A retailer offers a rewards program to its customers awarding points based on each recorded purchase as follows:

For every dollar spent over \$50 on the transaction, the customer receives one point.

In addition, for every dollar spent over \$100, the customer receives another point.

Ex: for a \$120 purchase, the customer receives

$(120 - 50) \times 1 + (120 - 100) \times 1 = 90$ points

Given a record of every transaction during a three-month period, calculate the reward points earned for each customer per month and total.

- **Make up a data set to best demonstrate your solution**
- **Check solution into GitHub and please share the GitHub URL.**

Write a REST API that calculates and returns the reward points in Java using Spring Boot as a microservice.

Additional Notes:

- All of the code should be provided in GitHub. Check-in all the code into master branch and squash and re-order all commits if necessary to have clean purposeful commit messages like you would normally in your job.
- Include a dataset in a table format that shows the customer list, transaction list, monthly total per customer, and 3 months summary it in, and add an image inside the README.md file with this data set and results.
- The client would be assessing your Coding Style and your personality when you join as a developer in their team, not just whether the code runs or logic is correct. So, please follow all of the Best practices that you know and have implemented in the past while doing this exercise.
- Please pay attention to the way you name your variables, classes, name of your project and package structure etc. Do not do silly mistakes like declaring variables but not using your variables anywhere in the code or using something from the Target folder.
- Externalize all the dependencies of your code nicely into property file or resource files in appropriate locations in package layout.
- Do NOT copy paste from GitHub, Stack Overflow or Leetcode. We run a plagiarism check and automatically reject all copy past jobs. Taking inspiration from something online is fine but blatant copy paste or renaming variables after copy pasting will be very easily flagged by our software. So, please do it on your own.
- You API should contain clear Documentation (a well formatted README.md and comments in code, sample Request Response formats, how to invoke your API by any client), a Health Check Endpoint to check whether the service is running or not by a monitoring tool.
- Write your own Unit Tests and provide instructions on How to run your Tests in your README.md file
- Show your ability to write well maintainable code by including custom Exception Handling (ability to identify problems within code easily by throwing custom Errors), custom Logging (different levels like Verbose etc)
- Additional brownie points if you can containerize your API with Dockerfile and provide instructions on How to run your dockerized API in REAMDE.md