Dev Challenge (Short)

Assembling the codebase

1. Download the base starter Spring Boot instance:
    - https://start.spring.io/#!type=gradle-project&language=java&platformVersion=2.6.6&packaging=jar&jvmVersion=11&groupId=com.dws&artifactId=challenge&name=challenge&description=DWS%20Dev%20Challenge&packageName=com.dws.challenge&dependencies=lombok,web,validation
2. Merge the download from *start.spring.io* with our starter code attachment, `src.zip`

The task

We are providing you with a simple REST service with some very basic functionality - to add and read an account.

It is a standard gradle project, running on Spring Boot. It uses Lombok and if you've not come across it before you'll need to configure your IDE to use it (otherwise code will not compile).

Your task is to **add functionality for a transfer of money between accounts**. Transfers should be specified by providing:

- accountFrom id
- accountTo id
- amount to transfer between accounts

The amount to transfer should always be a positive number. It should not be possible for an account to end up with negative balance (we do not support overdrafts!)

Whenever a transfer is made, a notification should be sent to both account holders, with a message containing id of the other account and amount transferred.

- For this purpose please use the NotificationService interface
- Do NOT provide implementation for this service - it is assumed another colleague would implement it.
- Do not use the provided (simple) implementation in your tests - it is provided for the main application to run. In your tests you should mock this service.

This feature should be implemented in a thread-safe manner. Your solution should never deadlock, should never result in corrupted account state, and should work efficiently for multiple transfers happening at the same time.

Additional guidance

Please provide us with the code, preferably via a private GitHub repository shared with *dws-paul* and *dws-neil-mckeown*.

We think this task should take you about an hour - take the **simplest possible approach that works**. It's not meant to be a trick question, and our expectations are set accordingly to the time we expect you to spend on this.

Please treat this as an opportunity to showcase how you work, quality of what you provide will matter much more than pure quantity of code, or adding features we do not ask for. Clean, elegant and simple code wins over feature rich every time.

Due to limited time we expect you to spend on this, we are happy for you to focus only on the code and tests to deliver the feature. ***However, please provide a short document (a few bullet points will suffice) describing any extra work you would consider important to do before this project was turned into a production application*** - i.e. what would you improve/add, given more time.

Please focus on the application being "production-and-support-ready" for requirements already provided - not on extra features/functionality that could be added.