

Java Developer Recruitment Task

You will develop a simple foreign exchange application which is one of the most common services used in financial applications. Requirements are as follows:

Mandatory Functional Requirements:

Exchange Rate Endpoint:

- **Input:** A pair of currency codes (e.g., USD to EUR).
- **Output:** The current exchange rate between the two currencies.

Currency Conversion Endpoint:

- **Input:** An amount in the source currency, source currency code, and target currency code.
- **Output:** The converted amount in the target currency and a unique transaction identifier.

Conversion History Endpoint:

- **Input:** A transaction identifier or a transaction date for filtering purposes (at least one must be provided).
- **Output:** A paginated list of currency conversions filtered by the provided criteria.

External Exchange Rate Integration:

- The application must utilize an external service provider for fetching exchange rates and optionally for performing the currency conversion calculations.

Error Handling:

- Errors must be handled gracefully, providing meaningful error messages and specific error codes.

Mandatory Technical Requirements:

Self-Contained Application:

- The application should require no additional setup or configuration to run.

RESTful API Design:

- Develop the service endpoints following REST principles using Spring Boot.

Build & Dependency Management:

- Use appropriate tools for building the project and managing dependencies.

Use of Design Patterns:

- Implement design patterns appropriately to enhance code quality.

Code Structure:

- The code should be organized to reflect a clear separation of concerns.

Unit Testing:

- Include unit tests to ensure the reliability and robustness of the application.

API Documentation:

- Provide complete and accurate documentation for the API, including request and response examples.

Docker:

- Containerize the application with Docker to ensure consistency across different environments.

Proper Use of a Git Repository:

- Maintain the code in a Git repository with a clear history of commits, adhering to best practices for version control.

Optional Features:**API Documentation Tooling:**

- Utilize tools like Swagger or OpenAPI for generating interactive API documentation.

Caching:

- Apply caching strategies to improve performance, particularly for exchange rate data.

Hints and Suggestions:**FX Rate Service Providers:**

- You may select any service provider for exchange rates such as currencylayer.com or fixer.io, considering the limitations of their free offerings.

Spring Boot Resources:

- Refer to Spring Boot's official guides for quick reference and tutorials.

In-Memory Database:

- An in-memory database can be employed for demonstration purposes and ease of setup.

The detailed API documentation is crucial for users to understand how to interact with the service effectively. Ensure that the Git repository reflects a professional approach to software development with meaningful commit messages, feature branches, and pull requests where applicable.

Please send it back to us within 5-7 days max from receiving the email.

many thanks & Good Luck!

OpenPayd Talent Team