Technical Exercise - Payment Optimization

Problem Statement

We have bank branches in multiple countries and due to regulatory restrictions we can't make direct payments from all bank branches.

Currently, we have connectivity between the following branches:

A to B

A to C

C to B

B to D

C to E

D to E

E to D

D to F

E to F

Therefore to make a payment between A and D we could use the following sequence: A to B, B to D.

Branches charge to process payments - any payment that leaves a branch is subject to charge. The receipt of funds into a branch is free of charge. The cost for each branch to transfer funds is shown below:

Branch	Cost
A	5
B	50
C	10
D	10
E	20
F	5

Your task is to develop a solution that is able to compute the cheapest way to make a payment between two branches.

Requirements

1	Your application should make use of Java and Maven

2	Your solution should define and implement the following interface:
	 `public interface PaymentService {`
	 `/**`
	 `* Process a payment returning the cheapest sequence of branches as comma separated String. Implementations are expected`
	 `* to be thread safe.`
	 `* @param originBranch the starting branch`
	 `* @param destinationBranch the destination branch`
	 `* @returns the cheapest sequence for the payment as a CSV (e.g. A,D,C) or null if no sequence is available`
	 `**/`
	 `String processPayment(String originBranch, String destinationBranch);`
	 `}`

3	Your solution should expose a REST API which exposes the functionality provided by the above interface. Consideration should be given to input validation

4	Your solution should be flexible to easily handle future additions of new branches and links without modification
5	You should consider the performance of your implementation and demonstrate how the implementation would scale when new branches and links are added
6	Your solution should have sufficient and appropriate functional test coverage
7	Your solution should be thread safe such that the above processPayment method can be called by multiple threads concurrently
8	Your implementation is free to use libraries however an assessment should be made on the credibility of the library
9	Your implementation and any supporting documentation (if required) should make no reference to any real company names or individuals - everything should be fictitious
10	Your completed solution should be uploaded to a publically available repository on GitHub. Once the interview process is complete, the repository should be deleted.