RASTIN 2.0

WYMIANA DANYCH

SPECYFIKACJA TECHNICZNA SYNCHRONICZNEGO PROTOKOŁU RASTIN 2.0

Wersja dokumentacji 1.1 z dnia 2023-05-10

Status dokumentu

Opracowanie wewnętrzne

Słowa kluczowe

BIG, Web Service, serwis, RASTIN, PESEL, RDO

Wszelkie korekty, komentarze oraz uwagi proszę przesyłać na adres pomocit@krd.pl

KRD BIG S.A., ul. Danuty Siedzikówny 12, 51-214 Wrocław
https://www.krd.pl

Atrybuty dokumentu

	Atrybut	Wartość
	A	B
1	Projekt	Rastin 2.0
2	Tytuł	Wymiana danych
3	Podtytuł	Specyfikacja techniczna synchronicznego protokołu Rastin 2.0
4	Wersja dokumentacji	1.1
5	Czas wersji	2023-05-10
6	Plik	Rastin 2.0 - Specyfikacja techniczna.docx/pdf
7	Autorzy	Michał Łeptuch, Maciej Łukasik
8	Nadzór	Justyna Grabek
9	Prawa autorskie	Copyright © Krajowy Rejestr Długów, 2023
10	Komentarz	

Historia dokumentu

	Atrybut	Wartość	Data
	A	B	C
1	Wersja dokumentacji	1.0	2023-04-06
2	Autor	Michał Łeptuch	
3	Sprawdził treść	Justyna Grabek	2023-04-06
4	Sprawdził formę	Maciej Łukasik	2023-04-06
5	Zatwierdził	Justyna Grabek	2023-04-06
6	Opis	Pierwsza wersja dokumentacji.	
1	Wersja dokumentacji	1.1	2023-05-10
2	Autor	Maciej Łukasik	
3	Sprawdził treść	Maciej Łukasik	2023-05-10
4	Sprawdził formę	Maciej Łukasik	2023-05-10
5	Zatwierdził	Maciej Łukasik	2023-05-10
6	Opis	Aktualizacja w zakresie limitów odpytań (dzienny i miesięczny).	
1	Wersja dokumentacji		
2	Autor		
3	Sprawdził treść		
4	Sprawdził formę		
5	Zatwierdził		
6	Opis		
1	Wersja dokumentacji		
2	Autor		
3	Sprawdził treść		
4	Sprawdził formę		
5	Zatwierdził		
6	Opis		

Spis treści

Wstęp

Korzystanie z usług KRD BIG S.A. (dalej KRD) może odbywać się m.in. za pomocą witryny WWW biura, bądź przez serwisy internetowe wykorzystujące protokół SOAP. Jednym z takich serwisów internetowych (ang. Web Service) jest serwis o nazwie RASTIN, wersja 2.0.

Interfejs SOAP umożliwia bezpośrednie połączenie aplikacji Klienta z systemem KRD tak, aby Klient wprost z używanej przez siebie aplikacji mógł w sposób synchroniczny (w czasie rzeczywistym, ze średnim czasem odpowiedzi wynoszącym zazwyczaj około 200-300 ms, nie licząc narzutu sieci, z max. timeout = 1 minuta) sprawdzić dane weryfikowanej osoby. Ten sposób jest bardzo wygodny dla Klientów, którzy mogą, albo dostosować aplikacje, z których już korzystają, albo skorzystać z aplikacji, które wcześniej zostały dostosowane do tego typu współpracy z KRD.

Dokument ten opisuje metody serwisu RASTIN 2.0 oraz sposoby połączenia Klientów z serwisem RASTIN 2.0.

Uwaga!

W dalszej części dokumentu serwis **RASTIN 2.0** będzie określany jako **RASTIN** (bez numeru wersji).

1. Metody serwisu RASTIN

1.1. Weryfikacje danych osób w rejestrach publicznych

RASTIN jest serwisem internetowym odpytującym rejestr PESEL oraz Rejestr Dowodów Osobistych (dalej RDO) będące częścią Systemu Rejestrów Państwowych (dalej SRP), w sposób i na warunkach określonych w obowiązujących przepisach (w szczególności w *Ustawie z dnia 9 kwietnia 2010 r. o udostępnianiu informacji gospodarczych i wymianie danych gospodarczych*, *Ustawie z dnia 24.09.2010 r. o ewidencji ludności* i *Ustawie z dnia z dnia 6 sierpnia 2010 r. o dowodach osobistych*) oraz zgodnie z warunkami technicznymi określonymi przez Ministerstwo Cyfryzacji jako właściciela rejestrów.

Przy pomocy serwisu RASTIN Klient KRD ma możliwość sprawdzenia poprawności przekazywanych danych przed umieszczeniem ich w bazie KRD. W tym celu może skorzystać z następujących usług:

- weryfikacja danych w rejestrze PESEL
- weryfikacja dowodu osobistego (dane)
- weryfikacja dowodu osobistego (ważność)
- weryfikacja w rejestrze PESEL, czy osoba żyje

2. Punkty końcowe

Serwis RASTIN dla zachowania kompatybilności z różnymi systemami klienckimi eksponuje dwa punkty końcowe dla każdej z metod: DefaultEndpoint oraz WsHttpBindingEndpoint:

- DefaultEndpoint: ten rodzaj punktów końcowych wykorzystuje wiązanie typu *basicHttpBinding* i jest zgodny z wersją 1.1 protokołu SOAP. Powinni z niego korzystać Klienci, którzy łączą się ze starszych systemów niewspierających SOAP 1.2, np. .NET w wersji 2.0 lub starszej.

  Adresy tego punktu to:

  - wersja produkcyjna (logowanie bez użycia certyfikatu): https://services.krd.pl/Rastin/v2/Verification.svc
  - wersja demonstracyjna (logowanie bez użycia certyfikatu): https://demo.krd.pl/Rastin/v2/Verification.svc
  - wersja produkcyjna (logowanie z użyciem certyfikatu): https://services.krd.pl/Rastin/v2/cert/Verification.svc
  - wersja demonstracyjna (logowanie z użyciem certyfikatu): https://demo.krd.pl/Rastin/v2/cert/Verification.svc

- WsHttpBindingEndpoint: ten punkt końcowy jest w pełni zgodny ze specyfikacją SOAP 1.2 oraz WS-Addressing. Wykorzystuje on wiązanie typu *wsHttpBinding*. **Zalecamy korzystać właśnie z niego**.

  Adresy tego punktu to:

  - wersja produkcyjna (logowanie bez użycia certyfikatu): https://services.krd.pl/Rastin/v2/Verification.svc/ws
  - wersja demonstracyjna (logowanie bez użycia certyfikatu): https://demo.krd.pl/Rastin/v2/Verification.svc/ws
  - wersja produkcyjna (logowanie z użyciem certyfikatu): https://services.krd.pl/Rastin/v2/cert/Verification.svc/ws
  - wersja demonstracyjna (logowanie z użyciem certyfikatu): https://demo.krd.pl/Rastin/v2/cert/Verification.svc/ws

WSDL produkcyjny: https://services.krd.pl/Rastin/v2/Verification.svc?singleWsdl
WSDL demonstracyjny: https://demo.krd.pl/Rastin/v2/Verification.svc?singleWsdl

3. Logowanie do systemu

3.1. Logowanie bez użycia certyfikatu

Logowanie do serwisu jest możliwe bez użycia certyfikatu uwierzytelniającego komunikację pomiędzy KRD a Klientem jedynie przez adres punktu końcowego.

W przypadku niekorzystania z certyfikatu możliwe są 3 sposoby zalogowania.

3.1.1. Login i hasło

Aby zalogować się do systemu przy użyciu loginu i hasła należy w sekcji *Authorization*, w węźle *AuthorizationType*, ustawić wartość *LoginAndPassword*. Dodatkowo należy przekazać login w węźle *Login* oraz hasło w węźle *Password*.

```
<aut:Authorization>
   <aut:AuthorizationType>LoginAndPassword</aut:AuthorizationType>
   <aut:Login>login</aut:Login>
   <aut:Password>hasło</aut:Password>
</aut:Authorization>
```

3.1.2. Login i zaszyfrowane hasło

Istnieje możliwość zalogowania się do serwisu, podając login oraz zaszyfrowane hasło. W tym celu w sekcji *Authorization*, w węźle *AuthorizationType* należy przekazać wartość *LoginAndPasswordHash*, w węźle *Login* należy przekazać login, a w węźle *PasswordHash* należy przekazać wartość zaszyfrowanego hasła. Skrót hasła można wyliczyć, używając poniższych metod:

```
public static string HashPassword(string key)
{
 byte[] _key = SHA1.Create().ComputeHash(Encoding.ASCII.GetBytes(key));
 return string.Concat(_key.Select(x => x.ToString("X2")));
}
```

lub:

```
public static string HashPassword(string key)
{
 byte[] _key = SHA1.Create().ComputeHash(Encoding.UTF8.GetBytes(key));
 return string.Concat(_key.Select(x => x.ToString("X2")));
}
```

```
   <aut:Authorization>
      <aut:AuthorizationType>LoginAndPasswordHash</aut:AuthorizationType>
      <aut:Login>login</aut:Login>
      <aut:PasswordHash>wynik funkcji skrótu hasła</aut:PasswordHash>
   </aut:Authorization>
```


3.1.3. Identyfikator trwającej sesji logowania

Sposób ten umożliwia zalogowanie do serwisu przy użyciu ciągu znaków identyfikujących poprzedzające logowanie (tzw. *ticket*). Sytuacja taka jest możliwa, jeśli Klient logował się do systemu przy użyciu innej metody logowania i w odpowiedzi dostał *ticket*. *Ticket* jest ważny przez 24 godziny od momentu jego utworzenia. Aby zalogować się w ten sposób, w sekcji *Authorization*, w węźle *AuthorizationType* należy podać wartość *Ticket*, a w węźle *Ticket* wartość ciągu otrzymaną przy poprzednim logowaniu.

```
<aut:Authorization>
   <aut:AuthorizationType>Ticket</aut:AuthorizationType>
   <aut:Ticket>wartość ciągu otrzymana przy poprzednim logowaniu</aut:Ticket>
</aut:Authorization>
```

3.2. Logowanie z użyciem certyfikatu

Uwierzytelnianie po stronie KRD może zostać zabezpieczone za pomocą certyfikatu. Aby używać certyfikatu, należy wykonywać zapytanie na odpowiedni punkt końcowy.

Certyfikat jest ważny przez rok. Pytania w tej kwestii należy kierować na pomocit@krd.pl lub do swojego Opiekuna Biznesowego po stronie KRD.

Przy korzystaniu z certyfikatu możliwych jest 6 sposobów zalogowania.

3.2.1. Login i hasło

Należy przekazać dane jak przy logowaniu bez certyfikatu.

3.2.2. Login i zaszyfrowane hasło

Należy przekazać dane jak przy logowaniu bez certyfikatu.

3.2.3. Identyfikator trwającej sesji logowania

Należy przekazać dane jak przy logowaniu bez certyfikatu.

3.2.4. Certyfikat

Aby zalogować się przy użyciu certyfikatu, w nagłówku autoryzacyjnym, w węźle *AuthorizationType*, należy przekazać wartość *Certificate*. Autoryzacja odbędzie się automatycznie na login zmapowany z certyfikatem (zazwyczaj główny login) po stronie KRD.

```
<aut:Authorization>
   <aut:AuthorizationType>Certificate</aut:AuthorizationType>
</aut:Authorization>
```


3.2.5. Certyfikat wraz z danymi uwierzytelniającymi – przelogowanie na inny login Klienta

Przy logowaniu z certyfikatem istnieje możliwość potwierdzenia tożsamości przy użyciu danych loginu zmapowanego z certyfikatem (zazwyczaj główny login), a następnie wykonywania operacji w kontekście jednego z subloginów Klienta. Aby doprowadzić do takiej sytuacji, należy ustawić typ autoryzacji jako *CertificateWithCredentials* w węźle *AuthorizationType*.

Sublogin Klienta, w którego kontekście wykonywane mają być operacje, należy podać w węźle *Login*.

Jeśli sublogin nie istnieje lub nie posiada uprawnień odpowiednich do wykonania operacji, zwrócony zostanie odpowiedni błąd.

```
<aut:Authorization>
   <aut:AuthorizationType>CertificateWithCredentials</aut:AuthorizationType>
   <aut:Login>login</aut:Login>
</aut:Authorization>
```

3.2.6. Certyfikat wraz z danymi uwierzytelniającymi – przelogowanie na inne konto Klienta

Przy logowaniu z certyfikatem istnieje możliwość potwierdzenia tożsamości przy użyciu danych loginu zmapowanego z certyfikatem (zazwyczaj główny login), a następnie wykonywania operacji w kontekście innego konta Klienta. Aby doprowadzić do takiej sytuacji, należy ustawić typ autoryzacji jako *CertificateWithCredentials* w węźle *AuthorizationType*.

Dane logowania innego konta Klienta należy przekazać w węzłach *Login* oraz *Password/PasswordHash*, analogicznie jak przy logowaniu przy użyciu tych danych.

Jeśli login nie istnieje lub nie posiada uprawnień odpowiednich do wykonania operacji, zwrócony zostanie odpowiedni błąd.

```
<aut:Authorization>
   <aut:AuthorizationType>CertificateWithCredentials</aut:AuthorizationType>
   <aut:Login>login</aut:Login>
   <aut:Password>hasło</aut:Password>
</aut:Authorization>

<aut:Authorization>
   <aut:AuthorizationType>CertificateWithCredentials</aut:AuthorizationType>
   <aut:Login>login</aut:Login>
   <aut:PasswordHash>wynik funkcji skrótu hasła</aut:PasswordHash>
</aut:Authorization>
```


4. (*Metoda 1 – VerifyConsumerIdentityNumber*) Metoda weryfikacji numeru PESEL

Dzięki tej metodzie Klient serwisu ma możliwość weryfikacji poprawności i zgodności przesłanych w zapytaniu danych dotyczących numeru PESEL, imienia, nazwiska i adresów weryfikowanej osoby, z danymi widniejącymi w udostępnionym rejestrze PESEL.

4.1. Wygląd zapytania

4.1.1. Dane autoryzacji

Autoryzacja w tego typu zapytaniu jest spójna z innymi zapytaniami i została opisana w punkcie **3**.

4.1.2. Dane zapytania

Ta sekcja zapytania służy do przekazania danych identyfikujących podmiot, który ma zostać zweryfikowany.

VerifyConsumerIdentityNumberRequest zawiera 5 sekcji (13 pól):

- *ConsumerIdentityNumber* (typu `string`) – *numer PESEL* osoby do weryfikacji – **pole wymagane**.

  Walidacja pola: 11 cyfr
  Przykład: 68061200023

- *FirstName* (typu `string`) – *imię* osoby do weryfikacji – **pole wymagane**.

  Walidacja pola: od 1 do 120 znaków
  Przykład: TERESA

- *Surname* (typu `string`) – *nazwisko* osoby do weryfikacji – **pole wymagane**.

  Walidacja pola: od 1 do 300 znaków
  Przykład: KOS-WIĘCEK

- *PermanentAddress* (typu `PermanentAddress`) – *adres stały* osoby do weryfikacji – **pola opcjonalne**.

  Walidacja pola: tylko kod pocztowy, 5 cyfr
  Przykład: 83000

  Sekcja zawiera następujące pola: *Street* (ulica), *Building* (numer domu), *Flat* (numer mieszkania), *ZipCode* [kod pocztowy (walidowany)], *City* (miasto)

- *TemporaryAddress* (typu `TemporaryAddress`) – *adres tymczasowy* osoby do weryfikacji – **pola opcjonalne**.

  Walidacja pola: tylko kod pocztowy, 5 cyfr
  Przykład: 83000

  Sekcja zawiera następujące pola: *Street* (ulica), *Building* (numer domu), *Flat* (numer mieszkania), *ZipCode* [kod pocztowy (walidowany)], *City* (miasto)

Uwaga!

1. Wielkość liter wprowadzanych w powyższych polach nie ma znaczenia.
2. Jeśli weryfikowana osoba ma dwa imiona, w polu *FirstName* wprowadzamy tylko pierwsze imię.
3. W przypadku wprowadzenia wartości **--** (2 myślniki) dla któregokolwiek elementu sekcji adresowej, tj.: *Street*; *Building*; *Flat*; *City*, nie dojdzie do odpytania rejestru PESEL. Jeśli zapytania zawierają jednak takie wartości, mimo braku faktycznych danych do odpytania dla danego elementu, to zalecamy podawanie dla nich myślników w zapytaniu (*request*) w ilości innej niż 2.
4. Definicja adresu stałego i tymczasowego:
   - adres stały (*PermanentAddress*) jest rejestrowany wtedy, gdy osoba melduje się z zamiarem pobytu stałego,
   - adres tymczasowy (*TemporaryAddress*) jest rejestrowany wtedy, gdy osoba deklaruje pobyt pod tym adresem do jakiejś daty.

   Osoba może mieć jednocześnie aktualny adres stały i tymczasowy. Przykładem takiej osoby może być student zameldowany na pobyt stały u rodziców, a na czas roku akademickiego zameldowany w akademiku.
5. Kody pocztowe – część adresów w rejestrze PESEL może nie mieć kodów pocztowych (*ZipCode*), ponieważ ta informacja jest tam gromadzona od 2005 r.
6. W polu *Street* (dotyczy: *PermanentAddress* i *TemporaryAddress*) podajemy tylko nazwę ulicy bez przedrostków/prefixów: **UL.**; **AL.**; **OS.** i **WS**. Jeśli w adresie weryfikowanej osoby nie ma ulicy, zalecamy usunąć pole *Street* z zapytania (request). Usunięcie tego pola (lub pozostawienie go pustego) i tak spowoduje zwrócenie dla niego przez system wartości "**false**" w odpowiedzi. Ministerstwo Cyfryzacji nie potwierdza odpowiedzią "**true**" danych, których nie ma w ich bazie. Taka sama zasada obowiązuje np. dla adresów bez numeru mieszkania, dla sytuacji z punktu **5.**

   Od 01-03-2015 roku podczas rejestracji adresu w SRP wykorzystywana jest walidacja zgodna ze słownikiem rejestru TERYT udostępnianym przez GUS (dodatkowo podczas wprowadzania i aktualizacji adresów sprawdzany jest kod TERYT gminy, określający województwo, powiat i gminę. Kody TERYT gmin publikowane są na stronie GUS w pliku TERC), dlatego nazwy ulic w rejestrze PESEL powinny być zgodne z nazwami zapisanymi w rejestrze TERYT. Istnieje również możliwość zarejestrowania w SRP adresu z nazwą ulicy, która nie występuje w rejestrze TERYT. Jeśli w rejestrze PESEL widnieje ulica „Kardynała Stefana Wyszyńskiego", natomiast w zapytaniu do bazy podamy tylko „Wyszyńskiego", to wynik weryfikacji będzie negatywny ("**false**") ze względu na niezgodność sposobu zapisania nazwy ulicy. Pisownię ulic ustala gmina dokonująca wpisu w SRP i w jednej gminie może to być np. „Kardynała Stefana Wyszyńskiego", a w drugiej „Stefana Wyszyńskiego".

   Dowody osobiste wydane przed 01-03-2015 roku wg starego wzoru mają wpisane adresy zameldowania, które nie muszą być zgodne z adresami w rejestrze PESEL. Dla przykładowego adresu ze wskazaniem numeru bloku, klatki i mieszkania (**BL.5 KL.1 M.1**) można wprowadzić **BL.5 KL.1** jako numer domu (*Building*), a **M.1** jako numer lokalu (*Flat*). Czasami konieczne może być kilka prób odpytania (różne kombinacje).
7. W SRP można także weryfikować obcokrajowców. Zalecamy robić to ***Metodą 1 – VerifyConsumerIdentityNumber***, ponieważ ***Metoda 2 – VerifyIDCard*** i ***Metoda 3 – VerifyIsIDCardCanceled*** obejmuje tylko dowody osobiste Rzeczypospolitej Polskiej.

W celu zweryfikowania obcokrajowca najlepiej w zapytaniu podać *numer PESEL* + *imię* + *nazwisko*. Obcokrajowiec przy wnioskowaniu o numer PESEL nie zawsze musi podawać swój adres stały/tymczasowy, więc odpowiedzi "**false**" przy danych adresowych nie zawsze będą gwarantowały, że podany w zapytaniu (w requeście) adres jest błędny, może go po prostu nie być w rejestrze PESEL.

8. Nazwisko wpisywane do rejestru PESEL jest ustalane na podstawie aktu stanu cywilnego (małżeństwa / urodzenia) albo dokumentu potwierdzającego tożsamość osoby (np. paszportu, karty pobytu).
9. Nazwisko dwuczłonowe umieszczane w polu *Surname* oddzielamy zazwyczaj myślnikiem (łącznikiem-minusem), np. *KOS-WIĘCEK*.
10. SRP stosuje kodowanie UTF-8. W sprawie wykazu dopuszczalnych znaków specjalnych należy wysłać zgłoszenie na pomocit@krd.pl lub skontaktować się ze swoim Opiekunem Biznesowym po stronie KRD.

```
<dto:ConsumerIdentityNumberRequest>
   <dto:ConsumerIdentityNumber>68061200023</dto:ConsumerIdentityNumber>
   <dto:FirstName>TERESA</dto:FirstName>
   <dto:Surname>KOS-WIĘCEK</dto:Surname>
   <dto:PermanentAddress>
      <dto:Street>NIEPODLEGŁOŚCI</dto:Street>
      <dto:Building>2</dto:Building>
      <dto:Flat>11</dto:Flat>
      <dto:ZipCode>83000</dto:ZipCode>
      <dto:City>Pruszcz Gdański</dto:City>
   </dto:PermanentAddress>
    <dto:TemporaryAddress>
      <dto:Street>KIRKORA</dto:Street>
      <dto:Building>20</dto:Building>
      <dto:Flat>1</dto:Flat>
      <dto:ZipCode>83000</dto:ZipCode>
      <dto:City>Pruszcz Gdański</dto:City>
   </dto:TemporaryAddress>
</dto:ConsumerIdentityNumberRequest>
```

4.1.3. Dane odpowiedzi

W odpowiedzi na zapytanie dostajemy element typu *VerifyConsumerIdentityNumberResponse*, zawierający 3 pola/sekcje:

- *ConsumerIdentytyNumberIsValid* (typu `bool`).
  Zwraca wynik weryfikacji ("**true**" lub "**false**")
  numeru PESEL + *imienia* + *nazwiska* weryfikowanej osoby,
- *PermanentAddressVerificationResult* (typu `AddressVerificationResult`).
  Zwraca wyniki weryfikacji ("**true**" lub "**false**")
  adresu stałego weryfikowanej osoby,
- *TemporaryAddressVerificationResult* (typu `AddressVerificationResult`).
  Zwraca wyniki weryfikacji ("**true**" lub "**false**")
  adresu tymczasowego weryfikowanej osoby.

Uwaga!

Jeśli wynik weryfikacji *numeru PESEL* + *imienia* + *nazwiska* weryfikowanej osoby = "**false**", w odpowiedzi na zapytanie dostajemy element typu *VerifyConsumerIdentityNumberResponse*, zawierający tylko jedno pole: *ConsumerIdentytyNumberIsValid*. W takiej sytuacji nie dochodzi do dalszej weryfikacji, czyli weryfikacji *adresu stałego* i/lub *tymczasowego*.

4.1.4. Przykład zapytania

```
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:aut="http://krd.pl/Authorization"
xmlns:dto="http://krd.pl/Rastin/Dto">
   <soap:Header>
      <aut:Authorization>
         <aut:AuthorizationType>LoginAndPassword</aut:AuthorizationType>
         <aut:Login>login</aut:Login>
         <aut:Password>hasło</aut:Password>
      </aut:Authorization>
   </soap:Header>
   <soap:Body>
      <dto:ConsumerIdentityNumberRequest>
         <dto:ConsumerIdentityNumber>68061200023</dto:ConsumerIdentityNumber>
         <dto:FirstName>TERESA</dto:FirstName>
         <dto:Surname>KOS-WIĘCEK</dto:Surname>
         <dto:PermanentAddress>
            <dto:Street>NIEPODLEGŁOŚCI</dto:Street>
            <dto:Building>2</dto:Building>
            <dto:Flat>11</dto:Flat>
            <dto:ZipCode>83000</dto:ZipCode>
            <dto:City>Pruszcz Gdański</dto:City>
         </dto:PermanentAddress>
          <dto:TemporaryAddress>
            <dto:Street>KIRKORA</dto:Street>
            <dto:Building>20</dto:Building>
            <dto:ZipCode>83000</dto:ZipCode>
            <dto:City>Pruszcz Gdański</dto:City>
         </dto:TemporaryAddress>
      </dto:ConsumerIdentityNumberRequest>
   </soap:Body>
</soap:Envelope>
```

4.1.5. Przykład odpowiedzi

```
<s:Envelope
xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
   <s:Header>
      <a:Action
s:mustUnderstand="1">
http://krd.pl/Rastin/IVerificationService/VerifyConsumerIdentityNumberResponse
      </a:Action>
      <h:Authorization
xmlns:h="http://krd.pl/Authorization/Response"
xmlns="http://krd.pl/Authorization/Response"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
         <Ticket>AAA1B1AA11AA111B1111AA111B1AA1111B1AA111</Ticket>
         <TicketExpirationDate>2022-02-16T12:08:59.11</TicketExpirationDate>
         <ShouldChangePassword>false</ShouldChangePassword>
         <PasswordExpirationDate>2021-05-23T00:00:00+02:00</PasswordExpirationDate>
      </h:Authorization>
   </s:Header>
```



```
   <s:Body
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <ConsumerIdentityNumberResponse
xmlns="http://krd.pl/Rastin/Dto">
         <ConsumerIdentytyNumberIsValid>true</ConsumerIdentytyNumberIsValid>
         <PermanentAddressVerificationResult>
            <StreetIsValid>true</StreetIsValid>
            <BuildingIsValid>true</BuildingIsValid>
            <FlatIsValid>true</FlatIsValid>
            <ZipCodeIsValid>true</ZipCodeIsValid>
            <CityIsValid>true</CityIsValid>
         </PermanentAddressVerificationResult>
         <TemporaryAddressVerificationResult>
            <StreetIsValid>false</StreetIsValid>
            <BuildingIsValid>false</BuildingIsValid>
            <FlatIsValid>false</FlatIsValid>
            <ZipCodeIsValid>false</ZipCodeIsValid>
            <CityIsValid>false</CityIsValid>
         </TemporaryAddressVerificationResult>
      </ConsumerIdentityNumberResponse>
   </s:Body>
</s:Envelope>
```


5. (*Metoda 2 – VerifyIDCard*) Metoda weryfikacji poprawności numeru i serii dowodu osobistego

Dzięki tej metodzie Klient serwisu, po odpytaniu o poprawność przesyłanych danych, ma możliwość weryfikacji ich zgodności z udostępnionym Rejestrem Dowodów Osobistych.

Z wykorzystaniem tej metody weryfikowana jest również ważność dowodu osobistego. Jeśli weryfikowany dokument będzie nieważny, a wszystkie pozostałe dane podane w zapytaniu będą poprawne, system zwróci odpowiedź "**false**" dla całego zapytania ze względu na nieważność dowodu. W celu weryfikacji wyłącznie ważności dowodu osobistego danej osoby zalecane jest skorzystanie z dedykowanej do tego metody weryfikacji, czy dowód osobisty jest unieważniony (***Metoda 3 – VerifyIsIDCardCanceled***).

5.1. Wygląd zapytania

5.1.1. Dane autoryzacji

Autoryzacja w tego typu zapytaniu jest spójna z innymi zapytaniami i została opisana w punkcie **3**.

5.1.2. Dane zapytania

Ta sekcja zapytania służy do przekazania danych wskazanych w dowodzie osobistym, który ma zostać zweryfikowany.

VerifyIDCardRequest zawiera 9 pól:

- *ConsumerIdentityNumber* (typu `string`) – *numer PESEL* – **pole wymagane**.

  Walidacja pola: 11 cyfr + suma kontrolna
  Przykład: 69022300022

- *FirstName* (typu `string`) – *pierwsze imię* – **pole wymagane**.

  Walidacja pola: od 1 do 120 znaków
  Przykład: ZENOBIA

- *SecondName* (typu `string`) – *drugie imię* – **pole opcjonalne**.

  Walidacja pola: od 1 do 120 znaków
  Przykład: JADWIGA

- *SurnameFirstPart* (typu `string`) – *pierwsza część nazwiska* – **pole wymagane**.

  Walidacja pola: od 1 do 300 znaków
  Przykład: PRITT

- *SurnameSecondPart* (typu `string`) – *druga część nazwiska* – **pole opcjonalne**.

  Walidacja pola: od 1 do 300 znaków
  Przykład: BITT

- *IDCardSeries* (typu `string`) – *seria dowodu osobistego* – **pole wymagane**.

  Walidacja pola: od 2 do 3 znaków, z zakresu A-Z
  Przykład: VUH

- *IDCardNumber* (typu `string`) – *numer dowodu osobistego* – **pole wymagane**.

  Walidacja pola: od 6 do 7 znaków, tylko cyfry
  Przykład: 952097

- *ProductionDate* (typu `Date`) – *data wydania dowodu osobistego* – **pole wymagane**.

  Walidacja pola: z zakresu od 1799-12-31 do 2200-01-01,
  data nie może być późniejsza niż data złożenia zapytania
  (ale może być zgodna z datą zapytania)
  Przykład: 2014-09-24

- *ExpirationDate* (typu `Date`) – *data utraty ważności dowodu osobistego* – **pole wymagane w przypadku gdy dowód osobisty ma datę ważności**.

  Walidacja pola: z zakresu od 1799-12-31 do 2200-01-01,
  data musi być późniejsza niż data złożenia zapytania
  Przykład: 2024-09-24

  Data utraty ważności dowodu osobistego jest **opcjonalna**. Wynika to z faktu, że dowody osobiste wydawane przed 1 stycznia 2015 roku osobom po 65 roku życia były ważne bezterminowo. Od 1 marca 2015 roku dokumenty ważne bezterminowo nie są już wydawane. W przypadku niepodania w zapytaniu daty utraty ważności dowodu osobistego, system doda do daty wydania 100 lat i taką datę wstawi w to pole. Pozwala to obsłużyć przypadki zapytań dotyczących dowodów osobistych ważnych bezterminowo. Aby zweryfikować dowody osobiste, które mają wpisaną datę ważności, wymagane jest podanie tej daty w zapytaniu.

Uwaga!

1. Wielkość liter wprowadzanych w powyższych polach nie ma znaczenia.
2. W przypadku nazwisk składających się z dwóch członów zalecamy przekazywać je w jednym polu (*SurnameFirstPart*) we wskazanej formie: CHRAPEK-PASEK, DE LORM itd. Czasami konieczne może być kilka prób odpytania (różne kombinacje, np. z użyciem także pola *SurnameSecondPart*). Nazwisko dwuczłonowe oddzielamy zazwyczaj myślnikiem (łącznikiem-minusem), np. *KOS-WIĘCEK*.
3. SRP stosuje kodowanie UTF-8. W sprawie wykazu dopuszczalnych znaków specjalnych należy wysłać zgłoszenie na pomocit@krd.pl lub skontaktować się ze swoim Opiekunem Biznesowym po stronie KRD.

```
<dto:IDCardRequest>
   <dto:ConsumerIdentityNumber>69022300022</dto:ConsumerIdentityNumber>
   <dto:FirstName>ZENOBIA</dto:FirstName>
   <dto:SecondName>JADWIGA</dto:SecondName>
   <dto:SurnameFirstPart>PRITT</dto:SurnameFirstPart>
   <dto:SurnameSecondPart>BITT</dto:SurnameSecondPart>
   <dto:IDCardSeries>VUH</dto:IDCardSeries>
   <dto:IDCardNumber>952097</dto:IDCardNumber>
   <dto:ProductionDate>2014-09-24</dto:ProductionDate>
   <dto:ExpirationDate>2024-09-24</dto:ExpirationDate>
</dto:IDCardRequest>
```


5.1.3. Dane odpowiedzi

W odpowiedzi na zapytanie dostajemy element typu *VerifyIDCardResponse*, zawierający 1 pole:

- *IsValid* (typu `bool`) – wartość logiczna ("**true**" lub "**false**") określająca czy wszystkie dane z dowodu osobistego są poprawne.

Uwaga!

Wartość "**true**" w polu *IsValid* świadczy także o tym, że dowód osobisty nie jest unieważniony, czyli jest ważny.

5.1.4. Przykład zapytania

```
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:aut="http://krd.pl/Authorization"
xmlns:dto="http://krd.pl/Rastin/Dto">
   <soap:Header>
      <aut:Authorization>
         <aut:AuthorizationType>Certificate</aut:AuthorizationType>
      </aut:Authorization>
   </soap:Header>
   <soap:Body>
      <dto:IDCardRequest>
         <dto:ConsumerIdentityNumber>69022300022</dto:ConsumerIdentityNumber>
         <dto:FirstName>ZENOBIA</dto:FirstName>
         <dto:SecondName>JADWIGA</dto:SecondName>
         <dto:SurnameFirstPart>PRITT</dto:SurnameFirstPart>
         <dto:SurnameSecondPart>BITT</dto:SurnameSecondPart>
         <dto:IDCardSeries>VUH</dto:IDCardSeries>
         <dto:IDCardNumber>952097</dto:IDCardNumber>
         <dto:ProductionDate>2014-09-24</dto:ProductionDate>
         <dto:ExpirationDate>2024-09-24</dto:ExpirationDate>
      </dto:IDCardRequest>
   </soap:Body>
</soap:Envelope>
```

5.1.5. Przykład odpowiedzi

```
<s:Envelope
xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
   <s:Header>
      <a:Action
s:mustUnderstand="1">
http://krd.pl/Rastin/IVerificationService/VerifyIDCardResponse
      </a:Action>
      <h:Authorization
xmlns:h="http://krd.pl/Authorization/Response"
xmlns="http://krd.pl/Authorization/Response"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
         <Ticket>AAA1B1AA11AA111B1111AA111B1AA1111B1AA111</Ticket>
         <TicketExpirationDate>2022-02-17T13:28:54.68</TicketExpirationDate>
         <ShouldChangePassword>false</ShouldChangePassword>
         <PasswordExpirationDate>2021-05-23T00:00:00+02:00</PasswordExpirationDate>
      </h:Authorization>
   </s:Header>
```



```
   <s:Body
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <IDCardResponse
xmlns="http://krd.pl/Rastin/Dto">
         <IsValid>false</IsValid>
      </IDCardResponse>
   </s:Body>
</s:Envelope>
```


6. (*Metoda 3* – *VerifyIsIDCardCanceled*) Metoda weryfikacji, czy dowód osobisty jest unieważniony

Dzięki tej metodzie Klient serwisu, po odpytaniu o poprawność przesyłanych danych, ma możliwość weryfikacji ich zgodności z udostępnionym **wykazem unieważnionych dowodów osobistych**, którym, zgodnie z art. 53 ustawy z dnia 6 sierpnia 2010 r. o dowodach osobistych (t.j. Dz. U. z 2022 r. poz. 671 ze zm.) jest wykaz zawierający serie i numery unieważnionych dokumentów oraz serie i numery błędnie spersonalizowanych lub utraconych blankietów dowodów osobistych.

6.1. Wygląd zapytania

6.1.1. Dane autoryzacji

Autoryzacja w tego typu zapytaniu jest spójna z innymi zapytaniami i została opisana w punkcie **3**.

6.1.2. Dane zapytania

Ta sekcja zapytania służy do przekazania danych identyfikujących dokument, którego ważność ma zostać zweryfikowana.

VerifyIsIDCardCanceledRequest zawiera 2 pola:

- *IDCardSeries* (typu `string`) – *seria dowodu osobistego* – **pole wymagane**.

  Walidacja pola: od 2 do 3 znaków, z zakresu A-Z
  Przykład: VRX

- *IDCardNumber* (typu `string`) – *numer dowodu osobistego* – **pole wymagane**.

  Walidacja pola: od 6 do 7 znaków, tylko cyfry
  Przykład: 938012

Uwaga!

Wielkość liter serii dowodu osobistego w polu *IDCardSeries* nie ma znaczenia.

```
<dto:IDCardCanceledRequest>
   <dto:IDCardSeries>VRX</dto:IDCardSeries>
   <dto:IDCardNumber>938012</dto:IDCardNumber>
</dto:IDCardCanceledRequest>
```

6.1.3. Dane odpowiedzi

W odpowiedzi na zapytanie dostajemy element typu *VerifyIsIDCardCanceledResponse*, zawierający jedno pole:

- *Canceled* (typu `bool`) – wartość logiczna ("**true**" lub "**false**") określająca czy dowód osobisty jest unieważniony.

Uwaga!

Wartość "**true**" w polu *Canceled* = dowód osobisty jest unieważniony. Wartość "**false**" w polu *Canceled* = dowód osobisty nie jest unieważniony, pod warunkiem że podane seria i numer są zgodne z dowodem osobistym.

Metoda 3 **–** ***VerifyIsIDCardCanceled*** zwraca wyłącznie informację o tym, czy dany dowód osobisty widnieje w wykazie unieważnionych dowodów osobistych.

- Jeśli w ***metodzie 3*** podamy błędną serię lub numer dowodu osobistego (albo obie wartości błędne), otrzymamy "**false**".
- Jeśli dowód osobisty nie został jeszcze wydany i nie został sklasyfikowany jako unieważniony, otrzymamy "**false**".
- Jeśli dowód osobisty został sklasyfikowany jako unieważniony, otrzymamy "**true**".

W celu weryfikacji zarówno poprawności danych wskazanych w dowodzie osobistym, tj. zgodności tych danych z Rejestrem Dowodów Osobistych, jak również jednoznacznego potwierdzenia, że dokument jest ważny, zalecane jest łączenie obydwu metod weryfikacji, tj. metody weryfikacji poprawności numeru i serii dowodu osobistego (***Metoda 2 – VerifyIDCard***) oraz metody weryfikacji, czy dowód osobisty został unieważniony (***Metoda 3 –VerifyIsIDCardCanceled***).

Dowód osobisty staje się dokumentem ważnym w momencie odebrania go przez właściciela z urzędu. Zgodnie z art. 50 ustawy z dnia 6 sierpnia 2010 r. o dowodach osobistych, w okresie ważności dowód osobisty może zostać unieważniony (wartość "**true**" w polu *Canceled*) m.in. w następujących przypadkach:

- zmiany danych zawartych w dowodzie osobistym,
- zmiany wizerunku twarzy posiadacza dowodu osobistego w stosunku do wizerunku twarzy zamieszczonego w dowodzie osobistym w stopniu utrudniającym lub uniemożliwiającym identyfikację jego posiadacza,
- utraty lub uszkodzenia dowodu osobistego w stopniu utrudniającym lub uniemożliwiającym identyfikację jego posiadacza,
- przekazania znalezionego dowodu osobistego przez osobę trzecią do organu gminy lub do konsula Rzeczypospolitej Polskiej,
- zgonu właściciela – w przypadku zgonu właściciela dowód osobisty jest unieważniany automatycznie po sporządzeniu przez właściwy Urząd Stanu Cywilnego aktu zgonu,
- żądania wymiany dowodu osobistego przez właściciela,
- utraty obywatelstwa polskiego przez właściciela,
- uprawdopodobnienia przez posiadacza dowodu osobistego podejrzenia nieuprawnionego wykorzystania jego danych osobowych, w tym serii i numeru dowodu osobistego – właściciel dowodu osobistego, w przypadku podejrzenia nieuprawnionego wykorzystania jego danych osobowych, może zgłosić osobiście ten fakt organowi dowolnej gminy w celu unieważnienia posiadanego dowodu osobistego.

6.1.4. Przykład zapytania

```
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:aut="http://krd.pl/Authorization"
xmlns:dto="http://krd.pl/Rastin/Dto">
   <soap:Header>
      <aut:Authorization>
         <aut:AuthorizationType>LoginAndPassword</aut:AuthorizationType>
         <aut:Login>login</aut:Login>
         <aut:Password>hasło</aut:Password>
      </aut:Authorization>
   </soap:Header>
   <soap:Body>
      <dto:IDCardCanceledRequest>
         <dto:IDCardSeries>VRX</dto:IDCardSeries>
         <dto:IDCardNumber>938012</dto:IDCardNumber>
      </dto:IDCardCanceledRequest>
   </soap:Body>
</soap:Envelope>
```

6.1.5. Przykład odpowiedzi

```
<s:Envelope
xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
   <s:Header>
      <a:Action
s:mustUnderstand="1">
http://krd.pl/Rastin/IVerificationService/VerifyIsIDCardCanceledResponse
      </a:Action>
      <h:Authorization
xmlns:h="http://krd.pl/Authorization/Response"
xmlns="http://krd.pl/Authorization/Response"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
         <Ticket>AAA1B1AA11AA111B1111AA111B1AA1111B1AA111</Ticket>
         <TicketExpirationDate>2022-02-17T14:24:19.23</TicketExpirationDate>
         <ShouldChangePassword>false</ShouldChangePassword>
         <PasswordExpirationDate>2021-05-23T00:00:00+02:00</PasswordExpirationDate>
      </h:Authorization>
   </s:Header>
   <s:Body
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <IDCardCanceledResponse
xmlns="http://krd.pl/Rastin/Dto">
         <Canceled>true</Canceled>
      </IDCardCanceledResponse>
   </s:Body>
</s:Envelope>
```


7. (*Metoda 4 – VerifyConsumerIsAlive*) Metoda weryfikacji w rejestrze PESEL, czy osoba żyje

Dzięki tej metodzie Klient serwisu ma możliwość weryfikacji, czy osoba, której dane przesyłane w zapytaniu, obejmujące: numer PESEL, imię, nazwisko, żyje zgodnie z danymi w udostępnionym rejestrze PESEL.

7.1. Wygląd zapytania

7.1.1. Dane autoryzacji

Autoryzacja w tego typu zapytaniu jest spójna z innymi zapytaniami i została opisana w punkcie **3**.

7.1.2. Dane zapytania

Ta sekcja zapytania służy do przekazania danych identyfikujących podmiot, który ma zostać zweryfikowany.

VerifyConsumerIsAliveRequest zawiera 3 pola:

- *ConsumerIdentityNumber* (typu `string`) – *numer PESEL* osoby do weryfikacji – **pole wymagane**.

  Walidacja pola: 11 cyfr
  Przykład: 68061200023

- *FirstName* (typu `string`) – *imię* osoby do weryfikacji – **pole wymagane**.

  Walidacja pola: od 1 do 120 znaków
  Przykład: TERESA

- *Surname* (typu `string`) – *nazwisko* osoby do weryfikacji – **pole wymagane**.

  Walidacja pola: od 1 do 300 znaków
  Przykład: KOS-WIĘCEK

Uwaga!

1. Wielkość liter wprowadzanych w powyższych polach nie ma znaczenia.
2. Nazwisko dwuczłonowe umieszczane w polu *Surname* oddzielamy zazwyczaj myślnikiem (łącznikiem-minusem), np. *KOS-WIĘCEK*.
3. SRP stosuje kodowanie UTF-8. W sprawie wykazu dopuszczalnych znaków specjalnych należy wysłać zgłoszenie na pomocit@krd.pl lub skontaktować się ze swoim Opiekunem Biznesowym po stronie KRD.

```
<dto:VerifyConsumerIsAliveRequest>
   <dto:ConsumerIdentityNumber>68061200023</dto:ConsumerIdentityNumber>
   <dto:FirstName>TERESA</dto:FirstName>
   <dto:Surname>KOS-WIĘCEK</dto:Surname>
</dto:VerifyConsumerIsAliveRequest>
```


7.1.3. Dane odpowiedzi

W odpowiedzi na zapytanie dostajemy element typu *VerifyConsumerIsAliveResponse*, zawierający 1 pole:

- *Status* (typu `string`) zwracający wynik weryfikacji, czy osoba żyje:
  - **Alive** – osoba żyje,
  - **NotAlive** – osoba nie żyje,
  - **IncorrectData** – dane przekazane w zapytaniu były niepoprawne, tzn. nie istnieje w rejestrze PESEL osoba o wskazanym numerze PESEL lub numer PESEL nie zgadza się z przekazanym w zapytaniu imieniem/nazwiskiem.

7.1.4. Przykład zapytania

```
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:aut="http://krd.pl/Authorization"
xmlns:dto="http://krd.pl/Rastin/Dto">
   <soap:Header>
      <aut:Authorization>
         <aut:AuthorizationType>LoginAndPassword</aut:AuthorizationType>
         <aut:Login>login</aut:Login>
         <aut:Password>hasło</aut:Password>
      </aut:Authorization>
   </soap:Header>
   <soap:Body>
      <dto:VerifyConsumerIsAliveRequest>
         <dto:ConsumerIdentityNumber>68061200023</dto:ConsumerIdentityNumber>
         <dto:FirstName>TERESA</dto:FirstName>
         <dto:Surname>KOS-WIĘCEK</dto:Surname>
      </dto:VerifyConsumerIsAliveRequest>
   </soap:Body>
</soap:Envelope>
```

7.1.5. Przykład odpowiedzi

```
<s:Envelope
xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
   <s:Header>
      <a:Action
s:mustUnderstand="1">
http://krd.pl/Rastin/IVerificationService/VerifyConsumerIsAliveResponse
      </a:Action>
      <h:Authorization
      xmlns:h="http://krd.pl/Authorization/Response"
      xmlns="http://krd.pl/Authorization/Response"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
         <Ticket>00AF70020063538E12A4409220FC180F5A7F52000</Ticket>
         <TicketExpirationDate>2022-12-20T15:21:52+01:00</TicketExpirationDate>
         <ShouldChangePassword>false</ShouldChangePassword>
         <PasswordExpirationDate>2022-12-22T00:00:00+01:00</PasswordExpirationDate>
      </h:Authorization>
   </s:Header>
```



```
    <s:Body
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
       <ConsumerIsAliveResponse
       xmlns="http://krd.pl/Rastin/Dto">
          <Status>Alive</Status>
       </ConsumerIsAliveResponse>
    </s:Body>
 </s:Envelope>
```


8. Plik WSDL

8.1. Definicje typów

8.1.1. Typy PermanentAddress i TemporaryAddress

Elementy tych typów są używane do zwrócenia informacji na temat *stałego* i *tymczasowego adresu* osoby, którą weryfikujemy w SRP. Oba elementy dziedziczą po typie `AddressBase`, który składa się z 5 pól:

- *Street* (typu `string`) – *ulica*,
- *Building* (typu `string`) – *numer domu*,
- *Flat* (typu `string`) – *numer mieszkania*,
- *ZipCode* (typu `string`) – *kod pocztowy*
  (format: ddddd, bez myślnika, np. 00000),
- *City* (typu `string`) – *miasto*.

```
<dto:PermanentAddress>
   <dto:Street>NIEPODLEGŁOŚCI</dto:Street>
   <dto:Building>2</dto:Building>
   <dto:Flat>11</dto:Flat>
   <dto:ZipCode>83000</dto:ZipCode>
   <dto:City>Pruszcz Gdański</dto:City>
</dto:PermanentAddress>
 <dto:TemporaryAddress>
   <dto:Street>KIRKORA</dto:Street>
   <dto:Building>20</dto:Building>
   <dto:Flat>1</dto:Flat>
   <dto:ZipCode>83000</dto:ZipCode>
   <dto:City>Pruszcz Gdański</dto:City>
</dto:TemporaryAddress>
```


9. Błędy zwracane w odpowiedzi

W przypadku wystąpienia błędu Klientowi zwracana jest odpowiedź, która zawiera *SOAP Fault Element* (*basicHttpBinding* / *wsHttpBinding*):

- element *faultcode* (basic)/*s:Subcode* (ws) zawiera nazwę błędu,
- element *faultstring* (basic)/*s:Reason* (ws) jest uzupełniony o wiadomość błędu,
- element *detail* (basic)/*s:Detail* (ws) zawiera różne typy błędów opisane poniżej.

Każda z metod dostępnych w serwisie RASTIN może zwrócić w elemencie *detail* (basic)/*s:Detail* (ws) jeden z 4 typów błędów:

- *DefaultFault*,
- *SecurityFault*,
- *ValidationFault*,
- *SchemaValidationFault*.

Przykładowa odpowiedź z błędem walidacji zwrócona przez serwis RASTIN (basic):

```
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   <s:Body>
      <s:Fault>
         <faultcode>s:IncorrectCredentials</faultcode>
    <faultstring xml:lang="pl-PL">Podane dane logowania są nieprawidłowe.</faultstring>
         <detail>
            <SecurityFault id="f11c0537-44da-4590-9207-03088ee3fab8" code="1"
xmlns="http://krd.pl/Faults"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
         </detail>
      </s:Fault>
   </s:Body>
</s:Envelope>
```

Przykładowa odpowiedź z błędem walidacji zwrócona przez serwis RASTIN (ws):

```
<s:Envelope
xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
   <s:Header>
      <a:Action
s:mustUnderstand="1">
http://krd.pl/Rastin/IVerificationService/VerifyConsumerIdentityNumber
      </a:Action>
   </s:Header>
   <s:Body>
      <s:Fault>
         <s:Code>
            <s:Value>s:Sender</s:Value>
            <s:Subcode>
               <s:Value>s:IncorrectCredentials</s:Value>
            </s:Subcode>
         </s:Code>
```



```
         <s:Reason>
            <s:Text xml:lang="pl-PL">Podane dane logowania są nieprawidłowe.</s:Text>
            <s:Text xml:lang="en-US">Specified credentials are invalid</s:Text>
         </s:Reason>
         <s:Detail>
            <SecurityFault id="bf3f81e3-1349-4b68-9c33-be8abb5897a1" code="1"
xmlns="http://krd.pl/Faults"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
         </s:Detail>
      </s:Fault>
   </s:Body>
</s:Envelope>
```

9.1. DefaultFault

DefaultFault jest typem, który jest wykorzystywany przez typ *SecurityFault*, *ValidationFault* i *SchemaValidationFault*, ale może też wystąpić samodzielnie, gdy np. dojdzie do usunięcia składowej SOAP (request), zmiany nazwy jakiegoś pola.

Posiada 2 atrybuty:

- *id* – zmienny identyfikator błędu (typu `guid`).
  Jest potrzebny do identyfikacji błędu po stronie systemu KRD,
- *code* – kod błędu (typu `int`).

```
<xs:complexType name="DefaultFault">
 <xs:attribute name="id" type="q1:guid" use="required"/>
 <xs:attribute name="code" type="xs:int" use="required"/>
</xs:complexType>
```

Przykład:

```
            <DefaultFault id="a4ba0698-96e1-4289-8bfb-891d007b82a6" code="0"
xmlns="http://krd.pl/Faults"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
```

9.2. SecurityFault

Typ zawiera w sobie typ *DefaultFault*. Reprezentuje błędy związane z brakiem uprawnień lub błędnymi danymi logowania. Zgłaszany jest w przypadku naruszenia zasad bezpieczeństwa.

```
<xs:complexType name="SecurityFault">
 <xs:complexContent mixed="false">
  <xs:extension base="tns:DefaultFault"/>
 </xs:complexContent>
</xs:complexType>
```

Przykład:

```
            <SecurityFault id="f11c0537-44da-4590-9207-03088ee3fab8" code="1"
xmlns="http://krd.pl/Faults"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
```


9.3. ValidationFault

Typ zawiera w sobie typ *DefaultFault*. Reprezentuje błędy związane z walidacją danych wprowadzonych do zapytania. Ten typ błędu pojawia się, gdy przekazane dane są niezgodne ze schematem WSDL lub nie spełniają reguł biznesowych (np. podano niepoprawny numer PESEL).

W typie *ValidationFault* znajduje się element *ValidationFaultDetail*, który zawiera w sobie szczegóły błędu walidacji.

```
<xs:complexType name="ValidationFault">
 <xs:complexContent mixed="false">
  <xs:extension base="tns:DefaultFault">
   <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Details"
type="tns:ArrayOfValidationFaultDetail"/>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

9.3.1. ValidationFaultDetail

Typ *ValidationFaultDetail* zawiera w sobie element:

- *Key* – klucz dla błędu walidacji (nazwa pola),
- *Message* – wiadomość dla błędu walidacji.

```
<xs:complexType name="ValidationFaultDetail">
 <xs:sequence>
  <xs:element minOccurs="0" maxOccurs="1" name="Key" type="xs:string"/>
  <xs:element minOccurs="0" maxOccurs="1" name="Message"
type="tns:ArrayOfValidationDetailMessageText"/>
 </xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="ValidationDetailMessageText">
 <xs:simpleContent>
  <xs:extension base="xs:string">
   <xs:attribute ref="xml:lang" use="optional"/>
  </xs:extension>
 </xs:simpleContent>
</xs:complexType>
```

Przykład:

```
            <ValidationFault id="2f031f4e-4775-44fe-9867-d4b2922fcbeb" code="13"
xmlns="http://krd.pl/Faults"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
               <Details>
                  <Detail>
                     <Key>ConsumerIdentityNumber</Key>
                     <Message>
                        <Text xml:lang="pl-PL">Podana wartość ma nieprawidłową długość.
Musi się składać z 11 znaków.</Text>
```



```
                        <Text xml:lang="en-US">The given value has an invalid length.
Must consist of 11 characters.</Text>
                    </Message>
                </Detail>
            </Details>
        </ValidationFault>
```

9.4. SchemaValidationFault

Typ zawiera w sobie typ *DefaultFault*. Ten typ błędu pojawia się, gdy wysłana wiadomość SOAP nie jest zgodna ze schematem WSDL (np. jako datę wydania lub ważności dowodu osobistego podano: 2023-01-00).

W typie *SchemaValidationFault* znajduje się element *SchemaValidationFaultDetail*, który zawiera w sobie szczegóły błędu walidacji.

```
<xs:complexType name="SchemaValidationFault">
 <xs:complexContent mixed="false">
  <xs:extension base="tns:DefaultFault">
   <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Details"
type="tns:ArrayOfSchemaValidationFaultDetail"/>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

9.4.1. SchemaValidationFaultDetail

Typ *SchemaValidationFaultDetail* zawiera w sobie 2 atrybuty:

- *line* – numer linii, w której wystąpił błąd,
- *column* – numer kolumny, w której wystąpił błąd.

```
<xs:complexType name="SchemaValidationFaultDetail">
 <xs:sequence>
  <xs:element minOccurs="0" maxOccurs="1" name="Message"
type="tns:ArrayOfValidationDetailMessageText"/>
 </xs:sequence>
 <xs:attribute name="line" type="xs:int" use="required"/>
 <xs:attribute name="column" type="xs:int" use="required"/>
</xs:complexType>
```

Oraz element:

- *Message* – wiadomość dla błędu walidacji.

```
<xs:complexType name="ValidationDetailMessageText">
 <xs:simpleContent>
  <xs:extension base="xs:string">
   <xs:attribute ref="xml:lang" use="optional"/>
  </xs:extension>
 </xs:simpleContent>
</xs:complexType>
```


Przykład:

```
            <SchemaValidationFault id="6cbc79b1-a127-439a-ac5f-abcdd3ce8ebb" code="14"
xmlns="http://krd.pl/Faults"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
                <Details>
                   <Detail line="19" column="42">
                      <Message>
                         <Text xml:lang="pl-PL">The
'http://krd.pl/Rastin/Dto:ProductionDate' element is invalid
- The value '2023-01-00' is invalid according to its datatype
'http://www.w3.org/2001/XMLSchema:date' - The string '2023-01-00'
is not a valid Date value.</Text>
                         <Text xml:lang="en-US">The
'http://krd.pl/Rastin/Dto:ProductionDate' element is invalid
- The value '2023-01-00' is invalid according to its datatype
'http://www.w3.org/2001/XMLSchema:date' - The string '2023-01-00'
is not a valid Date value.</Text>
                      </Message>
                   </Detail>
                </Details>
             </SchemaValidationFault>
```


10. Szczegółowy opis błędów + wykaz błędów

10.1. Atrybut code

Dla atrybutu code mogą pojawić się następujące wartości i odpowiadające im rodzaje błędów:

- **0** – „**Wystąpił wewnętrzny błąd systemu.**”

W przypadku wystąpienia takiego błędu należy sprawdzić poprawność konstrukcji zapytania (request) [błąd wywołuje np.: omyłkowo usunięta składowa SOAP, zmiana nazwy jakiegoś pola, podanie w typie autoryzacji (*AuthorizationType*) innej wartości niż ta określona w specyfikacji/WSDL, przerwa techniczna/awaria po stronie KRD lub po stronie Ministerstwa Cyfryzacji.

- **1** – „**Podane dane logowania są nieprawidłowe.**”

W przypadku wystąpienia takiego błędu należy sprawdzić poprawność loginu (*Login*) i hasła (*Password*).

- **2** – „**Brak wystarczających uprawnień do wykonania danej operacji.**”

W przypadku wystąpienia takiego błędu należy wysłać wiadomość na pomocit@krd.pl lub do swojego Opiekuna Biznesowego po stronie KRD.

- **3** – „**Miesięczny limit zapytań został osiągnięty!**”

Pojawia się, gdy zostanie przekroczony miesięczny limit odpytań baz RDO/PESEL (standardowy limit miesięczny = 100 000, możemy go zwiększać).

- **4** – „**Dzienny limit zapytań został osiągnięty!**”

Pojawia się, gdy zostanie przekroczony dzienny limit odpytań baz RDO/PESEL (standardowy limit dzienny = 10 000, możemy go zwiększać).

- **13** – „**Podane dane są nieprawidłowe. Sprawdź szczegóły, aby uzyskać więcej informacji.**”

W przypadku wystąpienia takiego błędu należy sprawdzić szczegóły w elemencie *Key* oraz *Message* i zweryfikować zapytanie (request) pod kątem poprawności wypełnionych pól [błąd wywołuje np.: podanie zbyt krótkiego numeru PESEL (*ConsumerIdentityNumber*), zbyt krótkiej serii dowodu osobistego (*IDCardSeries*), zbyt krótkiego numer dowodu osobistego (*IDCardNumber*); data wydania dowodu osobistego (*ProductionDate*) późniejsza niż data wysłania zapytania; zbyt krótki kod pocztowy (*ZipCode*); usunięcie wymaganego pola z zapytania (request)/ pozostawienie go pustego].

- **14** – „**Podane dane wiadomości SOAP są nieprawidłowe. Sprawdź szczegóły, aby uzyskać więcej informacji.**”

W przypadku wystąpienia takiego błędu należy sprawdzić szczegóły w atrybucie: *line*, *column* oraz w elemencie *Message* i zweryfikować zapytanie (request) pod kątem poprawności wypełnionych pól [błąd wywołuje np.: data wydania (*ProductionDate*) lub ważności (*ExpirationDate*) dowodu osobistego zapisana np. w ten sposób: 2023-01-00; umieszczenie w dacie wydania (*ProductionDate*) lub ważności (*ExpirationDate*) dowodu osobistego litery/znaku specjalnego, pozostawienie tych wartości pustych; usunięcie z zapytania (request) pola dotyczącego daty wydania (*ProductionDate*) dowodu osobistego; zmiana kolejności pól w zapytaniu (request)].

Pytania w sprawie błędów należy kierować na pomocit@krd.pl lub do swojego Opiekuna Biznesowego po stronie KRD.

Wykaz błędów									
Nr metody	*faultcode / s:Subcode*	*faultstring / s:Reason*	*detail / s:Detail*	*id (ErrorId)*	*code (ErrorSubCode)*	*line*	*column*	*Key*	*Message pl / pl-PL*
Nr metody	Nazwa błędu	Wiadomość błędu	Typ błędu	Id błędu (typu guid)	Kod błędu (typu int)	Przykład numeru linii, w której wystąpił błąd	Przykład numeru kolumny, w której wystąpił błąd	Przykład klucza dla błędu walidacji (nazwa pola)	Przykład wiadomości dla błędu walidacji
1, 2, 3, 4	s:InternalError	Wystąpił wewnętrzny błąd systemu.	DefaultFault	Zmienny	0	Nie dotyczy	Nie dotyczy	Nie dotyczy	Nie dotyczy
1, 2, 3, 4	s:IncorrectCredentials	Podane dane logowania są nieprawidłowe.	SecurityFault	Zmienny	1	Nie dotyczy	Nie dotyczy	Nie dotyczy	Nie dotyczy
1, 2, 3, 4	s:InsufficientRights	Brak wystarczających uprawnień do wykonania danej operacji.	SecurityFault	Zmienny	2	Nie dotyczy	Nie dotyczy	Nie dotyczy	Nie dotyczy
1, 2, 3, 4	s:LimitReached	Miesięczny limit zapytań został osiągnięty!	SecurityFault	Zmienny	3	Nie dotyczy	Nie dotyczy	Nie dotyczy	Nie dotyczy
1, 2, 3, 4	s:LimitReached	Dzienny limit zapytań został osiągnięty!	SecurityFault	Zmienny	4	Nie dotyczy	Nie dotyczy	Nie dotyczy	Nie dotyczy
1, 2, 3, 4	s:DataIsNotValid	Podane dane są nieprawidłowe. Sprawdź szczegóły, aby uzyskać więcej informacji.	ValidationFault	Zmienny	13	Nie dotyczy	Nie dotyczy	Szczegóły tutaj	Szczegóły tutaj
1, 2, 3, 4	s:MessageIsNotValid	Podane dane wiadomości SOAP są nieprawidłowe. Sprawdź szczegóły, aby uzyskać więcej informacji.	SchemaValidationFault	Zmienny	14	34	33	Nie dotyczy	The 'http://krd.pl/Rastin/Dto:ProductionDate' element is invalid - The value '2023-01-00' is invalid according to its datatype 'http://www.w3.org/2001/XMLSchema:date' – The string '2023-01-00' is not a valid Date value.

Wykaz błędów (code 13)	
Key	*Message pl / pl-PL*
Login	Login jest wymagany
Password	Hasło jest wymagane
Authorization	Element lub wartość są wymagane.
ConsumerIdentityNumber	Element lub wartość są wymagane.
ConsumerIdentityNumber	Podana wartość ma nieprawidłową długość. Musi się składać z 11 znaków.
ConsumerIdentityNumber	Podana wartość ma nieprawidłowy format. Dopuszczalne są tylko cyfry.
FirstName	Element lub wartość są wymagane.
FirstName	Podana wartość ma nieprawidłową długość. Maksymalna ilość znaków to 120.
SecondName	Podana wartość ma nieprawidłową długość. Maksymalna ilość znaków to 120.
Surname	Element lub wartość są wymagane.
Surname	Podana wartość ma nieprawidłową długość. Maksymalna ilość znaków to 300.
Surname (FirstPart)	Element lub wartość są wymagane.
Surname (FirstPart)	Podana wartość ma nieprawidłową długość. Maksymalna ilość znaków to 300.
SurnameSecondPart	Podana wartość ma nieprawidłową długość. Maksymalna ilość znaków to 300.
IDCardSeries	Element lub wartość są wymagane.
IDCardSeries	Podana wartość ma nieprawidłową długość. Musi się składać z 2 do 3 znaków.
IDCardSeries	Podana wartość ma nieprawidłowy format. Musi się składać z 2 do 3 liter.
IDCardNumber	Element lub wartość są wymagane.
IDCardNumber	Podana wartość ma nieprawidłową długość. Musi się składać z 6 do 7 znaków.
IDCardNumber	Podana wartość ma nieprawidłowy format. Dopuszczalne są tylko cyfry.
ProductionDate	Element lub wartość są wymagane.
ProductionDate	Data musi zawierać się w przedziale od 1799-12-31 do 2200-01-01.
ProductionDate	Data musi być z przeszłości.
ExpirationDate	Data musi zawierać się w przedziale od 1799-12-31 do 2200-01-01.
ExpirationDate	Data musi być z przyszłości.
ZipCode	Element lub wartość są wymagane.
ZipCode	Podana wartość ma nieprawidłową długość. Musi się składać z 5 znaków.
ZipCode	Podana wartość ma nieprawidłowy format. Dopuszczalne są tylko cyfry.

KONIEC DOKUMENTU

