

Komponent obsługi szkód medycznych

W ramach ćwiczenia budujemy komponent będący częścią systemu obsługi ubezpieczeń medycznych. Istnieje już komponent odpowiedzialny za zarządzanie polisami. Komponent ten, za każdym razem, gdy mają miejsce zmiany danych polisy publikuje zdarzenia PolicyVersionCreated.

Zdarzenie to zawiera następujące dane polisy:

```
{
  "policyNumber": "P1212121",
  "productCode": "Pakiet Gold",
  "policyHolder": {
    "firstName": "Jan",
    "lastName": "Nowak",
    "pesel": "1111111116",

    "accountNumber": "2738123834783247723",
    "address": {
      "country": "PL",
      "city": "Warszawa",
      "zipCode": "01-001",
      "street": "JaksTam 123 m 2"
    }
  },
  "policyValidFrom": "2018-01-01",
  "policyValidTo": "2018-12-31",
  "versionNumber": 1,
  "versionValidFrom": "2018-01-01",
  "versionValidTo": null,
  "covers": [
    {
      "coverCode" : "KONS",
      "services" : [
        {
          "code" : "KONS_INTERNISTA",
          "coPayment" : { "percent" : 0.25 },
          "limit" : { "maxQuantity" : null, "maxAmount" : 100, "limitPeriod" : "POLICY_YEAR" }
        },
        {
          "code" : "KONS_PEDIATRA",
          "coPayment" : { "amount" : 10 },
          "limit" : { "maxQuantity" : 20, "maxAmount" : 100, "limitPeriod" : "POLICY_YEAR" }
        }
      ]
    },
    {
      "coverCode" : "LAB",
      "services" : [
        {
          "code" : "LAB_KREW_OB",
          "coPayment" : { "percent" : 0.10 },
          "limit" : { "maxQuantity" : 5, "maxAmount" : 50, "limitPeriod" : "POLICY_YEAR" }
        },
        {
          "code" : "LAB_HDL",
          "coPayment" : { "amount" : 2 },
          "limit" : { "maxQuantity" : 2, "maxAmount" : 28, "limitPeriod" : "POLICY_YEAR" }
        }
      ]
    }
  ]
}
```

Komponent polisowy posiada też API pozwalające na pobranie danych polisy po podaniu jej numeru i daty, na którą mają być z

wrócone aktualne dane.

Naszym zadaniem jest zbudowanie komponentu do obsługi szkód.

Obsługa szkód składa się z:

1) Rejestracja szkody (ang. submit claim)

W ramach rejestracji szkody system musi otrzymać następujące dane: numer polisy, datę zdarzenia, kod placówki medycznej (ze słownika) oraz listę pozycji na szkodzie. Każda pozycja zawiera kod usługi (ze słownika), ilość usług, cenę.

System sprawdza, czy istnieje polisa o padnym numerze, jeśli nie to zgłasza błąd i proces się kończy.

Jeśli polisa istnieje to system zapisuje dane szkody a następnie sprawdza pokrycie ubezpieczeniem: jeśli data zdarzenia nie jest objęta okresem obowiązywania polisy, to szkoda zostaje automatycznie odrzucona z powodem: Zdarzenie nastąpiło poza okresem ochrony. Pełen koszt każdej usługi ponosi ubezpieczony.

W przeciwnym przypadku system wylicza dla każdej pozycji kwotę do wypłaty przez ubezpieczyciela. System robi to w następujący sposób dla każdej pozycji:

- sprawdzenie, czy na polisie w ochronach jest usługa o kodzie z pozycji, jeśli nie ma to cały koszt ponosi ubezpieczony,

- system wylicza udział własny na podstawie definicji co-payment z polisy (procentowy, lub kwotowy)

- następnie pozostała kwota porównywana jest z limitem na daną usługę i dotychczasowym zużyciem tego limitu np. "limit" : { "maxQuantity" : null, "maxAmount" : 100, "limitPeriod" : "POLICY_YEAR" } oznacza, że w ciągu roku polisowego ubezpieczonemu przysługuje zwrot maksymalnie 100PLN ale ma nieograniczoną liczbę wystąpienia usługi. W ten sposób powstaje kwota do zapłaty przez ubezpieczyciela

System rezerwuje sobie kwoty konsumpcji limitu wynikające z danej linii.

- na końcu system zapisuje kwotę do zapłaty przez ubezpieczyciela i przez ubezpieczonego, generuje unikalny numer szkody i zapisuje dane.

Na tym kończy się proces rejestracji.

2) Akceptacja, odrzucenia lub korekta zarejestrowanej szkody

Zarejestrowana szkoda może zostać odrzucona, zaakceptowana lub poddana korekcie (funkcji korekty możemy nie implementować w pierwszej fazie).

Akceptacja

Użytkownik z odpowiednim uprawnieniem może zaakceptować szkodę. Akcje to powoduje trwałą konsumpcję limitów.

Odrzucenia

Akcja to zwalnia konsumpcję limitów i powoduje, że cały koszt szkody ponosi ubezpieczony.

3) Wyszukiwanie szkody

System pozwala na wyszukiwanie szkody po: numerze, statusie, zakresie dat zdarzenia, zakresie kwot roszczenia, numerze polisy, kodzie usługi, kodzie placówki medycznej

4) Tworzenie poleceń zapłaty

Dla zaakceptowanych szkód uruchamiany jest proces tworzenia poleceń wypłaty. Dla każdej zaakceptowanej szkody tworzone jest polecenie wypłaty. Polecenie wypłaty ma wskazanie na osobę, której wypłacamy odszkodowanie – zawsze jest to policy holder, datę utworzenia, kwotę – wyliczoną na szkodzie, numer konta – konto policy holdera.

Celem zadania, jest zbudowanie komponentu, który zaprezentuje wzorcowe podejście do tworzenia komponentów WL i nasze coding guideliness / best practices.

W szczególności ważne jest zaprezentowanie zasad:

- 1) podziału kodu na pakiety
- 2) widoczności klas w pakietach
- 3) odpowiedniego nazewnictwa klas
- 4) właściwego użycia Springa (co wrzucamy do kontenera, a co nie)
- 5) zasad używania Optional (null handling)
- 6) obsługa błędów, tworzenie i rzucanie wyjątków
- 7) tworzenie złożonych encji
- 8) enkapsulacja
- 9) first class collections
- 10) użycie specyfikacji
- 11) Logowanie
- 12) tworzenie czytelnego i testowalnego kodu
- 13) zaprezentowanie implementacji typowych elementów:
 1. obsługa zdarzeń przychodzących
 2. implementacja logiki biznesowej
 3. implementacja zapisu / odczytu z bazy danych
 4. publikowanie zdarzeń
 5. komunikacja z innymi modułami
 6. implementacja wyszukiwania
 7. API REST
 8. procesy batchowe
 9. konfiguracja
 10. testy jednostkowe
 11. testy integracyjne