Service Requests: Case Study

Background

Product team has decided to develop a service request module for our clients. Their research shows that the users are interested in submitting a maintenance service request to the building staff.

This module will allow our product to provide a functionality to improve tenant experience in the building while improving communication between building staff and the end users.

The candidate is asked to develop CRUD operations for this module to get started.

Rules of the game

1. The candidate is expected to spend no more than **3 hours** on this exercise.
2. We will review and discuss the submitted solution during on-site technical deep-dive interview with the candidate
3. The candidate is allowed (and encouraged) to ask clarifying questions before they start the challenge.
4. The candidate can use in-memory list, in-memory database, file-as-a-database, or hosted databases to mimic database behavior
5. The candidate is encouraged to commit their code as frequently as possible to showcase their thought process in code.
6. The candidate does not have to have to complete all requirements to submit their code-challenge if they run out of time. Our team will review un-finished code without any penalties.

Current Requirements

- Create Service Request WebApi
- Add CRUD operations for service request module (data model provided below)
- **Bonus**: Send notification (e.g., email, text) to the user when Service request is closed.

Evaluation criteria

- Understanding of problem
- Use of Software fundamentals in the solution
    - OOP concepts
    - Dependency injection
    - Tests (i.e., Unit test, API Test)
- Clean code (i.e., readable and easy to maintain)
- Scalability considerations

Data model

Service Request model

```
{
```

```
    "id" : "guid",
    "buildingCode": "string",
    "description" : "string",
    "currentStatus" : "enum",
    "createdBy": "string",
    "createdDate" : "dateTime",
    "lastModifiedBy" : "string",
    "lastModifiedDate" : "dateTime"
}
```

CurrentStatus Enum

```
public enum CurrentStatus {
    NotApplicable,
    Created,
    InProgress,
    Complete,
    Canceled
}
```

Example

```
{
    "id" : "727b376b-79ae-498e-9cff-a9f51b848ea4",
    "buildingCode": "COH",
    "description" : "Please turn up the AC in suite 1200D. It is too hot
here.",
    "currentStatus" : "Created",
    "createdBy": "Nik Patel",
    "createdDate" : "2019-08-01T14:25:43.511Z",
    "lastModifiedBy" : "Jane Doe",
    "lastModifiedDate" : "2019-08-01T15:01:23.511Z"
}
```

WebApi end-points

- **GET**
  - `description`: Read all service requests
  - `route`: **api/servicerequest**
  - `Response`
    - `200`: list of service requests
    - `204`: empty content
- **GET** :
  - `description`: Read service request by `id`.

    - `route`: **api/servicerequest/{id}**
    - `Response`
        - `200`: single service request
        - `404`: not found
- **POST** :
    - `description`: Create new service request
    - `route`: **api/servicerequest**
    - `Response`
        - `201`: created service request with id
        - `400`: bad request
- **PUT**
    - `description`: update service request based on `id`
    - `route`: **api/servicerequest/{id}**
    - `Response`
        - `200`: updated service request
        - `400`: bad service request
        - `404`: not found
- **DELETE**
    - `description`: delete service request based on `id`
    - `route`: **api/servicerequest/{id}**
    - `Response`
        - `201`: successful
        - `404`: not found

Instructions to submit code repository

Option 1

1. Please create a `GitHub`/`BitBucket` repository and commit your code there.
2. Send link to the repository and allow read access (if private)

Options 2

1. Compress your code in `.zip` file
2. Send zip file with instructions to setup code

Best of luck!