Day 2

? Day 2 ? Authentication, Exchange Rate Sync, RBAC, and Orders with Business Rules

Objective

Introduce authentication and role-based access control using Keycloak. Integrate with a third-party API to fetch exchange rates. Implement recurring background jobs using Hangfire. Extend product logic with RBAC and implement order creation that applies business rules.

Tasks

1. Integrate Authentication Using Keycloak

- Use the existing keycloak service in the provided Docker setup.
- Configure your Web API to use Bearer Token Authentication via Keycloak.
- Set up at least two roles:
  - Admin
  - User
- Protect endpoints so that:
  - Admins can perform full CRUD operations on products.
  - Users can only read products.

2. Integrate with the Open Exchange Rates API

- Create a scheduled job to fetch exchange rates from the Open Exchange Rates API (or mock it if needed).
- Store the data in a PostgreSQL ExchangeRates table:
  - Id (GUID)
  - USD (decimal)

Day 2

  - ZAR (decimal)
  - Date (timestamp)

3. Schedule Recurring Sync Using Hangfire

- Set up Hangfire in your project to run background jobs.
- Use a cron expression to schedule the exchange rate sync (e.g., every 2 hours).
- Persist the fetched exchange rates to the database.

4. Extend Product Endpoints with RBAC

- Ensure all product endpoints are protected appropriately:
  - GET /products and GET /products/{id}: available to all authenticated users.
  - POST, PUT, DELETE: restricted to Admin users only.

5. Implement Order CRUD With Business Rules

- Create the following Order structure:
  - Id (GUID)
  - UserId (string from token or database)
  - OrderDate (timestamp)
  - Products (list of product references)
- Implement business rules:
  - A user can only place an order if all products are in stock.
  - When an order is created, reduce the stock quantity accordingly.
  - Associate the current exchange rate for ZAR ? USD at the time of order.

Notes

Day 2

- Use FluentValidation to validate order inputs (e.g., product count > 0, no duplicate product IDs).
- Exchange rates should be retrieved from your local database for order calculation, not from the external API at runtime.
- Consider seeding test users and roles into Keycloak or provide setup instructions for them.