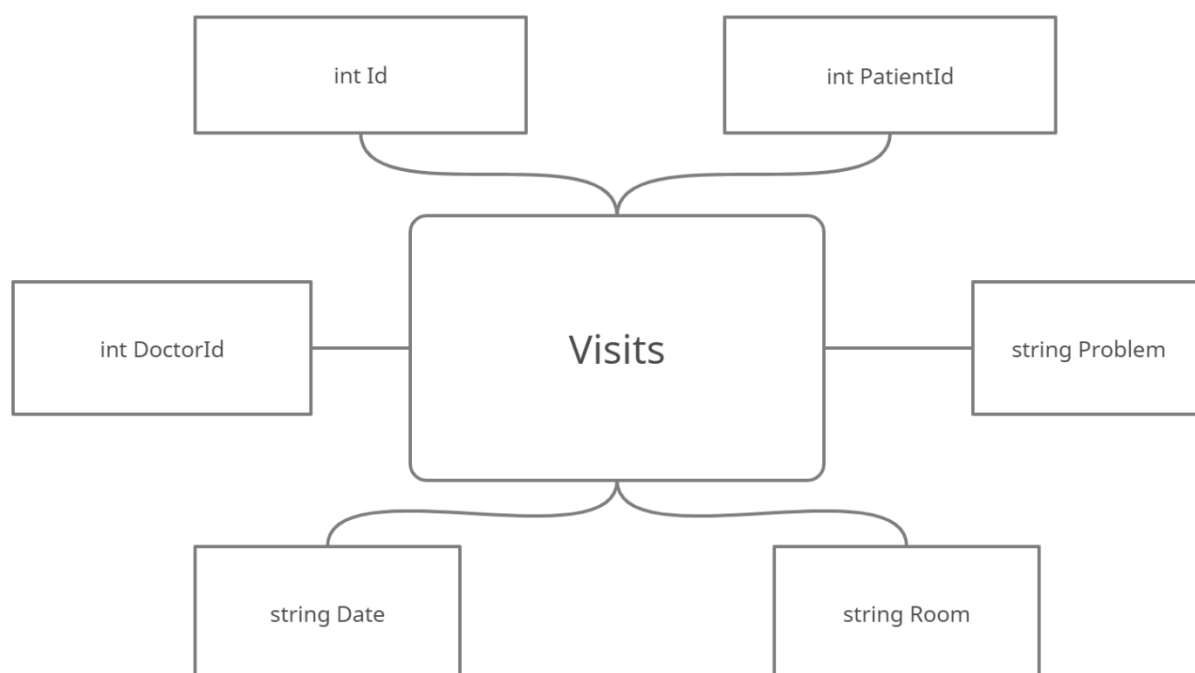


# Mikroustługa danych – Wizyty

## 1. Diagram encji



## 2. Formalny model danych

Wizyta ma następujące dane:

- Id – identyfikator, typ int
- DoctorId – identyfikator doktora, typ int
- PatientId – identyfikator pacjenta, typ int
- Problem – problem pacjenta - powód wizyty, typ string
- Date – data wizyty, typ string
- Room – numer gabinetu, typ string

### 3. Przykładowe dane

Id	DoctorId	PatientId	Problem	Date	Room
id="0"	doctorId="1"	patientId="0"	problem="ucho"	date="01.05.2021"	room="1"
id="1"	doctorId="1"	patientId="1"	problem="brzuch"	date="02.05.2021"	room="10"
id="2"	doctorId="2"	patientId="2"	problem="oko"	date="03.05.2021"	room="5"

### 4. Struktura rozwiązania

1. Projekt - Visits.Logic:
  - *VisitsFromXmlAndToXml* – klasa odpowiadająca za logikę mikrousługi, korzysta z czytnika i klasy zapisującej (Klasy *Reader* i *Writer*) do odczytania danych
  - *Reader* – klasa odpowiadająca za czytanie danych z pliku xml
  - *Writer* – klasa odpowiadająca za zapisywanie danych do pliku xml
2. Projekt - Visits.Model
  - *IVisits* – interfejs zawierający wzór metod dla klasy Visits
  - *Visit* – klasa definiująca model wizyty
3. Projekt - Visits.Rest
  - *Controller* – klasa tworząca kontroler serwisu Rest
4. Projekt - Visits.Rest.Model
  - *DataConverter* – klasa konwertująca dane odczytane z pliku xml do danych, które będą przesyłane w sieci
  - *VisitData* – klasa definiująca model danych (Visit) przesyłanych w sieci
  - *IVisitsService* - interfejs zawierający wzór metod dla kontrolera Rest

### 5. Interfejs

#### 1. Operacje usługowe:

1. Pobranie wszystkich wizyt z bazy danych - GetAllVisits():
  - Dane wejściowe: brak
  - Dane wyjściowe: tablica z danymi wszystkich wizyt
2. Dodanie wizyty do bazy danych – AddVisit(VisitData visitData):
  - Dane wejściowe: Id, DoctorId, PatientId, Problem, Date, Room
  - Dane wyjściowe: brak