

Zadanie rekrutacyjne

Komponent ten będzie częścią systemu bankowego. Wykorzystywany będzie w procesie rejestracji i przetwarzania wniosków kredytowych. Pracownik banku będzie wprowadzał dane wniosku: dane klienta takie, jak: imię, nazwisko, PESEL, data urodzenia, deklarowany dochód miesięczny oraz parametry kredytu: wysokość kredytu, ilość miesięcznych rat, oprocentowanie, data rozpoczęcia spłaty. Następnie system zapisze wniosek nadając mu unikalny numer oraz dokona oceny wniosku.

Ocena wniosku polega na sprawdzeniu przygotowanych przez departament Ryzyka warunków biznesowych. W pierwszej fazie projektu ocena wniosku będzie polegać na sprawdzeniu wieku klienta, sprawdzeniu, czy jego dochody są wystarczająco wysokie oraz na sprawdzeniu, czy nie jest on na liście dłużników.

Zależy nam, żeby w przyszłości móc szybko dodawać nowe warunki i modyfikować istniejące, żeby dopasować się do zmieniających przepisów prawnych i warunków panujących na rynku. Komponent będzie wykorzystywany przez inne elementy systemu przy użyciu API REST.

W startowym projekcie **src\loan-service** jest już zdefiniowany kontroler **LoanRequestController**. Zdefiniowane są także klasy DTO stanowiące request i response metod tego kontrolera.

W ramach zadania należy zaimplementować dwie funkcje:

1. **Register**

- a. Komponent musi sprawdzić, czy podano wszystkie wymagane dane. Wszystkie pola klasy **LoanRequestDto** powinny być wypełnione. Pole **customerBirthday** musi zawierać datę mniejszą od daty bieżącej. Pole **customerMonthlyIncome** musi zawierać wartość większą od zera. Pola **loanAmount** i **numberOfInstallments** muszą mieć wartości większe od zera. Pole **firstInstallmentDate** musi być datą większą od daty bieżącej.
- b. Komponent musi nadać wnioskowi unikalny numer.
- c. Komponent musi dokonać ewaluacji wniosku i wyznaczyć, czy może on zostać zaakceptowany czy odrzucony. Ewaluacja polega na sprawdzeniu czy spełnione są reguły biznesowe. Na obecnym etapie prac znane są trzy reguły, ale komponent powinien być tak zbudowany, by dodanie nowych reguł było łatwe. Reguły, które należy zaimplementować na tym etapie prac:
 - i. Wiek klienta w momencie zakończenia spłacania kredytu musi być mniejszy lub równy 65 lat. Raty kredytu są miesięczne. Wiek w latach na datę ostatniej raty zaokrąglamy do pełnych lat.
 - ii. Miesięczna rata kredytu nie może przekraczać 15% miesięcznych dochodów klienta. Oprocentowanie kredytu powinno być parametrem komponentu. Do celów testowych przyjmijmy 4% rocznie. Kredyt jest udzielany w modelu, w którym miesięczne raty mają stałą wysokość.
 - iii. Klient nie może znajdować się na liście dłużników. Aby to sprawdzić należy użyć usługi debtor-registry, która znajduje się w folderze **src\debtor-registry**.
- d. Komponent musi zapisać dane wniosku wraz z numerem i wynikiem ewaluacji do bazy danych.

2. **GetByNumber**

a. Komponent musi zwrócić dane wniosku z podanym numerem.

Zależy nam, aby kod był czytelny, testowalny, zgodny z dobrymi praktykami programowania obiektowego (SOLID, DDD).

W części odpowiadającej za ewaluację wniosku zależy nam na implementacji, która pozwoli nam w łatwy (dla programisty 😊) sposób dodawać nowe reguły i modyfikować już istniejące reguły.

Należy zaimplementować testy jednostkowe.