

Mikroustuga aplikacji – Lekarz

1. Struktura rozwiązania

1. Projekt – Doctor.AppService.Rest

Commands:

- AddPrescriptionCommand – (klasa) komenda określająca argumenty do stworzenia recepty
- ICommand – interfejs wskaźnikowy, określa które klasy będą komendami
- ICommandHandler – interfejs „handlera” komend
- PrescriptionsCommandHandler – klasa zawierająca „handler” komendy tworzenia recepty

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:

- DoctorWebServiceQueryHandler – (klasa) „handler” zapytań, wykorzystywana jest w kontrolerze

- IDoctorWebServiceQueryHandler - interfejs zawierający wzór metod dla klasy DoctorWebServiceQueryHandler

Controllers:

- Controller - klasa tworząca kontroler serwisu Rest

2. Dokumentacja interfejsu

1. Operacje usługowe:

1. Pobranie wszystkich pacjentów z bazy danych – GetAllPatients():
 - Dane wejściowe: brak
 - Dane wyjściowe: tablica z danymi wszystkich pacjentów
2. Pobranie wszystkich recept z bazy danych – GetAllPrescriptions():
 - Dane wejściowe: brak
 - Dane wyjściowe: tablica z receptami
3. Pobranie wybranej recepty z bazy danych – GetPrescriptions(int pesel):
 - Dane wejściowe: pesel pacjenta
 - Dane wyjściowe: wypisane recepty dla szukanego pacjenta
4. Dodanie recepty do bazy danych – AddPrescription(AddPrescriptionCommand prescriptionCommand):
 - Dane wejściowe: recepta (id, id doktora, id pacjenta, data wydania, data ważności, leki)
 - Dane wyjściowe: brak
5. Pobranie wizyt związanych z wybranym lekarzem z bazy danych GetVisitsAsync(int doctorId):
 - Dane wejściowe: id doktora
 - Dane wyjściowe: tablica z wizytami

3. Diagram sekwencji wiadomości

