Interfejsy WebService dla systemu Comarch ERP Optima

Spis treści

1. **Charakterystyka dokumentu** 3
1.1. Definicje 3
1.2. Zakres dokumentu 3
2. **Opis procesów i struktury interfejsów** 3
2.1. Wymagania techniczne i licencyjne 3
2.2. Zagadnienia bezpieczeństwa 3
2.3. Ogólne informacje o wymianie danych 4
Metoda Confirm 4
2.4. Słowniki 5
2.4.1. Kontrahenci 5
2.4.2. Towary 5
2.4.2.1. Grupy towarowe 5
GetProductGroups 5
2.4.2.2. Kartoteka towarowa 6
GetProductCodes 6
GetProduct 7
2.4.2.3. Polityka cenowa 9
GetPrices 9
2.4.2.4. Stany magazynowe 10
GetStockStates 10
2.5. Zamówienia 11
2.5.1. Dodanie zamówienia 11
InsertOrders 11
2.5.2. Potwierdzenie zamówienia 13
UpdateOrderStatus 13
2.5.3. Statusy zamówień 14
GetOrderStatuses 14
3. **Organizacja projektu** 15

3.1. Struktura projektu 15

1. Charakterystyka dokumentu

1.1. Definicje

ERP – system Comarch ERP Optima
SI – platforma sklepu internetowego
Webservice – interfejs komunikacyjny między systemem ERP i SC
Synchronizacja różnicowa – wymiana tylko danych zmienionych od ostatniego potwierdzenia poprawności synchronizacji

1.2. Zakres dokumentu

Niniejszy dokument zawiera propozycję rozwiązań w zakresie organizacji informacji systemie ERP i przygotowania dedykowanego webservice na potrzeby komunikacji między systemami ERP i SI. Ujęto w nim elementy funkcjonalne, jakie zostały ustalone w toku prac analitycznych.
Analiza dotyczy następujących zagadnień:

- Konfiguracja systemu ERP na potrzeby obsługi procesów
- Rejestracja danych w systemie ERP
- Interfejsów i sposobów wymiany danych przez webservice

Dokument został sporządzony przez konsultantów na podstawie otrzymanych materiałów w formie papierowej jak i elektronicznej, oraz na podstawie rozmów, które odbywały się telefonicznie.

2. Opis procesów i struktury interfejsów

2.1. Wymagania techniczne i licencyjne

Interfejs działa jako witryna na serwerze IIS (usługi internetowe) w technologii SOAP i udostępniana jest przez protokół HTTP. Aby usługa była dostępna dla systemu SI musi zostać udostępniona w sieci Internet na określonym porcie. Aby komunikacja była w pełni bezpieczna zaleca się zainstalowanie na serwerze IIS certyfikatu SSL.
Na potrzeby synchronizacji danych SI z systemem ERP niezbędne jest zapewnienie ciągłego dostępu do systemu ERP i licencji Kasa/Bank i Handel. W związku z tym proponuje się wyodrębnienie po jednej licencji powyższych modułów z klucza głównego na dedykowany klucz wirtualny, z którego będzie korzystać wyłącznie webservice.
Webservice jest przygotowany i testowany dla wersji systemu ERP: Comarch ERP Optima 2017.6.1

2.2. Zagadnienia bezpieczeństwa

Wszystkie komunikaty powinny być kodowane wg UTF8_polish_ci (dotyczy to całości tego dokumentu).
Każdy z przesyłanych komunikatów opatrzony będzie nagłówkiem nazwy użytkownika oraz hasła jako podstawowa metoda autoryzacji.
Nagłówek autoryzacji ma postać:

```
<Authentication >
    < Password >password</Password >
    < Username >user</Username >
```

<ShopId>2</ ShopId >
</Authentication >

Element	Typ	Opis
Password	varchar(64)	Hasło użytkownika
Username	varchar(64)	Nazwa użytkownika
ShopId	int	Identyfikator sklepu internetowego

2.3. Ogólne informacje o wymianie danych

Z założenia rozwiązanie będzie obsługiwać wiele SI w jednej bazie systemu ERP, i bazować będzie na konfiguracji dostępnej w systemie Optima.

Każdy metoda zwraca obiekt zawierający co najmniej 3 właściwości:

- LoginResult – w przypadku powodzenia logowania przyjmuje wartość „OK”, w przypadku błędu zawiera jego opis
- Success – w przypadku powodzenia operacji przyjmuje wartość prawda, w przypadku błędu fałsz
- ErrorDescription – opis ewentualnego błędu, w przypadku powodzenia opis jest pusty

Dodatkowo zwracany obiekt może zawierać dane specyficzne dla operacji.

Metoda Confirm

W przypadku wywołania metod, które umożliwiają przesyłanie danych różnicowych każdorazowo będzie wymagane potwierdzenie poprawnego odbioru przesłanych danych, czyli:

- GetProductsCodes,
- GetStockStates,
- GetPrices
- GetOrderStatus

Wysłanie komunikatu oznacza, że dane różnicowe zostały odebrane poprawnie i nie będą wysyłane przy kolejnych synchronizacjach różnicowych.

Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <Action s:mustUnderstand="1"
xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://tempuri.org/IService/Confirm</Action>
  </s:Header>
  <s:Body>
    <ConfirmRequest xmlns="OrderSoft">
      <Authentication xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Password />
        <Username>ADMIN</Username>
        <ShopId>2</ShopId>
      </Authentication>
      <ConfirmationType>StockStates</ConfirmationType>
    </ConfirmRequest>
  </s:Body>
</s:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
```

```
<s:Body>
  <ConfirmResponse xmlns="OrderSoft">
    <ErrorMessage />
    <LoginResult>OK</LoginResult>
    <Success>true</Success>
  </ConfirmResponse>
 </s:Body>
</s:Envelope>
```

Element	Typ	Opis
ConfirmationType	enum	Wartości: ProductsCodes (0), StockStates (1), Prices (2), OrderStatus (3)

2.4. Słowniki

2.4.1. Kontrahenci

Kartoteki klientów będą tworzone wyłącznie przy tworzeniu nowego zamówienia (dokumentu Rezerwacji Odbiorcy). Kartoteki klientów nie będą synchronizowane miedzy systemem ERP a SI.
Jeśli Klient przy składaniu zamówieni nie określi, że chce fakturę to w systemie ERP nie będzie tworzona kartoteka Klienta. Zamówienie zostanie zarejestrowane na kontrahenta NIEOKREŚLONEGO, dane do fakturowania i wysyłki będą rejestrowane każdorazowo na dokumencie Rezerwacji Odbiorcy.
Jeśli Klient określi na zamówieniu, że chce fakturę, to podczas wystawiania dokumentu w systemie ERP algorytm najpierw spróbuje odnaleźć Klienta w systemie ERP po NIPie, jeśli Klienta o takim NIPie nie będzie zostanie stworzona nowa kartoteka. Jako akronim Klienta będzie wpisany NIP.

2.4.2. Towary

2.4.2.1. Grupy towarowe

Grupy towarowe będą konfigurowane w systemie ERP. W konfiguracji będzie określane, która grupa towarowa oznacza początek drzewa grup dedykowaną dla danego SI. Do SI będzie zwracany Id grupy oraz Id grupy nadrzędnej. Dla korzenia drzewa jako id rodzic zwracane będzie 0.

GetProductGroups

Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <Action s:mustUnderstand="1"
xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://tempuri.org/IService/GetProductGroups</Action
>
  </s:Header>
  <s:Body>
    <GetProductGroupsRequest xmlns="OrderSoft">
      <Authentication xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Password />
        <Username>ADMIN</Username>
        <ShopId>2</ShopId>
      </Authentication>
    </GetProductGroupsRequest>
  </s:Body>
</s:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
    <GetProductGroupsResponse xmlns="OrderSoft">
      <ErrorMessage />
      <LoginResult>OK</LoginResult>
      <ProductGroups xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
       <ProductGroup>
            <Code>AMOR</Code>
            <Name>Amortyzatory</Name>
            <GroupId>12</Code>
            <ParentId>0</Code>
       </ProductGroup>
       <ProductGroup>
            <Code>AMOR1</Code>
            <Name>Amortyzatory małe</Name>
            <GroupId>23</Code>
            <ParentId>12</Code>
        </ProductGroup>
      </ProductGroups>
      <Success>true</Success>
    </GetProductGroupsResponse>
  </s:Body>
</s:Envelope>
```

2.4.2.2. Kartoteka towarowa

Identyfikacja towaru w obu systemach odbywać się będzie poprzez kod towaru. Towary, które mają być udostępniane w danym SI będą musiały być oznaczone flagą e-Sklep.
Do systemu SI będą wysyłane dane o kartotece towarowej:

- Kod towaru
- EAN
- Nazwa
- Jednostka miary
- Producent (Marka)
- Id grupy towarowej
- Atrybuty (np. kolekcja, kategoria, płeć, kolor) – oznaczone parametrem: udostępniaj w e-Sklep. Lista atrybutów dla danego towaru będzie pobierana dynamicznie, co pozwoli na rozszerzenie listy przesyłanych danych bez potrzeby modyfikacji integracji.

Synchronizacje towarów będzie można wywołać w 2 trybach:

- Pełna – synchronizowane są wszystkie udostępnione dla systemu SI kartoteki towarowe
- Różnicowa – synchronizowane są tylko kartoteki towarowe zmienione od czasu ostatniej synchronizacji

Niezależnie od trybu synchronizacja towarów odbywać się będzie dwueatpowo:
1. Pobranie listy kodów towaru towarów, które się zmieniły (getProductCodes)
2. Pobranie szczegółów każdego towaru z przesłanej listy (getProduct)

GetProductCodes

Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <Action s:mustUnderstand="1"
xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://tempuri.org/IService/GetProductCodes</Action>
```

```
  </s:Header>
  <s:Body>
   <GetProductCodesRequest xmlns="OrderSoft">
    <Authentication xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
     <Password />
     <Username>ADMIN</Username>
     <ShopId>2</ShopId>
    </Authentication>
    <IsDifferential>true</IsDifferential>
   </GetProductCodesRequest>
  </s:Body>
</s:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
   <GetProductCodesResponse xmlns="OrderSoft">
    <ErrorMessage />
    <LoginResult>OK</LoginResult>
    <ProductCodes xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
     <Code>AKUMULATOR</Code>
     <Code>AMORTYZATOR_PRZÓD</Code>
     <Code>AMORTYZATOR_TYŁ</Code>
     <Code>APTECZKA</Code>
     <Code>BAGAŻNIK_DACH</Code>
    </ProductCodes>
    <Success>true</Success>
   </GetProductCodesResponse>
  </s:Body>
</s:Envelope>
```

Element	Typ	Opis
IsDifferential	boolean	Tryb eksportu: false – pełny, true - różnicowy
Code	varchar(30)	Kod towaru

GetProduct

W komunikacie będą udostępniane podstawowe dane towaru oraz:

- Atrybuty – dla których ustawiono flagę udostępniania w SI
- Zestawy – elementy zestawu
- Zdjęcia – dla których ustawiono flagę udostępniania w SI
- Kod towaru grupującego warianty

Warianty:
Każdy wariant będzie osobną kartoteką produktową, na której zostanie wskazany wariant grupujący.

Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
   <Action s:mustUnderstand="1"
xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://tempuri.org/IService/GetProduct</Action>
  </s:Header>
  <s:Body>
   <GetProductRequest xmlns="OrderSoft">
    <Authentication xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
     <Password />
     <Username>ADMIN</Username>
     <ShopId>2</ShopId>
    </Authentication>
```

```
        <ProductCode>KOŁPAKI 16'</ProductCode>
      </GetProductRequest>
    </s:Body>
</s:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
    <GetProductResponse xmlns="OrderSoft">
      <ErrorMessage />
      <LoginResult>OK</LoginResult>
      <Product xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Attributes>
          <ProductAttribute>
            <Name>nazwa atrybutu KOLOR</Name>
            <Value>zielony</Value>
          </ProductAttribute>
          <ProductAttribute>
            <Name>nazwa atrybutu OPIS</Name>
            <Value>super ekstra tip top</Value>
          </ProductAttribute>
        </Set>
          <Element>
            <Code>AD-232</Code>
            <Quantity>2</Quantity>
          </Element>
        <Set>
        <Attributes>
        <Brand />
        <Code>KOŁPAKI 16’</Code>
        <ParentProductCode>KOŁPAKI</ ParentProductCode >
        <EAN />
        <Name>Kołpaki rozmiar 16’ komplet</Name>
        <Description>Kołpaki rozmiar 16’ komplet</Description >
        <Unit>szt.</Unit>
        <GroupId>123</GroupId>
        <Images>
          <Image>
            <Name>szampon.jpg</Name>
            <Size>16804</Size>
            <Data>0xFFDDDSFFSDSD</Data>
          </Image>
        </Images>
      </Product>
      <Success>true</Success>
    </GetProductResponse>
  </s:Body>
</s:Envelope>
```

Element	Typ	Opis
Code	varchar(30)	Kod towaru w systemie ERP
Brand	varchar(30)	Marka towaru
EAN	varchar(30)	Kod EAN towaru
Name	varchar(255)	Nazwa towaru
Description	varchar(255)	Opis towaru
Unit	varchar(5)	Jednostka miary (np. szt.)
GroupId	int	Id grupy towarowej, do której przypisany jest towar
ProductAttribute/Name	varchar(20)	Nazwa atrybutu
ProductAttribute/Value	varchar(255)	Wartość atrybutu
Set/Element/Code	varchar(30)	Kod towaru będącego elementem zestawu
Set/Element/Quantity	varchar(30)	Ilość towaru będącego elementem zestawu
Image/Name	varchar(255)	Nazwa pliku
Image/Size	int	Rozmiar pliku – maksymalny rozmiar 10485760

Image/Data	BINARY BASE64	Dane binarne
ParentProductCode	varchar(255)	Kod towaru grupującego warianty

2.4.2.3. Polityka cenowa

Informacje o cenie towaru będą pochodziły z systemu ERP. Do każdego SI będą wysyłane osobne cenniki:

- Cena regularna (cena katalogowa) M
- Cena detaliczna M
- Cena promocyjna M
- Cena wyprzedażowa M
- Cena hurtowa M
- Cena regularna (cena katalogowa) S
- Cena detaliczna S
- Cena promocyjna S
- Cena wyprzedażowa S
- Cena hurtowa S

Wysyłana cena będzie ceną brutto ze stawką VAT zgodną ze zdefiniowaną dla towaru stawką.
Synchronizacje cen będzie można wywołać w 2 trybach:

- Pełna – synchronizowane są ceny wszystkich towarów udostępnionych dla systemu SC
- Różnicowa – synchronizowane są tylko ceny zmienione od czasu ostatniej synchronizacji

Na potrzeby synchronizacji różnicowej przygotowany zostanie mechanizm do rejestracji zmiany ceny towaru. Informacje te będą przechowywane w bazie danych w dedykowanych tabelach.
Dodatkowe rabaty (np. dla grup klientów) będą konfigurowane w systemie SC, do systemu ERP na Zamówienia będą wprowadzane już ceny ostateczne.

GetPrices

Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <Action s:mustUnderstand="1"
xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://tempuri.org/IService/GetPrices</Action>
  </s:Header>
  <s:Body>
    <GetPricesRequest xmlns="OrderSoft">
      <Authentication xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Password />
        <Username>ADMIN</Username>
        <ShopId>2</ShopId>
      </Authentication>
      <IsDifferential>true</IsDifferential>
    </GetPricesRequest>
  </s:Body>
</s:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
    <GetPricesResponse xmlns="OrderSoft">
      <ErrorMessage />
      <LoginResult>OK</LoginResult>
      <Prices xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
```

```
      <Price>
        <Code>AKUMULATOR</Code>
        <Currency>PLN</Currency>
        <RegularPrice>405.61</RegularPrice>
        <RetailPrice>401.20</RetailPrice >
        <PromoPrice>401.20</PromoPrice >
        <SalePrice>401.20</SoldPrice >
        <DealersPrice>401.20</DealersPrice >
        <Tax>23.00</Tax>
      </Price>
    </Prices>
    <Success>true</Success>
  </GetPricesResponse>
 </s:Body>
</s:Envelope>
```

Element	Typ	Opis
IsDifferential	boolean	Tryb eksportu: False – pełny, True – różnicowy
Code	varchar(30)	Kod towaru
SalePrice	decimal(12,2)	Cena wyprzedażowa
RegularPrice	decimal(12,2)	Cena katalogowa
RetailPrice	decimal(12,2)	Cena detaliczna
PromoPrice	decimal(12,2)	Cena promocyjna
DealersPrice	decimal(12,2)	Cena hurtowa
Tax	decimal(12,2)	Stawka VAT sprzedaży
Currency	varchar(3)	Waluta, w której jest cena

2.4.2.4. Stany magazynowe

Udostępniane w SI stany będą dotyczyły określonych w konfiguracji magazynów systemu ERP. Synchronizowane stany będą uwzględniały rezerwacje wynikające z potwierdzonych, niezrealizowanych zamówień oraz zamówienia niepotwierdzone.
Synchronizacje stanów magazynowych będzie można wywołać w 2 trybach:

- Pełna – synchronizowane są stany wszystkich towarów udostępnionych dla systemu SC
- Różnicowa – synchronizowane są tylko stany zmienione od czasu ostatniej synchronizacji

GetStockStates

Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <Action s:mustUnderstand="1"
xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://tempuri.org/IService/GetStockStates</Action>
  </s:Header>
  <s:Body>
    <GetStockStatesRequest xmlns="OrderSoft">
      <Authentication xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Password />
        <Username>ADMIN</Username>
        <ShopId>2</ShopId>
      </Authentication>
      <IsDifferential>true</IsDifferential>
    </GetStockStatesRequest>
  </s:Body>
</s:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
    <GetStockStatesResponse xmlns="OrderSoft">
      <ErrorMessage />
```

```
<LoginResult>OK</LoginResult>
<StockStates xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <StockState>
    <Code>AMORTYZATOR_PRZÓD</Code>
    <Stock>MAGAZYN</Stock>
    <Value>0.0000</Value>
  </StockState>
  <StockState>
    <ProductCode>AMORTYZATOR_TYŁ</ProductCode>
    <Stock>MAGAZYN</Stock>
    <Value>12.0000</Value>
  </StockState>
</StockStates>
<Success>true</Success>
</GetStockStatesResponse>
</s:Body>
</s:Envelope>
```

Element	Typ	Opis
IsDifferential	Boolean	Tryb eksportu: false – pełny, true - różnicowy
Code	varchar(30)	Kod towaru
Stock	varchar(30)	Kod magazynu
Value	Decimal (10,4)	Ilość wolna do sprzedaży na magazynie

2.5. Zamówienia

2.5.1. Dodanie zamówienia

Synchronizacja zamówień będzie jednostronna: z systemu SI do ERP.
Na potrzeby integracji konieczne jest dodanie nowej definicji atrybutów: **Sposób dostawy** oraz wskazanie kartoteki usługowej, na której rejestrowane będą koszty dostawy.
Zamówienia będą rejestrowane jako zamówienia niepotwierdzone.

InsertOrders

Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <Action s:mustUnderstand="1"
xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://tempuri.org/IService/InsertOrders</Action>
  </s:Header>
  <s:Body>
    <InsertOrdersRequest xmlns="OrderSoft">
      <Authentication xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Password />
        <Username>ADMIN</Username>
        <ShopId>2</ShopId>
      </Authentication>
      <Orders xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Order>
          <BillCustomer>
            <Address>
              <City>Poznań</City>
              <FlatNo>4</FlatNo>
              <PostalCode>60-000</PostalCode>
              <Street>Akacjowa</Street>
              <StreetNo>22</StreetNo>
            </Address>
            <Company>Firma A</Company>
            <Email>artur.nowak@firmaa.pl</Email>
            <FirstName>Artur</FirstName>
            <LastName>Nowak</LastName>
            <NIP>7390291982</NIP>
            <Phone>600123456</Phone>
```

```
      </BillCustomer>
      <CreationDate>2017-09-22T09:29:00</CreationDate>
      <Description />
      <Id>00012</Id>
      <IsInvoice>true</IsInvoice>
      <Items>
        <Item>
          <Code>KIEROWNICA</Code>
          <Name>Kierownica</Name>
          <Quantity>1</Quantity>
          <RegularPrice>150</RegularPrice>
          <SoldPrice>135</SoldValue>
        </Item>
      </Items>
      <Payment>
        <Currency>PLN</Currency>
        <Form>gotówka</Form>
      </Payment>
      <Shipment>
        <Price>16.90</Price>
        <Type>kurier</Type>
        <Weight>2.50</ Weight >
      </Shipment>
      <ShippCustomer>
        <Address>
          <City>Warszawa</City>
          <FlatNo>5</FlatNo>
          <PostalCode>22-000</PostalCode>
          <Street>Bananowa</Street>
          <StreetNo>2</StreetNo>
        </Address>
        <Company>Firma B</Company>
        <Email>barbara.kowalska@firmab.pl</Email>
        <FirstName>Barbara</FirstName>
        <LastName>Kowalska</LastName>
        <NIP>8932533940</NIP>
        <Phone>500222333</Phone>
      </ShippCustomer>
    </Order>
  </Orders>
 </InsertOrdersRequest>
 </s:Body>
</s:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Header />
 <s:Body>
  <InsertOrdersResponse xmlns="OrderSoft">
   <ErrorMessage />
   <LoginResult>OK</LoginResult>
   <OrdersResults xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <OrderResult>
     <ErrorDescription />
     <OrderId>00012</OrderId>
     <Success>true</Success>
    </OrderResult>
   </OrdersResults>
   <Success>true</Success>
  </InsertOrdersResponse>
 </s:Body>
</s:Envelope>
```

Element	Typ	Opis
OrderId	varchar(255)	Numer zamówienia z SI
BillCustomer	-	Dane do faktury
ShippCustomer	-	Dane do wysyłki

CreationDate	timestamp	Czas złożenia zamówienia
Currency	varchar(3)	Waluta zamówienia
Form	varchar(10)	Forma płatności – słownik zgodny z systemem ERP
Code	varchar(255)	Kod towaru
Name	varchar(255)	Nazwa towaru w systemie SI
RegularPrice	decimal(12,2)	Cena regularna jednostki
SoldPrice	decimal(12,2)	Wartość produktów po rabatach
Quantity	int	Ilość
Description	varchar(255)	Uwagi klienta
IsInvoice	int	Określa czy dla klienta ma być faktura (1) czy nie (0). Determinuje tworzenie kartoteki Klienta w systemie.
FirstName	varchar(255)	Imię
LastName	varchar(255)	Nazwisko
Company	varchar(255)	Firma
NIP	varchar(255)	NIP
Email	varchar(255)	Email
Phone	varchar(255)	Nr. Telefonu
Street	varchar(255)	Ulica
StreetNo	varchar(255)	Numer domu
FlatNo	varchar(255)	Numer lokalu
PostalCode	varchar(255)	Kod pocztowy
City	varchar(255)	Miasto
Shipment/Price	Decimal(10,2)	Koszt dostawy
Shipment/Type	Decimal(10,2)	Sposób dostawy
Shipment/Weight	Decimal(10,2)	Waga paczki

2.5.2. Potwierdzenie zamówienia

Potwierdzenie zamówienia będzie oznaczało przekazanie go do realizacji. SI będzie wywoływał metodę potwierdzenia zamówienia w przypadku płatności za pobraniem oraz po pozytywnym przejściu procesu płatności przez serwisy pośredniczące w płatnościach.
W przypadku przelewów, po rejestracji wpłaty na koncie, zamówienia będą potwierdzane ręcznie przez operatora.

UpdateOrderStatus

Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <Action s:mustUnderstand="1"
xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://tempuri.org/IService/GetOrderStatus</Action>
  </s:Header>
  <s:Body>
    <UpdateOrderStatusRequest xmlns="OrderSoft">
      <Authentication xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Password />
        <Username>ADMIN</Username>
        <ShopId>2</ShopId>
      </Authentication>
      <OrderId>123456</OrderId>
      <Status>Potwierdzenie</Status>
    </UpdateOrderStatusRequest>
  </s:Body>
</s:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
    <UpdateOrderStatusResponse xmlns="OrderSoft">
      <ErrorDescription i:nil="true" xmlns:i="http://www.w3.org/2001/XMLSchema-instance" />
      <LoginResult>OK</LoginResult>
      <Success>true</Success>
    </UpdateOrderStatusResponse >
  </s:Body>
</s:Envelope>
```

Element	Typ	Opis
OrderId	varchar(255)	Numer zamówienia z SI
Status	varchar(30)	Dostępna wartość: Potwierdzenie

2.5.3. Statusy zamówień

Do systemu SI będą wysyłane statusy zamówień klienta, które zmieniły się od czasu ostatniej synchronizacji. Zwracane wartości to:

- Potwierdzone - zapłacone
- Spakowane – wystawiony dokument handlowy FA lub PA
- Wysłane – dokument FA/PA ma wpisany numer listu przewozowego
- Zrealizowane – dokument RO z atrybutem Status z wartością Zrealizowane
- Anulowane – rezygnacja z realizacji

GetOrderStatuses

Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <Action s:mustUnderstand="1"
xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://tempuri.org/IService/GetOrderStatuses</Action>
  </s:Header>
  <s:Body>
    <GetOrderStatusesRequest xmlns="OrderSoft">
      <Authentication xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Password />
        <Username>ADMIN</Username>
        <ShopId>2</ShopId>
      </Authentication>
    </GetOrderStatusesRequest>
  </s:Body>
</s:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
    <GetOrderStatusesResponse xmlns="OrderSoft">
      <ErrorMessage />
      <LoginResult>OK</LoginResult>
      <OrdersStatuses xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <OrderStatus>
          <Invoice/>
          <OrderId>00012</OrderId>
          <Status>POTWIERDZONE</Status>
          <WayBiill/>
        </OrderStatus>
        <OrderStatus>
          <Invoice > PA-123/09/17</Invoice>
```

```
        <OrderId>00099</OrderId>
        <Status>Wysłane</Status>
        <WayBill>342379237492734</WayBill>
      </OrderStatus>
    </OrdersStatuses>
    <Success>true</Success>
  </GetOrderStatusesResponse>
 </s:Body>
</s:Envelope>
```

Element	Typ	Opis
OrderId	varchar(255)	Numer zamówienia z SI
Status	varchar(30)	Status zamówienia w ERP
Invoice	varchar(30)	Numer paragonu/faktury wystawionego o zamówienia
WayBill	varchar(30)	Numer listu przewozowego

3. Organizacja projektu

3.1. Struktura projektu

	OrderSoft	Klient
Kierownik Projektu	Paweł Przytarski	
Członkowie zespołu		
Programista	Adam Pluciński	
Testy i konfiguracja	Dawid Bartkowiak	