

Dokumentacja interfejsu usługi danych doktora

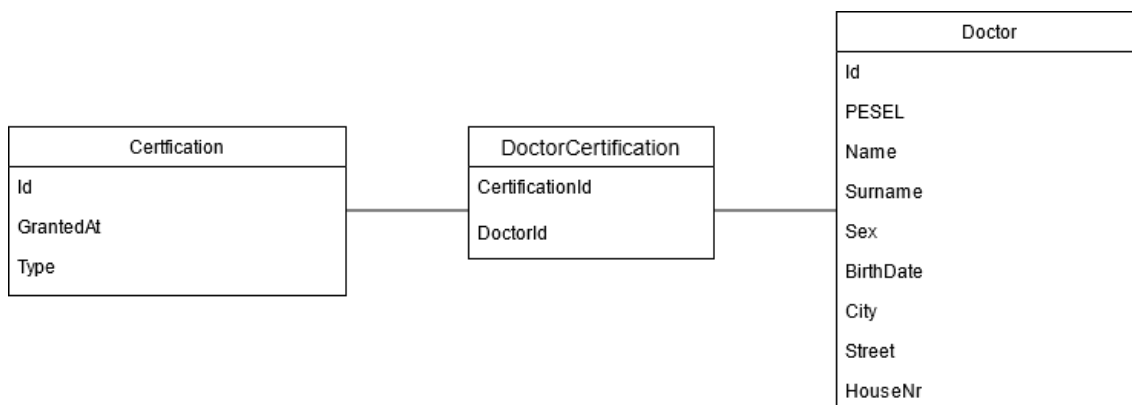
Olszewski Jakub, Wierzejska Izabela

Kwiecień 2021

Spis treści


1	Diagram encji	1
2	Formalny model danych	2
3	Przykładowe dane	3
4	Omówienie operacji usługowych	6
5	Struktura rozwiązania	7

1 Diagram encji




2 Formalny model danych



Model danych dla encji Doctor:

	Id	int
	PESEL	nvarchar(MAX)
	Name	nvarchar(MAX)
	Surname	nvarchar(MAX)
	Sex	nvarchar(MAX)
	BirthDate	datetime2(7)
	City	nvarchar(MAX)
	Street	nvarchar(MAX)
	HouseNr	nvarchar(MAX)

Model danych dla encji Certification:

	Id	int
	GrantedAt	datetime2(7)
	Type	int

Model danych dla encji DoctorCertification:

	CertificationId	int
	DoctorId	int

3 Przykładowe dane

Przykładowe dane dla encji Doctor:

```
{
  "Id": "9",
  "PESEL": "10801509300",
  "Name": "Noble",
  "Surname": "Sweeney",
  "Sex": "M",
  "BirthDate": "1969-12-31T04:12:00.0000000",
  "City": "Biarritz",
  "Street": "2123 Mauris, Rd.",
  "HouseNr": "62"
},
{
  "Id": "10",
  "PESEL": "47997383838",
  "Name": "Iris",
  "Surname": "Patton",
  "Sex": "M",
  "BirthDate": "1969-12-31T04:12:00.0000000",
  "City": "Mayerthorpe",
  "Street": "P.O. Box 899, 6608 Aliquam St.",
  "HouseNr": "133"
},
{
  "Id": "11",
  "PESEL": "27880768509",
  "Name": "Cheryl",
  "Surname": "Vargas",
  "Sex": "F",
  "BirthDate": "1969-12-31T04:12:00.0000000",
  "City": "Radicofani",
  "Street": "P.O. Box 284, 6990 Lacus. Rd.",
  "HouseNr": "38"
},
{
  "Id": "12",
  "PESEL": "49239723988",
  "Name": "Galena",
  "Surname": "Bird",
  "Sex": "M",
  "BirthDate": "1969-12-31T04:12:00.0000000",
  "City": "Orilla",
  "Street": "Ap #632-5836 Dolor. Ave",
  "HouseNr": "72"
},
{
  "Id": "13",
  "PESEL": "10526886186",
  "Name": "Kasimir",
  "Surname": "Bowman",
  "Sex": "M",
  "BirthDate": "1969-12-31T04:12:00.0000000",
  "City": "Shahjahanpur",
  "Street": "Ap #114-180 Sed Street",
  "HouseNr": "88"
}
```

Przykładowe dane dla encji Certification:

```
{
  "Id": "34",
  "GrantedAt": "2017-03-16T12:03:00.0000000",
  "Type": "10"
},
{
  "Id": "35",
  "GrantedAt": "2010-07-16T11:07:00.0000000",
  "Type": "6"
},
{
  "Id": "36",
  "GrantedAt": "2004-10-03T04:10:00.0000000",
  "Type": "7"
},
{
  "Id": "37",
  "GrantedAt": "1984-09-17T04:09:00.0000000",
  "Type": "2"
},
{
  "Id": "38",
  "GrantedAt": "1971-11-29T08:11:00.0000000",
  "Type": "9"
},
{
  "Id": "39",
  "GrantedAt": "1980-09-08T06:09:00.0000000",
  "Type": "8"
},
{
  "Id": "40",
  "GrantedAt": "2013-09-01T06:09:00.0000000",
  "Type": "10"
},
{
  "Id": "41",
  "GrantedAt": "2008-12-22T11:12:00.0000000",
  "Type": "3"
},
{
  "Id": "42",
  "GrantedAt": "2013-10-22T05:10:00.0000000",
  "Type": "1"
},
{
  "Id": "43",
  "GrantedAt": "1970-05-01T01:05:00.0000000",
  "Type": "8"
},
{
  "Id": "44",
  "GrantedAt": "1971-05-10T10:05:00.0000000",
  "Type": "4"
},
{
  "Id": "45",
  "GrantedAt": "1977-12-05T05:12:00.0000000",
  "Type": "2"
}
```

Przykładowe dane dla encji DoctorCertification:

```
{
  "CertificationId": "70",
  "DoctorId": "9"
},
{
  "CertificationId": "196",
  "DoctorId": "9"
},
{
  "CertificationId": "237",
  "DoctorId": "9"
},
{
  "CertificationId": "252",
  "DoctorId": "9"
},
{
  "CertificationId": "330",
  "DoctorId": "9"
},
{
  "CertificationId": "397",
  "DoctorId": "9"
},
{
  "CertificationId": "88",
  "DoctorId": "10"
},
{
  "CertificationId": "132",
  "DoctorId": "10"
},
{
  "CertificationId": "163",
  "DoctorId": "10"
},
{
  "CertificationId": "247",
  "DoctorId": "10"
},
{
  "CertificationId": "526",
  "DoctorId": "10"
},
{
  "CertificationId": "99",
  "DoctorId": "11"
},
{
  "CertificationId": "180",
  "DoctorId": "11"
},
{
  "CertificationId": "193",
  "DoctorId": "11"
},
{
  "CertificationId": "417",
  "DoctorId": "11"
}
```

4 Omówienie operacji usługowych

- Zapytania GET:
 - "doctors" - Nie przyjmuje żadnych parametrów, zwraca listę wszystkich doktorów
 - "getDoctorBySpecializations" - Jako parametr przyjmuje typ specjalizacji, zwraca listę wszystkich doktorów z daną specjalizacją
 - "getDoctorById" - Jako parametr przyjmuje identyfikator doktora, zwraca doktora z podanym identyfikatorem
- Zapytania POST:
 - "doctor" - Jako parametr przyjmuje określone atrybuty doktora, którego chcemy dodać do bazy danych; brak odpowiedzi zwrotnej.
 - "doctor-delete" - Jako parametr przyjmuje identyfikator doktora, którego chcemy z bazy danych usunąć; brak odpowiedzi zwrotnej.

5 Struktura rozwiązania

Solution Doctors

- Doctors.Domain
 - DoctorAggregate
 - * Certification.cs
 - * Doctor.cs
- Doctors.Infrastructure
 - Repositories
 - * IDoctorsRepository.cs
 - * DoctorsRepository.cs
 - Constants.cs
- Doctors.Web
 - Application
 - * Commands
 - AddDoctorCommand.cs
 - DeleteDoctorCommand.cs
 - DoctorsCommandHandler.cs
 - ICommand.cs
 - ICommandHandler.cs
 - * Dtos
 - DoctorDto.cs
 - * Mapper
 - Mapper.cs
 - * Queries
 - DoctorQueriesHandler.cs
 - IDoctorQueriesHandler.cs
 - Controllers
 - * DoctorsController.cs
 - Program.cs
 - Startup.cs

Solution Doctors dzieli się na trzy projekty: Domain, Infrastructure i Web.

Projekt Domain określa struktury danych wykorzystywane w mikrousłudze. W folderze DoctorAggregate zdefiniowane zostały dwie klasy publiczne Doctor i Certification - wyżej opisane struktury danych.

W projekcie Infrastructure zostały zdefiniowane polecenia, operacje na bazie danych. W klasie DoctorsRepository określamy metody które zwracają przez określone zapytania SQL wyselekcjonowane, filtrowane dane albo przeprowadzają na tych danych pewne działania (np. doctor delete). Dodatkowo w klasie Constants mamy zdefiniowany connectionString. Projekt korzysta z zdefiniowanych w Doctors.Domain struktur danych.

W projekcie Web znajdziemy logikę operacji i klasy odpowiadające za implementację klienta mikrousługi. W folderze Application/Commands zostały zdefiniowane templates komend. Ich logika rozbudowana została w Handlerze znajdującym się w DoctorsCommandHandler, który definiuje metody dla wcześniej utworzonych komend wykorzystując metody z klasy DoctorsRepository. W folderze Controllers zdefiniowany został kontroler API.

Projekt wykorzystuje oba pozostałe projekty: Domain i Infrastructure.