

Projekt Soap Web Service  
Sklep internetowy z kartami do gry Magic: The Gathering  
Specyfikacja Wejście-Wyjście  
wersja 1.0

Paweł Marczak i Łukasz Kosmaty

8 maja 2021

## Spis treści

<b>1</b>	<b>Wstęp</b>	<b>1</b>
1.1	Przeznaczenie . . . . .	1
1.2	Zakres . . . . .	2
<b>2</b>	<b>Specyfikacja usługi Web Service</b>	<b>2</b>
2.1	Adres usługi . . . . .	2
2.2	Specyfikacja wsdl . . . . .	2
2.3	Operacje usługi . . . . .	6
2.4	Zwracane obiekty . . . . .	6
2.4.1	Konto . . . . .	7
2.4.2	Stan . . . . .	7
2.4.3	Karta . . . . .	8
2.4.4	Magazyn . . . . .	8
2.4.5	Potw_zamowienia . . . . .	8
2.5	specyfikacja wejście-wyście operacji Web Service . . . . .	8
2.5.1	Operacja zaloguj . . . . .	8
2.5.2	Operacja zwrocMagazyn . . . . .	9
2.5.3	Operacja dodajDoKoszyka . . . . .	12
2.5.4	Operacja usunZKoszyka . . . . .	14
2.5.5	Operacja zlozZamowienie . . . . .	15

## 1 Wstęp

### 1.1 Przeznaczenie

Celem zaprojektowanego web service-u jest automatyzacja procesu składania zamówień na karty w sklepie sprzedającym single (karty na sztuki) do gry Magic: The Gathering.

W obecnej wersji, web service pozwala na przegląd magazynu sklepu (nawiazanie zdefiniowanego statycznie), zalogowanie na istniejące w bazie konto (w tym momencie bez możliwości otwartej rejestracji), uzupełnienie koszyka zamówienia przez użytkownika, złożenie zamówienia (z kontrolą poprawności zamówienia i odpowiednią aktualizacją stanu magazynu) oraz wydrukowanie potwierdzenia zamówienia w formacie pdf. By obsługiwać serwis można korzystać z przygotowanego przez nas klienta obsługiwanego za pośrednictwem html.

Serwis został stworzony z myślą o dalszym rozwoju. Przykładowymi ścieżkami rozwoju są: integracja z zewnętrznymi bazami danych, możliwość

otwartej rejestracji, szyfrowanie przesyłanych komunikatów SOAP (np. przy użyciu standardu SSL), integracja z zewnętrznymi systemami płatniczymi.

## 1.2 Zakres

Dokument opisuje standardy sieciowe, według których zbudowana jest usługa oraz prezentuje przykładowy sposób jej używania.

# 2 Specyfikacja usługi Web Service

Usługa „Sklep Mtg” zaimplementowana została jako usługa sieciowa (Web Service) z użyciem protokołu SOAP.

Usługa dostępna jest przez protokół HTTP za pośrednictwem zaprojektowanego w HTML klienta.

## 2.1 Adres usługi

## 2.2 Specyfikacja wsdl

Listing 1: Test

```
1 <?xml version='1.0' encoding='UTF-8'?><!-- Published by JAX-
   WS RI (http://jax-ws.java.net). RI's version is Metro
   /2.3.2-b608 (trunk-7979; 2015-01-21T12:50:19+0000) JAXWS-
   RI/2.2.11-b150120.1832 JAXWS-API/2.2.12 JAXB-RI/2.2.12-
   b141219.1637 JAXB-API/2.2.13-b141020.1521 svn-revision#
   unknown. --><!-- Generated by JAX-WS RI (http://jax-ws.
   java.net). RI's version is Metro/2.3.2-b608 (trunk-7979;
   2015-01-21T12:50:19+0000) JAXWS-RI/2.2.11-b150120.1832
   JAXWS-API/2.2.12 JAXB-RI/2.2.12-b141219.1637 JAXB-API
   /2.2.13-b141020.1521 svn-revision#unknown. -->
   definitions xmlns:wss="http://docs.oasis-open.org/wss
   /2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
   xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsp1_2="
   http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam=
   "http://www.w3.org/2007/05/addressing/metadata"
   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
   xmlns:tns="http://klasy/" xmlns:xsd="http://www.w3.org
   /2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
   targetNamespace="http://klasy/" name="
   Mtgsklep_serwerImplService">
2 <wsp:Policy wsu:Id="
   Mtgsklep_serwerImplPortBinding_MTOM_Policy-
   Mtgsklep_serwerImplPortBinding_MTOM_Policy">
```

```

3 <ns1:OptimizedMimeSerialization xmlns:ns1="http://schemas.xmlsoap.org/ws/2004/09/policy/optimizedmimeserialization"
  wsp:Optional="true"/>
4 </wsp:Policy>
5 <types>
6 <xsd:schema>
7 <xsd:import namespace="http://klasy/" schemaLocation="http://25.76.141.122:8080/ProjektSoapSerwer/Mtgsklep_serwerImplService?xsd=1"/>
8 </xsd:schema>
9 </types>
10 <message name="zloz_zamowienie">
11 <part name="parameters" element="tns:zloz_zamowienie"/>
12 </message>
13 <message name="zloz_zamowienieResponse">
14 <part name="parameters" element="tns:zloz_zamowienieResponse"/>
15 </message>
16 <message name="Exception">
17 <part name="fault" element="tns:Exception"/>
18 </message>
19 <message name="zaloguj">
20 <part name="parameters" element="tns:zaloguj"/>
21 </message>
22 <message name="zalogujResponse">
23 <part name="parameters" element="tns:zalogujResponse"/>
24 </message>
25 <message name="usun_z_koszyka">
26 <part name="parameters" element="tns:usun_z_koszyka"/>
27 </message>
28 <message name="usun_z_koszykaResponse">
29 <part name="parameters" element="tns:usun_z_koszykaResponse"/>
30 </message>
31 <message name="dodaj_do_koszyka">
32 <part name="parameters" element="tns:dodaj_do_koszyka"/>
33 </message>
34 <message name="dodaj_do_koszykaResponse">
35 <part name="parameters" element="tns:dodaj_do_koszykaResponse"/>
36 </message>
37 <message name="zwroc_magazyn">
38 <part name="parameters" element="tns:zwroc_magazyn"/>
39 </message>
40 <message name="zwroc_magazynResponse">
41 <part name="parameters" element="tns:zwroc_magazynResponse"/>
42 </message>
43 <portType name="Mtgsklep_serwerImpl">
44 <operation name="zloz_zamowienie">

```

```

45 <input wsam:Action="http://klasy/Mtgsklep_serwerImpl/
    zloz_zamowienieRequest" message="tns:zloz_zamowienie"/>
46 <output wsam:Action="http://klasy/Mtgsklep_serwerImpl/
    zloz_zamowienieResponse" message="
    tns:zloz_zamowienieResponse"/>
47 <fault message="tns:Exception" name="Exception" wsam:Action="
    http://klasy/Mtgsklep_serwerImpl/zloz_zamowienie/Fault/
    Exception"/>
48 </operation>
49 <operation name="zaloguj">
50 <input wsam:Action="http://klasy/Mtgsklep_serwerImpl/
    zalogujRequest" message="tns:zaloguj"/>
51 <output wsam:Action="http://klasy/Mtgsklep_serwerImpl/
    zalogujResponse" message="tns:zalogujResponse"/>
52 <fault message="tns:Exception" name="Exception" wsam:Action="
    http://klasy/Mtgsklep_serwerImpl/zaloguj/Fault/Exception"
    />
53 </operation>
54 <operation name="usun_z_koszyka">
55 <input wsam:Action="http://klasy/Mtgsklep_serwerImpl/
    usun_z_koszykaRequest" message="tns:usun_z_koszyka"/>
56 <output wsam:Action="http://klasy/Mtgsklep_serwerImpl/
    usun_z_koszykaResponse" message="
    tns:usun_z_koszykaResponse"/>
57 <fault message="tns:Exception" name="Exception" wsam:Action="
    http://klasy/Mtgsklep_serwerImpl/usun_z_koszyka/Fault/
    Exception"/>
58 </operation>
59 <operation name="dodaj_do_koszyka">
60 <input wsam:Action="http://klasy/Mtgsklep_serwerImpl/
    dodaj_do_koszykaRequest" message="tns:dodaj_do_koszyka"/>
61 <output wsam:Action="http://klasy/Mtgsklep_serwerImpl/
    dodaj_do_koszykaResponse" message="
    tns:dodaj_do_koszykaResponse"/>
62 <fault message="tns:Exception" name="Exception" wsam:Action="
    http://klasy/Mtgsklep_serwerImpl/dodaj_do_koszyka/Fault/
    Exception"/>
63 </operation>
64 <operation name="zwroc_magazyn">
65 <input wsam:Action="http://klasy/Mtgsklep_serwerImpl/
    zwroc_magazynRequest" message="tns:zwroc_magazyn"/>
66 <output wsam:Action="http://klasy/Mtgsklep_serwerImpl/
    zwroc_magazynResponse" message="tns:zwroc_magazynResponse"
    />
67 </operation>
68 </portType>
69 <binding name="Mtgsklep_serwerImplPortBinding" type="
    tns:Mtgsklep_serwerImpl">

```

```

70 <wsp:PolicyReference URI="#
    Mtgsklep_serwerImplPortBinding_MTOM_Policy-
    Mtgsklep_serwerImplPortBinding_MTOM_Policy"/>
71 <soap:binding transport="http://schemas.xmlsoap.org/soap/http
    " style="document"/>
72 <operation name="zloz_zamowienie">
73 <soap:operation soapAction=""/>
74 <input>
75 <soap:body use="literal"/>
76 </input>
77 <output>
78 <soap:body use="literal"/>
79 </output>
80 <fault name="Exception">
81 <soap:fault name="Exception" use="literal"/>
82 </fault>
83 </operation>
84 <operation name="zaloguj">
85 <soap:operation soapAction=""/>
86 <input>
87 <soap:body use="literal"/>
88 </input>
89 <output>
90 <soap:body use="literal"/>
91 </output>
92 <fault name="Exception">
93 <soap:fault name="Exception" use="literal"/>
94 </fault>
95 </operation>
96 <operation name="usun_z_koszyka">
97 <soap:operation soapAction=""/>
98 <input>
99 <soap:body use="literal"/>
100 </input>
101 <output>
102 <soap:body use="literal"/>
103 </output>
104 <fault name="Exception">
105 <soap:fault name="Exception" use="literal"/>
106 </fault>
107 </operation>
108 <operation name="dodaj_do_koszyka">
109 <soap:operation soapAction=""/>
110 <input>
111 <soap:body use="literal"/>
112 </input>
113 <output>
114 <soap:body use="literal"/>
115 </output>

```

```

116 <fault name="Exception">
117 <soap:fault name="Exception" use="literal"/>
118 </fault>
119 </operation>
120 <operation name="zwroc_magazyn">
121 <soap:operation soapAction=""/>
122 <input>
123 <soap:body use="literal"/>
124 </input>
125 <output>
126 <soap:body use="literal"/>
127 </output>
128 </operation>
129 </binding>
130 <service name="Mtgsklep_serwerImplService">
131 <port name="Mtgsklep_serwerImplPort" binding="
      tns:Mtgsklep_serwerImplPortBinding">
132 <soap:address location="http://25.76.141.122:8080/
      ProjektSoapSerwer/Mtgsklep_serwerImplService"/>
133 </port>
134 </service>
135 </definitions>

```

## 2.3 Operacje usługi

Usługa zaplanowana jest do obsługi operacji:

- zaloguj - zwraca informacje o koncie związanym z podanym loginem i hasłem
- zwroc magazyn - zwraca informacje o aktualnym stanie magazynu sklepu
- dodajDoKoszyka - ustawia w koszyku klienta podaną liczbę kopii danej karty
- usunZKoszyka - usuwa z koszyka daną pozycję (wszystkie karty o danej nazwie)
- zlozZamowienie - składa zamówienie (weryfikuje poprawność koszyka, aktualizuje magazyn i stan konta użytkownika, wysyła informacje o potwierdzeniu zamówienia)

## 2.4 Zwracane obiekty

Szczegółowiej opisane zostaną tylko elementy budzące wątpliwości.

### 2.4.1 Konto

Obiekty zawierające informacje o zalogowanym użytkowniku, tzn.:

- login (String)
- haslo (String)
- imie (String)
- nazwisko (String)
- stan\_konta (Float) - wirtualny stan konta służący do realizacji płatności
- miasto (String)
- kod\_pocztowy (String)
- adres (String)
- email (String)
- numer\_telefonu (String)
- ArrayList<Stan> koszyk- lista stanów w koszyku (karta+ liczba kopii karty+cena)

### 2.4.2 Stan

Obiekty zawierające informacje o danym stanie związanym z daną kartą (zarówno magazyn jak i koszyk klienta składają się z listy stanów ArrayList<Stan>), tzn.:

- karta (Karta)- obiekt karta zawierający informacje o karcie
- na\_stanie (int) -liczba kopii karty karta w danym stanie
- cena (float)- cena pojedynczej kopii karty karta brutto
- wartosc\_razem- cena wszystkich kart w danym stanie brutto ( $cena * na\_stanie$ )



### 2.4.3 Karta

Obiekty zawierające informacje o danej karcie, tzn.:

- nazwa (String)
- opis (String) - krótki opis karty: kolor, Set, rzadkość
- ilustracja (Image)- grafika karty

### 2.4.4 Magazyn

Obiekty zawierające dane o stanie magazynu, tzn.:

- ArrayList<Stan> lista\_st - lista stanów, z których złożony jest magazyn

### 2.4.5 Potw\_zamowienia

Obiekty zawierające informacje o statusie złożonego zamówienia, tzn.:

- kwota (Float)- kwota brutto w pln.
- kwota\_netto(Float)- kwota netto w pln.
- dane sklepu (Dane\_sklepu)- informacje o sprzedawcy.
- czy\_zatwierdzono(Boolean) -True, jeśli zamówienie zostało pomyślnie złożone, False- jeśli nie zostało pomyślnie złożone
- message (String)- komentarz dotyczący statusu zamówienia

## 2.5 specyfikacja wejście-wyjście opercji Web Service

By operacje działały poprawnie użytkownicy muszą trzymać się określonych niżej standardów w wysyłanych zapytaniach SOAP.

### 2.5.1 Operacja zaloguj

Dane wejściowe:

- login (String)- login skojarzony z kontem
- haslo (String)- hasło skojarzone z kontem

Dane wyjściowe:

- obiekt typu Konto skojarzony z podanymi loginem i hasłem
- zwraca wyjątek, jeżeli w bazie serwera nie ma konta o loginie i hasle podanych w zapytaniu

Listing 2: Przykładowy SOAP request operacji zaloguj

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
  soap/envelope/" xmlns:klas="http://klasy/">
3   <soapenv:Header/>
4   <soapenv:Body>
5     <klas:zaloguj>
6       <!--Optional:-->
7       <arg0>Lukasz</arg0>
8       <!--Optional:-->
9       <arg1>Kosmaty</arg1>
10    </klas:zaloguj>
11  </soapenv:Body>
12 </soapenv:Envelope>

```

Listing 3: Przykładowy SOAP response operacji zaloguj

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope
  /">
3   <S:Body>
4     <ns2:zalogujResponse xmlns:ns2="http://klasy/">
5       <return>
6         <login>Lukasz</login>
7         <haslo>Kosmaty</haslo>
8         <imie>Lukasz</imie>
9         <nazwisko>Kosmaty</nazwisko>
10        <stan_konta>566.45</stan_konta>
11        <miasto>Bialystok</miasto>
12        <kod_pocztowy>15-888</kod_pocztowy>
13        <adres>ul. szkolna 17</adres>
14        <email>ookosmaty@gmail.com</email>
15        <numer_telefonu>666662222</numer_telefonu>
16      </return>
17    </ns2:zalogujResponse>
18  </S:Body>
19 </S:Envelope>

```

### 2.5.2 Operacja zwrocMagazyn

Dane wejściowe:

- brak danych wejściowych

Dane wyjściowe:

- obiekt typu Magazyn zgodny z aktualnym stanem magazynu sklepu

Listing 4: Przykładowy SOAP request operacji zwrocMagazyn

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
  soap/envelope/" xmlns:klas="http://klasy/">
3   <soapenv:Header/>
4   <soapenv:Body>
5     <klas:zwroc_magazyn/>
6   </soapenv:Body>
7 </soapenv:Envelope>

```

Listing 5: Przykładowy SOAP response operacji zwrocMagazyn

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope
  /">
3   <S:Body>
4     <ns2:zwroc_magazynResponse xmlns:ns2="http://klasy/">
5       <return>
6         <lista_st>
7           <karta>
8             <nazwa>Drannith_Stinger</nazwa>
9             <opis>Kolor: czerwony; set: Ikorja , Lair of
              Behemoths; rzadkosc: common</opis>
10            <ilustracja>
11              <xop:Include href="cid:6ee14dda-65a6
                -4402-921d-c46c616077c6@example.
                jaxws.sun.com" xmlns:xop="http://www
                .w3.org/2004/08/xop/include"/>
12            </ilustracja>
13          </karta>
14          <na_stanie>70</na_stanie>
15          <cena>0.89</cena>
16          <wartosc_razem>62.3</wartosc_razem>
17        </lista_st>
18        <lista_st>
19          <karta>
20            <nazwa>Extinction_Event</nazwa>
21            <opis>Kolor: czarny; set: Ikorja , Lair of
              Behemoths; rzadkosc: rare</opis>
22          <ilustracja>

```

```

23         <xop:Include href="cid:5596a57d-6e95-45
                df-a0b7-65be1d1075b6@example.jaxws.
                sun.com" xmlns:xop="http://www.w3.
                org/2004/08/xop/include"/>
24     </ilustracja>
25 </karta>
26 <na_stanie>21</na_stanie>
27 <cena>6.83</cena>
28 <wartosc_razem>143.43</wartosc_razem>
29 </lista_st>
30 <lista_st>
31   <karta>
32     <nazwa>Gemrazer</nazwa>
33     <opis>Kolor: zielony; set: Ikorja , Lair of
        Behemoths; rzadkosc: rare</opis>
34     <ilustracja>
35       <xop:Include href="cid:804dd9f3-2ff5-41
                bf-b1ba-56731323d783@example.jaxws.
                sun.com" xmlns:xop="http://www.w3.
                org/2004/08/xop/include"/>
36     </ilustracja>
37   </karta>
38   <na_stanie>18</na_stanie>
39   <cena>5.44</cena>
40   <wartosc_razem>97.92</wartosc_razem>
41 </lista_st>
42 <lista_st>
43   <karta>
44     <nazwa>Luminous_Broodmoth</nazwa>
45     <opis>Kolor: bialy; set: Ikorja , Lair of
        Behemoths; rzadkosc: mythic</opis>
46     <ilustracja>
47       <xop:Include href="cid:95d1d9c3-1e4e-4
                eee-91c2-c0a255af5077@example.jaxws.
                sun.com" xmlns:xop="http://www.w3.
                org/2004/08/xop/include"/>
48     </ilustracja>
49   </karta>
50   <na_stanie>4</na_stanie>
51   <cena>8.99</cena>
52   <wartosc_razem>35.96</wartosc_razem>
53 </lista_st>
54 <lista_st>
55   <karta>
56     <nazwa>Ominous_Seas</nazwa>
57     <opis>Kolor: niebieski; set: Ikorja , Lair
        of Behemoths; rzadkosc: uncommon</opis>
58     <ilustracja>
59       <xop:Include href="cid:42b528f5-37bc

```

```

-4925-a113-ebce0d35aa18@example.
jaxws.sun.com" xmlns:xop="http://www
.w3.org/2004/08/xop/include"/>
60         </ilustracja>
61     </karta>
62     <na_stanie>43</na_stanie>
63     <cena>2.02</cena>
64     <wartosc_razem>86.86</wartosc_razem>
65 </lista_st>
66 </return>
67 </ns2:zwroc_magazynResponse>
68 </S:Body>
69 </S:Envelope>

```

### 2.5.3 Operacja dodajDoKoszyka

Operacja wbrew nazwie nie dodaje pozycji do koszyka, a ustawia w koszyku podaną przez klienta pozycję w podanej liczbie sztuk.

Dane wejściowe:

- login (String)- login skojarzony z kontem, do którego koszyka się odnosimy
- haslo (String)- hasło skojarzone z kontem, do którego koszyka się odnosimy
- nazwa\_karty (String)- nazwa karty, którą chcemy włożyć do koszyka (nazwa musi być identyczna z odpowiednią nazwą w magazynie)
- liczba (int)- liczba kopii karty, którą chcemy ustawić w koszyku

Dane wyjściowe:

- obiekt typu Konto ze zaktualizowanym koszykiem
- zwraca wyjątek, jeżeli w bazie serwera nie ma konta o loginie i hasle podanych w zapytaniu, lub nie ma karty o podanej nazwie w magazynie.

Listing 6: Przykładowy SOAP request operacji dodajDoKoszyka

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
  soap/envelope/" xmlns:klas="http://klasy/">
3   <soapenv:Header/>

```

```

4    <soapenv:Body>
5        <klas:dodaj_do_koszyka>
6            <!--Optional:-->
7            <arg0>Lukasz</arg0>
8            <!--Optional:-->
9            <arg1>Kosmaty</arg1>
10           <!--Optional:-->
11           <arg2>Extinction_Event</arg2>
12           <arg3>5</arg3>
13       </klas:dodaj_do_koszyka>
14   </soapenv:Body>
15 </soapenv:Envelope>

```

Listing 7: Przykładowy SOAP response operacji dodajDoKoszyka

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope
  /">
3      <S:Body>
4          <ns2:dodaj_do_koszykaResponse xmlns:ns2="http://klasy/"
5          >
6              <return>
7                  <login>Lukasz</login>
8                  <haslo>Kosmaty</haslo>
9                  <imie>Lukasz</imie>
10                 <nazwisko>Kosmaty</nazwisko>
11                 <stan_konta>566.45</stan_konta>
12                 <miasto>Bialystok</miasto>
13                 <kod_pocztowy>15-888</kod_pocztowy>
14                 <adres>ul. szkolna 17</adres>
15                 <email>oookosmaty@gmail.com</email>
16                 <numer_telefonu>666662222</numer_telefonu>
17                 <koszyk>
18                     <karta>
19                         <nazwa>Extinction_Event</nazwa>
20                         <opis>Kolor: czarny; set: Ikorcia , Lair of
21                         Behemoths; rzadkosc: rare</opis>
22                         <ilustracja>
23                             <xop:Include href="cid:ca80dfab-5900-426
24                             e-9802-a9499384a0be@example.jaxws.
25                             sun.com" xmlns:xop="http://www.w3.
26                             org/2004/08/xop/include"/>
27                         </ilustracja>
28                     </karta>
29                     <na_stanie>5</na_stanie>
30                     <cena>6.83</cena>
31                     <wartosc_razem>34.15</wartosc_razem>
32                 </koszyk>
33             </return>

```

```

29         </ns2:dodaj_do_koszykaResponse>
30     </S:Body>
31 </S:Envelope>

```

#### 2.5.4 Operacja usunZKoszyka

Operacja usuwa podaną pozycję z koszyka podanego konta.

Dane wejściowe:

- login (String)- login skojarzony z kontem, do którego koszyka się odnosimy
- haslo (String)- hasło skojarzone z kontem, do którego koszyka się odnosimy
- nazwa\_karty (String)- nazwa karty, którą chcemy usunąć z koszyka (nazwa musi być identyczna z odpowiednią nazwą w magazynie). Usuwa całą pozycję (wszystkie sztuki).

Dane wyjściowe:

- obiekt typu Konto ze zaktualizowanym koszykiem
- zwraca wyjątek, jeżeli w bazie serwera nie ma konta o loginie i hasle podanych w zapytaniu, lub nie ma karty o podanej nazwie w koszyku.

Listing 8: Przykładowy SOAP request operacji usunZKoszyka

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
   soap/envelope/" xmlns:klas="http://klasy/">
3     <soapenv:Header/>
4     <soapenv:Body>
5         <klas:usun_z_koszyka>
6             <!--Optional:-->
7             <arg0>Lukasz</arg0>
8             <!--Optional:-->
9             <arg1>Kosmaty</arg1>
10            <!--Optional:-->
11            <arg2>Extinction_Event</arg2>
12        </klas:usun_z_koszyka>
13    </soapenv:Body>
14 </soapenv:Envelope>

```

Listing 9: Przykładowy SOAP response operacji usunZKoszyka

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope
  /">
3   <S:Body>
4     <ns2:usun_z_koszykaResponse xmlns:ns2="http://klasy/">
5       <return>
6         <login>Lukasz</login>
7         <haslo>Kosmaty</haslo>
8         <imie>Lukasz</imie>
9         <nazwisko>Kosmaty</nazwisko>
10        <stan_konta>566.45</stan_konta>
11        <miasto>Bialystok</miasto>
12        <kod_pocztowy>15-888</kod_pocztowy>
13        <adres>ul. szkolna 17</adres>
14        <email>oookosmaty@gmail.com</email>
15        <numer_telefonu>666662222</numer_telefonu>
16      </return>
17    </ns2:usun_z_koszykaResponse>
18  </S:Body>
19 </S:Envelope>
```

### 2.5.5 Operacja zlozZamowienie

Operacja sprawdza poprawność koszyka (spójność z magazynem), stan konta użytkownika i zwraca potwierdzenie transakcji oraz aktualizuje stan magazynu i opróżnia koszyk.

Dane wejściowe:

- login (String)- login skojarzony z kontem, do którego koszyka się odnosimy
- haslo (String)- hasło skojarzone z kontem, do którego koszyka się odnosimy

Dane wyjściowe:

- obiekt typu Potw\_zamowienia z wszelkimi informacjami dotyczącymi transakcji
- zwraca wyjątek, jeżeli w bazie serwera nie ma konta o loginie i hasle podanych w zapytaniu.
- jeżeli zamówienie nie zostało potwierdzone, wtedy w potwierdzeniu zwracane jest czy\_zatwierdzono = False i komentarz w polu message.



Listing 10: Przykładowy SOAP request operacji zlozZamowienie

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
  soap/envelope/" xmlns:klas="http://klasy/">
3   <soapenv:Header/>
4   <soapenv:Body>
5     <klas:zloz_zamowienie>
6       <!--Optional:-->
7       <arg0>Lukasz</arg0>
8       <!--Optional:-->
9       <arg1>Kosmaty</arg1>
10    </klas:zloz_zamowienie>
11  </soapenv:Body>
12 </soapenv:Envelope>

```

Listing 11: Przykładowy SOAP response operacji zlozZamowienie

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope
  /">
3   <S:Body>
4     <ns2:zloz_zamowienieResponse xmlns:ns2="http://klasy/">
5       <return>
6         <kwota>34.15</kwota>
7         <kwota_netto>27.76</kwota_netto>
8         <dane_sklepu>
9           <adres>Boboli 10</adres>
10          <nazwa>Znowu trzeba w magica grac.Magiaaa!</
            nazwa>
11          <miasto>Bialystok</miasto>
12          <kod_pocztowy>15-431</kod_pocztowy>
13          <email>Magiaaa@gmail.com</email>
14          <NIP>68830299831</NIP>
15          <wlasciciel>Pawel Roszkowski</wlasciciel>
16        </dane_sklepu>
17        <czy_zatwierdzono>true</czy_zatwierdzono>
18        <message>zamowienie potwierdzone</message>
19      </return>
20    </ns2:zloz_zamowienieResponse>
21  </S:Body>
22 </S:Envelope>

```