

Technical Hands-on Assessment

Money Changer

A businessman wants to get into money changer business. He will hire people to run each branch.

His business handles exchange from Singapore Dollars (SGD), to US Dollars (USD), Hong Kong Dollar (HKD), other currencies, and vice-versa. The exchange rates may be represented as:

Currency	Buy	Sell
USD	1.3392	1.3574
HKD	0.1738	0.1698

- Buy: The rate which the money changer accepts in foreign currency, and exchanges to SGD. For example, Customer gives **1 USD** and get 1.3392 SGD.
- Sell: The rate which the money changer accepts in SGD, and exchanges to foreign currency. For example, Customer gives 1.3574 SGD and get **1 USD**.

Each branch has a monitor to show rates of its chosen currencies. Each branch manager needs to

1. Answer customer enquiries, i.e. the exchange rate & the exchanged amount,
2. Record the currency note exchange transaction details.

Here's a typical scenario how the money changer would attend to his customer:

1. A customer looks at the rate displayed in the monitor and indicates that he would like to exchange SGD 200 to USD.
2. The money changer informs the customer that he will get USD 147 at the sell rate of 1.3574.
3. If the customer accepts the rate, the money changer will go ahead with the transaction.

The businessman intends to build a central backend to provide the services. Each branch will be running a frontend application and integrating with this central backend through internet.

All branches do not have small notes or cents for each foreign currency. So for a Sell transaction, the system help branch manager to provide customer the alternative SGD amounts result in the nearest multiple of the minimum currency unit available. The SGD amount should be rounded to nearest 10 cents. For example, a customer intends to exchange 200 SGD to HKD. He should get 1177.86 HKD at the sell rate of 0.1698. The branch manager could suggest to get 1200 HKD by exchanging 203.80 SGD (203.76 rounded to 203.80).

Your Tasks

- 1) Create a Console or Web API application that is able to fulfil the requirement with Java Spring Boot or C#

Things to note

- Handle all exceptions and return friendly error messages.
- Persistence can choose from file, in-memory DB or purely memory, as long as the entity structures are clear.
- Write Unit Test. No need to achieve high coverage if time does not allow.
- Enough logging for troubleshooting.
- Any additional feature that will help business grow.