

Coding exercise

Solution Developer

Introduction

Thank you for being interested to join our team here at SEB. This exercise will help us better understand your strengths as a developer and will provide a starting point.

Whilst we use a range of the libraries and frameworks to help us build great applications, in these exercises we are going to try and keep things a bit more basic. Partly, this is to save your time, partly it's so that we can focus on some core skills like mastery of the language, code structure, readability, testability, and ease of reasoning.

Description

The bank is producing a tool to convert currencies.

- The customer may convert amount from one currency to another.
The customer may choose from available currencies.
- Conversion rate may be displayed after selecting two currencies.
- Result may be displayed after conversion.
- If one of currencies is missing validation error may be shown.
- If no rates were fetched message may be shown.
- Currencies may be taken from https://www.ecb.europa.eu/stats/policy_and_exchange_rates/euro_reference_exchange_rates/html/index.en.html below on the page are links to the files (pdf, csv or xml), or any other legal & publicly available source.
- Currency rates may be fetched daily or frequency may be configurable from config file.
- Code may be covered with Unit tests
- Code must be uploaded to Github or Bitbucket

Description

Additional:

- Historical rates may be retrieved as well.
- Historical rates may be displayed.
- Conversion may be done by preferred history rate.

Guidance

All work must be your own. You will be expected to explain, discuss, and extend your code in the interview.

The purpose of this exercise is to help us understand how you design solutions, and your core C# skills. Often there isn't a single best answer so write your code how you see best and be prepared to discuss your choices in the interview.

You may use the latest version of C#, HTML, JavaScript, or preferably Angular for UI. You may use any testing library. Your code should have object oriented design and should be easy maintainable, extendable and configurable.