

# SIDDIN 2.1

# WYMIANA DANYCH

## SPECYFIKACJA TECHNICZNA WEB SERWISU SIDDIN 2.1

Wersja dokumentacji 1.2 z dnia 2022-04-28

Status dokumentu

Opracowanie wewnętrzne

Słowa kluczowe

BIG, Transza, WebService, SIDDIN, NICCI

*Copyright © Krajowy Rejestr Długów, 2006-2022*

Wszelkie korekty, komentarze oraz uwagi proszę przysyłać na adres [pomocit@krd.pl](mailto:pomocit@krd.pl)



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

Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

## Atrybuty dokumentu

	Atrybut A	Wartość B
1	Projekt	Siddin 2.1
2	Tytuł	Wymiana danych
3	Podtytuł	Specyfikacja techniczna web serwisu Siddin 2.1
4	Wersja dokumentacji	1.2
5	Czas wersji	2022-04-28
6	Plik	Siddin 2.1 - Specyfikacja techniczna.docx/pdf
7	Autorzy	Rafał Stramski
8	Nadzór	Sebastian Tkocz
9	Prawa autorskie	Copyright © Krajowy Rejestr Długów, 2006-2022
10	Komentarz	

## Historia dokumentu

	Atrybut A	Wartość B	Data C
1	Wersja dokumentacji	1.0	2006-10-13
2	Autor	Rafał Stramski	
3	Sprawdził treść		
4	Sprawdził formę		
5	Zatwierdził		
6	Opis		
1	Wersja dokumentacji	1.1	2020-06-08
2	Autor	Maciej Łukasik	
3	Sprawdził treść	Maciej Łukasik	2020-06-08
4	Sprawdził formę	Maciej Łukasik	2020-06-08
5	Zatwierdził	Maciej Łukasik	2020-06-08
6	Opis	1. Aktualizacja/odświeżenie opisu treści technicznych.	
1	Wersja dokumentacji	1.2	2022-04-28
2	Autor	Maciej Łukasik	
3	Sprawdził treść	Maciej Łukasik	2022-04-28
4	Sprawdził formę	Maciej Łukasik	2022-04-28
5	Zatwierdził	Maciej Łukasik	2022-04-28
6	Opis	1. Aktualizacja/odświeżenie opisu treści technicznych. 2. Stworzenie nowego automatycznego spisu treści. 3. Modyfikacja/uzupełnienie/dodanie opisu zawartości adresu WSDL. 4. Dodanie informacji o obsłudze protokołu Nicci 3.2 (Wstęp, Typ NicciVersionEnum).	

Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

# Spis treści

<b>SPIS TREŚCI .....</b>	<b>3</b>
<b>WSTĘP .....</b>	<b>5</b>
<b>1. METODY SERWISU SIDDIN .....</b>	<b>6</b>
1.1. DANE OGÓLNE .....	6
1.2. UWIERZYTELNIANIE KLIENTA.....	6
1.3. ŁADOWANIE DANYCH .....	6
1.4. PRZETWORZENIE TRANSZY ZLECEŃ.....	6
1.5. POBRANIE ODPOWIEDZI Z SERWISU.....	8
1.6. ZABEZPIECZENIE DANYCH .....	8
1.7. OBSŁUGA BŁĘDÓW.....	8
<b>2. PLIK WSDL.....</b>	<b>9</b>
2.1. DEFINICJE TYPÓW .....	9
2.1.1. <i>Typ SignedRequest</i> .....	9
2.1.2. <i>Typ LoginRequest</i> .....	9
2.1.3. <i>Typ LogoutRequest</i> .....	9
2.1.4. <i>Typ ChunkBag</i> .....	9
2.1.5. <i>Typ NicciVersionEnum</i> .....	10
2.1.6. <i>Typ Job</i> .....	10
2.1.7. <i>Typ JobStatusEnum</i> .....	10
2.1.8. <i>Typ UploadChunkRequest</i> .....	11
2.1.9. <i>Typ CloseChunkRequest</i> .....	11
2.1.10. <i>Typ GetChunkBagRequest</i> .....	12
2.1.11. <i>Typ DownloadChunkRequest</i> .....	12
2.1.12. <i>Typ GetJobsRequest</i> .....	12
2.1.13. <i>Typ ArrayOfJob</i> .....	13
2.1.14. <i>Typ CancelJobRequest</i> .....	13
2.1.15. <i>Typ RemoveChunkBagRequest</i> .....	13
2.1.16. <i>Typ guid</i> .....	13
2.2. DEFINICJE WIADOMOŚCI.....	14
2.2.1. <i>Wiadomości LoginSoapIn i LoginSoapOut</i> .....	15
2.2.2. <i>Wiadomości LogoutSoapIn i LogoutSoapOut</i> .....	15
2.2.3. <i>Wiadomości UploadChunkSoapIn i UploadChunkSoapOut</i> .....	15
2.2.4. <i>Wiadomości CloseChunkBagSoapIn i CloseChunkBagSoapOut</i> .....	15
2.2.5. <i>Wiadomości GetChunkBagSoapIn i GetChunkBagSoapOut</i> .....	15
2.2.6. <i>Wiadomości DownloadChunkSoapIn i DownloadChunkSoapOut</i> .....	16
2.2.7. <i>Wiadomości GetJobsSoapIn i GetJobsSoapOut</i> .....	16
2.2.8. <i>Wiadomości CancelJobSoapIn i CancelJobSoapOut</i> .....	16
2.2.9. <i>Wiadomości RemoveChunkBagSoapIn i RemoveChunkBagSoapOut</i> .....	17

Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

2.3.	ELEMENT PORTTYPE .....	17
2.4.	ELEMENT BINDING .....	18
2.5.	DEFINICJA SERWISU.....	20

Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

## Wstęp

System KRD realizuje usługi Biura Informacji Gospodarczej (BIG) w myśl ustawy o udostępnianiu informacji gospodarczych i wymianie danych gospodarczych z dnia 9 kwietnia 2010 roku. Umożliwia on dodawanie, aktualizowanie i usuwanie informacji gospodarczych oraz udostępnianie tych informacji osobom trzecim.

Informacje gospodarcze mogą być wprowadzane do systemu KRD na jeden z dwóch sposobów: interakcji na witrynie WWW biura – Panel Klienta KRD (manualnie lub poprzez pliki CSV) bądź przez serwisy internetowe wykorzystujące protokół SOAP. Jednym z takich serwisów internetowych (ang. Webservice) jest serwis o nazwie SIDDIN, wersja 2.1.

Webservice SIDDIN 2.1 jest serwisem przeznaczonym do przetwarzania transz protokołu NICCI. Potrafi on przetwarzać wszystkie opublikowane wersje protokołu NICCI. Protokół NICCI (wersja 2.1, 3.0, 3.1, 3.2) opisany jest w osobnych specyfikacjach technicznych.

Wersja 2.1 serwisu zapewnia możliwość asynchronicznego przetwarzania dużej ilości danych. Serwis wspiera przyjmowanie dużych danych XML (zgodnych z protokołem NICCI) w „kawałkach” oraz przetwarzanie ich w tle. Użytkownik ma także możliwość monitorowania postępu prac oraz pobierania danych wynikowych w częściach.

Dokument ten opisuje metody serwisu SIDDIN 2.1 oraz sposoby połączenia Klientów z serwerem SIDDIN 2.1. W dalszej części dokumentu serwis SIDDIN 2.1 będzie określany jako SIDDIN (bez numeru wersji). Opisane w dokumencie metody nie dotyczą wcześniejszych wersji serwisu.

### UWAGA!

#### **Prosimy o wysyłanie do nas jednej zbiorczej transzy Nicci/maksymalnie kilku zbiorczych transz Nicci.**

Wysyłanie wielu pojedynczych transz ze zleceniem np. dodania tylko jednej sprawy powoduje, że ich oczekiwanie na przetworzenie/przetworzenie po naszej stronie trwa dłużej (czasami znacznie dłużej) niż przetworzenie jednej zbiorczej transzy z wieloma zleceniami np. dodania spraw.

Właśnie dlatego tutoriala [np.: **Przykładowe wiadomości SOAP (Siddin 2.1 - Nicci 3.1, Nicci 3.2)**, dostępny także [tutaj](#)], które Państwo od nas otrzymują przy wdrożeniach, pokazują dodanie wielu spraw i przynależnych do nich zobowiązań poprzez jedną zbiorczą transzę. Zlecenie: aktualizacji, usunięcia, zawieszenia, odwieszenia itd. powinno odbywać się analogicznie. Wszystkie zlecenia mogą, a nawet powinny występować w jednej zbiorczej transzy.

#### **Dodatkowo prosimy o:**

- agregowanie zobowiązań negatywnych tego samego dłużnika w jednej sprawie negatywnej zamiast np. w kilku sprawach negatywnych,
- agregowanie zobowiązań pozytywnych tego samego kontrahenta w jednej sprawie pozytywnej zamiast np. w kilku sprawach pozytywnych.

Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

# 1. Metody serwisu SIDDIN

## 1.1. Dane ogólne

Serwis SIDDIN posługuje się wiadomościami do wykonywania metod oraz zwracania ich wyników. Wiadomości wchodzące opatrzone są przyrostkiem "Request". Wynikami metod są określone struktury danych lub w niektórych przypadkach – pojedyncze dane.

## 1.2. Uwierzytelnianie Klienta

Serwer SIDDIN uwierzytelnia Klienta za pomocą sprawdzenia poprawności loginu użytkownika oraz jego hasła. Poprawne uwierzytelnienie w systemie powoduje przekazanie Klientowi biletu (ang. Ticket), który przez określony czas (24h) może być użyty do korzystania z serwera SIDDIN.

W celu uwierzytelnienia Klient wywołuje metodę "Login" z parametrem "LoginRequest". Wiadomość "LoginRequest" posiada dwie właściwości w postaci ciągów znaków: login użytkownika ("UserName") oraz hasło ("Password"). Metoda ta zwraca ciąg znaków zawierający uzyskany bilet (co najmniej 40 znaków) bądź brak danych elementu, jeśli uwierzytelnienie się nie powiodło. Po zakończeniu pracy z serwerem należy wyrejestrować (unieważnić) uzyskany bilet, podając jako parametr metody "Logout", wiadomość "LogoutRequest" z aktualnym biletem. Jeżeli wylogowanie nie nastąpi, bilet będzie ważny przez ograniczony czas (24h), a następnie zostanie automatycznie unieważniony.

## 1.3. Ładowanie danych

Serwis wspiera ładowanie dużych danych XML w mniejszych częściach o nieograniczonej ilości, ale z zachowaniem kolejności. Ładowanie części pliku może odbywać się w dowolnych odstępach czasu. Poszczególne części nie muszą być jednakowego rozmiaru. Jedynym ograniczeniem jest ich maksymalny rozmiar.

Ładowanie pliku rozpoczyna wysłanie do serwera pierwszej jego części za pomocą metody "UploadChunk", której parametrem jest wiadomość "UploadChunkRequest". Wiadomość ta z kolei musi zawierać aktywny bilet, ciąg znaków XML oraz element "ChunkBag" będący „pojemnikiem” (ang. chunk bag) na kolejne części pliku. Wysyłając pierwszy kawałek pliku, element "ChunkBag" należy pozostawić pusty, co spowoduje utworzenie nowego „pojemnika” i wysłanie go w odpowiedzi metody. "ChunkBag" jest to struktura zawierająca informacje o ilości załadowanych części, identyfikator „pojemnika” oraz całkowity rozmiar załadowanych danych.

Aby umieścić kolejne części w istniejącym „pojemniku”, należy wykorzystać ponownie metodę "UploadChunk", tym razem podając jako parametr wiadomość "UploadChunkRequest" z podanymi wszystkimi danymi, łącznie z „pojemnikiem” "ChunkBag". W wyniku serwis zwróci zmodyfikowaną strukturę "ChunkBag".

## 1.4. Przetworzenie transzy zleceń

Przetworzenie transzy zleceń następuje w wyniku przekazania metodzie "CloseChunkBag" serwisu Siddin parametru "CloseChunkBagRequest". Parametr ten powinien zawierać następujące dane: bilet uzyskany podczas uwierzytelniania, „pojemnik”, do którego były załadowane dane, wersję

Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

protokołu NICCI, według której poskładany plik jest przetwarzany oraz opcjonalny własny opis transzy.

Po wywołaniu tej metody dane załadowane przez użytkownika są konsolidowane i sprawdzana jest ich poprawność (zgodność ze schematem XSD). W przypadku błędu generowany jest wyjątek protokołu SOAP (tzw. SOAP FAULT).

Skonsolidowana transza musi być zgodna z protokołem NICCI w wersji 2.1 lub wyższej. Protokół NICCI opisany jest w osobnych specyfikacjach technicznych.

W wyniku wywołania "CloseChunkBag", po sprawdzeniu poprawności transzy, użytkownik otrzymuje identyfikator zleconej pracy (ang. Job Id).

Posługując się metodą "GetJobs" użytkownik może śledzić stan prac z wysłanym plikiem. Metoda ta jako parametr przyjmuje zlecenie "GetJobsRequest", której parametrami są aktywny bilet oraz opcjonalnie np. identyfikator zlecenia. Wynikiem wywołania metody jest wieloelementowa tablica struktur "Job" zawierająca informacje o postępie, pozycji w kolejce zleconych transz (jeżeli wartość tego parametru jest równa 0, oznacza to, że transza jest aktualnie przetwarzana), status (przerwano, wystąpił błąd, zakończono, itp.), kod statusu, datę zlecenia, identyfikator, login zlecającego oraz opis użytkownika. Jeżeli w zleceniu "GetJobsRequest" podano identyfikator konkretnego zlecenia, wówczas metoda zwróci jednoelementową kolekcję elementów "Job" z danymi wskazanego zlecenia.

Przetwarzanie pliku transzy może zostać przerwane poprzez wywołanie metody "CancelJob" z parametrem "CancelJobRequest", która zawiera aktywny bilet oraz identyfikator wykonywanej pracy (Job Id). Anulowanie zlecenia może nastąpić tylko dla transzy oczekującej na przetworzenie.

Serwer SIDDIN przetwarza transzę i przygotowuje odpowiedź w postaci pliku XML zawierającego raport z wykonanych zleceń. Format takiego raportu jest zgodny ze specyfikacją protokołu NICCI w wersji, którą podano podczas zlecenia transzy.

W momencie, gdy status wykonania pobrany za pomocą metody "GetJobs" będzie miał wartość "2" ("Processed"), możliwe jest wywołanie metody "GetChunkBag" z parametrem "GetChunkBagRequest" zawierającym oprócz aktywnego biletu, identyfikator zadania oraz rozmiar części odpowiedzi. Ostatni parametr określa maksymalny rozmiar jednorazowo pobranej części odpowiedzi. W szczególnym przypadku będzie to tylko jedna część.

Po wywołaniu metody odpowiedź zostanie podzielona i przygotowana do ściągnięcia przez użytkownika (metody opisano w następnym podrozdziale). W rezultacie zwrócony zostanie „pojemnik” zawierający informację o tym ile części i o jakim rozmiarze przygotowano do pobrania.

W przypadku, gdy transza nie została przetworzona, a wywołano metodę "GetChunkBag", wygenerowany zostanie błąd.

W przypadku błędów krytycznych (np. nierozpoznana wersja protokołu NICCI) serwer NICCI zwraca wyjątek protokołu SOAP (tzw. SOAP FAULT), a zlecony plik zostaje opatrzony statusem "FAILED".

Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

## **1.5. Pobranie odpowiedzi z serwisu**

Aby pobrać z serwisu wynik przetworzenia transzy, należy się posłużyć „pojemnikiem” („ChunkBag”) otrzymanym po wywołaniu „GetChunkBag” oraz metodą „DownloadChunk”. Metoda ta przyjmuje wiadomość „DownloadChunkRequest” z trzema parametrami: aktywnym biletem, „pojemnikiem”, którego część ma zostać pobrana oraz numerem części do pobrania. Wynikiem wywołania metody jest ciąg znaków zawierający fragment (lub w szczególnym przypadku całość) odpowiedzi. Jest to raport z wykonanych zleceń (format takiego raportu opisany jest w protokole NICCI).

Pobrane części nie są natychmiast usuwane z serwera, więc możliwe jest ich wielokrotne pobranie, jeżeli zajdzie taka konieczność.

## **1.6. Zabezpieczenie danych**

Przechowywany w systemie KRD dokument XML, przesłany wcześniej przez użytkownika, poddawany jest przed przetworzeniem sprawdzeniu zgodności ze schematem XSD protokołu NICCI, a następnie jest podpisywany cyfrowo (wraz z nazwą użytkownika, który go umieścił w systemie), tak, aby pewne było, że w czasie oczekiwania na przetworzenie, dokument nie zostanie zmodyfikowany.

## **1.7. Obsługa błędów**

Wszystkie błędy zgłaszane przez serwis SIDDIN są zapamiętywane na serwerze, a użytkownik otrzymuje ogólny błąd „SOAP FAULT”. W opisie tego błędu znajduje się jednak identyfikator błędu, który pozwoli na zlokalizowanie opisu błędu przez dział IT Krajowego Rejestru Długów po stronie serwera. Zatem w razie problemu należy skontaktować się z konsultantem KRD (np. pomocit@krd.pl), podając identyfikator błędu.



Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

## 2. Plik WSDL

Plik WSDL (ang. Web Services Description Language) opisuje rodzaje operacji, które mogą zostać wykonane w serwisie internetowym z wykorzystaniem SOAP. W pliku opisane są typy danych (types), rodzaje komunikatów (messages) akceptowanych przez serwis, nazwy i parametry operacji (operations) oraz atrybuty samego serwisu. Plik WSDL dla serwisu SIDDIN dostępny jest pod adresami:

- <https://demo.krd.pl/Siddin/2.1/Import.asmx?WSDL> – wersja demo (wniosek o bezpłatne konto demo do testów można wysłać [tutaj](#))
- <https://services.krd.pl/Siddin/2.1/Import.asmx?WSDL> – wersja produkcyjna

### 2.1. Definicje typów

#### 2.1.1. Typ SignedRequest

Typ ten jest podstawą większości typów definiujących parametry metod serwisu, z wyjątkiem metody "Login". Zawiera on informację o aktywnym bilecie użytkownika.

```
<s:complexType name="SignedRequest">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="Ticket" type="s:string"/>
  </s:sequence>
</s:complexType>
```

#### 2.1.2. Typ LoginRequest

Typ "LoginRequest" jest parametrem metody "Login" serwisu i służy do przekazania informacji o nazwie użytkownika i hasle.

```
<s:complexType name="LoginRequest">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="UserName" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="Password" type="s:string"/>
  </s:sequence>
</s:complexType>
```

#### 2.1.3. Typ LogoutRequest

Ten typ służy do wylogowania z serwisu i „opakowuje” jedynie typ *SignedRequest*.

```
<s:complexType name="LogoutRequest">
  <s:complexContent mixed="false">
    <s:extension base="tns:SignedRequest"/>
  </s:complexContent>
</s:complexType>
```

#### 2.1.4. Typ ChunkBag

Typ "ChunkBag" opisuje parametry „pojemnika” na mniejsze części dużego pliku transzy. Zawiera takie elementy jak "ID", czyli identyfikator obiektu, "Count" określający ilość załadowanych części, "Size" przynoszący informację o łącznym rozmiarze załadowanych części w bajtach oraz "NicciVersion" z informacją o wersji protokołu. Typ ten wykorzystywany jest w metodach "UploadChunkBag", "DownloadChunk", a także jako parametr zwrotny metody "GetChunkBag".

Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

```
<s:complexType name="ChunkBag">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="ID" type="s2:guid"/>
    <s:element minOccurs="1" maxOccurs="1" name="Count" type="s:int"/>
    <s:element minOccurs="1" maxOccurs="1" name="Size" type="s:long"/>
    <s:element minOccurs="1" maxOccurs="1" name="NicciVersion" type="s1:NicciVersionEnum"/>
  </s:sequence>
</s:complexType>
```

## 2.1.5. Typ NicciVersionEnum

Typ wyliczeniowy, który określa wersje protokołu NICCI, które mogą być przekazywane przez serwis SIDDIN. Wersja 2.1 serwisu wspiera wszystkie dostępne specyfikacje NICCI – 1.1 (protokół wycofany z użytku), 1.2 (protokół wycofany z użytku), 2.0 (protokół wycofany z użytku), 2.1, 3.0, 3.1 i 3.2.

```
<s:simpleType name="NicciVersionEnum">
  <s:restriction base="s:string">
    <s:enumeration value="NotSpecified"/>
    <s:enumeration value="Version_1_1"/>
    <s:enumeration value="Version_1_2"/>
    <s:enumeration value="Version_2_0"/>
    <s:enumeration value="Version_2_1"/>
    <s:enumeration value="Version_3_0"/>
    <s:enumeration value="Version_3_1"/>
    <s:enumeration value="Version_3_2"/>
  </s:restriction>
</s:simpleType>
```

## 2.1.6. Typ Job

Zmienna typu "Job" występuje jako element kolekcji zwracanej w wyniku wywołania metody "GetJobs". Zawarte w niej informacje to numer zlecenia (JobID), kolejny numer zlecenia (JobNumber), numer w kolejce do przetworzenia (QueueNumber), procent zaawansowania (Progress), login zlecającego użytkownika (QueueLogin), data zlecenia (QueueDate), kod statusu zlecenia (StatusCode) zgodny z typem *JobStatusCodeEnum*, status w formie słownej (Status) oraz własny (opcjonalny) opis transzy/wykonanego zlecenia (Description).

```
<s:complexType name="Job">
  <s:sequence>
    <s:element minOccurs="1" maxOccurs="1" name="JobNumber" type="s:int"/>
    <s:element minOccurs="1" maxOccurs="1" name="QueueNumber" type="s:int"/>
    <s:element minOccurs="1" maxOccurs="1" name="QueueDate" type="s:dateTime"/>
    <s:element minOccurs="0" maxOccurs="1" name="Status" type="s:string"/>
    <s:element minOccurs="1" maxOccurs="1" name="StatusCode" type="s1:JobStatusCodeEnum"/>
    <s:element minOccurs="1" maxOccurs="1" name="Progress" type="s:int"/>
    <s:element minOccurs="0" maxOccurs="1" name="QueueLogin" type="s:string"/>
    <s:element minOccurs="1" maxOccurs="1" name="JobID" type="s2:guid"/>
    <s:element minOccurs="0" maxOccurs="1" name="Description" type="s:string"/>
  </s:sequence>
</s:complexType>
```

## 2.1.7. Typ JobStatusEnum

Typ definiuje możliwe stany, w jakich znajduje się zlecona przez użytkownika transza. Status zlecenia jest częścią struktury "Job" zwracanej po wywołaniu metody "GetJobs".

Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

Stan "Queued" oznacza, że transza czeka na przetworzenie (element "QueueNumber" zwracany obok statusu przez "QueryJob" określa jej numer w kolejce).

Stan "Canceled" oznacza, że użytkownik usunął z kolejki transzę za pomocą metody "Cancel" serwera.

Stan "Started" oznacza, że transza jest aktualnie przetwarzana.

Stan "Processed" oznacza, że przetwarzanie transzy zostało zakończone (w całości lub w części po wywołaniu metody "Cancel" serwera).

Stan "Failed" oznacza, że transzy nie udało się przetworzyć z przyczyn niezależnych od użytkownika. Stan ten pojawia się tylko w przypadku błędów krytycznych. Błędy w przetwarzaniu poszczególnych zleceń transzy są zwracane w raporcie zgodnie ze schematem XSD wybranej wersji NICCI.

```
<s:simpleType name="JobStatusCodeEnum">
  <s:restriction base="s:string">
    <s:enumeration value="CANCELED"/>
    <s:enumeration value="FAILED"/>
    <s:enumeration value="PROCESSED"/>
    <s:enumeration value="QUEUED"/>
    <s:enumeration value="STARTED"/>
  </s:restriction>
</s:simpleType>
```

### 2.1.8. Typ UploadChunkRequest

*UploadChunkRequest* jest typem służącym do wysłania części pliku do serwera. Składa się z parametru "Data", który zawiera ciąg znaków wysyłanej części pliku oraz parametru "ChunkBag" opisanego w punkcie [2.1.4](#). Wysyłając pierwszą z części pliku, element "ChunkBag" pozostawiamy pusty, co powoduje wygenerowanie nowego „pojemnika” na pliki. Nowy „pojemnik” otrzymujemy jako wynik przetwarzania metody serwisu. Ponieważ typ ten rozszerza typ *SignedRequest*, wymaga również podania aktywnego biletu.

```
<s:complexType name="UpladChunkRequest">
  <s:complexContent mixed="false">
    <s:extension base="tns:SignedRequest">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="ChunkBag" type="s1:ChunkBag"/>
        <s:element minOccurs="0" maxOccurs="1" name="Data" type="s:string"/>
      </s:sequence>
    </s:extension>
  </s:complexContent>
</s:complexType>
```

### 2.1.9. Typ CloseChunkRequest

Ten typ wykorzystujemy jako parametr metody "CloseChunkBag" kończącej przesyłanie danych do serwisu. Parametrami typu *CloseChunkRequest*, oprócz aktywnego biletu, jest „pojemnik” "ChunkBag", "Description" zawierający opcjonalny opis wykonanego zlecenia (opis zlecenia powinien składać się maksymalnie z 512 znaków i pojawi się na liście zleceń w elemencie "Job") oraz "GetDifference" (tzw. transza różnicowa, współgra z Nicci 2.1).

```
<s:complexType name="CloseChunkBagRequest">
  <s:complexContent mixed="false">
    <s:extension base="tns:SignedRequest">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="ChunkBag" type="s1:ChunkBag"/>
        <s:element minOccurs="0" maxOccurs="1" name="Description" type="s:string"/>
        <s:element minOccurs="1" maxOccurs="1" name="GetDifference" type="s:boolean"/>
      </s:sequence>
    </s:extension>
  </s:complexContent>
</s:complexType>
```

Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

```

</s:sequence>
</s:extension>
</s:complexContent>
</s:complexType>

```

### 2.1.10. Typ GetChunkBagRequest

Wykorzystywany jest do podzielenia pliku wyjściowego na części o zdefiniowanej wielkości oraz pobrania danych „pojemnika” „ChunkBag”, do którego trafiły części podzielonego pliku wyjściowego. Poza parametrem „Ticket” wynikającym z rozrzerzenia typu *SignedRequest*, typ *GetChunkBagRequest* posiada parametr „JobId” z identyfikatorem zlecenia oraz „ChunkSize” z rozmiarem, jaki ma przybrać pojedynczy fragment pobieranego pliku.

```

<s:complexType name="GetChunkBagRequest">
  <s:complexContent mixed="false">
    <s:extension base="tns:SignedRequest">
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="JobId" type="s2:guid"/>
        <s:element minOccurs="1" maxOccurs="1" name="ChunkSize" type="s:int"/>
      </s:sequence>
    </s:extension>
  </s:complexContent>
</s:complexType>

```

### 2.1.11. Typ DownloadChunkRequest

Ten typ używany jest w metodzie „DownloadChunk” i służy do przekazania informacji o „pojemniku” na pliki („ChunkBag”) oraz numerze części pliku do pobrania („Number”). Oczywiście wymaga również podania aktywnego biletu Klienta, gdyż jest rozszerzeniem typu *SignedRequest*.

```

<s:complexType name="DownloadChunkRequest">
  <s:complexContent mixed="false">
    <s:extension base="tns:SignedRequest">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="ChunkBag" type="s1:ChunkBag"/>
        <s:element minOccurs="1" maxOccurs="1" name="Number" type="s:int"/>
      </s:sequence>
    </s:extension>
  </s:complexContent>
</s:complexType>

```

### 2.1.12. Typ GetJobsRequest

Typ *GetJobsRequest* służy do pobrania listy zleceń przechowywanych na serwerze. Jego dodatkowym parametrem (poza biletem „Ticket”) jest „JobId” z identyfikatorem zlecenia, którego status ma zostać pobrany. Element „JobId” jest wymagany, jednak może pozostać pusty. Wówczas metoda zwróci listę wszystkich zleceń Klienta znajdujących się na serwerze.

Podobnie działa element „GetAll” – pobiera całą kolejkę zleconych plików XML (w ramach jednego konta Klienta). Aby pobrać całą kolejkę zleconych plików XML, należy w elemencie „GetAll” podać wartość *true*. W wyniku przetworzenia otrzymamy listę elementów *Job*. Można zawęzić ten wynik korzystając z elementu „LastGet”, wtedy po przetworzeniu otrzymamy listę elementów *Job*, których status zmienił się od daty podanej w elemencie „LastGet”. Zamiennie można skorzystać z elementu „LastNumber”, w którym podajemy numer zlecenia („JobNumber”). W wyniku przetworzenia otrzymamy listę elementów *Job*, których status zmienił się od czasu przesłania podanego zlecenia.

Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

```
<s:complexType name="GetJobsRequest">
  <s:complexContent mixed="false">
    <s:extension base="tns:SignedRequest">
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="GetAll" type="s:boolean"/>
        <s:element minOccurs="1" maxOccurs="1" name="LastGet" type="s:dateTime"/>
        <s:element minOccurs="0" maxOccurs="1" name="LastNumber" type="s:int"/>
        <s:element minOccurs="1" maxOccurs="1" name="JobId" type="s2:guid"/>
      </s:sequence>
    </s:extension>
  </s:complexContent>
</s:complexType>
```

### 2.1.13. Typ ArrayOfJob

Typ *ArrayOfJob* to nic innego jak tablica typów *Job* zawierających informacje o zleceniach. W przypadku, gdy w typie *GetJobsRequest* podamy identyfikator zlecenia, *ArrayOfJob* będzie zawierać tylko jeden element *Job*.

```
<s:complexType name="ArrayOfJob">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="Job" nillable="true" type="s1:Job"/>
  </s:sequence>
</s:complexType>
```

### 2.1.14. Typ CancelJobRequest

Ten typ służy do anulowania jeszcze nieprzetworzonego zlecenia. Parametrem tego typu jest identyfikator zlecenia "JobId" oraz wynikający z dziedziczenia po *SignedRequest*, aktywny bilet użytkownika "Ticket".

```
<s:complexType name="CancelJobRequest">
  <s:complexContent mixed="false">
    <s:extension base="tns:SignedRequest">
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="JobId" type="s2:guid"/>
      </s:sequence>
    </s:extension>
  </s:complexContent>
</s:complexType>
```

### 2.1.15. Typ RemoveChunkBagRequest

Ten typ służy do usunięcia przesłanego „pojemnika” ("ChunkBag"). Parametrem tego typu jest identyfikator „pojemnika” ("ChunkBagId") oraz aktywny bilet użytkownika ("Ticket").

```
<s:complexType name="RemoveChunkBagRequest">
  <s:complexContent mixed="false">
    <s:extension base="tns:SignedRequest">
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="ChunkBagId" type="s2:guid"/>
      </s:sequence>
    </s:extension>
  </s:complexContent>
</s:complexType>
```

### 2.1.16. Typ guid

Typ unikalnych ciągów znaków służących do jednoznacznego identyfikowania zadań oraz „pojemników”. Elementy tego typu są generowane przez serwer (numer paczki lub numer zlecenia) i użytkownik nie musi się troszczyć o ich unikatowość.

Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

```
<s:simpleType name="guid">
  <s:restriction base="s:string">
    <s:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}"/>
  </s:restriction>
</s:simpleType>
```

## 2.2. Definicje wiadomości

Plik WSDL dla serwisu SIDDIN definiuje 26 wiadomości: wejściową i wyjściową wiadomość dla każdej metody serwisu. Są to wiadomości:

- LoginSoapIn oraz LoginSoapOut dla metody Login
- LoginExSoapIn oraz LoginExSoapOut dla metody LoginEx
- LogoutSoapIn oraz LogoutSoapOut dla metody Logout
- UploadChunkSoapIn oraz UploadChunkSoapOut dla metody UploadChunk
- CloseChunkBagSoapIn oraz CloseChunkBagSoapOut dla metody CloseChunkBag
- CloseChunkBagExSoapIn oraz CloseChunkBagExSoapOut dla metody CloseChunkBagEx
- GetChunkBagSoapIn oraz GetChunkBagSoapOut dla metody GetChunkBag
- GetChunkBagExSoapIn oraz GetChunkBagExSoapOut dla metody GetChunkBagEx
- GetRepeatFileSoapIn oraz GetRepeatFileSoapOut dla metody GetRepeatFile
- DownloadChunkSoapIn oraz DownloadChunkSoapOut dla metody DownloadChunk
- GetJobsSoapIn oraz GetJobsSoapOut dla metody GetJobs
- CancelJobSoapIn oraz CancelJobSoapOut dla metody CancelJob
- RemoveChunkBagSoapIn oraz RemoveChunkBagSoapOut dla metody RemoveChunkBag

Dla każdej z tych wiadomości definiuje się parametry wywołania oraz typy zwracanych wartości (m.in. zdefiniowane w poprzednim rozdziale). Poniżej znajduje się opis podstawowych metod (metody LoginEx, CloseChunkBagEx, GetChunkBagEx powstały swego czasu na potrzeby jednej z naszych aplikacji – Sabar i protokołu XML Nicci 2.1/CSV Yonick 2.0. Podobnie jest z metodą GetRepeatFile, która pobiera tzw. transzę powtórkową tylko z błędnymi pozycjami, które system odrzucił, przetwarzając transzę wsadową – tyle że współgra również z innymi wersjami protokołu Nicci).



### 2.2.1. Wiadomości LoginSoapIn i LoginSoapOut

Wiadomość wejściowa dla metody "Login", typu *LoginRequest* zawiera informacje o identyfikatorze ("UserName") oraz hasło użytkownika ("Password").

Wiadomość wyjściowa zwraca *LoginResult* jako ciąg znaków – bilet (ang. Ticket).

```
<wsdl:message name="LoginSoapIn">
  <wsdl:part name="LoginRequest" element="tns:LoginRequest"/>
</wsdl:message>
<wsdl:message name="LoginSoapOut">
  <wsdl:part name="LoginResult" element="tns:LoginResult"/>
</wsdl:message>
```

### 2.2.2. Wiadomości LogoutSoapIn i LogoutSoapOut

Wiadomość wejściowa dla metody "Logout" wymaga podania typu *LogoutRequest* z aktywnym biletem. Metoda nie zwraca żadnych parametrów.

```
<wsdl:message name="LogoutSoapIn">
  <wsdl:part name="LogoutRequest" element="tns:LogoutRequest"/>
</wsdl:message>
<wsdl:message name="LogoutSoapOut"/>
```

### 2.2.3. Wiadomości UploadChunkSoapIn i UploadChunkSoapOut

*UploadChunkSoapIn* jest wiadomością metody "UploadChunk" służącej do przekazywania części pliku XML zlecanej transzy NICCI. Wymaga podania typu *UploadChunkRequest* opisanego w punkcie [2.1.8](#). Jako odpowiedź otrzymujemy element *UploadChunkResult* zawierający „pojemnik” do umieszczania części transzy.

```
<wsdl:message name="UploadChunkSoapIn">
  <wsdl:part name="UploadChunkRequest" element="tns:UploadChunkRequest"/>
</wsdl:message>
<wsdl:message name="UploadChunkSoapOut">
  <wsdl:part name="UploadChunkResult" element="s1:UploadChunkResult"/>
</wsdl:message>
```

### 2.2.4. Wiadomości CloseChunkBagSoapIn i CloseChunkBagSoapOut

Aby zakończyć przesyłanie części pliku i zlecić go do wykonania, należy wywołać metodę "CloseChunkBag", której parametrem jest typ *CloseChunkBagRequest*.

Wynikiem wywołania metody jest element typu *CloseChunkBagRequest* zawierający identyfikator zlecenia "JobId".

```
<wsdl:message name="CloseChunkBagSoapIn">
  <wsdl:part name="CloseChunkBagRequest" element="tns:CloseChunkRequest"/>
</wsdl:message>
<wsdl:message name="CloseChunkBagSoapOut">
  <wsdl:part name="CloseChunkBagResult" element="tns:CloseChunkBagResult"/>
</wsdl:message>
```

### 2.2.5. Wiadomości GetChunkBagSoapIn i GetChunkBagSoapOut

Aby po zakończeniu przetwarzania transzy pobrać plik odpowiedzi, należy w pierwszej kolejności wywołać metodę "GetChunkBag". Parametrem wywołania jest *GetChunkBagRequest*. Wynikiem operacji jest element typu

Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

*GetChunkBagResult* zawierający "ChunkBag" niezbędny do pobrania odpowiedzi za pomocą metody "DownloadChunk".

```
<wsdl:message name="GetChunkBagSoapIn">
  <wsdl:part name="GetChunkBagRequest" element="tns:GetChunkBagRequest"/>
</wsdl:message>
<wsdl:message name="GetChunkBagSoapOut">
  <wsdl:part name="GetChunkBagResult" element="s1:GetChunkBagResult"/>
</wsdl:message>
```

### 2.2.6. Wiadomości DownloadChunkSoapIn i DownloadChunkSoapOut

Aby otrzymać wyniki przetwarzania, użytkownik powinien wywołać metodę "DownloadChunk". Parametrem wywołania metody jest "DownloadChunkRequest", typu *DownloadChunkRequest* opisany w punkcie [2.1.11](#). Wynikiem wywołania metody jest *DownloadChunkResult* jako ciąg znaków zawierający fragment (lub w szczególności cały) pliku XML.

```
<wsdl:message name="DownloadChunkSoapIn">
  <wsdl:part name="downloadChunkRequest" element="tns:DownloadChunkRequest"/>
</wsdl:message>
<wsdl:message name="DownloadChunkSoapOut">
  <wsdl:part name="DownloadChunkResult" element="tns:DownloadChunkResult"/>
</wsdl:message>
```

### 2.2.7. Wiadomości GetJobsSoapIn i GetJobsSoapOut

Użytkownik ma możliwość sprawdzić status zleconego zadania poprzez wywołanie metody "GetJobs". Parametrem jej wywołania jest element typu *GetJobsRequest*, na który składają się: aktywny bilet ("Ticket") oraz identyfikator elementu w kolejce ("JobId").

Parametrem wynikowym jest element typu *GetJobsResult* zawierający tablicę elementów typu *Job*.

```
<wsdl:message name="GetJobsSoapIn">
  <wsdl:part name="GetJobsRequest" element="tns:GetJobsRequest"/>
</wsdl:message>
<wsdl:message name="GetJobsSoapOut">
  <wsdl:part name="GetJobsResult" element="tns:GetJobsResult"/>
</wsdl:message>
```

### 2.2.8. Wiadomości CancelJobSoapIn i CancelJobSoapOut

Wiadomość *CancelJobSoapIn* służy jako parametr metody "CancelJob" wykluczającej zlecone zadanie z kolejki i zawiera typ *CancelJobRequest* z aktywnym biletem ("Ticket") oraz identyfikatorem elementu w kolejce ("JobId").

Metoda nie zwraca żadnych danych, zatem w celu sprawdzenia, czy zlecenie zostało anulowane, należy skorzystać z metody "GetJobs", podając identyfikator zlecenia.

```
<wsdl:message name="CancelJobSoapIn">
  <wsdl:part name="CancelJobRequest" element="tns:CancelJobRequest"/>
</wsdl:message>
<wsdl:message name="CancelJobSoapOut"/>
```



Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

## 2.2.9. Wiadomości RemoveChunkBagSoapIn i RemoveChunkBagSoapOut

Wiadomość *RemoveChunkBagSoapIn* służy jako parametr metody "RemoveChunkBag" usuwającej przesłany „pojemnik” ("ChunkBag").

Aby usunąć przesłany „pojemnik”, wywołać należy metodę "RemoveChunkBag". Aby operacja się powiodła, należy podać aktywny bilet ("Ticket") oraz Id „pojemnika”, który chcemy usunąć. Metoda nie zwraca żadnych parametrów. Informacyjnie – „pojemniki” i tak są cyklicznie usuwane po naszej stronie.

```
<wsdl:message name="RemoveChunkBagSoapIn">
  <wsdl:part name="removeChunkBagRequest" element="tns:RemoveChunkBagRequest"/>
</wsdl:message>
<wsdl:message name="RemoveChunkBagSoapOut"/>
```

## 2.3. Element portType

Następny element pliku WSDL (portType) definiuje zestaw dostępnych w serwisie operacji – każdy z elementów składowych zawiera nazwę oraz powiązane z nim wiadomości (wejściową i wyjściową), których definicje przedstawiono w poprzednim rozdziale.

```
<wsdl:portType name="Import">
  <wsdl:operation name="Login">
    <wsdl:input message="tns:LoginSoapIn"/>
    <wsdl:output message="tns:LoginSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="LoginEx">
    <wsdl:input message="tns:LoginExSoapIn"/>
    <wsdl:output message="tns:LoginExSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="Logout">
    <wsdl:input message="tns:LogoutSoapIn"/>
    <wsdl:output message="tns:LogoutSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="UploadChunk">
    <wsdl:input message="tns:UploadChunkSoapIn"/>
    <wsdl:output message="tns:UploadChunkSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="CloseChunkBag">
    <wsdl:input message="tns:CloseChunkBagSoapIn"/>
    <wsdl:output message="tns:CloseChunkBagSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="CloseChunkBagEx">
    <wsdl:input message="tns:CloseChunkBagExSoapIn"/>
    <wsdl:output message="tns:CloseChunkBagExSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetChunkBag">
    <wsdl:input message="tns:GetChunkBagSoapIn"/>
    <wsdl:output message="tns:GetChunkBagSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetChunkBagEx">
    <wsdl:input message="tns:GetChunkBagExSoapIn"/>
    <wsdl:output message="tns:GetChunkBagExSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetRepeatFile">
    <wsdl:input message="tns:GetRepeatFileSoapIn"/>
    <wsdl:output message="tns:GetRepeatFileSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="DownloadChunk">
    <wsdl:input message="tns:DownloadChunkSoapIn"/>
    <wsdl:output message="tns:DownloadChunkSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
```

Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

```

</wsdl:operation>
<wsdl:operation name="GetJobs">
  <wsdl:input message="tns:GetJobsSoapIn"/>
  <wsdl:output message="tns:GetJobsSoapOut"/>
</wsdl:operation>
<wsdl:operation name="CancelJob">
  <wsdl:input message="tns:CancelJobSoapIn"/>
  <wsdl:output message="tns:CancelJobSoapOut"/>
</wsdl:operation>
<wsdl:operation name="RemoveChunkBag">
  <wsdl:input message="tns:RemoveChunkBagSoapIn"/>
  <wsdl:output message="tns:RemoveChunkBagSoapOut"/>
</wsdl:operation>
</wsdl:portType>

```

## 2.4. Element binding

Następny element pliku WSDL (binding) definiuje powiązanie między usługami serwisu a protokołem transportowym. Za pomocą elementu `soap:binding` wybierany jest schemat oraz styl transportu. Następnie dla każdej metody serwisu (operation) określone są dalsze parametry (np. adres i styl akcji).

```

<wsdl:binding name="Import" type="tns:Import">
  <wsdl:documentation>
    <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.1"/>
  </wsdl:documentation>
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="Login">
    <soap:operation soapAction="http://Siddin.ServiceContracts/2006/09/Login"
style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="LoginEx">
    <soap:operation soapAction="http://Siddin.ServiceContracts/2006/09/LoginEx"
style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="Logout">
    <soap:operation soapAction="http://Siddin.ServiceContracts/2006/09/Logout"
style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="UploadChunk">
    <soap:operation soapAction="http://Siddin.ServiceContracts/2006/09/UploadChunk"
style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>

```

Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

```

    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CloseChunkBag">
  <soap:operation soapAction="http://Siddin.ServiceContracts/2006/09/CloseChunkBag"
style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CloseChunkBagEx">
  <soap:operation soapAction="http://Siddin.ServiceContracts/2006/09/CloseChunkBagEx"
style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetChunkBag">
  <soap:operation soapAction="http://Siddin.ServiceContracts/2006/09/GetChunkBag"
style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetChunkBagEx">
  <soap:operation soapAction="http://Siddin.ServiceContracts/2006/09/GetChunkBagEx"
style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetRepeatFile">
  <soap:operation soapAction="http://Siddin.ServiceContracts/2006/09/GetRepeatFile"
style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="DownloadChunk">
  <soap:operation soapAction="http://Siddin.ServiceContracts/2006/09/DownloadChunk"
style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetJobs">

```

Siddin 2.1	Wersja dokumentacji: 1.2
Specyfikacja techniczna web serwisu Siddin 2.1	Z dnia: 2022-04-28

```

    <soap:operation soapAction="http://Siddin.ServiceContracts/2006/09/GetJobs"
style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="CancelJob">
    <soap:operation soapAction="http://Siddin.ServiceContracts/2006/09/CancelJob"
style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="RemoveChunkBag">
    <soap:operation soapAction="http://Siddin.ServiceContracts/2006/09/RemoveChunkBag"
style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

```

## 2.5. Definicja serwisu

W definicji serwisu zawarte jest określenie portu – miejsca, w jakim usługa jest udostępniana (location) oraz wybranie dla danego portu jednego ze zdefiniowanych wcześniej powiązań.

```

<wsdl:service name="Import">
  <wsdl:port name="Import" binding="tns:Import">
    <soap:address location="https://services.krd.pl/Siddin/2.1/Import.asmx"/>
  </wsdl:port>
</wsdl:service>

```

**KONIEC DOKUMENTU**