Coding Exercise

.NET Developer

2016-06-21

Contents

Contents

Introduction

Thank you for wanting join our team here at SEB. We've developed this coding exercise to help us understand a little better your strengths as a developer, and to provide a great starting point.

Whilst we use a range of the latest libraries and frameworks to help us build great applications, in these exercises we are going to try and keep things a bit more basic, just C# and ASP.NET Web API. Partly, this is to save your time, partly it's so that we can focus on some core skills, skills that don't change as often as the libraries and frameworks do. Skills like mastery of the language, code structure, readability, testability, and ease of reasoning.

We expect that this exercise will take between 2-3 hours to complete.

Contents

- Introduction
- **Exercise**

Problem Description

The bank is producing a tool to recommend a 'bundle' of products to prospective customers.

The customer may then customise this bundle further, by adding, removing, or upgrading products.

There are rules which govern what products a customer may choose, based upon:

1. Answers to questions that the customer has given

2. Other products that they have selected

Additional Rules

A customer may only have one account (eg not Current Account and Pensioner Account)

Where a customer is eligible for more than one bundle then they must be offered the highest value bundle (where 3 is the highest value, and 0 is the lowest).

Question	Possible Answers
Age	0-17, 18-64, 65+
Student	Yes, No
Income	0, 1-12000,12001-40000,40001+

Product	Rules
Current Account	Income > 0 & Age > 17
Current Account Plus	Income > 40000 & Age > 17
Junior Saver Account	Age < 18
Student Account	Student = Yes & Age > 17
Debit Card	Bundle must include one of: Current Account, Current Account Plus, Student Account, or Pensioner Account
Credit Card	Income > 12000 & Age > 17
Gold Credit Card	Income > 40000 & Age > 17

Bundle	Products Included	Rules	Value
Junior Saver	Junior Saver Account	Age < 18	0
Student	Student Account, Debit Card, Credit Card	Age > 17 & Student = Yes	0
Classic	Current Account, Debit Card	Age > 17 & Income > 0	1
Classic Plus	Current Account, Debit Card, Credit Card	Income > 12000 & Age > 17	2
Gold	Current Account Plus, Debit Card, Gold Credit Card	Income > 40000 & Age > 17	3

Guidance

Write a unit tested ASP.NET Web API application with Restful API that:

1. Given answers to the questions, returns a 201 Created together with a JSON representation of the recommended bundle
2. Given a representation of the bundle and the answers to the questions validates the bundle and returns a 200 OK if valid, or an appropriate HTTP response if it is not that explains why. All of the rules for product availability must be respected.

All work must be your own. You will be expected to explain, discuss, and extend your code in the interview.

The purpose of this exercise is to help us understand how you design solutions. Often there isn't a single best answer so write your code how you see best and be prepared to discuss your choices in the interview.

You may use the latest versions of C# and ASP.NET Web API. You may use any testing library. You may embed any application server, but one should be embedded.

Persistence is required for this solution (use MS SQL).

Prepare documentation for your solution.