

Bankdata Code Challenge

Purpose

The purpose of this assignment is for us to talk about code that you've written and designed yourself based on a set of requirements.

Expectations

We expect that you create a **private repository (Github, Bitbucket, Gitlab)** - where you host it does not matter, as long as you can send us a link, that allows us to clone the repo and run it locally.

It's important that you include a README file which describes how we start your application, together with a description of how to hit your endpoints (e.g. cURL).

Assignment

We do not expect you to spend days on this assignment. Feel free to leave rough edges as you see fit, we can talk about your decisions during the interview. Exception and error handling should be considered, but does not have to be fully implemented either.

Task 1 - Bank account system

Implement in Java using preferably Quarkus (<https://quarkus.io/>) or in a similar Java framework, a small rest API that allows the following:

1. Create a new Account
2. Deposit money to an Account
3. Transfer money between two Accounts
4. Get Balance for an Account

Account should have at least the following attributes:

- Account number
- Balance
- First name
- Last name

For dependency management you can use either Gradle or Maven, and for persisting data H2 (in-memory DB) could be an option.

Your API only has to support DKK currency.

Do not use any real world data, use only mock data!

Create tests as you see fit, are you comfortable making code changes with the level of testing you've included?

Task 2 - API integration with 3rd party

Create a new GET endpoint in your service, which can calculate the exchange rate between DKK and USD. The endpoint should return a JSON structure like so:

```
{ "DKK": 100,00, "USD": 14,61 }
```

You should *not* calculate the exchange rate internally in your service, but use a 3rd party provider with an API that can do it for you.

A potential provider could be <https://www.exchangerate-api.com/docs/pair-conversion-requests> - it's free.

Task 3 - Currency conversion frontend

Develop a simple Angular, React or Vue.js app which allows the user to calculate currency exchange rates using the backend developed in Task 2

Additional

If you know of any software design principles, take note when you implement them, you'll have your chance to shine during the interview.

Assessment

You'll be asked to present your solution to the interviewers where they'll be asking questions about your design decisions.