**A small test to demonstrate a multithreaded payment API!**

This test is to develop a small program in C# that exposes a REST endpoint with two methods. The program should simulate initiation of a payment where one method initiates the payment, and the second method returns the complete transaction of any payments made. The program should simulate that a payment takes two seconds to complete from initiation to the transaction is available.

Please note that the program must support multithreading and safely handle many unique parallel payments.

*<u>Method – Initiate Payment</u>*

POST /payments

This method is called to start a unique payment. The method requires HTTP header Client-ID that contains a unique ID for each client. The method returns a HTTP status code “Created” for a successful initiated payment. The response contains a unique, server-side generated, Payment-ID.

The method should guarantee that max one payment initiation is processing at any given time for each Client-ID. Consequently, the method should return HTTP status code “Conflict” if the method is called for a Client-ID that already has a payment processing (e.g. less than two seconds).

Request parameters:

| Name | Type | Comment |
|---|---|---|
| Debtor Account | IBAN (String up to 34 alphanumeric characters) | From account |
| Creditor Account | IBAN (String up to 34 alphanumeric characters) | To account |
| Instructed Amount | -?[0-9]{1,14}(\.[0-9]{1,3})? | |
| Currency | ISO 4217 Alpha 3 currency code | |

*<u>Method – Get Transactions</u>*

GET /accounts/{iban}/transactions

This method returns transactions (payments) for a given account number (iban) as soon as they are completed (e.g. should include all payments initiated more than two seconds ago on the account). If no payment has been completed (e.g. less than two seconds after initiation), the method should return HTTP status code “No Content”.

Response parameters:

| Name | Type | Comment |
|---|---|---|
| Payment-ID | String | Unique identifier for the payment |
| Debtor Account | IBAN (String up to 34 alphanumeric characters) | From account |
| Creditor Account | IBAN (String up to 34 alphanumeric characters) | To account |
| Transaction Amount | -?[0-9]{1,14}(\.[0-9]{1,3})? | |
| Currency | ISO 4217 Alpha 3 currency code | |

The program should be developed in .NET Core’s standard libraries, and no third-party libraries are allowed. No persistent storage is required.

The best of luck!