

Mikroustuga aplikacji – Pacjent

1. Struktura rozwiązania

1. Projekt – Patient.AppService.Rest

Commands:

- ICommand – interfejs wskaźnikowy, określa które klasy będą komendami
- ICommandHandler – interfejs „handlera” komend

DataServiceClient:

- IPatientsServiceClient – interfejs zawierający wzór metod dla klasy PatientsServiceClient
- IPrescriptionsServiceClient - interfejs zawierający wzór metod dla klasy PrescriptionsServiceClient
- IVisitsServiceClient - interfejs zawierający wzór metod dla klasy VisitsServiceClient
- PatientsServiceClient – klasa odpowiadająca za pobieranie danych z mikroustugi danych - Pacjenci
- PrescriptionsServiceClient - klasa odpowiadająca za pobieranie danych z mikroustugi danych – Recepty
- VisitsServiceClient - klasa odpowiadająca za pobieranie danych z mikroustugi danych - Wizyty

Dtos:

- PatientDto – klasa definiująca model danych (Pacjent) przesyłanych w sieci
- PrescriptionDto – klasa definiująca model danych (Recepta) przesyłanych w sieci
- MedicineDto – klasa definiująca model danych (Leki) przesyłanych w sieci
- VisitsDto – klasa definiująca model danych (Wizyta) przesyłanych w sieci

Queries:

- PatientWebServiceQueryHandler – (klasa) „handler” zapytań, wykorzystywana jest w kontrolerze
- IPatientWebServiceQueryHandler - interfejs zawierający wzór metod dla klasy PatientWebServiceQueryHandler

Controllers:

- Controller - klasa tworząca kontroler serwisu Rest

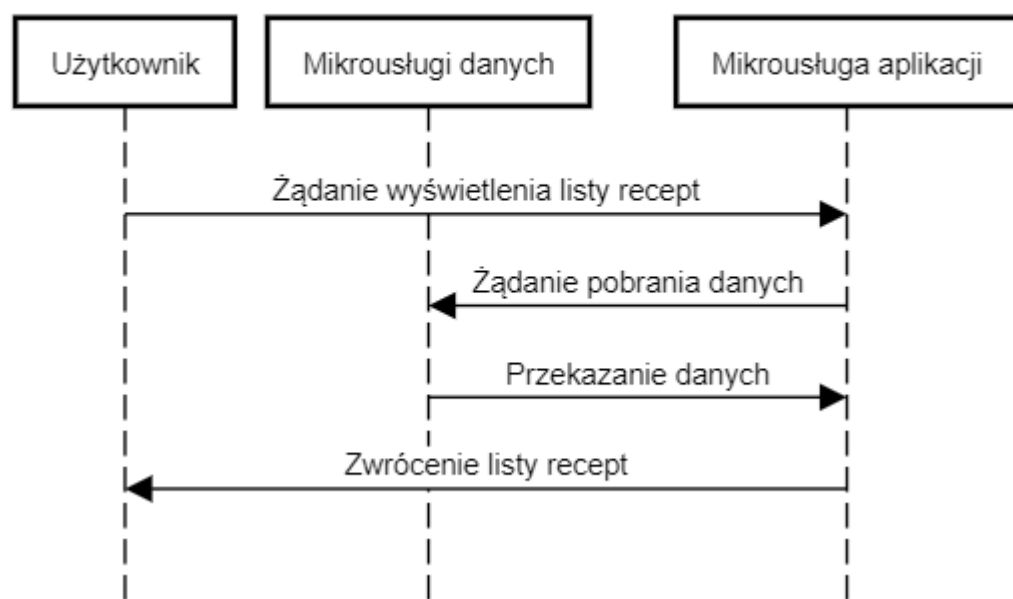
2. Dokumentacja interfejsu

1. Operacje usługowe:

1. Pobranie wybranej recepty z bazy danych – GetAllPrescriptions (int patientId):
 - Dane wejściowe: id pacjenta
 - Dane wyjściowe: wypisane recepty dla szukanego pacjenta
2. Pobranie wizyt związanych z wybranym pacjentem z bazy danych - GetAllVisits(int patientId):
 - Dane wejściowe: id pacjenta
 - Dane wyjściowe: tablica z wizytami

3. Diagram sekwencji wiadomości

Pobranie listy recept



Pobranie listy wizyt

