Backend Engineer Code Assessment

Overview

You will be working with a pre-existing Spring Boot Application developed with Java 21 and leveraging the Temporal Workflow Engine for orchestrating business logic. Your main task involves integrating the Stripe payment processing service to manage customer data upon user signup.

Project Setup

Base Application: You are provided with a Spring Boot Application setup with Java 21.
External Service: The application uses the Temporal Workflow Engine to handle workflow tasks.
Task Focus: Your primary focus will be on integrating the Stripe service for customer creation and managing associated data.

Setup

The setup instruction is provided in the GitHub repository

Link: https://github.com/Midas-Labs/backend-engineer-assessment

Tasks

1. **Stripe Integration for Customer Creation using Temporal Workflow**
    - Upon a new user signup, integrate the Stripe Create Customer API to create a new customer in Stripe.
    - The Stripe SDK has already been added to the project's Gradle dependencies, and some boilerplate code is provided to get you started and everything is bootstrap ready except for the implementation.
    - You have to implement the required workflow to create the account using **Temporal.**

2. **Add the following fields in the APIs**
    - Add a new providerType field to the user model with an enum type with values: stripe
    - Add a providerId field to store the generated Stripe customer ID.

- Update the application controller to handle these new fields, ensuring that the providerId is generated and stored appropriately during the user signup process.

3. **API Implementation**
   - For convenience and testing purposes, a GET /accounts endpoint is already implemented within the codebase. You may use this endpoint to verify the integration and functionality of your changes.

4. **Bonus: Writing Tests**
   - As a bonus challenge, write tests for your implementation. This includes unit tests for the Stripe integration and the new fields in the user model, as well as integration tests that cover the signup process and the GET /accounts endpoint functionality.

Project Submission Guidelines

Code Quality: Your code should be clean, well-documented, and adhere to standard Java and Spring Boot best practices.
Testing: Include tests for all new functionality. Bonus points are awarded for comprehensive testing that ensures reliability and robustness.
Documentation: Provide a README file that includes:

- Setup instructions.
- How to run your tests.
- A brief explanation of your implementation approach and any assumptions made.

Evaluation Criteria

Functionality: Your submission should meet all the specified requirements and work as expected.
Code Quality: Your code should be well-organised, clean, and easy to read with proper usage of patterns and practices.
Testing: Your tests should cover critical paths and edge cases, ensuring the application works correctly under various conditions.
Documentation: Your documentation should provide clear setup instructions and insights into your development process.

We look forward to reviewing your project and seeing how you approach and solve these tasks. Good luck!