



1. Pobranie danych pacjenta wraz z wizytami

1. Przygotowujemy końcówkę, która jako zwróci nam dane pacjenta wraz z informacjami na temat przypisanych do niego wizyt.

```
{
  "firstName": "Jan",
  "lastName": "Kowalski",
  "birthdate": "1980-01-01", //zwracany format daty może być inny, można użyć typu
danych datetime
  "totalAmountMoneySpent": "1200 zł",
  "numberOfVisit": 10,
  "visits": [
    {
      "IdVisit": 1,
      "Doctor": "John Doe",
      "Date": "2024-03-12 12:30",
      "Price": "100 zł"
    },
    ...
  ]
}
```

3. Kolumna totalAmountMoneySpent zwraca sumę wydanych przez klienta pieniędzy.
4. Kolumna numberOfVisit zwraca liczbę wizyt przypisanych do pacjenta.

2. Wstawienie nowej wizyty

1. Implementujemy końcówkę, która pozwala na wstawienie nowej wizyty.
2. Końcówka powinna przyjmować następujące parametry: IdPatient, IdDoctor i Date.
3. Powinniśmy upewnić się, że:
 1. Pacjent istnieje.
 2. Doktor istnieje.
 3. Przed przypisaniem upewniamy się, że klient nie ma już umówionej żadnej innej wizyty. Sprawdzamy jest w bazie mamy jakąkolwiek inną wizytę z Date>Now.
 4. Sprawdzamy czy wybrany lekarz pracuje danego dnia w odpowiednich godzinach na podstawie tabeli Schedule. Jeśli doktor nie jest dostępny, zwracamy odpowiedni kod błędu.
 5. Wstawiamy rekord do tabeli Visit.
 6. Zwracamy wygenerowaną wartość Id.
 7. Pamiętamy o zwracaniu odpowiednich kodów błędu.

Dodatkowe informacje:

- Pamiętaj o zasadach związanych z projektowaniem aplikacji typu REST.
- Pamiętaj o dobrych praktykach związanych z architekturą aplikacji.
- Upewnij się, że aplikacja się kompiluje.
- Pamiętaj o zwracaniu odpowiednich kodów HTTP i obsłudze błędów.
- W pierwszej kolejności ważne jest działanie aplikacji, w drugiej kolejności jakość kodu (oddzielenie logiki biznesowej, prezentacji i infrastruktury).