Test – Backend Developer

Należy napisać prostą aplikację w języku C# (.NET Framework lub .NET Core) przechowującą w wybranej bazie danych informacje o firmach łącznie z listą pracowników oraz z możliwością zarządzania poprzez RESTful Web API. Usługa powinna się uruchamiać w trybie self-host czyli jako aplikacja konsolowa i nasłuchiwać na dowolnym porcie. W projekcie należy zastosować mapowanie obiektowo-relacyjne (ORM) dla bazy danych firm – preferowany framework to NHibernate / Fluent NHibernate ale wedle preferencji można użyć Entity Framework lub innego, również własnego rozwiązania.

Dane należy przechować w dwóch tabelach z 64-bitowym kluczem głównym (ID), połączonych ze sobą relacją jeden do wielu. Kolumny powinny posiadać odpowiedni typ danych w zależności od rodzaju przechowywanego pola. Pojedyncza encja składa się z nazwy firmy, roku założenia oraz listy pracowników: imię, nazwisko, data urodzenia, stanowisko. W odzwierciedleniu obiektowym stanowisko powinno być typem wyliczeniowym (enum) przechowywanym w postaci nazwy tego typu. Możliwe typy stanowisk: *Administrator*, *Developer*, *Architect*, *Manager*.

Aplikacja musi obsługiwać następujące akcje HTTP:

1. POST: /company/create

```
{
  "Name": "<string>",
  "EstablishmentYear”: <integer>,
  "Employees": [{
    "FirstName": "<string>",
    "LastName": "<string>",
    "DateOfBirth": "<DateTime>",
    "JobTitle": "<string(enum)>"
  }, …]
}
```

Odpowiedź:

```
{
  "Id”: <long>
}
```

Tworzy nowy wpis z kompletem informacji oraz zwraca jego wygenerowany ID. Wszystkie pola są wymagane (!NULL), lista może być pusta.

Pumox GmbH
John-F.-Kennedy-Str. 5
34128 Kassel

Tel.: +49 561 4739533-0
info@pumox.com
www.pumox.com

IBAN:DE87520400210311048300
BIC: COBADEFFXXX
Ust-IdNr.: DE304314815

2. POST: /company/search

```
{
  "Keyword": "<string>",
  "EmployeeDateOfBirthFrom": "<DateTime?>",
  "EmployeeDateOfBirthTo": "<DateTime?>",
  "EmployeeJobTitles": ["<string(enum)>", …]
}
```

Odpowiedź:

```
{
  "Results": [{
    "Name": "<string>",
    "EstablishmentYear": <integer>,
    "Employees": [{
      "FirstName": "<string>",
      "LastName": "<string>",
      "DateOfBirth": "<DateTime>",
      "JobTitle": "<string(enum)>"
    }, …]
  }, …]
}
```

Zwraca listę wszystkich wpisów, które spełniają dane kryteria. Każde z pól zapytania jest opcjonalne i wartość NULL powinna skutkować pominięciem tego pola w procesie filtrowania. Pole „Keyword" jest frazą (operator SQL „LIKE") dla nazwy firmy, imienia oraz nazwiska pracownika. Jeśli fraza występuje w jednym z tych pól, firma powinna być obecna na liście wynikowej. Pola „EmployeeDateOfBirthFrom" oraz „EmployeeDateOfBirthTo" określają zakres daty urodzenia dowolnego pracownika w firmie. Podobnie „EmployeeJobTitles" – wystarczy aby jeden pracownik posiadał stanowisko obecne na liście.

3. PUT: /company/update/<id>

```
{
  "Name": "<string>",
  "EstablishmentYear": <integer>,
  "Employees": [{
    "FirstName": "<string>",
    "LastName": "<string>",
    "DateOfBirth": "<DateTime>",
    "JobTitle": "<string(enum)>"
  }, …]
}
```

Aktualizuje komplet informacji dla istniejącego wpisu o podanym ID. Wszystkie pola są wymagane (!NULL), lista może być pusta.

Pumox GmbH
John-F.-Kennedy-Str. 5
34128 Kassel
Tel.: +49 561 4739533-0
info@pumox.com
www.pumox.com
IBAN:DE87520400210311048300
BIC: COBADEFFXXX
Ust-IdNr.: DE304314815

4. DELETE: /company/delete/<id>

Usuwa wpis o podanym ID z bazy danych.

Aplikacja powinna być odporna na przesłanie brakujących lub nieprawidłowych danych w zapytaniach i odpowiadać odpowiednim statusem HTTP wraz z informacją o powodzie błędu. Wszystkie akcje oprócz wyszukiwania (3.) muszą być chronione metodą „Basic Authentication" czyli dodatkowym nagłówkiem HTTP zawierającym zakodowanego w Base64 użytkownika i hasło. Serwer musi odpowiedzieć błędem 401 – Unauthorized jeśli zapytanie nie posiada takiego nagłówka lub jest niepoprawny. Dla uproszczenia użytkownik i hasło mogą być zdefiniowane na stałe w kodzie.

W realizacji istotne jest optymalne wykorzystanie gotowych rozwiązań czyli integracja z .NET Framework / .NET Core, a także jakość i przejrzystość samego kodu (zgodnie z: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions).

Do kodu źródłowego w postaci solucji Visual Studio należy dołączyć skrypt SQL inicjujący schemat bazy danych.

Pumox GmbH
John-F.-Kennedy-Str. 5
34128 Kassel

Tel.: +49 561 4739533-0
info@pumox.com
www.pumox.com

IBAN:DE87520400210311048300
BIC: COBADEFFXXX
Ust-IdNr.: DE304314815