ledata

FactSet Digital Solutions GmbH

FACTSET DigitalSolutions

Application

2nd Round Programming Task

Currency calculator

For the second-round interview please prepare the following programming task and then present it to us.

The task is to implement a currency calculator. It will include a REST API as well as a modern Graphical User Interface.

The first part which is the API. It should be implemented in PHP Symfony (https://symfony.com) or Node.js and should have an endpoint which after receiving the base currency, the target currency and the amount that needs to be converted will return the converted value. Whilst implementing please keep in mind that the provided currencies and exchange rates might change dynamically and that it could be necessary to add new currencies and rates to the application. This means that additional endpoints to manage various CRUD (Create – Read – Update - Delete) operations for the above will be needed. Before users can make requests to your API some form of authentication and authorization would be needed. Please implement a simple mechanism that would achieve this. It is not mandatory that all endpoints should require authorization and you are free to decide which ones should and which should not.

Summary:

- Use PHP Symfony framework or Node.js.
- Create API endpoints for converting currencies plus performing CRUD operations on currencies and their exchange ratios.
- Add some simple form of Authentication / Authorization that will make some of the endpoints private. (e.g. The ones responsible for the CRUD operations.)

The second part which is the GUI (Graphical User Interface) should be implemented in either of the following three major front-end frameworks Vue.js, React or Angular. The interface should be simple and should leverage the endpoints created in the API to make it possible for the user to convert currencies as well as add or update additional or existing ones. It should also have a form for the user to log in to access the protected endpoints of the API.

Summary:

- Use Vue.js, React or Angular
- Create the user interface that the user will communicate with the API and the results will be presented.
- Data will come from the API

Below you will find some example currencies and exchange ratios that you could use as a starting point.

Euro -> US Dollar (1.3764)

Euro -> Swiss Franc (1.2079)

Euro -> British Pound (0.8731)

US Dollar -> JPY (76.7200)

Swiss Franc -> US Dollar (1.1379)

British Pound -> CAD (1.5648)

Do not forget to include any necessary instructions or documentations regarding setting up and running the application.

Please send us the source code via mail to Dimitrios Dimitriadis (Dimitrios.Dimitriadis@intragon.de).

Thank you.