Senior Software Engineer (12-month contract) – assessment

Situation

You have been assigned a customer whose legacy system requires a solution to enable them to send their customers notifications through various communication channels. These channels can include any one of the following: SMS, Email, HTTP, MS Queue, FTP, SFTP, etc.

There are currently thousands of customers in the system which has been configured with multiple channels of communication, each with it's accepted format of data. The only thing which is pre-defined is the message types, the field names, and their data types (Refer to the **Message structure** below).

Background on the task

The challenge is to look for a solution that supports all existing (and new /future integrations) partied with ***plug and play formatting*** services. Note that there are countless customers, with different preferences in their communication channel (handled by MessagingService and provided in the **attached customer notification** zip file).

For example, one customer may accept a message via SMS, whose body is of type JSON, whereas another customer may accept XML over an HTTP call or CSV over FTP, etc.

Message structure:

The structure is simple, it is a string format that has a specified Type for the message and Data which contains the various fields we need for each message.

As an example, in json format our data may be presented as follow

```
{
   "messageType": "UserDeleted",
   "data": {
            "customerId": "9f9b1a81-2f94-44b7-994d-50cb60738f93"
    }
}
```

While the same message can be presented in the CSV format as

```
"UserDeleted","9f9b1a81-2f94-44b7-994d-50cb60738f93"
```

Message types and content:

All messages should contain the following information –

- A message type that indicates the context of the notification. It is expected that the API should support the following Types
    - NewUserRegistered
    - UserDeleted
    - UserBlocked
- A message body that may contain different information depends on the message type.

The following list illustrates the potential content of these messages

1. **Message when a user registered**

    Type: NewUserRegistered

    Body type:

string UserId

string Email

string Firstname

string LastName

2. **Message when a user deleted**

Type: UserDeleted
Body type:
string UserId

3. **Message when a user blocked**

Type: UserAccessBlocked
Body type:
string UserId

The Task to complete

1. Complete the attached Web API customer notification solution, using dotnet Core and C#
   - Implement the REST API to receive http calls with the right payload.
   - Design the API schema.
   - Generate the customer's desired format (refer to Task 1) and then send the message to them using the mock "MessagingService".
   - When the implementation is done please upload the solution (with document) to a github repository and share the repository details with us via the provided email address.

2. Using OOP design patterns and software best practices, design and Implement functionality using C# to support at least two different message formats of choice for some fictitious customers.

Things to note

- Nothing needs to be implemented for the messaging service, so please use the provided MessagingService class as it is (can be found in the attached solution code). This service has only one public method which accepts a CustomerID and a string for message body (data passed to the message body parameter has to be already as the accepted format of the corresponding customer id)
- Do not worry about Security or Authorization / Authentication
- You can choose your own message format (e.g. JSON, CSV, XML) to implement.

- Please pay extra attention to make the solution easy to understand and test.
- You may also want to provide a README document in MD format to help us understand your solution.

Please feel free to contact Kia Geraeli, Senior Software Engineer if you have any questions or require any support: Kia.Geraeli@ExportFinance.gov.au or call +61 474 068 262. Good Luck!