



InSystem Sp. z o.o.

ul. Kijowska 1,
03-738 Warszawa

tel.: +48 22 2050710-18
fax.: +48 22 2050719

<http://www.insystem.com.pl>,
office@insystem.com.pl

CF ESB

Dokumentacja projektu

Historia zmian

Wersja	Data	Autor	Opis
1.0	28-09-2018	Marcin Warda	Stworzenie pierwszej wersji
2.0	29-10-2021	Marcin Warda	Aktualizacja

Regon: 142282110, NIP: 701-022-55-66

Sąd Rejonowy dla m.st. Warszawy, XII Wydział Gospodarczy Krajowego Rejestru Sądowego nr KRS 0000352995

Kapitał zakładowy spółki: 6 250,00 zł , opłacony w całości.

Spis treści

Historia zmian.....	1
Wstęp	4
Cel projektu	4
Zakres projektu.....	4
Kluczowe pojęcia	5
Wymagania biznesowe.....	6
Opis systemu	7
Schemat architektury systemu.....	8
Systemy zewnętrzne.....	8
COOLZAM	8
PRESTA.....	9
Interfejsy systemowe	9
CreateCustomer	10
CreateOrder.....	12
GetColorName.....	13
GetVendorName	14
Interfejsy zewnętrzne.....	15
COOLZAM	15
v_presta_klient.....	15
v_presta_rezerwacje_coolzam_aktualne.....	17
v_presta_produkty.....	18

v_presta_lista_dostaw	20
v_presta_lista_zamowien.....	21
v_presta_zestaw_kategoria_produkt.....	23
v_presta_kategoria_nowa.....	23
v_presta_podkategoria_nowa	24
v_presta_podpodkategoria_nowa	25
v_presta_klient_dodawanie.....	26
v_presta_zamowienie	27
v_presta_kolory.....	29
v_presta_mixkolor.....	29
PRESTA.....	30
CreateOrUpdateCategory.....	30
CreateOrUpdateCustomer	31
CreateOrUpdateOrder.....	33
CreateOrUpdateProduct	34
Moduły synchronizacji.....	37
CoolzamSyncService	37
PrestaSync	42
Skrypt bazy danych.....	44
Specyfikacje interfejsów.....	72

Wstęp

Cel projektu

Projekt CF ESB został stworzony w odpowiedzi na potrzeby wymiany danych pomiędzy dwoma niezależnymi systemami COOLZAM i PRESTA. Oba te systemy posiadają swoje własne bazy danych, które różnią się strukturą i zakresem danych w nich przetrzymywanych, jak również sposobem ich przetwarzania. Pomimo tych różnic, ww. systemy zewnętrzne współuczestniczą w pewnym podzbiorze procesów biznesowych. Celem projektu jest zapewnienie zewnętrznym systemom dostępu do odpowiednich danych (poprzez jawne przesłanie ich do systemu lub udostępnianiu na żądanie) oraz kontroli przepływu tych danych. System ESB, w pewnym zakresie, ma również na celu zapewnienie kontroli nad opóźnieniami pomiędzy pojawieniem się danych w jednym systemie zewnętrznym, a ich dostępnością w drugim systemie zewnętrznym. Dodatkową zaletą wprowadzenia centralnego punktu komunikacji dla zewnętrznych systemów, jest uniezależnienie od siebie systemów zewnętrznych, co umożliwia łatwe zastąpienie ich przez inne systemy, jeżeli tylko będą one posiadać takie same interfejsy komunikacyjne jak oryginalny system.

Zakres projektu

Projekt obejmuje wymianę danych pomiędzy zewnętrznymi systemami, w zakresie danych, jakie te systemy udostępniają.

W zakres projektu wchodzi:

- Pozyskiwanie danych z systemów zewnętrznych
- Rejestracja danych w lokalnej bazie systemu ESB, z możliwością ich dalszego przetwarzania czy modyfikacji ich struktur
- Kontrola poprawności danych

-
- Kontrola przepływu danych pod względem ich treści jak i częstotliwości aktualizacji
 - Udostępnianie danych systemom zewnętrznym

Projekt nie zajmuje się funkcjonalnością realizowaną przez interfejsy użytkownika zewnętrznych systemów oraz nie wchodzi w szczegóły realizowanych przez te interfejsy procesów biznesowych.

Kluczowe pojęcia

- **ESB (Szyna danych)** – system pośredniczący w wymianie danych pomiędzy systemami zewnętrznymi, centralny punkt komunikacyjny
- **PRESTA** – system zewnętrzny udostępniający klientom CF funkcjonalność sklepu internetowego/katalogu produktów, komunikuje się z systemem ESB za pomocą usług Web (Web Service) w technologii SOAP
- **COOLZAM** – system zewnętrzny, używany wyłącznie przez pracowników CF do wewnętrznej obsługi procesów sprzedażowych, komunikuje się z systemem ESB za pomocą tabel interfejsowych (widoków) udostępnianych przez bazę danych COOLZAM
- **CF ESB** – Projekt związany ze stworzeniem systemu ESB na potrzeby firmy CoolFlowers
- **Klient** – Klient firmy CF, który może być lub nie być klientem w systemie PRESTA
- **Zamówienie klienta (Rezerwacja)** – zamówienie złożone przez klienta firmy CF, może pochodzić zarówno z systemu PRESTA jak i z systemu COOLZAM, zamówienie zawiera wyłącznie jeden produkt
- **Produkt** – konkretny produkt sprzedawany przez firmę CF w konkretnym wariantcie kolorystycznym, różne warianty kolorystyczne liczą się jak różne produkty
- **Dostawa produktu** – dostawa produktu(ów) od dostawcy do firmy CF, każda pozycja dostawy ma powiązane zamówienie produktu (Termin), posiada konkretną datę dostawy

-
- **Zamówienie produktu (Termin)** – zamówienie produktów od firmy CF do dostawcy, na podstawie zamówienia mogą zostać utworzone różne dostawy, posiada orientacyjny termin realizacji zamówienia

Wymagania biznesowe

System powinien realizować następujące funkcje:

- Przekazywanie z systemu COOLZAM do systemu PRESTA informacji o nowych/zaktualizowanych/usuniętych produktach
- Przekazywanie z systemu COOLZAM do systemu PRESTA informacji o nowych/zaktualizowanych/usuniętych kategoriach produktowych
- Przekazywanie z systemu COOLZAM do systemu PRESTA informacji o nowych/zaktualizowanych/usuniętych dostawach produktów
- Przekazywanie z systemu COOLZAM do systemu PRESTA informacji o nowych/zaktualizowanych/usuniętych zamówieniach produktów
- Przekazywanie z systemu COOLZAM do systemu PRESTA informacji o nowych/zaktualizowanych klientach
- Przekazywanie z systemu COOLZAM do systemu PRESTA informacji o nowych/zaktualizowanych zamówieniach klienta
- Przekazywanie z systemu COOLZAM do systemu PRESTA informacji o dostawcach produktu (na żądanie)
- Przekazywanie z systemu COOLZAM do systemu PRESTA informacji o kolorach produktu i o jego składowych w przypadku mieszanki kolorów (na żądanie)
- Przekazywanie z systemu PRESTA do systemu COOLZAM informacji o nowych klientach

-
- Przekazywanie z systemu PRESTA do systemu COOLZAM informacji o nowych zamówieniach klientów

Opis systemu

System ESB został stworzony w oparciu o platformę Talend ESB, która wykorzystuje język programowania Java oraz udostępnia narzędzia usprawniające procesy integracji danych. System korzysta z kontenera usług Apache Karaf. Jako systemowa baza danych, został użyty DBMS PostgreSQL.

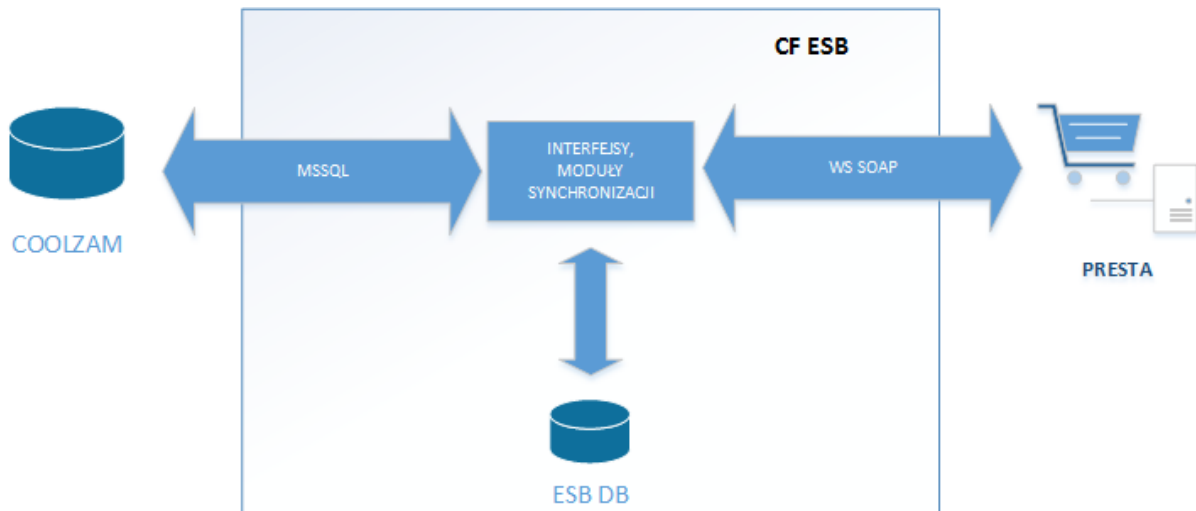
System ESB znajduje się na wirtualnej maszynie działającej w infrastrukturze CF. Użyty został system operacyjny Windows Server 2012R2, który został dołączony do domeny AD CF.local.

System dostępny jest pod adresem:

DNS: **cf-esb-prod.cf.local**

IP: **192.168.11.30**

Schemat architektury systemu



Systemy zewnętrzne

COOLZAM

System do obsługi zamówień w firmie CF.

Działa w oparciu o bazę danych MSSQL na serwerze: **CF-SQLN.CF.local [192.168.11.10]** (domyślna instancja)

Nazwa bazy danych, przez którą wystawione są interfejsy: „coolzam”

Komunikacja z tym systemem jest inicjowana zawsze ze strony ESB (odczyt, zapis). Baza danych systemu COOLZAM udostępnia mechanizm wykrywania zmian oparty na znaczniku czasowym modyfikacji/utworzenia rekordu, co pozwala na synchronizację bez każdorazowego pobierania pełnej tabeli danych.



InSystem Sp. z o.o.

ul. Kijowska 1,
03-738 Warszawa

tel.: +48 22 2050710-18
fax.: +48 22 2050719

<http://www.insystem.com.pl>,
office@insystem.com.pl

PRESTA

System umożliwiający klientom CF przeglądanie produktów oraz składanie zamówień.

Komunikacja z systemem odbywa się dwukierunkowo poprzez interfejsy wystawione po stronie systemu PRESTA i po stronie systemu ESB.

Wymiana danych może być inicjowana zarówno z systemu ESB jak i systemu PRESTA. Odbywa się poprzez Webservice'y w technologii SOAP

Interfejsy PRESTA dostępne są pod adresem:

<http://katalog.coolflowers.pl/integration/server.php>

Techniczna specyfikacja interfejsu znajduje się pod adresem:

<http://katalog.coolflowers.pl/integration/server.php?wsdl>

Interfejsy systemowe

System ESB udostępnia interfejsy Webservice w technologii SOAP. Interfejsy dostępne są pod adresem:

- **<https://192.168.11.30:9002/services>**

LUB

- **<https://cf-esb-prod.cf.local:9002/services>**

Poniżej opisane zostały wszystkie interfejsy udostępnione przez system ESB.

CreateCustomer

Interfejs przekazujący do systemu COOLZAM informacje o nowych klientach tworzonych w systemie PRESTA

Interfejs: **<https://192.168.11.30:9002/services/CreateCustomer>**

Specyfikacja: **<https://192.168.11.30:9002/services/CreateCustomer?wsdl>**

Kierunek wywoływania: PRESTA -> ESB

Kierunek przepływu danych: PRESTA -> ESB (dane klienta), ESB -> PRESTA (identyfikator COOLZAM nowo utworzonego klienta)

Pola dostępne w zapytaniu:

- **nip** – numer NIP klienta, może być w jednym z dwóch formatów: [0-9]{10} LUB [A-Z][A-Z][A-Z0-9]{8,12}
- **companyName** – Nazwa firmy klienta
- **emailAddress** – adres e-mail klienta, jest to główny adres e-mail danego klienta powiązany z kontem w systemie PRESTA
- **tel1** – pierwsze pole z numerem telefonu klienta
- **tel1Desc** – opis pierwszego pola z numerem telefonu klienta
- **tel2** – drugie pole z numerem telefonu klienta
- **tel2Desc** – opis drugiego pola z numerem telefonu klienta
- **tel3** – trzecie pole z numerem telefonu klienta
- **tel3Desc** – opis trzeciego pola z numerem telefonu klienta
- **deliveryAddress**
 - **firstname** – imię pracownika klienta dla adresu dostawy
 - **lastname** – nazwisko pracownika klienta dla adresu dostawy

-
- **companyName** – nazwa firmy, do której ma być realizowana dostawa
 - **addressLine** – linia adresowa, zawierająca ulicę, numer i numer lokalu miejsca dostawy
 - **postalCode** – kod pocztowy miejsca dostawy
 - **city** – miasto adresu dostawy
 - **invoiceAddress**
 - **firstname** – imię pracownika klienta dla adresu faktury (kontakt płatnika faktury)
 - **lastname** – nazwisko pracownika klienta dla adresu faktury (kontakt płatnika faktury)
 - **companyName** – nazwa firmy, na którą ma być wystawiona faktura (płatnik faktury)
 - **addressLine** – linia adresowa, zawierająca ulicę, numer i numer lokalu płatnika faktury
 - **postalCode** – kod pocztowy płatnika faktury
 - **city** – miasto płatnika faktury
 - **status** – status klienta, aktualnie nie wykorzystywane, nowy klient pochodzący z systemu PRESTA jest zawsze przesyłany dalej jako niezweryfikowany (statyczna wartość „NIEZWERYFIKOWANY”)

Pola dostępne w odpowiedzi:

- **result** – wynik wykonania operacji, pozytywny lub negatywny
- **resultCode** – dodatkowy kod rezultatu, może być przydatny do automatycznej obsługi błędów
- **resultMsg** – czytelny dla człowieka wynik wykonania operacji lub powód jej niewykonania
- **customerId** – w przypadku pozytywnego wyniku operacji, w tym polu zwracany jest identyfikator COOLZAM klienta przydzielony podczas tworzenia nowego klienta na tabeli interfejsowej COOLZAM

CreateOrder

Interfejs przekazujący do systemu COOLZAM informacje o nowych zamówieniach klienta tworzonych w systemie PRESTA

Interfejs: **<https://192.168.11.30:9002/services/CreateOrder>**

Specyfikacja: **<https://192.168.11.30:9002/services/CreateOrder?wsdl>**

Kierunek wywoływania: PRESTA -> ESB

Kierunek przepływu danych: PRESTA -> ESB

Pola dostępne w zapytaniu:

- **orderId** – identyfikator zamówienia, tworzony w systemie PRESTA, współdzielony pomiędzy systemami
- **createDate** – data utworzenia zamówienia klienta
- **customerId** – identyfikator COOLZAM klienta, który składa zamówienie
- **ean** – numer EAN produktu, na który klient składa zamówienie (aktualnie nie wykorzystywane)
- **prodId** – identyfikator COOLZAM produktu, na który klient składa zamówienie
- **colorId** – identyfikator COOLZAM wariantu kolorystycznego produktu, na który klient składa zamówienie
- **productOrderId** – pole opcjonalne, identyfikator COOLZAM zamówienia produktu (od CF do dostawcy), wypełniany, gdy produkt ma pochodzić z konkretnego zamówienia produktu do dostawcy
- **productDeliveryId** – pole opcjonalne, identyfikator COOLZAM dostawy produktu (od dostawcy do CF), wypełniany, gdy produkt ma pochodzić z konkretnej dostawy produktu od dostawcy
- **quantity** – ilość jednostek danego produktu zamówionego przez klienta

-
- **status** – status zamówienia klienta, system ESB nie ingeruje w treść statusu, status jest przekazywany w takiej samej formie pomiędzy systemami

Pola dostępne w odpowiedzi:

- **result** – wynik wykonania operacji, pozytywny lub negatywny
- **resultCode** – dodatkowy kod rezultatu, może być przydatny do automatycznej obsługi błędów
- **resultMsg** – czytelny dla człowieka wynik wykonania operacji lub powód jej niewykonania

GetColorName

Interfejs udostępniający informacje o danym wariantcie kolorystycznym na podstawie jego identyfikatora. W przypadku mieszanek kolorów pozwala na rekurencyjne wywołania, co do kolorów składowych, dzięki temu obsługuje dowolnie zagnieżdżone drzewa kolorów i kolorów składowych.

Interfejs: **<https://192.168.11.30:9002/services/GetColorName>**

Specyfikacja: **<https://192.168.11.30:9002/services/GetColorName?wsdl>**

Kierunek wywoływania: PRESTA -> ESB

Kierunek przepływu danych: ESB -> PRESTA

Pola dostępne w zapytaniu:

- **colorId** – identyfikator COOLZAM wariantu kolorystycznego produktu, o którym chcemy uzyskać szczegółowe informacje

Pola dostępne w odpowiedzi:

-
- **colorFound** – wynik operacji, w tym przypadku informacja czy kolor o podanym identyfikatorze został odnaleziony w COOLZAM
 - **colorName** – gdy kolor został odnaleziony w tym polu zwracana jest jego nazwa (lub nazwa mieszanki kolorystycznej)
 - **isMix** – flaga określająca czy dany kolor jest pojedynczym kolorem lub mieszanką kolorystyczną
 - **componentColors** – w przypadku, gdy kolor jest mieszanką różnych kolorów, element ten zawiera listę kolorów składowych
 - **colorId** – identyfikator(y) kolorów składowych mieszanki kolorów

GetVendorName

Interfejs udostępniający informacje o danym dostawcy produktu na podstawie jego identyfikatora, aktualnie zwraca nazwę dostawcy.

Interfejs: **<https://192.168.11.30:9002/services/GetVendorName>**

Specyfikacja: **<https://192.168.11.30:9002/services/GetVendorName?wsdl>**

Kierunek wywoływania: PRESTA -> ESB

Kierunek przepływu danych: ESB -> PRESTA

Pola dostępne w zapytaniu:

- **vendorId** – identyfikator dostawcy produktu, o którym informacje chcemy uzyskać

Pola dostępne w odpowiedzi:

- **vendorFound** – wynik wykonania operacji, informacja czy dostawca o danym identyfikatorze został odnaleziony

-
- **vendorId** – identyfikator dostawcy
 - **vendorName** – nazwa dostawcy

Interfejsy zewnętrzne

COOLZAM

v_presta_klient

Widok „v_presta_klient” zawiera listę klientów systemu COOLZAM wraz z wymaganymi na potrzeby integracji ich danymi.

Odczyt/zapis: tylko odczyt

Kierunek wywoływania: ESB -> COOLZAM

Kierunek przepływu danych: COOLZAM -> ESB

Lista pól na widoku:

[KL_ID] – identyfikator COOLZAM klienta

[NIP] – numer NIP klienta

[EMAIL] – adres e-mail klienta

[ADRE_DOST_IMIE] – imię pracownika klienta dla adresu dostawy

[ADRE_DOST_NAZWISKO] – nazwisko pracownika klienta dla adresu dostawy

[ADRE_DOST_FIRMA] – nazwa firmy, do której ma być realizowana dostawa

[ADRE_DOST_ADRES] – linia adresowa, zawierająca ulicę, numer i numer lokalu miejsca dostawy

[ADRE_DOST_KOD_POCZTOWY] – kod pocztowy miejsca dostawy

[ADRE_DOST_MIASTO] – miasto adresu dostawy

[ADRE_FAKT_IMIE] – imię pracownika klienta dla adresu faktury (kontakt płatnika faktury)

[ADRE_FAKT_NAZWISKO] – nazwisko pracownika klienta dla adresu faktury (kontakt płatnika faktury)

[ADRE_FAKT_FIRMA] – nazwa firmy, na którą ma być wystawiona faktura (płatnik faktury)

[ADRE_FAKT_ADRES] – linia adresowa, zawierająca ulicę, numer i numer lokalu płatnika faktury

[ADRE_FAKT_KOD_POCZTOWY] – kod pocztowy płatnika faktury

[ADRE_FAKT_MIASTO] – miasto płatnika faktury

[KL_TIMESTAMP_V2] – znacznik czasowy utworzenia/modyfikacji rekordu, używany do wykrywania modyfikacji widoku

[KL_DO_PRESTA] – flaga określająca czy klient jest klientem, który ma być synchronizowany z PRESTA, aktualnie nieużywane, wszyscy klienci, którzy pojawią się w tabeli interfejsowej oraz spełniają wymagania poprawności są przesyłani do PRESTA

[KL_STATUS] – status klienta, aktualnie nieużywane, wszyscy klienci do trafiają do PRESTY z wartością "NIEAKTYWNY" w polu status

[KL_TEL1] – pierwsze pole z numerem telefonu klienta

[KL_TEL2] – drugie pole z numerem telefonu klienta

[KL_TEL3] – trzecie pole z numerem telefonu klienta

[KL_OP1] – opis pierwszego pola z numerem telefonu klienta

[KL_OP2] – opis drugiego pola z numerem telefonu klienta

[KL_OP3] – opis trzeciego pola z numerem telefonu klienta

[v_presta_rezerwacje_coolzam_aktualne](#)

Widok „v_presta_rezerwacje_coolzam_aktualne” zawiera listę zamówień od klientów, zarówno utworzone z PRESTY jak i z COOLZAM

Odczyt/zapis: tylko odczyt

Kierunek wywoływania: ESB -> COOLZAM

Kierunek przepływu danych: COOLZAM -> ESB

Lista pól na widoku:

[ID_REZ] – identyfikator zamówienia klienta, może być wygenerowany przez system COOLZAM lub system PRESTA w zależności od tego gdzie powstało zamówienie

[DATA_REZ] – data utworzenia zamówienia klienta

[ZA_MIE_ID] – pole nieużywane, wartość określona przez identyfikator zamówienia do dostawcy (ZA_ID)

[ZA_TY_ID] – pole nieużywane, wartość określona przez identyfikator zamówienia do dostawcy (ZA_ID)

[ZA_ROK] – pole nieużywane, wartość określona przez identyfikator zamówienia do dostawcy (ZA_ID)

[KL_ID] – identyfikator COOLZAM klienta, na którego jest zamówienie

[PR_ID] – identyfikator COOLZAM produktu, na który jest zamówienie

[KO_ID] – identyfikator COOLZAM wariantu kolorystycznego produktu, na który jest zamówienie

[Ilosc] – ilość produktu na zamówieniu

[Rezerwa] – wirtualna ilość zamówienia, zawiera różnicę pomiędzy oryginalną ilością na zamówieniu a aktualną ilością, po jej zmniejszeniu lub rozbiciu zamówienia

[Status] – status zamówienia, przekazywany bez zamian pomiędzy systemami

[DATA_MAX_INS_UPD] – znacznik czasowy utworzenia/modyfikacji rekordu, używany do wykrywania modyfikacji widoku

[Oryginalny_ID] – oryginalny identyfikator zamówienia, używany w przypadku rozbicia zamówień na pod zamówienia np. z różnymi datami, ilościami czy statusami

[DYSPR_ID] – pole nieużywane

[DYSPR_ILOSC_WYDANA] – pole nieużywane

[DYSPR_ILOSC_ODSTAWIONA] – pole nieużywane

[ZA_ID] – identyfikator COOLZAM zamówienia produktu do dostawcy, z którego pochodzi zamówiony przez klienta produkt

[DOS_ID] – identyfikator COOLZAM dostawy produktu od dostawcy, z której pochodzi zamówiony przez klienta produkt

v_presta_produkt_2

Widok „v_presta_produkt_2” zawiera listę produktów z COOLZAM wraz z informacjami o nich, wymaganymi na potrzeby integracji

Odczyt/zapis: tylko odczyt

Kierunek wywoływania: ESB -> COOLZAM

Kierunek przepływu danych: COOLZAM -> ESB

Lista pól na widoku:

[ID_KOLORU] – identyfikator COOLZAM wariantu kolorystycznego produktu

[ID_PRODUKTU] – identyfikator COOLZAM produktu

[DOSTEPNY] – flaga określająca czy produkt jest w danym momencie na magazynie

[EAN] – numer EAN produktu

[PR_NUMERPRODUKTU] – krótka nazwa produktu/kod produktu

[PR_NAZWAPL] – pełna polska nazwa produktu

[PR_NAZWAENG] – pełna angielska nazwa produktu

[PR_CENA] – cena produktu

[PR_TIMESTAMP_v2] – znacznik czasowy utworzenia/modyfikacji rekordu, używany do wykrywania modyfikacji widoku

[PAKOWANIE] – skrótowy opis pakowania produktu

[PAKOWANIE_DOD] – opcjonalne dodatkowe pakowanie produktu

[PR_SMH] – wysokość produktu/rozmiar

[PR_SMO] – średnica produktu

[DO_NAZWADOSTAWCY] – nazwa dostawcy, od którego pochodzi dany produkt

[pr_do_id] – identyfikator COOLZAM danego dostawcy produktu

[NieAktywne] – flaga określająca czy dana kombinacja identyfikatora produktu i koloru została zdezaktywowana/usunięta

[PR_KAT_ID] – kategoria produktu pierwszego poziomu (obsługa kategorii została zmieniona, pole przestało być wykorzystywane w COOLZAM, lecz pozostało na interfejsach i nadal bierze udział w komunikacji pomiędzy systemami)

[PR_KAT_POD_ID] – kategoria produktu pierwszego poziomu (uwaga j.w.)

[PR_KAT_POD_POD_ID] – kategoria produktu pierwszego poziomu (uwaga j.w.)

v_presta_lista_dostaw

Widok „v_presta_lista_dostaw” zawiera listę dostaw produktu do CF, w rozbiciu na konkretne pozycje dostawy dla konkretnego produktu w konkretnym wariantcie kolorystycznym

Odczyt/zapis: tylko odczyt

Kierunek wywoływania: ESB -> COOLZAM

Kierunek przepływu danych: COOLZAM -> ESB

Lista pól na widoku:

[ID] – unikalny identyfikator pozycji dostawy (potrzebny gdyż kombinacja [DOS_ID + KO_ID + PR_ID] może się powtórzyć), aktualnie nie przesyłany do PRESTA

[DOS_ID] – identyfikator dostawy produktu do CF

[NR_Faktury] – numer danej dostawy/numer dokumentu powiązanego

[KO_ID] – identyfikator COOLZAM wariantu kolorystycznego produktu w danej dostawie

[PR_ID] – identyfikator COOLZAM produktu w danej dostawie

[DATA_MAX_INS_UPD] – znacznik czasowy utworzenia/modyfikacji rekordu, używany do wykrywania modyfikacji widoku

[DOS_DATA_DOSTAWY] – planowana data dostawy danej dostawy produktu do CF

[ILOSC] – ilość danego produktu w danym wariantcie kolorystycznym dla danej dostawy

[Cena] – cena danego produktu w danym wariantcie kolorystycznym dla danej dostawy

[ZA_ID] – identyfikator COOLZAM zamówienia produktu do dostawcy, z którego pochodzi dana pozycja produktowa w dostawie

[Doplynelo] – flaga określająca czy dana dostawa produktu została dostarczona

[Sprzedane] – flaga określająca czy dany produkt z danej dostawy został wysprzedany

[PAKOWANIE] – informacja o sposobie pakowania produktu w tej konkretnej dostawie

[PAKOWANIE_DOD] – dodatkowa informacja o sposobie pakowania dla produktu w tej konkretnej dostawie

[USUNIETO] – flaga określająca czy dana pozycja produktowa została skasowana z tej dostawy

[DOS_DATA_ETA] – szacowana data przybycia dostawy

[ILOSC_SZT] – całkowita ilość danego produktu w danej pozycji dostawy

[ILOSC_DOSTEPNE] – ilość danego produktu w danej pozycji dostawy, która pozostała dostępna do sprzedaży

[v_presta_lista_zamowien](#)

Widok „v_presta_lista_zamowien” zawiera listę zamówień produktów do dostawców, w rozbiciu na konkretne pozycje zamówienia dla konkretnego produktu w konkretnym wariantcie kolorystycznym.

Odczyt/zapis: tylko odczyt

Kierunek wywoływania: ESB -> COOLZAM

Kierunek przepływu danych: COOLZAM -> ESB

Lista pól na widoku:

-
- [ID]** – nieużywane, wewnętrzny identyfikator COOLZAM
- [ZA_ID]** – identyfikator COOLZAM zamówienia do dostawcy
- [DO_ID]** – nieużywane
- [MIESIAC]** – planowany miesiąc dostarczenia zamówienia w danym roku
- [TYDZIEN]** – planowany numer tygodnia dostarczenia zamówienia w danym miesiącu danego roku
- [ROK]** – planowany rok dostarczenia zamówienia
- [KO_ID]** – identyfikator COOLZAM wariantu kolorystycznego produktu w zamówieniu
- [PR_ID]** – identyfikator COOLZAM produktu w zamówieniu
- [DATA_MAX_INS_UPD]** – znacznik czasowy utworzenia/modyfikacji rekordu, używany do wykrywania modyfikacji widoku
- [ILOSC_DOSTEPNE]** – aktualnie dostępna ilość produktu dla danego zamówienia
- [CO_CENA]** – cena produktu w danym zamówieniu
- [ZAPOZ_ILOSC_BEGIN]** – całkowita ilość produktu w danym zamówieniu
- [PAKOWANIE]** – informacja o sposobie pakowania produktu w tym konkretnym zamówieniu
- [PAKOWANIE_DOD]** – dodatkowa informacja o sposobie pakowania dla produktu w tym konkretnym zamówieniu
- [ZA_NUMERZAMOWIENIA]** – numer zamówienia do dostawcy
- [USUNIETO]** – flaga określająca czy dana pozycja produktowa została skasowana z tego zamówienia
- [SPRZEDANE]** – flaga określająca czy dana pozycja zamówienia dostawcy została wysprzedana
- [ILOSC_SZT]** – „Ilość sztuk w pakowaniu”
-

[CZY_AKTYWNY] – flaga określająca czy dana pozycja zamówienia „jest aktywna”

[v_presta_zestaw_kategoria_produkt](#)

Widok „v_presta_zestaw_kategoria_produkt” zawiera zestawy kategorii produktowych. Kategorie zostały wydzielone do nowego widoku w COOLZAM wraz z koniecznością przypisania wielu hierarchii kategorii dla pojedynczego produktu. Dana hierarchia (kategoria 1-2-3) jest powiązana z identyfikatorem produktu bez względu na wariant kolorystyczny.

Odczyt/zapis: tylko odczyt

Kierunek wywoływania: ESB -> COOLZAM

Kierunek przepływu danych: COOLZAM -> ESB

Lista pól na widoku:

[KATZ_ID] – identyfikator COOLZAM zestawu kategorii

[KATZ_PR_ID] – identyfikator COOLZAM produktu, którego dotyczy zestaw kategorii

[KATZ_KAT_ID] – identyfikator kategorii pierwszego poziomu

[KATZ_PODK_ID] – identyfikator kategorii drugiego poziomu

[KATZ_PODPODK_ID] – identyfikator kategorii trzeciego poziomu

[KATZ_TIMESTAMP_V2] – znacznik czasowy utworzenia/modyfikacji rekordu, używany do wykrywania modyfikacji rekordu

[KATZ_AKTYWNY] – flaga określająca czy dany zestaw kategorii jest aktywny, nieaktywny
zestaw traktowany jest jako usunięty

[v_presta_kategoria_nowa](#)

Widok „v_presta_kategoria_nowa” zawiera nazwy kategorii produktowych pierwszego poziomu.

Odczyt/zapis: tylko odczyt

Kierunek wywoływania: ESB -> COOLZAM

Kierunek przepływu danych: COOLZAM -> ESB

Lista pól na widoku:

[KA_ID] – identyfikator COOLZAM kategorii pierwszego poziomu

[KA_NAZWAKATEGORI] – nazwa kategorii

[KA_TIMESTAMP_V2] – znacznik czasowy utworzenia/modyfikacji rekordu, używany do wykrywania modyfikacji rekordu

[KA_USUNIETO] – flaga określająca czy dany zestaw kategorii jest aktywny, nieaktywny zestaw traktowany jest jako usunięty

[v_presta_podkategoria_nowa](#)

Widok „v_presta_podkategoria_nowa” zawiera nazwy kategorii produktowych drugiego poziomu.

Odczyt/zapis: tylko odczyt

Kierunek wywoływania: ESB -> COOLZAM

Kierunek przepływu danych: COOLZAM -> ESB

Lista pól na widoku:

[POD_ID] – identyfikator COOLZAM kategorii drugiego poziomu

[POD_NAZWAPODKATEGORI] – nazwa kategorii

[POD_TIMESTAMP_V2] – znacznik czasowy utworzenia/modyfikacji rekordu, używany do wykrywania modyfikacji rekordu

[POD_KA_ID] – identyfikator COOLZAM kategorii pierwszego poziomu, z którą powiązana jest dana kategoria drugiego poziomu, inaczej nad-kategoria w hierarchii

[KA_USUNIETO] – flaga określająca czy dany zestaw kategorii jest aktywny, nieaktywny zestaw traktowany jest jako usunięty

[v_presta_podpodkategoria_nowa](#)

Widok „v_presta_podpodkategoria_nowa” zawiera nazwy kategorii produktowych trzeciego poziomu.

Odczyt/zapis: tylko odczyt

Kierunek wywoływania: ESB -> COOLZAM

Kierunek przepływu danych: COOLZAM -> ESB

Lista pól na widoku:

[POD_POD_ID] – identyfikator COOLZAM kategorii trzeciego poziomu

[POD_POD_NAZWAKATEGORI] – nazwa kategorii

[POD_POD_LACZ_ID] – identyfikator COOLZAM kategorii drugiego poziomu, z którą powiązana jest dana kategoria trzeciego poziomu, inaczej nad-kategoria w hierarchii

[POD_POD_TIMESTAMP_V2] – znacznik czasowy utworzenia/modyfikacji rekordu, używany do wykrywania modyfikacji rekordu

v_presta_klient_dodawanie

Widok „v_presta_klient_dodawanie” służy do przesyłania do COOLZAM nowo utworzonych klientów z systemu PRESTA. Po załadowaniu klienta do widoku przydzielany jest mu nowy identyfikator COOLZAM, który natychmiastowo jest przekazywany do systemu PRESTA

Odczyt/zapis: tylko zapis

Kierunek wywoływania: ESB -> COOLZAM

Kierunek przepływu danych: ESB -> COOLZAM (dane klienta), COOLZAM -> ESB (identyfikator COOLZAM klienta)

Lista pól na widoku:

[KL_ID] – identyfikator COOLZAM klienta

[NIP] – numer NIP klienta

[EMAIL] – adres e-mail klienta

[ADRE_DOST_IMIE] – imię pracownika klienta dla adresu dostawy

[ADRE_DOST_NAZWISKO] – nazwisko pracownika klienta dla adresu dostawy

[ADRE_DOST_FIRMA] – nazwa firmy, do której ma być realizowana dostawa

[ADRE_DOST_ADRES] – linia adresowa, zawierająca ulicę, numer i numer lokalu miejsca dostawy

[ADRE_DOST_KOD_POCZTOWY] – kod pocztowy miejsca dostawy

[ADRE_DOST_MIASTO] – miasto adresu dostawy

[ADRE_FAKT_IMIE] – imię pracownika klienta dla adresu faktury (kontakt płatnika faktury)

[ADRE_FAKT_NAZWISKO] – nazwisko pracownika klienta dla adresu faktury (kontakt płatnika faktury)

[ADRE_FAKT_FIRMA] – nazwa firmy, na którą ma być wystawiona faktura (płatnik faktury)

[ADRE_FAKT_ADRES] – linia adresowa, zawierająca ulicę, numer i numer lokalu płatnika faktury

[ADRE_FAKT_KOD_POCZTOWY] – kod pocztowy płatnika faktury

[ADRE_FAKT_MIASTO] – miasto płatnika faktury

[KL_TIMESTAMP_V2] – znacznik czasowy utworzenia klienta w bazie COOLZAM

[KL_DO_PRESTA] – flaga określająca czy klient jest klientem PRESTA, ponieważ do widoku są zapisywani wyłącznie nowi klienci z PRESTA, to pole ma stałą wartość

[KL_STATUS] – status klienta, aktualnie nieużywane, wszyscy klienci z PRESTA trafiają do COOLZAM ze statusem „NIEZWERYFIKOWANY”

[KL_TEL1] – pierwsze pole z numerem telefonu klienta

[KL_TEL2] – drugie pole z numerem telefonu klienta

[KL_TEL3] – trzecie pole z numerem telefonu klienta

[KL_OP1] – opis pierwszego pola z numerem telefonu klienta

[KL_OP2] – opis drugiego pola z numerem telefonu klienta

[KL_OP3] – opis trzeciego pola z numerem telefonu klienta

[v_presta_zamowienie](#)

Widok „v_presta_zamowienie” służy do przesyłania do COOLZAM nowo utworzonych zamówień (od klientów) z systemu PRESTA

Odczyt/zapis: tylko zapis

Kierunek wywoływania: ESB -> COOLZAM

Kierunek przepływu danych: ESB -> COOLZAM

Lista pól na widoku:

[ID_PRE_ZAM] – nieużywane

[ID_ZAMOWIENIA_PRESTA] – identyfikator zamówienia pochodzący z PRESTA

[DATA_DOSTAWY] – nieużywane

[NIP_KLIENTA] – numer nip klienta, na którego jest zamówienie

[EAN] – numer EAN zamawianego produktu

[ID_KOLORU_GLOWNY] – identyfikator COOLZAM wariantu kolorystycznego zamawianego produktu

[ILOSC_Z_PRESTA] – ilość zamawianego produktu

[ILOSC_Z_COOLZAM_POTW] – nieużywane

[STATUS] – status nowego zamówienia z PRESTA, przekazywany w niezmienionej formie

[ID_ZAMOWIENIA_PRESTA_ORYG] – nieużywane, nowe zamówienie nie posiada nigdy

[ID_PRODUKT] – identyfikator COOLZAM zamawianego produktu

[KL_ID] – identyfikator COOLZAM klienta, na którego jest zamówienie

[ZA_TY_ID] – nieużywane

[ZA_MIE_ID] – nieużywane

[ZA_ROK] – nieużywane

[DATA_ZAMOWIENIA] – data utworzenia zamówienia

[DATA_TIMESTAMP_V2] – nieużywane

[ZA_ID] – identyfikator COOLZAM zamówienia produktu, do którego odnosi się zamówienie klienta

[DOS_ID] – identyfikator COOLZAM dostawy produktu, do której odnosi się zamówienie klienta

[v_presta_kolory](#)

Widok „v_presta_kolory” zawiera listę kolorów głównych (wariantów kolorystycznych), zarówno pojedynczych kolorów jak i mieszanek kolorystycznych

Odczyt/zapis: tylko odczyt

Kierunek wywoływania: ESB -> COOLZAM

Kierunek przepływu danych: COOLZAM -> ESB

Lista pól na widoku:

[KO_ID] – identyfikator COOLZAM wariantu kolorystycznego

[KO_NAZWAKOLORU] – nazwa koloru/mieszanki kolorów

[MIX] – flaga określająca czy dany wariant kolorystyczny jest pojedynczym kolorem lub mieszanką kolorów

[DATA_MAX] – nieużywane

[v_presta_mixkolor](#)

Widok „v_presta_mixkolor” zawiera listę identyfikatorów kolorów składowych dla wariantów produktowych, które są mieszankami kolorystycznymi

Odczyt/zapis: tylko odczyt

Kierunek wywoływania: ESB -> COOLZAM

Kierunek przepływu danych: COOLZAM -> ESB

Lista pól na widoku:

[MI_ID] – nieużywane, wewnętrzny identyfikator COOLZAM

[MI_KO_GLOWNY_ID] – Identyfikator COOLZAM wariantu kolorystycznego, który jest mieszanką kolorów

[MI_KO_ID] – identyfikator COOLZAM wariantu kolorystycznego, który jest kolorem składowym mieszanki kolorów

[DATA_MAX] – nieużywane

PRESTA

CreateOrUpdateCategory

Interfejs przekazujący do systemu PRESTA informacje o nowych i zaktualizowanych kategoriach

Kierunek wywoływania: ESB -> PRESTA

Kierunek przepływu danych: ESB -> PRESTA

Pola dostępne w zapytaniu:

- **atrybut action** – atrybut przyjmujący wartości „upsert” lub „delete” – zmiana wdrożona na ESB, nigdy nie uruchomiona w systemie PRESTA

-
- **catLevel** – poziom kategorii, aktualnie istnieją trzy poziomy – pole przyjmuje numeryczne wartości: 1,2,3
 - **catId** – identyfikator kategorii danego poziomu, unikalny tylko w połączeniu z poziomem kategorii
 - **catName** – nazwa kategorii
 - **supercatId** – identyfikator COOLZAM kategorii wyższego poziomu, nad-kategoria dla danej kategorii

Pola dostępne w odpowiedzi:

- **result** – wynik wykonania operacji, pozytywny lub negatywny
- **message** – czytelny dla człowieka wynik wykonania operacji lub powód jej niewykonania

CreateOrUpdateCustomer

Interfejs przekazujący do systemu PRESTA informacje o nowych i zaktualizowanych klientach

Kierunek wywoływania: ESB -> PRESTA

Kierunek przepływu danych: ESB -> PRESTA

Pola dostępne w zapytaniu:

- **customerId** – identyfikator COOLZAM klienta
- **nip** – numer NIP klienta
- **companyName** – Nazwa firmy klienta
- **emailAddress** – adres e-mail klienta, jest to główny adres e-mail danego klienta powiązany z kontem w systemie PRESTA
- **tel1** – pierwsze pole z numerem telefonu klienta

-
- **tel1Desc** – opis pierwszego pola z numerem telefonu klienta
 - **tel2** – drugie pole z numerem telefonu klienta
 - **tel2Desc** – opis drugiego pola z numerem telefonu klienta
 - **tel3** – trzecie pole z numerem telefonu klienta
 - **tel3Desc** – opis trzeciego pola z numerem telefonu klienta
 - **deliveryAddress**
 - **firstname** – imię pracownika klienta dla adresu dostawy
 - **lastname** – nazwisko pracownika klienta dla adresu dostawy
 - **companyName** – nazwa firmy, do której ma być realizowana dostawa
 - **addressLine** – linia adresowa, zawierająca ulicę, numer i numer lokalu miejsca dostawy
 - **postalCode** – kod pocztowy miejsca dostawy
 - **city** – miasto adresu dostawy
 - **invoiceAddress**
 - **firstname** – imię pracownika klienta dla adresu faktury (kontakt płatnika faktury)
 - **lastname** – nazwisko pracownika klienta dla adresu faktury (kontakt płatnika faktury)
 - **companyName** – nazwa firmy, na którą ma być wystawiona faktura (płatnik faktury)
 - **addressLine** – linia adresowa, zawierająca ulicę, numer i numer lokalu płatnika faktury
 - **postalCode** – kod pocztowy płatnika faktury
 - **city** – miasto płatnika faktury
 - **status** – pole nieużywane, status klienta jest zarządzany przez PRESTA, pole aktualnie zawiera statyczną wartość „NIEAKTYWNY”

Pola dostępne w odpowiedzi:

- **result** – wynik wykonania operacji, pozytywny lub negatywny
- **message** – czytelny dla człowieka wynik wykonania operacji lub powód jej niewykonania

CreateOrUpdateOrder

Interfejs przekazujący do systemu PRESTA informacje o nowych i zaktualizowanych zamówieniach

Kierunek wywoływania: ESB -> PRESTA

Kierunek przepływu danych: ESB -> PRESTA

Pola dostępne w zapytaniu:

- **orderId** – identyfikator zamówienia, pochodzący z systemu PRESTA lub systemu COOLZAM
- **createDate** – data utworzenia zamówienia klienta
- **customerId** – identyfikator COOLZAM klienta, który składa zamówienie
- **ean** – numer EAN produktu, na który klient składa zamówienie (aktualnie nie wykorzystywane)
- **prodId** – identyfikator COOLZAM produktu, na który klient składa zamówienie
- **colorId** – identyfikator COOLZAM wariantu kolorystycznego produktu, na który klient składa zamówienie
- **productOrderId** – pole opcjonalne, identyfikator COOLZAM zamówienia produktu (od CF do dostawcy), wypełniany, gdy produkt ma pochodzić z konkretnego zamówienia produktu do dostawcy
- **productDeliveryId** – pole opcjonalne, identyfikator COOLZAM dostawy produktu (od dostawcy do CF), wypełniany, gdy produkt ma pochodzić z konkretnej dostawy produktu od dostawcy
- **quantity** – ilość jednostek danego produktu zamówionego przez klienta
- **status** – status zamówienia klienta, system ESB nie ingeruje w treść statusu, status jest przekazywany w takiej samej formie pomiędzy systemami
- **originalId** – oryginalny identyfikator zamówienia, używany jedynie w przypadku rozbicia zamówień na pod-zamówienia np. z różnymi datami, ilościami czy statusami
- **qtyDiff** – wirtualna ilość zamówienia, zawiera różnicę pomiędzy oryginalną ilością na zamówieniu a aktualną ilością, po jej zmniejszeniu lub rozbiciu zamówienia

Pola dostępne w odpowiedzi:

- **result** – wynik wykonania operacji, pozytywny lub negatywny
- **message** – czytelny dla człowieka wynik wykonania operacji lub powód jej niewykonania

CreateOrUpdateProduct

Interfejs przekazujący do systemu PRESTA informacje o nowych, zaktualizowanych i usuniętych produktach. Elementy „categorySets”, „productDeliveries”, „productOrders” mogą w przesłanej strukturze pojawiać się wielokrotnie, lub nie pojawiać się w ogóle (0..*). Istotną kwestią wynikającą ze specyfiki działania systemu źródłowego jest fakt, że dla danego produktu (**idProduct**, **idColor**) może pojawić się wiele pozycji dostawy z takim samym identyfikatorem dostawy, w takich przypadkach jednak identyfikator zamówienia, do którego odnosi się pozycja dostawy będzie zawsze różny – unikalna jest wyłącznie para (**id**, **productOrderId**). Interfejs ten od swojej początkowej wersji był wielokrotnie rozbudowywany o kolejne elementy, jego aktualna forma ma znaczny wpływ na wydajność komunikacji i powoduje zbędne przetwarzanie dużej ilości danych, przy najbliższej możliwości sugerowane jest jego rozbitcie tego interfejsu na kilka mniejszych interfejsów, wydzielając co najmniej przesyłanie informacji o dostawach i terminach (zamówieniach do dostawców) do osobnych interfejsów.

Kierunek wywoływania: ESB -> PRESTA

Kierunek przepływu danych: ESB -> PRESTA

Pola dostępne w zapytaniu:

- **atrybut action** – określa rodzaj operacji na produkcie „upsert” – stworzenie lub aktualizacja produkt, „delete” – usunięcie produktu, w przypadku akcji „delete” wysyłane są jedynie dane

produktu najwyższego poziomu, bez elementów tj. „categorySets”, „productDeliveries”, „productOrders”

- **idProduct** – identyfikator COOLZAM produktu
- **idColor** – identyfikator COOLZAM wariantu kolorystycznego produktu
- **ean** – numer EAN produktu
- **sku** – krótka nazwa produktu/kod produktu
- **namePL** – pełna polska nazwa produktu
- **nameEN** – pełna angielska nazwa produktu
- **package** – skrótowy opis pakowania produktu
- **available** – flaga określająca czy produkt jest w danym momencie na magazynie
- **price** – cena produktu
- **addIPackage** – opcjonalne dodatkowe pakowanie produktu
- **height** – wysokość produktu/rozmiar
- **diameter** – średnica produktu
- **idVendor** – identyfikator COOLZAM danego dostawcy produktu
- **idCatL1** – identyfikator kategorii pierwszego poziomu wg wcześniejszych założeń (jeden produkt – jedna hierarchia), aktualnie używane są zestawy kategorii („categorySets”)
- **idCatL2** – identyfikator kategorii drugiego poziomu wg wcześniejszych założeń (uwagi j.w.)
- **idCatL3** – identyfikator kategorii trzeciego poziomu wg wcześniejszych założeń (uwagi j.w.)
- **categorySets**
 - **idCatSet** – identyfikator COOLZAM zestawu kategorii
 - **idCatL1** – identyfikator kategorii pierwszego poziomu w zestawie kategorii
 - **idCatL2** – identyfikator kategorii drugiego poziomu w zestawie kategorii
 - **idCatL3** – identyfikator kategorii trzeciego poziomu w zestawie kategorii
- **productDeliveries**
 - **id** – identyfikator COOLZAM dostawy produktu do CF, w której ten produkt się znajduje

-
- **productOrderId** – identyfikator COOLZAM zamówienia do dostawcy, powiązanego z tą dostawą produktu
 - **deliveryNo** – numer dostawy/ numer dokumentu powiązanego
 - **date** – planowana data dostarczenia dostawy produktu
 - **dateETAozarow** – planowana data dostarczenia dostawy produktu do magazynu końcowego (transport lądowy do Ożarowa)
 - **quantity** – aktualna pozostała do sprzedaży ilość produktu w danej pozycji dostawy
 - **totalQuantity** – całkowita początkowa ilość produktu w danej pozycji dostawy
 - **price** – cena produktu w danej pozycji dostawy
 - **arrived** – flaga określająca czy dana dostawa została dostarczona
 - **soldOut** – flaga określająca czy produkt z danej pozycji dostawy został wysprzedany
 - **package** – informacja o sposobie pakowania produktu w tej konkretnej dostawie
 - **addlPackage** – dodatkowa informacja o sposobie pakowania dla produktu w tej konkretnej dostawie
 - **pcsInBox** – ilość sztuk w pakowaniu
- **productOrders**
 - **id** – identyfikator COOLZAM zamówienia do dostawcy
 - **orderNo** – numer zamówienia do dostawcy
 - **date** – planowany termin dostarczenia zamówienia, w formacie RRRR-MM-T (rok, miesiąc, tydzień)
 - **quantity** – aktualnie dostępna ilość produktu dla danego zamówienia
 - **totalQuantity** – całkowita ilość produktu w danym zamówieniu
 - **price** – cena produktu w danym zamówieniu
 - **soldOut** – flaga określająca czy produkt z danego zamówienia został wysprzedany
 - **package** – informacja o sposobie pakowania produktu w tym konkretnym zamówieniu
 - **addlPackage** – dodatkowa informacja o sposobie pakowania dla produktu w tym konkretnym zamówieniu
-

-
- **pcsInBox** – ilość sztuk w pakowaniu
 - **isActive** – flaga określająca czy dane zamówienie jest aktywne

Pola dostępne w odpowiedzi:

- **result** – wynik wykonania operacji, pozytywny lub negatywny
- **message** – czytelny dla człowieka wynik wykonania operacji lub powód jej niewykonania

Moduły synchronizacji

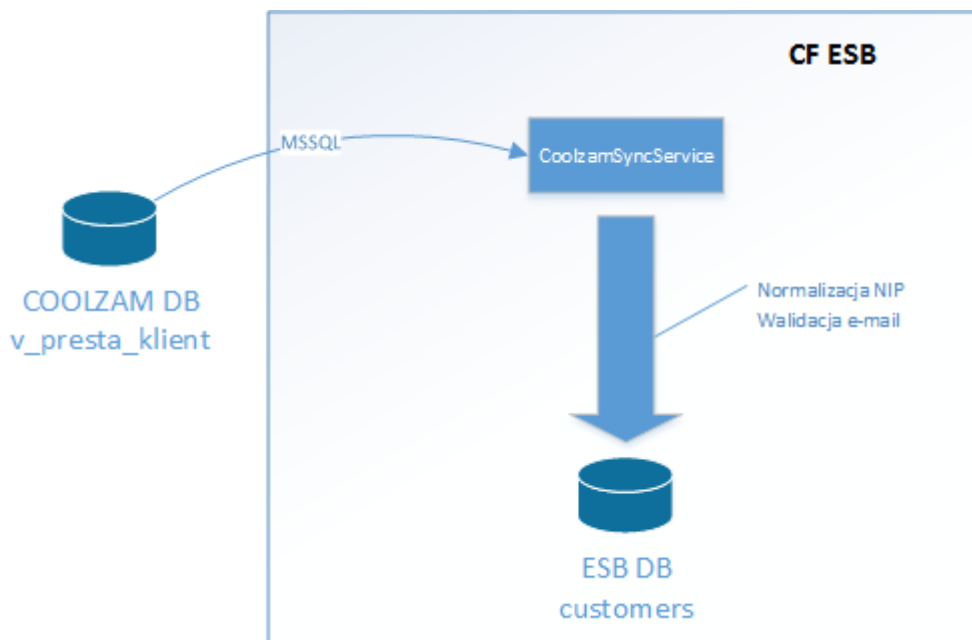
CoolzamSyncService

Moduł odpowiada za synchronizację z ESB zmian występujących w COOLZAM. Zajmuje się też wstępną walidacją i transformacją danych. Działa na zasadzie nasłuchiwanie zmian na tabelach interfejsowych COOLZAM, czekając na pojawienie się rekordów zmodyfikowanych lub utworzonych po danym progu czasowym (na podstawie znaczników czasowych z tabel interfejsowych). Próg czasowy dla nasłuchiwanie nowych zmian, określony jest na podstawie najświeższego znacznika czasowego ostatniej pomyślnej synchronizacji. Dzięki ciągłemu nasłuchowi zmian średnie opóźnienie pomiędzy pojawieniem się zmian w COOLZAM a wprowadzeniem ich do ESB jest zminimalizowane do czasów poniżej jednej minuty. Dzięki znacznikom czasowym na tabelach interfejsowych COOLZAM nie ma

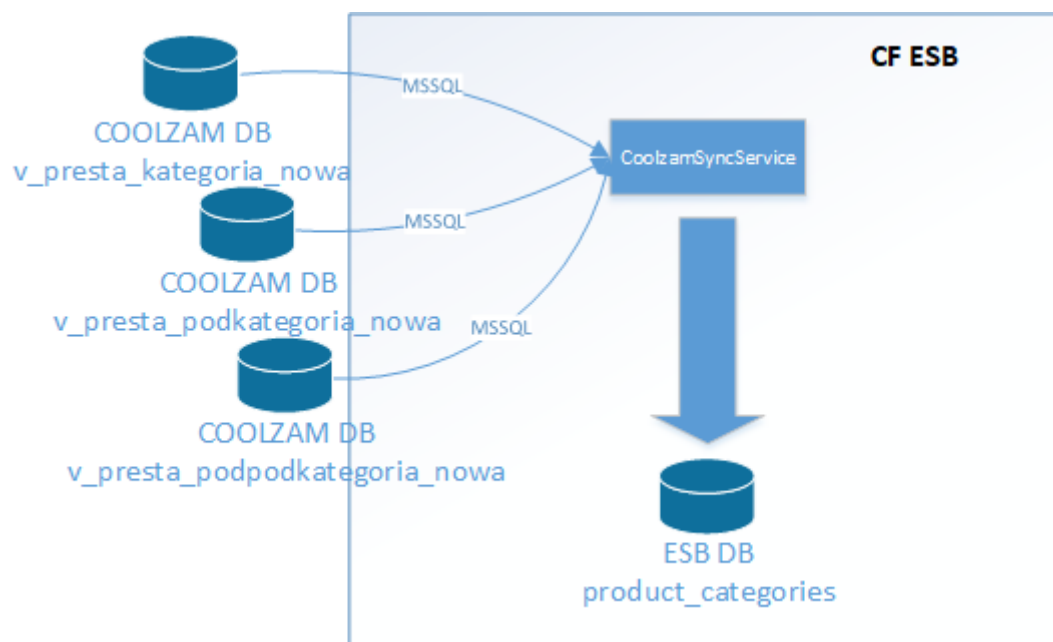
konieczności każdorazowego pobierania pełnej tabeli danych, co zmniejsza objętość komunikacji i obciążenie systemów.

Usługa synchronizacji COOLZAM podzielona jest na podmoduły, które działają równolegle w swoich własnych pętlach.

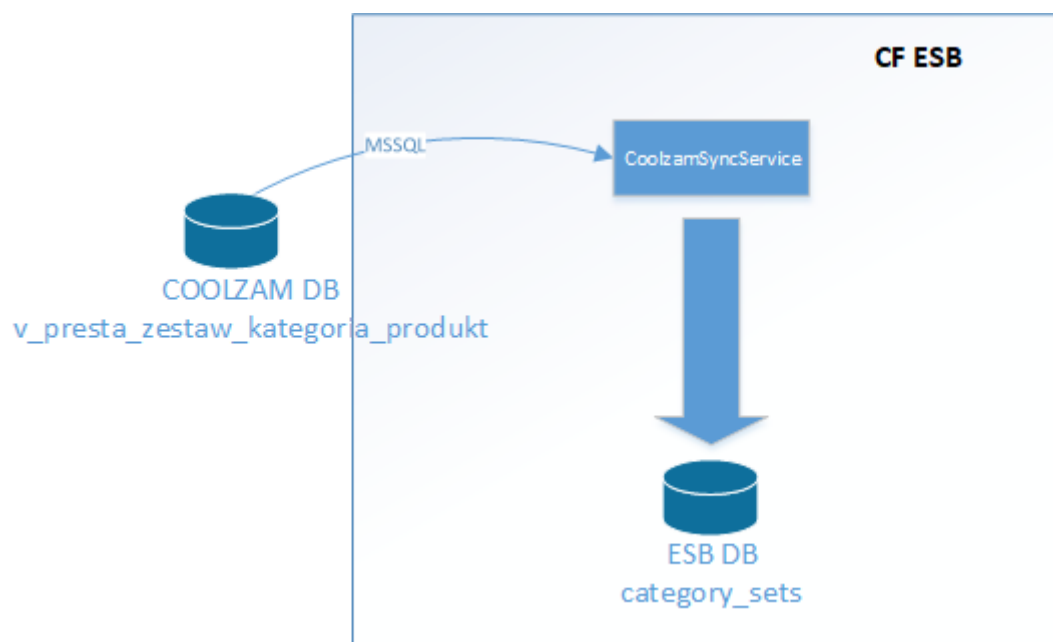
Klienci:



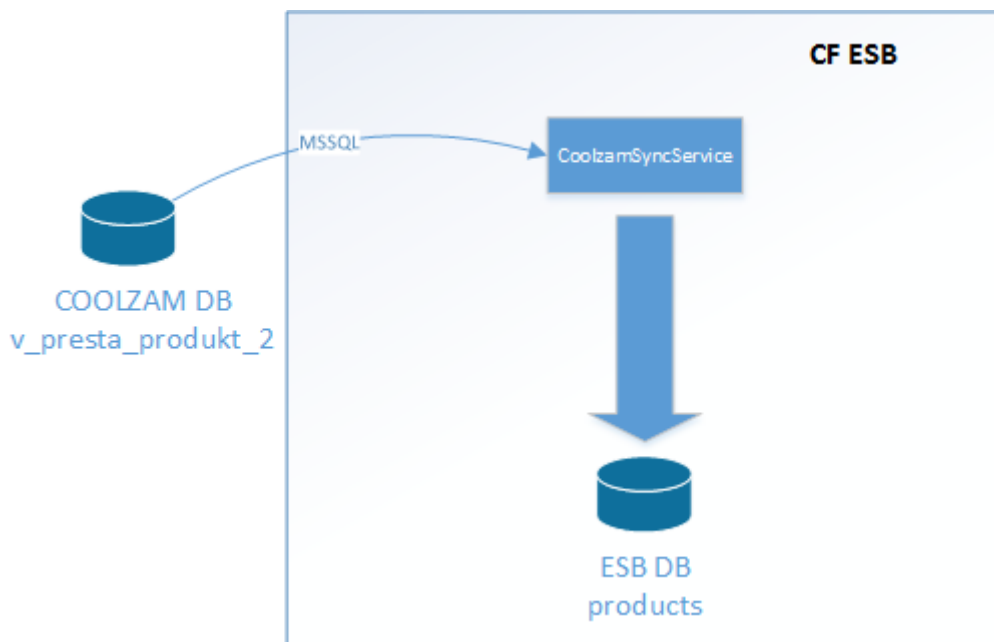
Kategorie produktów:



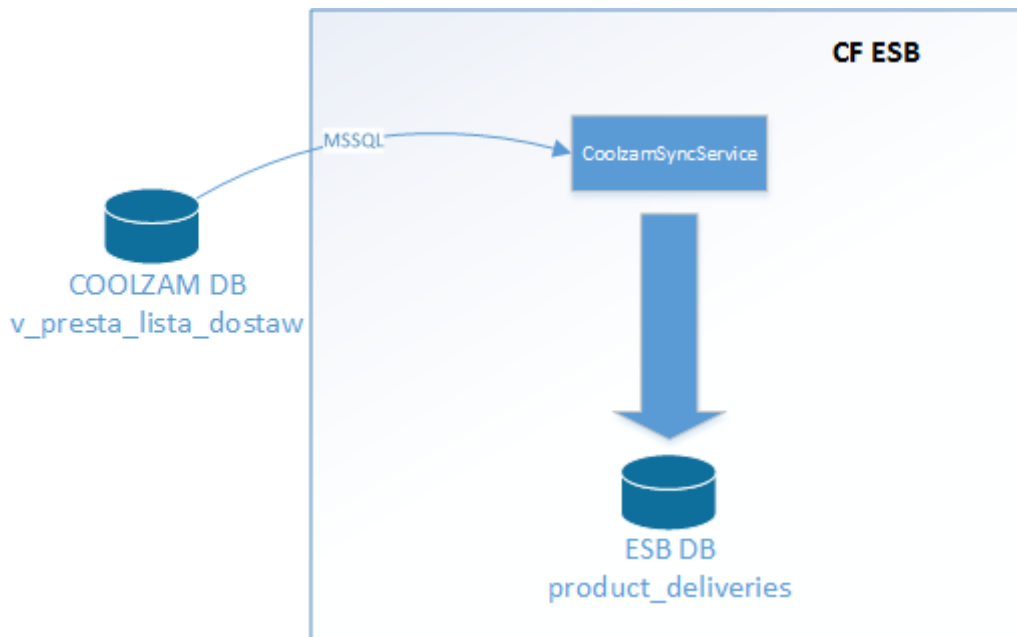
Zestawy kategorii:



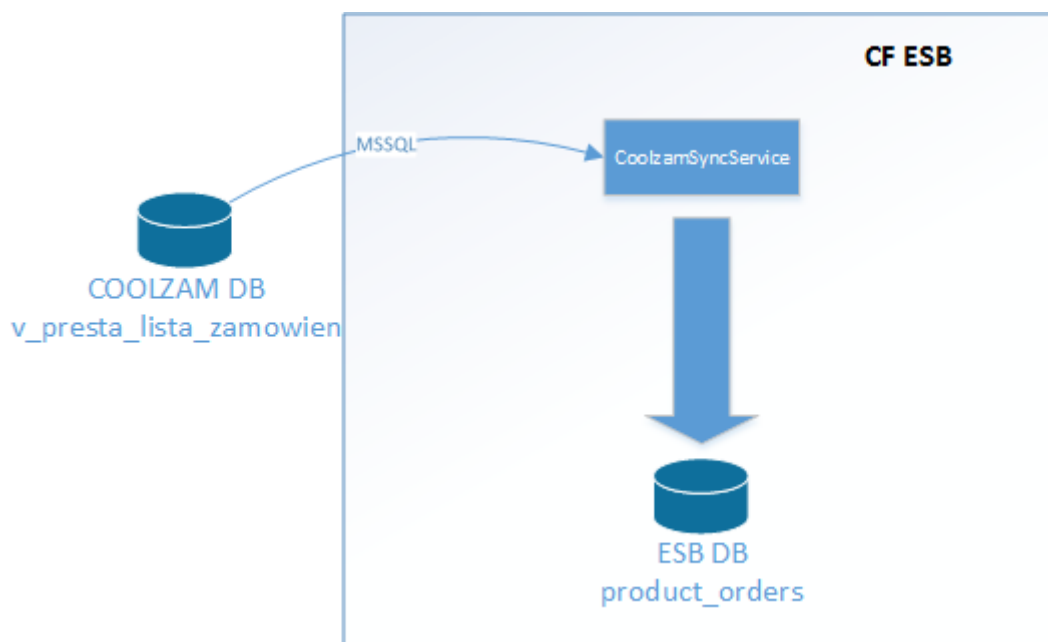
Produkty:



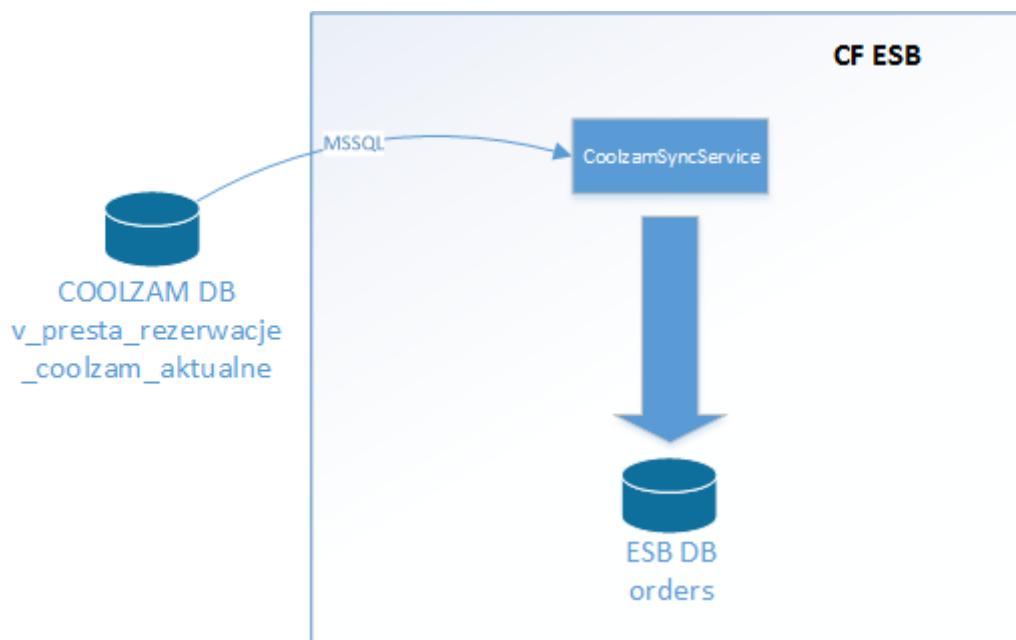
Dostawy produktów:



Zamówienia produktów:

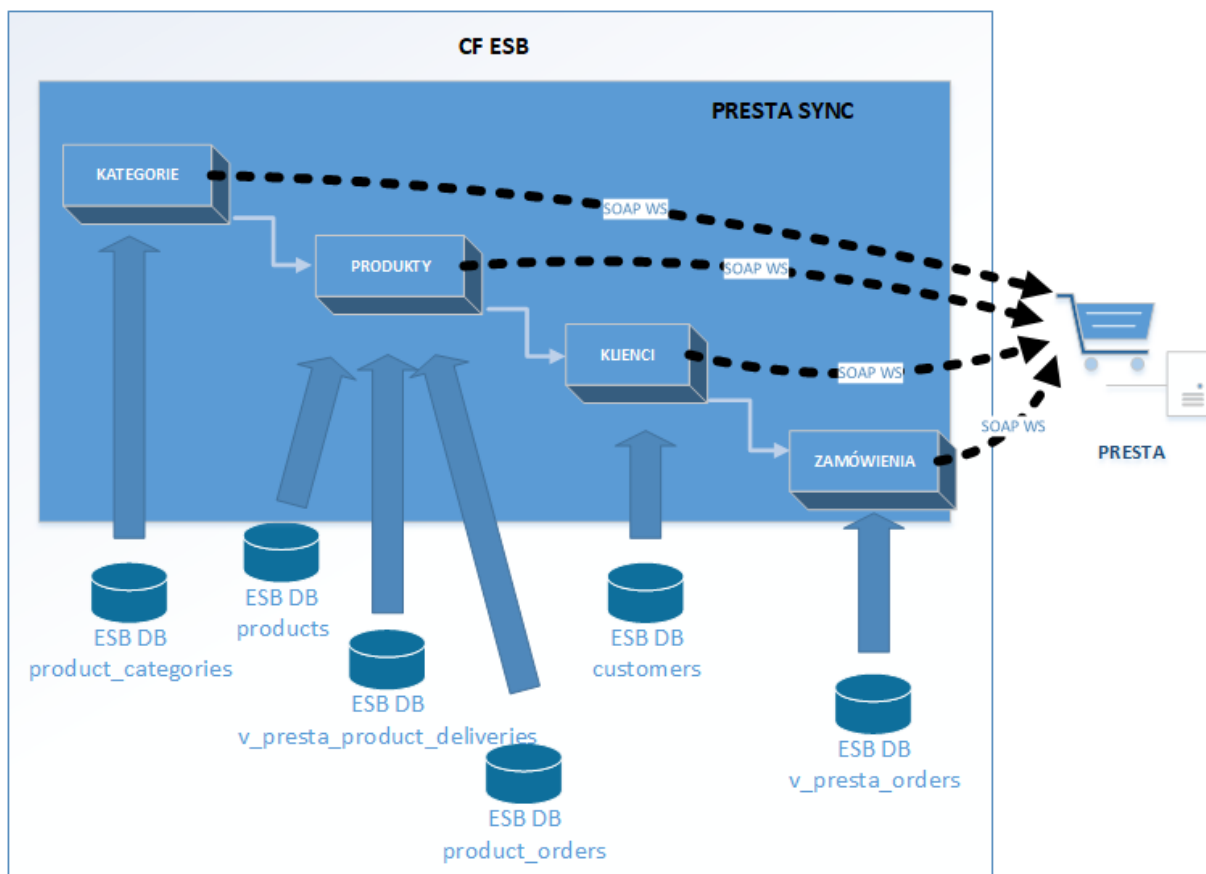


Zamówienia klienta:



PrestaSync

Moduł odpowiada za synchronizację zmian pomiędzy ESB a systemem PRESTA. Zajmuje się również filtrowaniem i kwalifikacją danych, które w danym momencie nadają się do przesłania do PRESTA. Proces synchronizacji uruchamiany jest okresowo, co 15 minut. Synchronizacja działa zawsze sekwencyjnie tzn. w pierwszej kolejności synchronizowane są zmiany produktów wraz z dostawami i zamówieniami produktów, następnie zmiany klientów, a w ostatniej kolejności zmiany zamówień.



Warunki, które muszą być spełnione, aby dany klient był synchronizowany z PRESTA:

- Klient w swoim polu adresu e-mail, musi posiadać pojedynczy i poprawny adres e-mail
- Klient musi posiadać prawidłowy numer NIP tzn. musi być w jednym z dwóch formatów:

-
- [0-9]{10} - polskie NIPy
 - [A-Z][A-Z][A-Z0-9]{8,12} – zagraniczne

System dokonuje wstępnej standaryzacji numeru NIP (usuwanie '-', usuwanie przedrostka 'PL', ...) i dopiero po tej operacji weryfikowana jest zgodność z powyższymi formatami. Ma to na celu zapewnienie pomyślnej synchronizacji klienta, gdy numer NIP jest poprawny, ale nie występuje w standardowym formacie.

Warunki, które muszą być spełnione, aby dostawa produktu, była przesłana razem z produktem do PRESTA:

- Dostawa nie może być oznaczona jako usunięta
- Zamówienie produktu, z którego pochodzi dana dostawa musi istnieć i nie może być oznaczone, jako usunięte

Warunki, które muszą zostać spełnione, aby zamówienie klienta było synchronizowane z PRESTA:

- Klient, na którego jest zamówienie musi istnieć i być klientem, który kwalifikuje się do synchronizacji z PRESTA
- Aktualne dane klienta, zostały wcześniej zsynchronizowane
- Aktualne dane produktu zostały wcześniej zsynchronizowane
- Jeżeli zamówienie powołuje się na daną dostawę produktu, ta dostawa musi istnieć i zawierać pozycję produktową dla produktu z zamówienia, która nie została oznaczona, jako usunięta
- Jeżeli zamówienie powołuje się na dane zamówienie produktu, to zamówienie musi istnieć i zawierać pozycję produktową dla produktu z zamówienia, która nie została oznaczona, jako usunięta
- Wymaganie powiązania z dostawą i terminem nie obowiązuje jeżeli zamówienie ma status „ANULOWANO”

Wszystkie powyższe warunki mają na celu zapewnienie spójności danych przesyłanych do PRESTA

Skrypt bazy danych

```
--  
-- PostgreSQL database dump  
--  
  
-- Dumped from database version 9.6.6  
-- Dumped by pg_dump version 9.6.6  
  
SET statement_timeout = 0;  
SET lock_timeout = 0;  
SET idle_in_transaction_session_timeout = 0;  
SET client_encoding = 'UTF8';  
SET standard_conforming_strings = on;  
SET check_function_bodies = false;  
SET client_min_messages = warning;  
SET row_security = off;  
  
--  
-- Name: talend; Type: SCHEMA; Schema: -; Owner: talend  
--  
  
CREATE SCHEMA talend;  
  
  
ALTER SCHEMA talend OWNER TO talend;  
  
--  
-- Name: plpgsql; Type: EXTENSION; Schema: -; Owner:  
--  
  
CREATE EXTENSION IF NOT EXISTS plpgsql WITH SCHEMA pg_catalog;
```



InSystem Sp. z o.o.

ul. Kijowska 1,
03-738 Warszawa

tel.: +48 22 2050710-18
fax.: +48 22 2050719

<http://www.insystem.com.pl>,
office@insystem.com.pl

```
--
-- Name: EXTENSION plpgsql; Type: COMMENT; Schema: -; Owner:
--

COMMENT ON EXTENSION plpgsql IS 'PL/pgSQL procedural language';

SET search_path = public, pg_catalog;

SET default_tablespace = '';

SET default_with_oids = false;

--
-- Name: archive_products; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE archive_products (
    product_id integer NOT NULL,
    clz_prod_id integer NOT NULL,
    clz_color_id integer NOT NULL,
    ean character varying(20),
    sku character varying(100),
    name_pl character varying(150),
    name_en character varying(150),
    packaging character varying(100),
    price numeric(7,2),
    is_on_stock boolean,
    clz_modified_on timestamp without time zone,
    modified_on timestamp without time zone,
    presta_is_synced boolean,
    presta_last_sync_on timestamp without time zone
);

ALTER TABLE archive_products OWNER TO postgres;

--
-- Name: category_sets; Type: TABLE; Schema: public; Owner: cf_esb
--

CREATE TABLE category_sets (
```

Regon: 142282110, NIP: 701-022-55-66

Sąd Rejonowy dla m.st Warszawy, XII Wydział Gospodarczy Krajowego Rejestru Sądowego nr KRS 0000352995

Kapitał zakładowy spółki: 6 250,00 zł , opłacony w całości.

```
categoryset_id integer NOT NULL,  
clz_catset_id integer NOT NULL,  
clz_prod_id integer,  
clz_cat_l1_id integer,  
clz_cat_l2_id integer,  
clz_cat_l3_id integer,  
clz_is_deleted boolean,  
clz_modified_on timestamp without time zone,  
modified_on timestamp without time zone,  
presta_is_synced boolean,  
presta_last_sync_on timestamp without time zone  
);
```

```
ALTER TABLE category_sets OWNER TO cf_esb;
```

```
--  
-- Name: category_sets_categoryset_id_seq; Type: SEQUENCE; Schema: public; Owner: cf_esb  
--
```

```
CREATE SEQUENCE category_sets_categoryset_id_seq  
  START WITH 1  
  INCREMENT BY 1  
  NO MINVALUE  
  NO MAXVALUE  
  CACHE 1;
```

```
ALTER TABLE category_sets_categoryset_id_seq OWNER TO cf_esb;
```

```
--  
-- Name: category_sets_categoryset_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: cf_esb  
--
```

```
ALTER SEQUENCE category_sets_categoryset_id_seq OWNED BY category_sets.categoryset_id;
```

```
--  
-- Name: customers; Type: TABLE; Schema: public; Owner: cf_esb  
--
```

```
CREATE TABLE customers (  
  customer_id integer NOT NULL,  
  clz_cust_id integer,
```

```
clz_nip character varying(50),
nip character varying(50),
email character varying(100),
del_addr_firstname character varying(50),
del_addr_lastname character varying(50),
del_addr_company character varying(250),
del_addr_street character varying(200),
del_addr_postalcode character varying(20),
del_addr_city character varying(100),
inv_addr_firstname character varying(50),
inv_addr_lastname character varying(50),
inv_addr_company character varying(250),
inv_addr_street character varying(100),
inv_addr_postalcode character varying(20),
inv_addr_city character varying(100),
status character varying(50),
is_presta_customer boolean,
clz_modified_on timestamp without time zone,
modified_on timestamp without time zone,
presta_is_synced boolean,
presta_last_sync_on timestamp without time zone,
email_valid boolean,
customer_from_presta boolean,
tel1 character varying(100),
tel1_desc character varying(100),
tel2 character varying(100),
tel2_desc character varying(100),
tel3 character varying(100),
tel3_desc character varying(100)
);
```

```
ALTER TABLE customers OWNER TO cf_esb;
```

```
--
```

```
-- Name: customers_customer_id_seq; Type: SEQUENCE; Schema: public; Owner: cf_esb
```

```
--
```

```
CREATE SEQUENCE customers_customer_id_seq
START WITH 1
INCREMENT BY 1
NO MINVALUE
NO MAXVALUE
CACHE 1;
```

```
ALTER TABLE customers_customer_id_seq OWNER TO cf_esb;
```

```
--  
-- Name: customers_customer_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: cf_esb  
--
```

```
ALTER SEQUENCE customers_customer_id_seq OWNED BY customers.customer_id;
```

```
--  
-- Name: ins_products; Type: TABLE; Schema: public; Owner: cf_esb  
--
```

```
CREATE TABLE ins_products (  
    clz_prod_id integer NOT NULL,  
    ins_prod_id integer NOT NULL,  
    ins_prod_symbol character varying(20) NOT NULL,  
    ins_prod_nazwa character varying(50),  
    ins_price_kartotekowa numeric(19,4),  
    ins_price_detal numeric(19,4),  
    ins_price_hurt numeric(19,4),  
    ins_price_spec numeric(19,4),  
    ins_is_deleted boolean DEFAULT false,  
    modified_on timestamp without time zone,  
    ins_price_modified_on timestamp without time zone,  
    presta_is_synced boolean DEFAULT false,  
    presta_last_sync_on timestamp without time zone  
);
```

```
ALTER TABLE ins_products OWNER TO cf_esb;
```

```
--  
-- Name: orders; Type: TABLE; Schema: public; Owner: cf_esb  
--
```

```
CREATE TABLE orders (  
    order_id integer NOT NULL,  
    clz_order_id character varying(50) NOT NULL,  
    pre_orig_id character varying(50),  
    customer_id integer,  
    clz_cust_id integer,
```

```
product_id integer,  
clz_prod_id integer,  
clz_col_id integer,  
order_date timestamp without time zone,  
clz_prod_ord_id integer,  
clz_del_id integer,  
amount integer,  
status character varying(50),  
clz_modified_on timestamp without time zone,  
modified_on timestamp without time zone,  
presta_is_synced boolean,  
presta_last_sync_on timestamp without time zone,  
qty_diff integer  
);
```

```
ALTER TABLE orders OWNER TO cf_esb;
```

```
--
```

```
-- Name: orders_order_id_seq; Type: SEQUENCE; Schema: public; Owner: cf_esb
```

```
--
```

```
CREATE SEQUENCE orders_order_id_seq  
START WITH 1  
INCREMENT BY 1  
NO MINVALUE  
NO MAXVALUE  
CACHE 1;
```

```
ALTER TABLE orders_order_id_seq OWNER TO cf_esb;
```

```
--
```

```
-- Name: orders_order_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: cf_esb
```

```
--
```

```
ALTER SEQUENCE orders_order_id_seq OWNED BY orders.order_id;
```

```
--
```

```
-- Name: product_categories; Type: TABLE; Schema: public; Owner: cf_esb
```

```
--
```

```
CREATE TABLE product_categories (
```

```
category_id integer NOT NULL,  
clz_cat_id integer NOT NULL,  
cat_lvl smallint NOT NULL,  
cat_name character varying(150),  
clz_supercat_id integer,  
supercat_id integer,  
clz_modified_on timestamp without time zone,  
modified_on timestamp without time zone,  
presta_is_synced boolean,  
presta_last_sync_on timestamp without time zone  
);
```

```
ALTER TABLE product_categories OWNER TO cf_esb;
```

```
--  
-- Name: product_categories_category_id_seq; Type: SEQUENCE; Schema: public; Owner: cf_esb  
--
```

```
CREATE SEQUENCE product_categories_category_id_seq  
  START WITH 1  
  INCREMENT BY 1  
  NO MINVALUE  
  NO MAXVALUE  
  CACHE 1;
```

```
ALTER TABLE product_categories_category_id_seq OWNER TO cf_esb;
```

```
--  
-- Name: product_categories_category_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: cf_esb  
--
```

```
ALTER SEQUENCE product_categories_category_id_seq OWNED BY product_categories.category_id;
```

```
--  
-- Name: product_deliveries; Type: TABLE; Schema: public; Owner: cf_esb  
--
```

```
CREATE TABLE product_deliveries (  
  product_delivery_id integer NOT NULL,  
  clz_del_entry_id integer NOT NULL,  
  clz_del_id integer NOT NULL,
```

```
clz_prod_ord_id integer NOT NULL,  
clz_prod_id integer NOT NULL,  
clz_color_id integer NOT NULL,  
del_number character varying(150),  
del_date date,  
quantity integer,  
price numeric(8,4),  
clz_modified_on timestamp without time zone,  
modified_on timestamp without time zone,  
presta_is_synced boolean,  
presta_last_sync_on timestamp without time zone,  
arrived boolean,  
sold_out boolean,  
packaging character varying(100),  
addl_packaging character varying(100),  
clz_is_deleted boolean,  
del_date_eta_ozarow date,  
pcs_in_box integer,  
total_quantity integer  
);
```

```
ALTER TABLE product_deliveries OWNER TO cf_esb;
```

```
--  
-- Name: product_deliveries_product_delivery_id_seq; Type: SEQUENCE; Schema: public; Owner: cf_esb  
--
```

```
CREATE SEQUENCE product_deliveries_product_delivery_id_seq  
START WITH 1  
INCREMENT BY 1  
NO MINVALUE  
NO MAXVALUE  
CACHE 1;
```

```
ALTER TABLE product_deliveries_product_delivery_id_seq OWNER TO cf_esb;
```

```
--  
-- Name: product_deliveries_product_delivery_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner:  
cf_esb  
--
```



InSystem Sp. z o.o.

ul. Kijowska 1,
03-738 Warszawa

tel.: +48 22 2050710-18
fax.: +48 22 2050719

<http://www.insystem.com.pl>,
office@insystem.com.pl

```
ALTER          SEQUENCE          product_deliveries_product_delivery_id_seq      OWNED      BY
product_deliveries.product_delivery_id;
```

```
--
-- Name: product_orders; Type: TABLE; Schema: public; Owner: cf_esb
--
```

```
CREATE TABLE product_orders (
  product_order_id integer NOT NULL,
  clz_ord_entry_id integer NOT NULL,
  clz_prod_id integer NOT NULL,
  clz_color_id integer NOT NULL,
  clz_ord_id integer NOT NULL,
  ord_year smallint,
  ord_month smallint,
  ord_week smallint,
  clz_modified_on timestamp without time zone,
  modified_on timestamp without time zone,
  presta_is_synced boolean,
  presta_last_sync_on timestamp without time zone,
  quantity integer,
  price numeric(8,4),
  total_quantity integer,
  packaging character varying(100),
  addl_packaging character varying(100),
  ord_no character varying(100),
  clz_is_deleted boolean,
  sold_out boolean,
  pcs_in_box integer,
  is_active boolean
);
```

```
ALTER TABLE product_orders OWNER TO cf_esb;
```

```
--
-- Name: product_orders_product_order_id_seq; Type: SEQUENCE; Schema: public; Owner: cf_esb
--
```

```
CREATE SEQUENCE product_orders_product_order_id_seq
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
```

Regon: 142282110, NIP: 701-022-55-66

Sąd Rejonowy dla m.st Warszawy, XII Wydział Gospodarczy Krajowego Rejestru Sądowego nr KRS 0000352995

Kapitał zakładowy spółki: 6 250,00 zł , opłacony w całości.

NO MAXVALUE
CACHE 1;

ALTER TABLE product_orders_product_order_id_seq OWNER TO cf_esb;

--

-- Name: product_orders_product_order_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: cf_esb

--

ALTER SEQUENCE product_orders_product_order_id_seq OWNED BY product_orders.product_order_id;

--

-- Name: products; Type: TABLE; Schema: public; Owner: cf_esb

--

CREATE TABLE products (
 product_id integer NOT NULL,
 clz_prod_id integer NOT NULL,
 clz_color_id integer NOT NULL,
 ean character varying(20),
 sku character varying(100),
 name_pl character varying(150),
 name_en character varying(150),
 packaging character varying(100),
 price numeric(8,4),
 is_on_stock boolean,
 clz_modified_on timestamp without time zone,
 modified_on timestamp without time zone,
 presta_is_synced boolean,
 presta_last_sync_on timestamp without time zone,
 clz_is_deleted boolean,
 addl_packaging character varying(100),
 height double precision,
 diameter double precision,
 vendor_id integer,
 vendor_name character varying(100),
 clz_cat_l1_id integer,
 clz_cat_l2_id integer,
 clz_cat_l3_id integer
);

```
ALTER TABLE products OWNER TO cf_esb;
```

```
--
```

```
-- Name: products_product_id_seq; Type: SEQUENCE; Schema: public; Owner: cf_esb
```

```
--
```

```
CREATE SEQUENCE products_product_id_seq
START WITH 1
INCREMENT BY 1
NO MINVALUE
NO MAXVALUE
CACHE 1;
```

```
ALTER TABLE products_product_id_seq OWNER TO cf_esb;
```

```
--
```

```
-- Name: products_product_id_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: cf_esb
```

```
--
```

```
ALTER SEQUENCE products_product_id_seq OWNED BY products.product_id;
```

```
--
```

```
-- Name: v_presta_ins_price; Type: VIEW; Schema: public; Owner: cf_esb
```

```
--
```

```
CREATE VIEW v_presta_ins_price AS
SELECT ins_products.clz_prod_id,
       ins_products.ins_price_hurt AS ins_price
FROM ins_products;
```

```
ALTER TABLE v_presta_ins_price OWNER TO cf_esb;
```

```
--
```

```
-- Name: v_presta_not_synced; Type: VIEW; Schema: public; Owner: cf_esb
```

```
--
```

```
CREATE VIEW v_presta_not_synced AS
SELECT ( SELECT count(*) AS prod_cat_sync_wait
        FROM product_categories
        WHERE (product_categories.presta_is_synced = false)) AS prod_cat_sync_wait,
       ( SELECT count(*) AS cust_sync_wait
```

```
FROM customers
WHERE (customers.presta_is_synced = false)) AS cust_sync_wait,
( SELECT count(*) AS prod_sync_wait
FROM products
WHERE (products.presta_is_synced = false)) AS prod_sync_wait,
( SELECT count(*) AS prod_del_sync_wait
FROM product_deliveries
WHERE (product_deliveries.presta_is_synced = false)) AS prod_del_sync_wait,
( SELECT count(*) AS prod_ord_sync_wait
FROM product_orders
WHERE (product_orders.presta_is_synced = false)) AS prod_ord_sync_wait,
( SELECT count(*) AS ord_sync_wait
FROM orders
WHERE (orders.presta_is_synced = false)) AS ord_sync_wait;

ALTER TABLE v_presta_not_synced OWNER TO cf_esb;

--
-- Name: v_presta_orders; Type: VIEW; Schema: public; Owner: cf_esb
--

CREATE VIEW v_presta_orders AS
SELECT o.clz_order_id,
       o.pre_orig_id,
       o.customer_id,
       c.clz_cust_id,
       c.nip,
       o.product_id,
       p.clz_prod_id,
       p.clz_color_id,
       p.ean,
       o.order_date,
       o.clz_prod_ord_id,
       o.clz_del_id,
       o.amount,
       o.status,
       o.clz_modified_on,
       o.modified_on,
       o.presta_last_sync_on,
       o.qty_diff
FROM (((orders o
JOIN customers c ON (((o.customer_id = c.customer_id) AND (c.presta_is_synced = true))))
JOIN products p ON (((o.product_id = p.product_id) AND (p.presta_is_synced = true)))))
```

```
LEFT JOIN ( SELECT DISTINCT product_orders.clz_ord_id,
    product_orders.clz_prod_id,
    product_orders.clz_color_id
FROM product_orders
    WHERE (NOT (product_orders.clz_is_deleted = true))) po ON (((po.clz_ord_id = o.clz_prod_ord_id) AND
(po.clz_prod_id = p.clz_prod_id) AND (po.clz_color_id = p.clz_color_id))))
LEFT JOIN ( SELECT DISTINCT product_deliveries.clz_del_id,
    product_deliveries.clz_prod_id,
    product_deliveries.clz_color_id
FROM product_deliveries
    WHERE (NOT (product_deliveries.clz_is_deleted = true))) pd ON (((pd.clz_del_id = o.clz_del_id) AND
(pd.clz_prod_id = p.clz_prod_id) AND (pd.clz_color_id = p.clz_color_id))))
    WHERE ((o.presta_is_synced = false) AND (((((o.clz_prod_ord_id IS NULL) AND (po.clz_ord_id IS NULL)) OR
(o.clz_prod_ord_id = po.clz_ord_id)) AND (((o.clz_del_id IS NULL) AND (pd.clz_del_id IS NULL)) OR (o.clz_del_id
= pd.clz_del_id))) OR ((o.status)::text = 'ANULOWANO'::text)));
```

```
ALTER TABLE v_presta_orders OWNER TO cf_esb;
```

```
--
-- Name: v_presta_orders_old; Type: VIEW; Schema: public; Owner: cf_esb
--
```

```
CREATE VIEW v_presta_orders_old AS
SELECT o.clz_order_id,
    o.pre_orig_id,
    o.customer_id,
    c.clz_cust_id,
    c.nip,
    o.product_id,
    p.clz_prod_id,
    p.clz_color_id,
    p.ean,
    o.order_date,
    o.clz_prod_ord_id,
    o.clz_del_id,
    o.amount,
    o.status,
    o.clz_modified_on,
    o.modified_on,
    o.presta_last_sync_on,
    o.qty_diff
FROM ((orders o
    JOIN customers c ON ((o.customer_id = c.customer_id)))
```

```
JOIN products p ON ((o.product_id = p.product_id))
WHERE ((o.presta_is_synced = false) AND (c.presta_is_synced = true) AND (p.presta_is_synced = true) AND
(NOT (p.clz_is_deleted = true)) AND ((o.clz_prod_ord_id IS NULL) OR (EXISTS ( SELECT 1
FROM product_orders
WHERE ((product_orders.clz_ord_id = o.clz_prod_ord_id) AND (product_orders.clz_prod_id =
p.clz_prod_id) AND (product_orders.clz_color_id = p.clz_color_id) AND (NOT (product_orders.clz_is_deleted =
true)))))) AND ((o.clz_del_id IS NULL) OR (EXISTS ( SELECT 1
FROM product_deliveries
WHERE ((product_deliveries.clz_del_id = o.clz_del_id) AND (product_deliveries.clz_prod_id =
p.clz_prod_id) AND (product_deliveries.clz_color_id = p.clz_color_id) AND (NOT
(product_deliveries.clz_is_deleted = true))))))));
```

```
ALTER TABLE v_presta_orders_old OWNER TO cf_esb;
```

```
--
-- Name: v_presta_orders_old2; Type: VIEW; Schema: public; Owner: cf_esb
--
```

```
CREATE VIEW v_presta_orders_old2 AS
SELECT o.clz_order_id,
       o.pre_orig_id,
       o.customer_id,
       c.clz_cust_id,
       c.nip,
       o.product_id,
       p.clz_prod_id,
       p.clz_color_id,
       p.ean,
       o.order_date,
       o.clz_prod_ord_id,
       o.clz_del_id,
       o.amount,
       o.status,
       o.clz_modified_on,
       o.modified_on,
       o.presta_last_sync_on,
       o.qty_diff
FROM (((orders o
JOIN customers c ON (((o.customer_id = c.customer_id) AND (c.presta_is_synced = true))))
JOIN products p ON (((o.product_id = p.product_id) AND (p.presta_is_synced = true) AND (NOT
(p.clz_is_deleted = true))))))
LEFT JOIN ( SELECT DISTINCT product_orders.clz_ord_id,
                           product_orders.clz_prod_id,
```

```
product_orders.clz_color_id
FROM product_orders
WHERE (NOT (product_orders.clz_is_deleted = true))) po ON (((po.clz_ord_id = o.clz_prod_ord_id) AND
(po.clz_prod_id = p.clz_prod_id) AND (po.clz_color_id = p.clz_color_id)))
LEFT JOIN ( SELECT DISTINCT product_deliveries.clz_del_id,
product_deliveries.clz_prod_id,
product_deliveries.clz_color_id
FROM product_deliveries
WHERE (NOT (product_deliveries.clz_is_deleted = true))) pd ON (((pd.clz_del_id = o.clz_del_id) AND
(pd.clz_prod_id = p.clz_prod_id) AND (pd.clz_color_id = p.clz_color_id)))
WHERE ((o.presta_is_synced = false) AND (((o.clz_prod_ord_id IS NULL) AND (po.clz_ord_id IS NULL)) OR
(o.clz_prod_ord_id = po.clz_ord_id)) AND (((o.clz_del_id IS NULL) AND (pd.clz_del_id IS NULL)) OR (o.clz_del_id
= pd.clz_del_id)));
```

```
ALTER TABLE v_presta_orders_old2 OWNER TO cf_esb;
```

```
--
-- Name: v_presta_orders_old20210923; Type: VIEW; Schema: public; Owner: cf_esb
--
```

```
CREATE VIEW v_presta_orders_old20210923 AS
SELECT o.clz_order_id,
o.pre_orig_id,
o.customer_id,
c.clz_cust_id,
c.nip,
o.product_id,
p.clz_prod_id,
p.clz_color_id,
p.ean,
o.order_date,
o.clz_prod_ord_id,
o.clz_del_id,
o.amount,
o.status,
o.clz_modified_on,
o.modified_on,
o.presta_last_sync_on,
o.qty_diff
FROM (((orders o
JOIN customers c ON (((o.customer_id = c.customer_id) AND (c.presta_is_synced = true))))
JOIN products p ON (((o.product_id = p.product_id) AND (p.presta_is_synced = true) AND (NOT
(p.clz_is_deleted = true))))))
```

```
LEFT JOIN ( SELECT DISTINCT product_orders.clz_ord_id,
    product_orders.clz_prod_id,
    product_orders.clz_color_id
FROM product_orders
WHERE (NOT (product_orders.clz_is_deleted = true))) po ON (((po.clz_ord_id = o.clz_prod_ord_id) AND
(po.clz_prod_id = p.clz_prod_id) AND (po.clz_color_id = p.clz_color_id))))
LEFT JOIN ( SELECT DISTINCT product_deliveries.clz_del_id,
    product_deliveries.clz_prod_id,
    product_deliveries.clz_color_id
FROM product_deliveries
WHERE (NOT (product_deliveries.clz_is_deleted = true))) pd ON (((pd.clz_del_id = o.clz_del_id) AND
(pd.clz_prod_id = p.clz_prod_id) AND (pd.clz_color_id = p.clz_color_id))))
WHERE ((o.presta_is_synced = false) AND (((((o.clz_prod_ord_id IS NULL) AND (po.clz_ord_id IS NULL)) OR
(o.clz_prod_ord_id = po.clz_ord_id)) AND (((o.clz_del_id IS NULL) AND (pd.clz_del_id IS NULL)) OR (o.clz_del_id
= pd.clz_del_id))) OR ((o.status)::text = 'ANULOWANO'::text))));
```

```
ALTER TABLE v_presta_orders_old20210923 OWNER TO cf_esb;
```

```
--
```

```
-- Name: v_presta_product_deliveries; Type: VIEW; Schema: public; Owner: cf_esb
```

```
--
```

```
CREATE VIEW v_presta_product_deliveries AS
```

```
SELECT pd.product_delivery_id,
    pd.clz_del_id,
    pd.clz_prod_ord_id,
    pd.clz_prod_id,
    pd.clz_color_id,
    pd.del_number,
    pd.del_date,
    pd.del_date_eta_ozarow,
    pd.quantity,
    pd.total_quantity,
    pd.price,
    pd.arrived,
    pd.sold_out,
    pd.packaging,
    pd.addl_packaging,
    pd.pcs_in_box,
    pd.clz_is_deleted,
    pd.clz_modified_on,
    pd.modified_on,
    pd.presta_is_synced,
```

```
pd.presta_last_sync_on
FROM (product_deliveries pd
JOIN product_orders po ON (((pd.clz_prod_ord_id = po.clz_ord_id) AND (pd.clz_prod_id = po.clz_prod_id)
AND (pd.clz_color_id = po.clz_color_id) AND ((po.clz_is_deleted IS NULL) OR (po.clz_is_deleted = false)))));
```

```
ALTER TABLE v_presta_product_deliveries OWNER TO cf_esb;
```

```
--
-- Name: v_presta_vendors; Type: VIEW; Schema: public; Owner: cf_esb
--
```

```
CREATE VIEW v_presta_vendors AS
SELECT DISTINCT products.vendor_id,
       products.vendor_name
FROM products
ORDER BY products.vendor_id;
```

```
ALTER TABLE v_presta_vendors OWNER TO cf_esb;
```

```
SET search_path = talend, pg_catalog;
```

```
--
-- Name: event_id; Type: SEQUENCE; Schema: talend; Owner: postgres
--
```

```
CREATE SEQUENCE event_id
START WITH 1
INCREMENT BY 1
NO MINVALUE
NO MAXVALUE
CACHE 1;
```

```
ALTER TABLE event_id OWNER TO postgres;
```

```
--
-- Name: events; Type: TABLE; Schema: talend; Owner: talend
--
```

```
CREATE TABLE events (
  id bigint NOT NULL,
  ei_timestamp timestamp without time zone,
```

```
ei_event_type character varying(255),
orig_custom_id character varying(255),
orig_process_id character varying(255),
orig_hostname character varying(128),
orig_ip character varying(64),
orig_principal character varying(255),
mi_port_type character varying(255),
mi_operation_name character varying(255),
mi_message_id character varying(255),
mi_flow_id character varying(64),
mi_transport_type character varying(255),
message_content text,
content_cut boolean
);
```

```
ALTER TABLE events OWNER TO talend;
```

```
--
-- Name: events_copy; Type: TABLE; Schema: talend; Owner: postgres
--
```

```
CREATE TABLE events_copy (
    id bigint,
    ei_timestamp timestamp without time zone,
    ei_event_type character varying(255),
    orig_custom_id character varying(255),
    orig_process_id character varying(255),
    orig_hostname character varying(128),
    orig_ip character varying(64),
    orig_principal character varying(255),
    mi_port_type character varying(255),
    mi_operation_name character varying(255),
    mi_message_id character varying(255),
    mi_flow_id character varying(64),
    mi_transport_type character varying(255),
    message_content text,
    content_cut boolean
);
```

```
ALTER TABLE events_copy OWNER TO postgres;
```

```
--
```

```
-- Name: events_custominfo; Type: TABLE; Schema: talend; Owner: talend
--
```

```
CREATE TABLE events_custominfo (
  id bigint NOT NULL,
  event_id bigint NOT NULL,
  cust_key character varying(255),
  cust_value character varying(255)
);
```

```
ALTER TABLE events_custominfo OWNER TO talend;
```

```
SET search_path = public, pg_catalog;
```

```
--
-- Name: category_sets categoryset_id; Type: DEFAULT; Schema: public; Owner: cf_esb
--
```

```
ALTER TABLE ONLY category_sets ALTER COLUMN categoryset_id SET DEFAULT
nextval('category_sets_categoryset_id_seq'::regclass);
```

```
--
-- Name: customers customer_id; Type: DEFAULT; Schema: public; Owner: cf_esb
--
```

```
ALTER TABLE ONLY customers ALTER COLUMN customer_id SET DEFAULT
nextval('customers_customer_id_seq'::regclass);
```

```
--
-- Name: orders order_id; Type: DEFAULT; Schema: public; Owner: cf_esb
--
```

```
ALTER TABLE ONLY orders ALTER COLUMN order_id SET DEFAULT nextval('orders_order_id_seq'::regclass);
```

```
--
-- Name: product_categories category_id; Type: DEFAULT; Schema: public; Owner: cf_esb
--
```

```
ALTER TABLE ONLY product_categories ALTER COLUMN category_id SET DEFAULT
nextval('product_categories_category_id_seq'::regclass);
```

**InSystem Sp. z o.o.**

ul. Kijowska 1,
03-738 Warszawa

tel.: +48 22 2050710-18
fax.: +48 22 2050719

<http://www.insystem.com.pl>,
office@insystem.com.pl

```
--
-- Name: product_deliveries product_delivery_id; Type: DEFAULT; Schema: public; Owner: cf_esb
--

ALTER TABLE ONLY product_deliveries ALTER COLUMN product_delivery_id SET DEFAULT
nextval('product_deliveries_product_delivery_id_seq'::regclass);

--
-- Name: product_orders product_order_id; Type: DEFAULT; Schema: public; Owner: cf_esb
--

ALTER TABLE ONLY product_orders ALTER COLUMN product_order_id SET DEFAULT
nextval('product_orders_product_order_id_seq'::regclass);

--
-- Name: products product_id; Type: DEFAULT; Schema: public; Owner: cf_esb
--

ALTER TABLE ONLY products ALTER COLUMN product_id SET DEFAULT
nextval('products_product_id_seq'::regclass);

--
-- Name: category_sets category_sets_clz_catset_id_key; Type: CONSTRAINT; Schema: public; Owner: cf_esb
--

ALTER TABLE ONLY category_sets
  ADD CONSTRAINT category_sets_clz_catset_id_key UNIQUE (clz_catset_id);

--
-- Name: category_sets category_sets_pkey; Type: CONSTRAINT; Schema: public; Owner: cf_esb
--

ALTER TABLE ONLY category_sets
  ADD CONSTRAINT category_sets_pkey PRIMARY KEY (categoryset_id);

--
-- Name: customers customers_pkey; Type: CONSTRAINT; Schema: public; Owner: cf_esb
```

Regon: 142282110, NIP: 701-022-55-66

Sąd Rejonowy dla m.st Warszawy, XII Wydział Gospodarczy Krajowego Rejestru Sądowego nr KRS 0000352995

Kapitał zakładowy spółki: 6 250,00 zł , opłacony w całości.



InSystem Sp. z o.o.

ul. Kijowska 1,
03-738 Warszawa

tel.: +48 22 2050710-18
fax.: +48 22 2050719

<http://www.insystem.com.pl>,
office@insystem.com.pl

--

ALTER TABLE ONLY customers

ADD CONSTRAINT customers_pkey PRIMARY KEY (customer_id);

--

-- Name: ins_products ins_products_pkey; Type: CONSTRAINT; Schema: public; Owner: cf_esb

--

ALTER TABLE ONLY ins_products

ADD CONSTRAINT ins_products_pkey PRIMARY KEY (clz_prod_id);

--

-- Name: orders orders_pkey; Type: CONSTRAINT; Schema: public; Owner: cf_esb

--

ALTER TABLE ONLY orders

ADD CONSTRAINT orders_pkey PRIMARY KEY (order_id);

--

-- Name: product_categories product_categories_pkey; Type: CONSTRAINT; Schema: public; Owner: cf_esb

--

ALTER TABLE ONLY product_categories

ADD CONSTRAINT product_categories_pkey PRIMARY KEY (category_id);

--

-- Name: product_deliveries product_deliveries_pkey; Type: CONSTRAINT; Schema: public; Owner: cf_esb

--

ALTER TABLE ONLY product_deliveries

ADD CONSTRAINT product_deliveries_pkey PRIMARY KEY (product_delivery_id);

--

-- Name: product_orders product_orders_pkey; Type: CONSTRAINT; Schema: public; Owner: cf_esb

--

ALTER TABLE ONLY product_orders

ADD CONSTRAINT product_orders_pkey PRIMARY KEY (product_order_id);

Regon: 142282110, NIP: 701-022-55-66

Sąd Rejonowy dla m.st Warszawy, XII Wydział Gospodarczy Krajowego Rejestru Sądowego nr KRS 0000352995

Kapitał zakładowy spółki: 6 250,00 zł , opłacony w całości.



InSystem Sp. z o.o.

ul. Kijowska 1,
03-738 Warszawa

tel.: +48 22 2050710-18
fax.: +48 22 2050719

<http://www.insystem.com.pl>,
office@insystem.com.pl

```
--
-- Name: products products_pkey; Type: CONSTRAINT; Schema: public; Owner: cf_esb
--

ALTER TABLE ONLY products
  ADD CONSTRAINT products_pkey PRIMARY KEY (product_id);

--

-- Name: product_categories unique_clz_categories; Type: CONSTRAINT; Schema: public; Owner: cf_esb
--

ALTER TABLE ONLY product_categories
  ADD CONSTRAINT unique_clz_categories UNIQUE (clz_cat_id, cat_lvl);

--

-- Name: customers unique_clz_cust_id; Type: CONSTRAINT; Schema: public; Owner: cf_esb
--

ALTER TABLE ONLY customers
  ADD CONSTRAINT unique_clz_cust_id UNIQUE (clz_cust_id);

--

-- Name: orders unique_clz_order_id; Type: CONSTRAINT; Schema: public; Owner: cf_esb
--

ALTER TABLE ONLY orders
  ADD CONSTRAINT unique_clz_order_id UNIQUE (clz_order_id);

--

-- Name: product_deliveries unique_deliveries; Type: CONSTRAINT; Schema: public; Owner: cf_esb
--

ALTER TABLE ONLY product_deliveries
  ADD CONSTRAINT unique_deliveries UNIQUE (clz_del_entry_id);

--

-- Name: product_orders unique_orders; Type: CONSTRAINT; Schema: public; Owner: cf_esb
```

Regon: 142282110, NIP: 701-022-55-66

Sąd Rejonowy dla m.st. Warszawy, XII Wydział Gospodarczy Krajowego Rejestru Sądowego nr KRS 0000352995

Kapitał zakładowy spółki: 6 250,00 zł , opłacony w całości.



InSystem Sp. z o.o.

ul. Kijowska 1,
03-738 Warszawa

tel.: +48 22 2050710-18
fax.: +48 22 2050719

<http://www.insystem.com.pl>,
office@insystem.com.pl

--

ALTER TABLE ONLY product_orders
ADD CONSTRAINT unique_orders UNIQUE (clz_ord_entry_id);

--

-- Name: products unique_products; Type: CONSTRAINT; Schema: public; Owner: cf_esb

--

ALTER TABLE ONLY products
ADD CONSTRAINT unique_products UNIQUE (clz_prod_id, clz_color_id);

SET search_path = talend, pg_catalog;

--

-- Name: events_custominfo events_custominfo_pkey; Type: CONSTRAINT; Schema: talend; Owner: talend

--

ALTER TABLE ONLY events_custominfo
ADD CONSTRAINT events_custominfo_pkey PRIMARY KEY (id);

--

-- Name: events events_pkey; Type: CONSTRAINT; Schema: talend; Owner: talend

--

ALTER TABLE ONLY events
ADD CONSTRAINT events_pkey PRIMARY KEY (id);

--

-- Name: events_custominfo_event_id_idx; Type: INDEX; Schema: talend; Owner: talend

--

CREATE INDEX events_custominfo_event_id_idx ON events_custominfo USING btree (event_id);

--

-- Name: events_flow_timestamp_idx; Type: INDEX; Schema: talend; Owner: talend

--

CREATE INDEX events_flow_timestamp_idx ON events USING btree (mi_flow_id, ei_timestamp DESC);

Regon: 142282110, NIP: 701-022-55-66

Sąd Rejonowy dla m.st Warszawy, XII Wydział Gospodarczy Krajowego Rejestru Sądowego nr KRS 0000352995

Kapitał zakładowy spółki: 6 250,00 zł , opłacony w całości.

```
--
-- Name: events_mi_port_type_idx; Type: INDEX; Schema: talend; Owner: talend
--

CREATE INDEX events_mi_port_type_idx ON events USING btree (mi_port_type);

--
-- Name: events_timestamp_idx; Type: INDEX; Schema: talend; Owner: talend
--

CREATE INDEX events_timestamp_idx ON events USING btree (ei_timestamp);

SET search_path = public, pg_catalog;

--
-- Name: orders orders_customer_id_fkey; Type: FK CONSTRAINT; Schema: public; Owner: cf_esb
--

ALTER TABLE ONLY orders
  ADD CONSTRAINT orders_customer_id_fkey FOREIGN KEY (customer_id) REFERENCES
customers(customer_id) ON DELETE SET NULL;

--
-- Name: orders orders_product_id_fkey; Type: FK CONSTRAINT; Schema: public; Owner: cf_esb
--

ALTER TABLE ONLY orders
  ADD CONSTRAINT orders_product_id_fkey FOREIGN KEY (product_id) REFERENCES products(product_id) ON
DELETE SET NULL;

--
-- Name: product_categories supercategory_fkey; Type: FK CONSTRAINT; Schema: public; Owner: cf_esb
--

ALTER TABLE ONLY product_categories
  ADD CONSTRAINT supercategory_fkey FOREIGN KEY (supercat_id) REFERENCES
product_categories(category_id);
```



InSystem Sp. z o.o.

ul. Kijowska 1,
03-738 Warszawa

tel.: +48 22 2050710-18
fax.: +48 22 2050719

<http://www.insystem.com.pl>,
office@insystem.com.pl

```
SET search_path = talend, pg_catalog;
```

```
--  
-- Name: event_id; Type: ACL; Schema: talend; Owner: postgres  
--
```

```
GRANT ALL ON SEQUENCE event_id TO talend;
```

```
--  
-- Name: events_copy; Type: ACL; Schema: talend; Owner: postgres  
--
```

```
GRANT ALL ON TABLE events_copy TO talend;
```

```
--  
-- Name: DEFAULT PRIVILEGES FOR SEQUENCES; Type: DEFAULT ACL; Schema: talend; Owner: postgres  
--
```

```
ALTER DEFAULT PRIVILEGES FOR ROLE postgres IN SCHEMA talend REVOKE ALL ON SEQUENCES FROM postgres;  
ALTER DEFAULT PRIVILEGES FOR ROLE postgres IN SCHEMA talend GRANT SELECT,USAGE ON SEQUENCES TO talend;
```

```
--  
-- Name: DEFAULT PRIVILEGES FOR TABLES; Type: DEFAULT ACL; Schema: talend; Owner: postgres  
--
```

```
ALTER DEFAULT PRIVILEGES FOR ROLE postgres IN SCHEMA talend REVOKE ALL ON TABLES FROM postgres;  
ALTER DEFAULT PRIVILEGES FOR ROLE postgres IN SCHEMA talend GRANT ALL ON TABLES TO talend;
```

```
--  
-- PostgreSQL database dump complete  
--  
--  
-- PostgreSQL database dump  
--
```

```
-- Dumped from database version 9.6.6  
-- Dumped by pg_dump version 9.6.6
```

Regon: 142282110, NIP: 701-022-55-66

Sąd Rejonowy dla m.st. Warszawy, XII Wydział Gospodarczy Krajowego Rejestru Sądowego nr KRS 0000352995

Kapitał zakładowy spółki: 6 250,00 zł , opłacony w całości.



InSystem Sp. z o.o.

ul. Kijowska 1,
03-738 Warszawa

tel.: +48 22 2050710-18
fax.: +48 22 2050719

<http://www.insystem.com.pl>,
office@insystem.com.pl

```
SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET row_security = off;

--
-- Name: talend; Type: SCHEMA; Schema: -; Owner: talend
--

CREATE SCHEMA talend;

ALTER SCHEMA talend OWNER TO talend;

--
-- Name: plpgsql; Type: EXTENSION; Schema: -; Owner:
--

CREATE EXTENSION IF NOT EXISTS plpgsql WITH SCHEMA pg_catalog;

--
-- Name: EXTENSION plpgsql; Type: COMMENT; Schema: -; Owner:
--

COMMENT ON EXTENSION plpgsql IS 'PL/pgSQL procedural language';

SET search_path = talend, pg_catalog;

--
-- Name: event_id; Type: SEQUENCE; Schema: talend; Owner: talend
--

CREATE SEQUENCE event_id
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;
```

Regon: 142282110, NIP: 701-022-55-66

Sąd Rejonowy dla m.st. Warszawy, XII Wydział Gospodarczy Krajowego Rejestru Sądowego nr KRS 0000352995

Kapitał zakładowy spółki: 6 250,00 zł, opłacony w całości.

```
ALTER TABLE event_id OWNER TO talend;

SET default_tablespace = '';

SET default_with_oids = false;

--
-- Name: events; Type: TABLE; Schema: talend; Owner: talend
--

CREATE TABLE events (
    id bigint NOT NULL,
    ei_timestamp timestamp without time zone,
    ei_event_type character varying(255),
    orig_custom_id character varying(255),
    orig_process_id character varying(255),
    orig_hostname character varying(128),
    orig_ip character varying(64),
    orig_principal character varying(255),
    mi_port_type character varying(255),
    mi_operation_name character varying(255),
    mi_message_id character varying(255),
    mi_flow_id character varying(64),
    mi_transport_type character varying(255),
    message_content text,
    content_cut boolean
);

ALTER TABLE events OWNER TO talend;

--
-- Name: events_custominfo; Type: TABLE; Schema: talend; Owner: talend
--

CREATE TABLE events_custominfo (
    id bigint NOT NULL,
    event_id bigint NOT NULL,
    cust_key character varying(255),
    cust_value character varying(255)
);
```



InSystem Sp. z o.o.

ul. Kijowska 1,
03-738 Warszawa

tel.: +48 22 2050710-18
fax.: +48 22 2050719

<http://www.insystem.com.pl>,
office@insystem.com.pl

```
ALTER TABLE events_custominfo OWNER TO talend;
```

```
--  
-- Name: events_custominfo events_custominfo_pkey; Type: CONSTRAINT; Schema: talend; Owner: talend  
--
```

```
ALTER TABLE ONLY events_custominfo  
  ADD CONSTRAINT events_custominfo_pkey PRIMARY KEY (id);
```

```
--  
-- Name: events events_pkey; Type: CONSTRAINT; Schema: talend; Owner: talend  
--
```

```
ALTER TABLE ONLY events  
  ADD CONSTRAINT events_pkey PRIMARY KEY (id);
```

```
--  
-- Name: events_custominfo_event_id_idx; Type: INDEX; Schema: talend; Owner: talend  
--
```

```
CREATE INDEX events_custominfo_event_id_idx ON events_custominfo USING btree (event_id);
```

```
--  
-- Name: events_flow_timestamp_idx; Type: INDEX; Schema: talend; Owner: talend  
--
```

```
CREATE INDEX events_flow_timestamp_idx ON events USING btree (mi_flow_id, ei_timestamp DESC);
```

```
--  
-- Name: events_mi_port_type_idx; Type: INDEX; Schema: talend; Owner: talend  
--
```

```
CREATE INDEX events_mi_port_type_idx ON events USING btree (mi_port_type);
```

```
--  
-- Name: events_timestamp_idx; Type: INDEX; Schema: talend; Owner: talend  
--
```

Regon: 142282110, NIP: 701-022-55-66

Sąd Rejonowy dla m.st. Warszawy, XII Wydział Gospodarczy Krajowego Rejestru Sądowego nr KRS 0000352995

Kapitał zakładowy spółki: 6 250,00 zł , opłacony w całości.

```
CREATE INDEX events_timestamp_idx ON events USING btree (ei_timestamp);
```

```
--  
-- PostgreSQL database dump complete  
--
```

Specyfikacje interfejsów

```
<wsdl:definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="urn:insystem.com.pl:services:CF" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" name="CreateCustomer" targetNamespace="urn:insystem.com.pl:services:CF">  
  <wsdl:types>  
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="urn:insystem.com.pl:services:CF" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" targetNamespace="urn:insystem.com.pl:services:CF">  
      <xsd:element name="CreateCustomerOperationRequest">  
        <xsd:complexType>  
          <xsd:sequence>  
            <xsd:element name="nip" type="tns:NipType"/>  
            <xsd:element name="companyName" type="xsd:string"/>  
            <xsd:element name="emailAddress" type="xsd:string"/>  
            <xsd:element name="tel1" type="xsd:string"/>  
            <xsd:element name="tel1Desc" type="xsd:string"/>  
            <xsd:element name="tel2" type="xsd:string"/>  
            <xsd:element name="tel2Desc" type="xsd:string"/>  
            <xsd:element name="tel3" type="xsd:string"/>  
            <xsd:element name="tel3Desc" type="xsd:string"/>  
            <xsd:element maxOccurs="1" minOccurs="0" name="deliveryAddress" type="tns:AddressType"/>  
            <xsd:element maxOccurs="1" minOccurs="0" name="invoiceAddress" type="tns:AddressType"/>  
            <xsd:element name="status" type="tns:CustomerStatusType"/>  
          </xsd:sequence>  
        </xsd:complexType>  
      </xsd:element>  
      <xsd:element name="CreateCustomerOperationResponse">  
        <xsd:complexType>  
          <xsd:sequence>  
            <xsd:element name="result" type="xsd:boolean"/>  
          </xsd:sequence>  
        </xsd:complexType>  
      </xsd:element>  
    </xsd:schema>  
  </wsdl:types>  
</wsdl:definitions>
```

```
<xsd:element name="resultCode" type="xsd:string"/>
<xsd:element name="resultMsg" type="xsd:string"/>
<xsd:element maxOccurs="1" minOccurs="0" name="customerId" type="xsd:int">
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:simpleType name="NipType">
<xsd:restriction base="xsd:string">
<xsd:pattern value="[0-9]{10}"/>
<xsd:pattern value="[A-Z][A-Z][A-Z0-9]{8,12}"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="AddressType">
<xsd:sequence>
<xsd:element name="firstname" type="xsd:string"/>
<xsd:element name="lastname" type="xsd:string"/>
<xsd:element name="companyName" type="xsd:string"/>
<xsd:element name="addressLine" type="xsd:string"/>
<xsd:element name="postalCode" type="xsd:string"/>
<xsd:element name="city" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="CustomerStatusType">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="ZWERYFIKOWANY"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:schema>
</wsdl:types>
<wsdl:message name="CreateCustomerOperationResponse">
<wsdl:part element="tns:CreateCustomerOperationResponse" name="parameters">
</wsdl:part>
</wsdl:message>
<wsdl:message name="CreateCustomerOperationRequest">
<wsdl:part element="tns:CreateCustomerOperationRequest" name="parameters">
</wsdl:part>
</wsdl:message>
<wsdl:portType name="CreateCustomerPortType">
<wsdl:operation name="CreateCustomerOperation">
<wsdl:input message="tns:CreateCustomerOperationRequest"> </wsdl:input>
<wsdl:output message="tns:CreateCustomerOperationResponse"> </wsdl:output>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="CreateCustomerBinding" type="tns:CreateCustomerPortType">
</wsdl:binding>
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/h
ttp"/>
```

```
<wsdl:operation name="CreateCustomerOperation">
<soap:operation soapAction="urn:insystem.com.pl:services:CF/CreateCustomerO
peration"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="CreateCustomer">
<wsdl:port binding="tns:CreateCustomerBinding" name="CreateCustomerPort">
<soap:address location="https://192.168.11.30:9002/services/CreateCustomer"
/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

```
<wsdl:definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="
http://schemas.xmlsoap.org/wsdl/" xmlns:tns="urn:insystem.com.pl:services:C
F" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" name="CreateOrder" ta
rgetNamespace="urn:insystem.com.pl:services:CF">
<wsdl:types>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http:/
/schemas.xmlsoap.org/wsdl/" xmlns:tns="urn:insystem.com.pl:services:CF" xml
ns:soap="http://schemas.xmlsoap.org/wsdl/soap/" targetNamespace="urn:insyst
em.com.pl:services:CF">
<xsd:element name="CreateOrderOperationRequest">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="orderId" type="xsd:string"/>
<xsd:element name="createDate" type="xsd:dateTime"/>
<xsd:element name="customerId" type="xsd:int"/>
<xsd:element maxOccurs="1" minOccurs="0" name="ean" nillable="true" type="t
ns:Ean13Type"/>
<xsd:element name="prodId" type="xsd:int"/>
<xsd:element name="colorId" type="xsd:int"/>
<xsd:element maxOccurs="1" minOccurs="0" name="productOrderId" type="xsd:in
t"> </xsd:element>
<xsd:element maxOccurs="1" minOccurs="0" name="productDeliveryId" type="xsd
:int"> </xsd:element>
<xsd:element name="quantity" type="xsd:nonNegativeInteger"/>
<xsd:element name="status" type="tns:OrderStatusType"/>
```



```
<wsdl:operation name="CreateOrderOperation">
<soap:operation soapAction="urn:insystem.com.pl:services:CF/CreateOrderOperation"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="CreateOrder">
<wsdl:port binding="tns:CreateOrderBinding" name="CreateOrderPort">
<soap:address location="https://192.168.11.30:9002/services/CreateOrder"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

```
<wsdl:definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="
http://schemas.xmlsoap.org/wsdl/" xmlns:tns="urn:insystem.com.pl:services:CF"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" name="GetColorName" targetNamespace="urn:insystem.com.pl:services:CF">
<wsdl:types>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://
schemas.xmlsoap.org/wsdl/" xmlns:tns="urn:insystem.com.pl:services:CF" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" targetNamespace="urn:insystem.com.pl:services:CF">
<xsd:element name="GetColorNameOperationRequest">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="colorId" type="xsd:nonNegativeInteger"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="GetColorNameOperationResponse">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="colorFound" type="xsd:boolean"/>
<xsd:element maxOccurs="1" minOccurs="0" name="colorName" type="xsd:string"/>
<xsd:element maxOccurs="1" minOccurs="0" name="isMix" type="xsd:boolean"/>
<xsd:element maxOccurs="1" minOccurs="0" name="componentColors">
<xsd:complexType>
<xsd:sequence>
```

```
<xsd:element maxOccurs="unbounded" minOccurs="1" name="colorId" type="xsd:n
onNegativeInteger"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</wsdl:types>
<wsdl:message name="GetColorNameOperationRequest">
<wsdl:part element="tns:GetColorNameOperationRequest" name="parameters"> </
wsdl:part>
</wsdl:message>
<wsdl:message name="GetColorNameOperationResponse">
<wsdl:part element="tns:GetColorNameOperationResponse" name="parameters"> <
/wsdl:part>
</wsdl:message>
<wsdl:portType name="GetColorNamePortType">
<wsdl:operation name="GetColorNameOperation">
<wsdl:input message="tns:GetColorNameOperationRequest"> </wsdl:input>
<wsdl:output message="tns:GetColorNameOperationResponse"> </wsdl:output>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="GetColorNameBinding" type="tns:GetColorNamePortType">
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/h
ttp"/>
<wsdl:operation name="GetColorNameOperation">
<soap:operation soapAction="urn:insystem.com.pl:services:CF/GetColorNameOpe
ration"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="GetColorName">
<wsdl:port binding="tns:GetColorNameBinding" name="GetColorNamePort">
<soap:address location="https://192.168.11.30:9002/services/GetColorName"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

```
<wsdl:definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="urn:insystem.com.pl:services:CF" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" name="GetVendorName" targetNamespace="urn:insystem.com.pl:services:CF">
  <wsdl:types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="urn:insystem.com.pl:services:CF" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" targetNamespace="urn:insystem.com.pl:services:CF">
      <xsd:element name="GetVendorNameOperationRequest">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="1" name="vendorId" type="xsd:nonNegativeInteger"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="GetVendorNameOperationResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="1" name="vendorFound" type="xsd:boolean"/>
            <xsd:element maxOccurs="1" minOccurs="1" name="vendorId" type="xsd:nonNegativeInteger"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="vendorName" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="GetVendorNameOperationRequest">
    <wsdl:part element="tns:GetVendorNameOperationRequest" name="parameters"> </wsdl:part>
  </wsdl:message>
  <wsdl:message name="GetVendorNameOperationResponse">
    <wsdl:part element="tns:GetVendorNameOperationResponse" name="parameters"> </wsdl:part>
  </wsdl:message>
  <wsdl:portType name="GetVendorNamePortType">
    <wsdl:operation name="GetVendorNameOperation">
      <wsdl:input message="tns:GetVendorNameOperationRequest"> </wsdl:input>
      <wsdl:output message="tns:GetVendorNameOperationResponse"> </wsdl:output>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="GetVendorNameBinding" type="tns:GetVendorNamePortType">
```



InSystem Sp. z o.o.

ul. Kijowska 1,
03-738 Warszawa

tel.: +48 22 2050710-18
fax.: +48 22 2050719

<http://www.insystem.com.pl>,
office@insystem.com.pl

```
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/h
ttp"/>
<wsdl:operation name="GetVendorNameOperation">
<soap:operation soapAction="urn:insystem.com.pl:services:CF/GetVendorNameOp
eration"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="GetVendorName">
<wsdl:port binding="tns:GetVendorNameBinding" name="GetVendorNamePort">
<soap:address location="https://192.168.11.30:9002/services/GetVendorName"/
>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

Regon: 142282110, NIP: 701-022-55-66

Sąd Rejonowy dla m.st Warszawy, XII Wydział Gospodarczy Krajowego Rejestru Sądowego nr KRS 0000352995

Kapitał zakładowy spółki: 6 250,00 zł , opłacony w całości.