

Adam Burzyński
INF, SSI, IDSI 2

Projekt z przedmiotu **Platforma .NET**

Temat: Przychodnia

1. Technologie:

Projekt został wykonany w technologii ASP.NET. Wykorzystano w nim biblioteki takie jak Entity Framework, Newtonsoft JSON czy pakiet Swashbuckle.

Projekt, ze względu na podział na warstwy aplikacji:

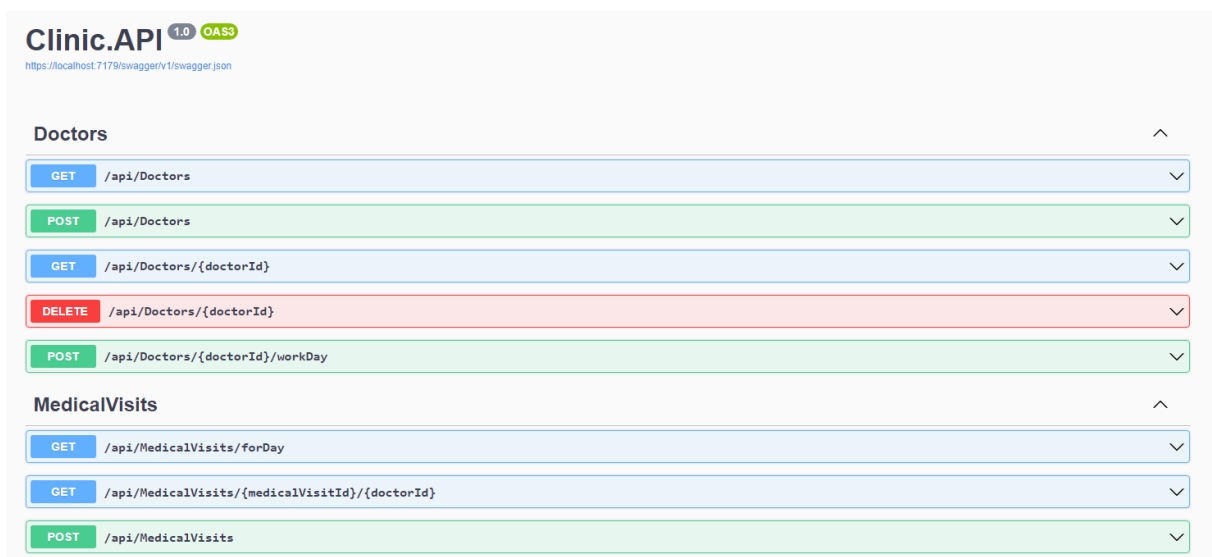
- API
- Contracts
- Domain

Można w łatwy sposób rozszerzyć, używając dodatkowych technologii takich jak AutoMapper.

API oparto o styl REST API.

W projekcie wykorzystano wzorzec architektoniczny CQRS (Command Query Responsibility Segregation) z wykluczeniem rozbicia bazy danych na bazę do odczytu i zapisu – wszystko zapisywane i odczytywane jest z tej samej lokalnej bazy danych SQL Server, z wykorzystaniem Entity Framework.

2. Interfejs



Projekt wykorzystuje interfejs biblioteki Swashbuckle.Swagger, dzięki której korzystanie z funkcjonalności API jest prostsze – bardziej przejrzyste.

2.1 Omówienie interfejsu

API posiada dwa kontrolery: Doctors oraz MedicalVisits.

Kontroler doktorów (Doctors), ma następujące możliwości:

- Get:

/api/Doctors -> zwraca listę wszystkich doktorów

/api/Doctors/{doctorId} -> zwraca dane wybranego doktora (filtracja po numerze id doktora)

Example Value	Schema
<pre>{ "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "firstName": "string", "lastName": "string", "specialization": 1, "email": "string", "mobilePhone": "string", "description": "string" }</pre>	

- Post:

/api/Doctors -> dodanie nowego doktora

Request body
<pre>{ "firstName": "string", "lastName": "string", "specialization": 1, "email": "string", "mobilePhone": "string", "description": "string" }</pre>

/api/Doctors/{doctorId}/workDay -> dodanie nowego dnia pracy dla doktora (dzień pracy to możliwe terminy wizyt pacjentów z interwałem co 30 minut. Dzień pracy trwa od 8:00 do 16:30 – ostatnia wizyta możliwa do umówienia o 15:30).

POST `/api/Doctors/{doctorId}/workDay`

Parameters

Name	Description
doctorId * required string(\$uuid) (path)	doctorId

Request body

Example Value | Schema

```
"2023-01-14T14:27:25.873Z"
```

- Delete:

/api/Doctors/{doctorId} -> usunięcie wybranego doktora

Kontroler wizyt (MedicalVisits), ma następujące możliwości:

- Get:

/api/MedicalVisits/forDay -> zwraca listę lekarzy wraz z dostępnymi godzinami wizyt dla danej specjalizacji lekarza

Name	Description
date string(\$date-time) (query)	date
doctorSpecialization integer(\$int32) (query)	Available values : 1 --

Responses

Code	Description
200	Success

Media type
text/plain

Controls Accept header.

Example Value | Schema

```
[
  {
    "doctorId": "3fa85f64-5717-4562-b3fc-2c963f66afae",
    "doctorFirstName": "string",
    "doctorLastName": "string",
    "availableHours": [
      "2023-01-14T14:23:13.241Z"
    ]
  }
]
```

/api/MedicalVisits/{medicalVisitId}/{doctorId} -> zwraca wizytę o podanym id podlegającą doktorowi o podanym id

```
{
  "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "doctorId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "patientFirstName": "string",
  "patientLastName": "string",
  "patientPESEL": "string",
  "type": 1,
  "description": "string"
}
```

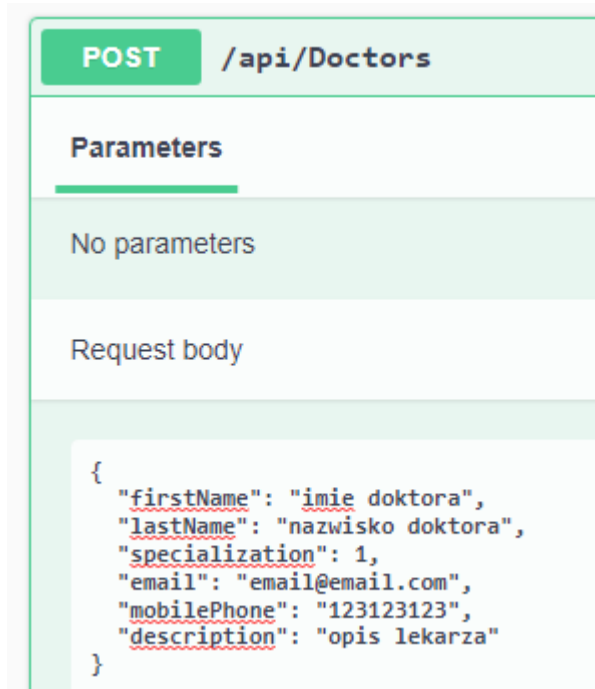
○ Post:

/api/MedicalVisits -> tworzy wizytę.

```
{
  "patientFirstName": "string",
  "patientLastName": "string",
  "patientPESEL": "string",
  "description": "string",
  "doctorId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "term": "2023-01-14T14:35:58.115Z",
  "type": 1
}
```

3. Przykład działania

1. Tworzymy doktora:



W odpowiedzi otrzymujemy id nowo utworzonego doktora:

```
"78195db8-a564-4c44-b82e-8f532880f522"
```

2. Dodajemy mu nowe godziny pracy na dzień 14.01.2023:

POST /api/Doctors/{doctorId}/workDay

Parameters

Name	Description
doctorId * required string(\$uuid) (path)	78195db8-a564-4c44-b82e-8f532880f522

Request body

"2023-01-14T14:40:46.346Z"

3. Pobieramy listę dostępnych lekarzy ze specjalizacją = 1 (GeneralPractitioner) dla dnia 14.01.2023r.

GET /api/MedicalVisits/forDay

Parameters

Name	Description
date string(\$date-time) (query)	2023-01-14T14:40:46.346Z
doctorSpecialization integer(\$int32) (query)	1

200

Response body

```
[
  {
    "doctorId": "78195db8-a564-4c44-b82e-8f532880f522",
    "doctorFirstName": "imie doktora",
    "doctorLastName": "nazwisko doktora",
    "availableTerms": [
      "2023-01-14T08:00:00+01:00",
      "2023-01-14T08:30:00+01:00",
      "2023-01-14T09:00:00+01:00",
      "2023-01-14T09:30:00+01:00",
      "2023-01-14T10:00:00+01:00",
      "2023-01-14T10:30:00+01:00",
      "2023-01-14T11:00:00+01:00",
      "2023-01-14T11:30:00+01:00",
      "2023-01-14T12:00:00+01:00",
      "2023-01-14T12:30:00+01:00",
      "2023-01-14T13:00:00+01:00",
      "2023-01-14T13:30:00+01:00",
      "2023-01-14T14:00:00+01:00",
      "2023-01-14T14:30:00+01:00",
      "2023-01-14T15:00:00+01:00",
      "2023-01-14T15:30:00+01:00"
    ]
  }
]
```

4. Rejestrujemy się na termin 14.01.2023 13:30



W odpowiedzi otrzymujemy id nowo utworzonej wizyty:

```
"f3f6feda-5982-4b2e-851c-ad6c2056276f"
```

5. Sprawdźmy ponownie listę dostępnych wizyt dla dnia 14.01.2023:



Wizyta na godzinę 13:30 zniknęła z listy, gdyż została zajęta.

6. Pobierzmy dane wizyty:

GET

/api/MedicalVisits/{medicalVisitId}/{doctorId}

Parameters

Name	Description
doctorId * required string(\$uuid) (path)	78195db8-a564-4c44-b82e-8f532880f522
medicalVisitId * required string(\$uuid) (path)	f3f6feda-5982-4b2e-851c-ad6c2056276f

Code

Details

200

Response body

```
{
  "id": "f3f6feda-5982-4b2e-851c-ad6c2056276f",
  "doctorId": "78195db8-a564-4c44-b82e-8f532880f522",
  "patientFirstName": "imie pacjenta",
  "patientLastName": "nazwisko pacjenta",
  "patientPESEL": "12312312312",
  "type": 1,
  "description": "opis wizyty"
}
```