

# .Net Core Test

## Introduction

Thank you for your interest in Strategic Financial Solutions! 😊

This test will help us gauge your expertise with .Net Core



Feel free to use any data storage or libraries you are comfortable with as long as the required script or schema is described in the documentation. For example, in-memory store, Mongo (mongoose), SQL Server etc.

## Requirements:

Please design and implement two integrated web services as follows:

### Service #1

- Given that an HTTP GET request includes the following parameters, return the corresponding credit report data
  - Parameters
    - applicationId
    - source
    - bureau



Bonus Credit: Make it as an Azure Function

Initial test data can be found here:

<https://raw.githubusercontent.com/StrategicFS/Recruitment/master/creditData.json>

Credit data model:

```
"applicationId": number,  
"customerName": string,  
"source": string,  
"bureau": string,  
"minPaymentPercentage": number,  
"tradelines": array
```

Trade line model:

```
"tradelineId": number,  
"accountNumber": string,  
"balance": number,  
"monthlyPayment": number,  
"type": string,  
"isMortgage": bool
```

## Service #2

- Given an HTTP GET request that includes an application id and annual income, return a result that includes:
  - Number of unsecured trade lines
  - Total unsecured debt balance
  - DTI (Debt-To-Income)
- Further details:
  - Service #2 should consume the credit data from Service #1
  - The service will always use 'ABC' as the source and 'EFX' for bureau. It will not support other sources, bureaus
  - The service should retrieve the trade lines for the given application id
  - Two calculations must be performed:
    - DTI (Debt-To-Income)
      - Formula: Total the monthly payment for all secured and unsecured trade lines that are not mortgages. Divide by the monthly income
    - Total unsecured debt balance

- Total the debt balances for trade lines where the type is “UNSECURED”

## Follow-Up Question (please discuss in your submission video)

Given the following scenario, what design approach would you recommend?

- The data source by which Service #1 connects to for retrieving the trade lines takes approximately 5 seconds to return a response
  - Note that in the original exercise, you read the source data from a JSON file but in reality you would most likely be retrieving data from a credit bureau or third party via web service calls.
- The data source only allows up to two concurrent requests
- Service #2 must be enhanced to support a single bulk request of up to 5000
  - Assume there is flexibility with the consumer of Service #2. For example, the consumer could fire a single HTTP POST containing all 5000 request and receive the results later/asynchronously. For the purposes of this exercise you can assume that as the tech lead you will advise the consumer on the best architectural approach.

## Upon Completion

i. Please provide documentation on the structure of the application and provide details about how to run and test the application step by step.

Once you finish your coding challenge, please

1. Create a repository on **github** and make sure it is **public** so that we can review your code.
2. Create a 10 minutes screenshare video showing us your code on the left and it running locally on the right, while talking through the below questions. For example:
  - i. How did you design the app.
  - ii. How did you store and retrieve data?
  - iii. What did you find challenging? Please explain.
3. Finally, after finishing steps, please shoot an email to [pharris@stratfs.com](mailto:pharris@stratfs.com) with the github repo link.

***Happy coding!!***