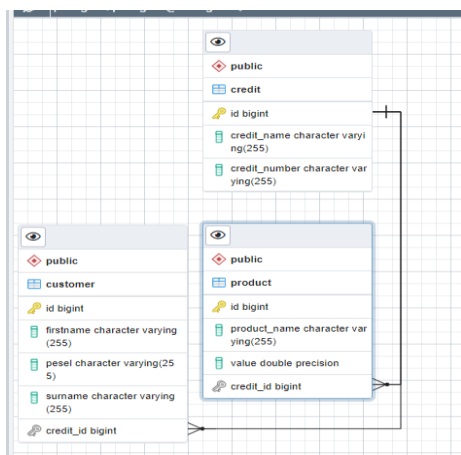


Projekt CreditApp

Stworzono projekt CreditApp łączący się z bazą danych i polegający na komunikacji komponentów służących do umieszczania danych o kredycie, produktach i kliencie. Oparto o microserwisy i usługi restowe.



Rys1. Schemat ERD bazy danych

W projekcie użyto trzech tabel pozostających ze sobą w relacji wiele do jednego.

W projekcie skorzystano z architektury trójwarstwowej aplikacji webowej. Użyto warstwy prezentacji, dostępu do danych i biznesowej. Do mapowania użyto technologii hibernate.

Opisano metody w controlerze łączących polecenia z zadania.

Dla Diagram sekwencji dla UC01: Złożenie informacji o kredycie:

1. Klient wywołuje usługę CreateCredit w celu złożenia informacji o nowym kredycie - Klasa CreditController i metoda add.
2. Komponent Credit nadaje nowy numer kredytu – Klasa CreditController i metoda add
3. Komponent Credit wywołuje usługę CreateProduct komponentu Product i przekazuje informacje o produkcie oraz numer kredytu– klasa ProductController i metoda add.
4. Komponent Product składa informacje o nowym produkcie w bazie danych – klasa ProductController i metoda add
5. Komponent Credit wywołuje usługę CreateCustomer komponentu Customer i przekazuje informacje o kliencie oraz numer kredytu – klasa CreditController i metoda getCreditNumber
6. Komponent Customer składa informacje o kliencie w bazie danych - klasa CustomerController i metoda add.
7. Komponent Credit składa informacje o kredycie w bazie danych – klasa CreditController i metoda getAllCredit
8. Komponent Credit zwraca informacje o nadanym numerze kredytu klientowi. - klasa CreditController i metoda getAllClientWithTheSameCreditNumber.

Diagram sekwencji dla UC02: Pobranie informacji o kredycie:

1. Klient wywołuje usługę GetCredits
2. Komponent Credit pobiera informacje o wszystkich kredytach z bazy danych – klasa CreditController i metoda getAllCredits.
3. Komponent Credit wywołuje usługę GetCustomers komponentu Customer i przekazuje numery kredytów, dla których należy zwrócić klientów – klasa CreditController i metoda getAllClientWithTheSameCreditNumber
4. Komponent Customer pobiera informacje o klientach z bazy danych i zwraca – Klasa CustomerController i getAllCustomer

5. Komponent Credit wywołuje usługę GetProducts komponentu Product i przekazuje numery kredytów, dla których należy zwrócić produkty. – Klasa CreditController i metoda getAllProductWithTheSameCreditNumber
6. Komponent Product pobiera informacje o produktach z bazy danych i zwraca - Klasa ProductController i metoda getAllProduct
7. Komponent Credit agreguje dane zwrócone z usług GetCustomers oraz GetProducts i zwraca listę kredytów Klientowi. - Klasa CreditController i metody getListCreditsWithCustomerContains i getListCreditWithProductContains
8. Komponent product sumuje informacje o wszystkich cenach produktów Klasa ProductController i metoda getSumProductValue

Dla czytelności kodu skorzystano z pluginu lombok.

UWAGA!!!

By zapisać product i klienta należy utworzyć kredyt i w requescie dla zapisu producta i klienta użyć nadany numer kredytu utworzonego podczas tworzenia nowego kredytu.

Projekt uruchamiać z poziomu intellij oraz za pomocą docker-compose.