AT autoridade tributária e aduaneira

SDSCAU – VIA POSTAL

WebService Externo

MOD 12.2

Av. Eng.º Duarte Pacheco n.º 28 Apartado 1852 1099-013 LISBOA PORTUGAL Tel: (+351) 213834200 Fax: (+351) 213834646
Email: asa@at.gov.pt www.portaldasfinancas.gov.pt Centro de Atendimento Telefónico: (+351) 217 206 707

VERSÃO

DATA	AUTOR	VERSÃO	COMENTÁRIO
31-10-2019	AT	1.0	Versão inicial
08-02-2021	AT	1.1	Versão revista
02-09-2021	AT	1.2	Versão revista
02-11-2021	AT	1.3	Versão revista. Alteração do número mínimo de ocorrências nas respostas.

Índice

1 Introdução

1.1 Objetivo

O objetivo deste documento é descrever o WebService disponibilizados para a interligação do sistema SDSCAU com os CTT (no âmbito da Via Postal), sendo descritas todas as funcionalidades e modo de utilização do mesmo.

1.2 Âmbito

O âmbito deste documento contempla a especificação técnica do sistema SDSCAU.

2 Definições e Siglas

2.1 Siglas

SDSCAU – Sistema Integrado dos Meios de Transporte e das Mercadorias – Código Aduaneiro da União

XML – Formato das mensagens enviadas e geradas pelo sistema (eXtensible Markup Language)

3 WebService

De forma a utilizar o WebService o cliente necessitará de se autenticar no Portal da AT, autenticação esta cujo mecanismo deverá estar de acordo com o já definido pelo Portal da AT.

Sob determinadas condições (sobrecarga do sistema, reenvio repetido de mensagens, entre outras), o WebService poderá rejeitar os pedidos como política de garantia de estabilidade do sistema, ficando o cliente responsável, nestes casos, de efetuar novamente esses pedidos mais tarde.

Para a interligação entre o sistema SDSCAU e os CTT é disponibilizado o seguinte WebService.

3.1 Cabeçalho do Pedido SOAP

A estrutura do cabeçalho do pedido SOAP está desenhada de forma a garantir a confidencialidade dos dados de autenticação e a impossibilidade de reutilização dos mesmos em possíveis ataques Man-In-The-Middle (MITM). Assim, só serão aceites os pedidos que assegurem os procedimentos de encriptação e codificação dos dados.

O SOAP:Header terá de ser construído com base no standard WS-Security, definido pela OASIS e recorrendo à definição do Username Token Profile 1.1.

A tabela abaixo descreve como terão de ser construídos cada um dos campos presentes no cabeçalho:

Campo	Descrição	Obrig.	Tipo de Dados
H.1 - Username	NIF do utilizador que vai submeter os dados, composto da seguinte forma e de acordo com a autenticação do portal das finanças. Exemplos: 1. 500500500 2. 500500500/0002 (subutilizador n.º 2) 3. 500500500/1234 (subutilizador n.º 1234)	Sim	String
H.2 - Nonce	Chave simétrica gerada a cada pedido e para cifrar o conteúdo dos campos H.3 - Password e H.4 - Created. Cada invocação do Webservice deverá conter esta chave gerada aleatoriamente e a qual não pode ser repetida. Para garantir a confidencialidade, a chave simétrica tem de ser cifrada com a chave pública do Sistema de Autenticação de acordo com o algoritmo RSA e codificada em Base 64. A chave pública do sistema de autenticação do portal das finanças deve ser obtida por solicitação própria e através do endereço de email asi-cd@at.gov.pt. O campo é construído de acordo com o seguinte procedimento	Sim	String (base 64)

	$Nonce := Base64(C_{RSA,Kpub_{SA}}(K_s))$ **K_S** := array de bytes com a chave simétrica de 128 bits, produzida de acordo com a norma AES. **$C_{RSA,KpubSA}$** := Função de cifra da chave simétrica com o algoritmo RSA utilizando a chave pública do sistema de autenticação (K_{pubSA}). **Base64** := Codificação em Base 64 do resultado.		
H.3 - Password	O campo Password deverá conter a senha do utilizador, a mesma que é utilizada para entrar no Portal das Finanças. Esta Password tem de ser cifrada através da chave simétrica do pedido (ver campo H.2 - Nonce) e codificado em Base64. $Password := Base64(C_{K_s}^{AES,ECB,PKCS5Padding}(SenhaPF))$ **SenhaPF** := Senha do utilizador definido no campo H.1 - Username ***$C_{Ks}^{AES, ECB, PKCS 5 Padding}$*** := Função de cifra utilizando o algoritmo AES, Modelo ECB, PKCS5Padding e a chave simétrica do pedido (K_S). **Base64** := Codificação em Base 64 do resultado.	Sim	String (base 64)
H.4 - Data de sistema (Created)	O campo Created deverá conter a data e hora de sistema da aplicação que está a invocar o webservice. Esta data é usada para validação temporal do pedido, pelo que é crucial que o sistema da aplicação cliente tenha o seu relógio certo. Sugere-se a sincronização com o Observatório Astronómico de Lisboa: • http://oal.ul.pt/hora-legal/como-acertar/ A zona temporal deste campo deverá estar definida para UTC+0 e formatado de acordo com a norma ISO 8601 tal como é definido pelo W3C: • https://www.w3.org/QA/Tips/iso-date • https://www.w3.org/TR/NOTE-datetime (e.g.: 2021-08-16T15:31:22.163Z) Este campo é cifrado com a chave de pedido (K_S) e codificada em Base64. $Created := Base64(C_{K_s}^{AES,ECB,PKCS5Padding}(Timestamp))$ **Timestamp** := data hora do sistema (UTC+0); ***$C_{Ks}^{AES, ECB, PKCS 5 Padding}$*** := Função de cifra utilizando o algoritmo AES, Modelo ECB, PKCS5Padding e a chave simétrica do pedido (K_S). **Base64** := Codificação em Base 64 do resultado.	Sim	String (base 64)

Exemplo:

```
<soap:Header>
  <wss:Security xmlns:wss="http://schemas.xmlsoap.org/ws/2002/12/secext">
    <wss:UsernameToken>
      <wss:Username>500077568</wss:Username>
      <wss:Password>ld4cF6/M8MT5L5ftSsbyjg==</wss:Password>

<wss:Nonce>XLrfW8ks/Nl6UAp+UXucW76lt4kSwx1ZsgrhtiMBXrccm1Yfg0wkkqkTfkRL7CSHtnRyjCMQx+
ch5OMbc1qGuwOOPFtt57MtQKoV3oMubttAWvDwaPJWmPvOUrA5K65MQ6uGcYIuzTiqmqlEowhZlUES
ah1q2NtisCthqpSe0qHxsHJypa2Be4igxg6a3KliEJPLTSBpRa5e7njBmS0gZG3t5eHv6QllaYSBTMoKV2uh
76JHeoayYezXK1REpxjjcH4Pr45/iRfuQX6oFnsoCwjpsRPDZ5Sbpk47NnSeyKbJ27DIpNKct34zVioICb8R
MGQ9T/5EL6RPUke4+2ZYSQ==</wss:Nonce>
      <wss:Created>6BA7Xy1lWfB9zPtvHP3i+ppY8LGP26HrV0e/RP2eKDE=</wss:Created>
    </wss:UsernameToken>
  </wss:Security>
</soap:Header>
```

3.2 WebService PostalTraders

São disponibilizados três métodos no WebService:

- ***submitMessages*** – Este método permite o envio de uma ou mais mensagens em simultâneo e retorna "*OK*" no caso do sistema SDSCAU ter recebido as mensagens com sucesso ou "*NOK*" em caso de existir algum erro na sua receção. O retorno da invocação deste método não está relacionado com o processamento do conteúdo das mensagens que será efetuado num momento posterior.
- ***retrieveNewOutputMessagesIds*** – Este método retorna a lista de IDs das mensagens novas que o sistema SDSCAU gerou para os CTT. Esses IDs serão passados na invocação do próximo método *retrieveOutputMessagesByIds*.
- ***retrieveOutputMessagesByIds*** – Este método retorna as mensagens XML indicadas na lista de IDs.

3.2.1 Método: submitMessages

Este método é responsável pela receção e posterior processamento de mensagens enviadas pelos CTT.

Parâmetros de Input:

- *String sender* – EORI correspondente à entidade (CTT) que está a efetuar a invocação deste método. Pode, opcionalmente, ser concatenado no final o caracter "/" mais o utilizador orgânico (por ex: PT123456789/1). As mensagens de resposta e de notificação resultantes serão geradas contendo o *receiver* igual ao *sender*.
- *List<String> messages* – Lista de mensagens que os CTT pretendem enviar para o SDSCAU. Estas mensagens deverão estar codificadas em UTF8 e no formato base64.

Retorno:

- *List<String>* – Lista com o resultado da receção de cada mensagem enviada:
    - *"OK"* – Quando a mensagem em causa foi recebida com sucesso (não deverá ser reenviada ao SDSCAU);
    - *"NOK <message>"* – Quando a mensagem em causa não foi recebida com sucesso (deverá ser reenviada ao SDSCAU posteriormente). Por ex, é retornado "NOK Messages are undefined" caso o parâmetro *messages* seja vazio.

3.2.2 Método: retrieveNewOutputMessagesIds

Este método é responsável por retornar aos CTT quais são os IDs das mensagens novas que o sistema SDSCAU gerou e que estão disponíveis para serem obtidas pelo mesmo.

Parâmetros de Input:

- *String receiver* – EORI correspondente aos CTT que pretende obter a lista de IDs das suas mensagens novas. Pode, opcionalmente, ser concatenado no final o caracter "/" mais o utilizador orgânico (por ex: PT123456789/1).

Retorno:

- Lista com os IDs das mensagens novas que estão disponíveis para serem obtidas com o método *retrieveOutputMessagesByIds*. Caso tenha sido indicado o utilizador orgânico no parâmetro *receiver*, só serão retornados os IDs destinados a esse utilizador orgânico. Caso tenha sido indicado só o EORI coletivo, serão retornados todos os IDs destinados a esse EORI (incluindo todos os utilizadores orgânicos).

Esta invocação não produz qualquer alteração de estado no sistema SDSCAU e deve ser invocada periodicamente (por ex: de 1 em 1 minuto) para que os CTT tenham conhecimento de que têm novas mensagens para vir buscar ao sistema SDSCAU.

É da responsabilidade dos CTT guardar os IDs recebidos no retorno desta invocação antes de efetuar a invocação ao próximo método *retrieveOutputMessagesByIds*, para que eventuais problemas de comunicações não resultem em perda de dados entre os sistemas.

3.2.3 Método: retrieveOutputMessagesByIds

Este método é responsável por retornar aos CTT as mensagens de saída identificadas pelos IDs indicados e marcar as mesmas como já enviadas (para deixarem de constar no retorno do método *retrieveNewOutputMessagesIds*).

Parâmetros de Input:

- *String receiver* – EORI correspondente aos CTT que pretende obter as mensagens XML com os IDs indicados. Pode, opcionalmente, ser concatenado no final o caracter "/" mais o utilizador orgânico (por ex: PT123456789/1).
- *List<String> outputMessagesIds* – Lista dos IDs das mensagens de saída que os CTT pretendem obter.

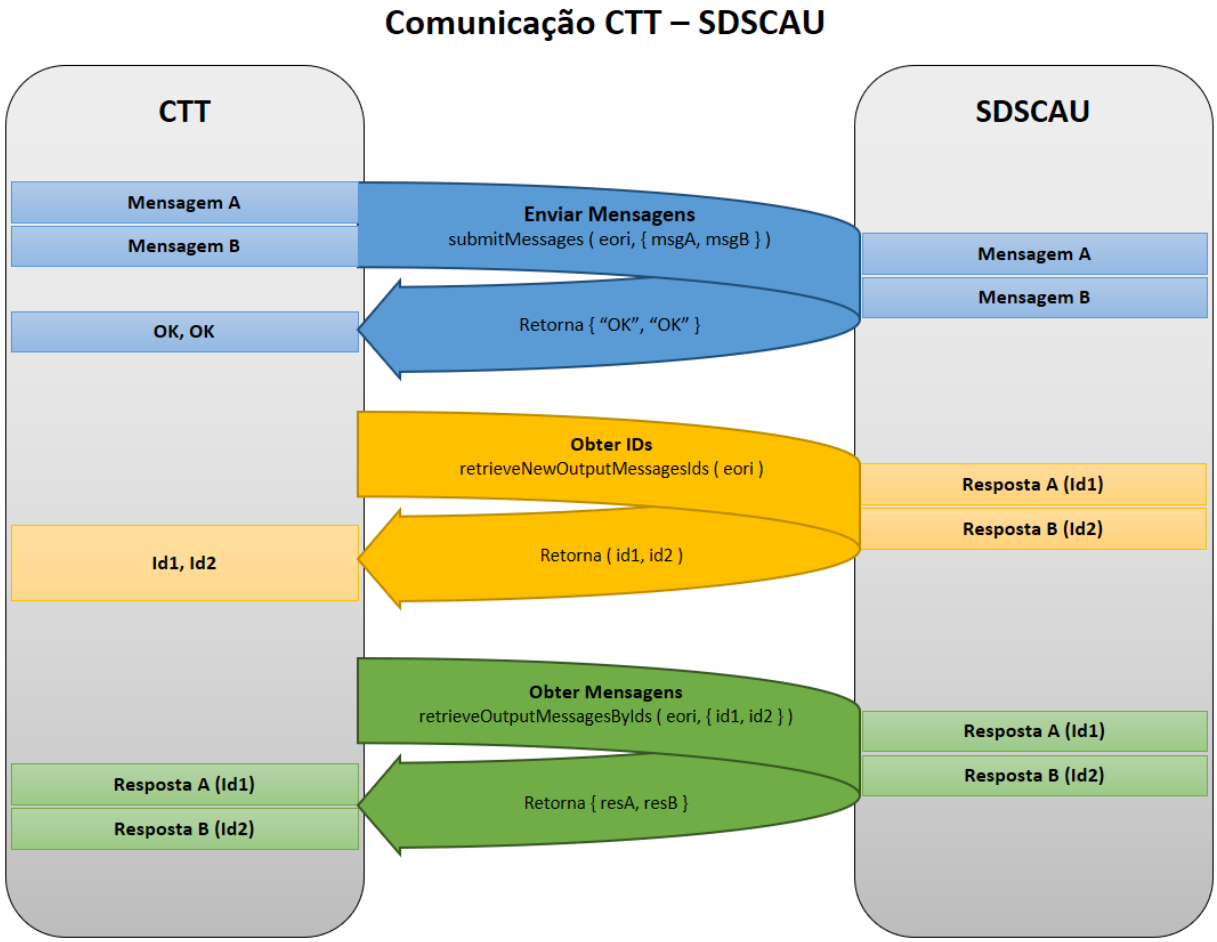
Retorno:

- List<String> - Lista com as mensagens XML de saída cujos IDs foram indicados no input. Estas mensagens encontram-se codificadas em base64.

Caso haja problemas de comunicações no retorno das mensagens entre o sistema SDSCAU e os CTT e as mesmas não cheguem ao destino, os referidos IDs já poderão ter sido marcados como enviados. De qualquer das formas, este método retornará sempre as mensagens com os IDs indicados, pelo que poderá ser invocado mais do que uma vez. É da responsabilidade dos CTT manter os IDs retornados na invocação do método *retrieveNewOutputMessagesIds* e só os descartar quando realmente consegue receber com sucesso as respetivas mensagens XML.

3.2.4 Fluxo

De seguida ilustra-se um exemplo de troca de mensagens entre os CTT e o SDSCAU.



Neste exemplo cronologicamente são realizadas as seguintes ações:

- Os CTT enviam duas Mensagens ao SDSCAU invocando o método *submitMessages* passando como argumentos o seu EORI e as duas mensagens XML em base64. O SDSCAU retorna uma lista de duas Strings com "OK", "OK" significando que o SDSCAU recebeu as mensagens com sucesso.
- Logo após, o SDSCAU processa as duas mensagens recebidas, e cria duas mensagens de resposta.
- Os CTT invocam o método *retrieveNewOutputMessagesIds* que retorna 2 números de identificação de mensagens (as duas respostas).
- Os CTT recebem esses 2 identificadores, e guardam na sua Base de Dados para salvaguardar problemas de comunicação que possam ocorrer na invocação seguinte.

- Os CTT invocam o método *retrieveOutputMessagesByIds* indicando os 2 IDs, e o SDSCAU retorna o XML das 2 mensagens em causa (em base64). Neste momento o SDSCAU marca os 2 IDs como já respondidos, deixando de constar a partir desse momento do retorno do método *retrieveNewOutputMessagesIds*.
- Os CTT guardam as 2 mensagens XML recebidas e procede ao seu processamento.

4 WSDL

Para a troca de mensagens com os CTT é disponibilizado o seguinte WSDL:

```
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:sch="http://sdscau.at.gov.pt/WS/PostalTraders"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:tns="http://sdscau.at.gov.pt/WS/PostalTraders"
targetNamespace="http://sdscau.at.gov.pt/WS/PostalTraders">
  <wsdl:types>
    <xs:schema xmlns:ns="http://sdscau.at.gov.pt/WS/PostalTraders"
xmlns:xs="http://www.w3.org/2001/XMLSchema" attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="http://sdscau.at.gov.pt/WS/PostalTraders">

        <!-- submitMessages -->

        <xs:element name="submitMessagesRequest" type="ns:submitMessagesRequest"/>
        <xs:element name="submitMessagesResponse" type="ns:submitMessagesResponse"/>

        <xs:complexType name="submitMessagesRequest">
                <xs:sequence>
                        <xs:element name="sender" type="ns:EoriType">
                                <xs:annotation>
                                        <xs:documentation>EORI number of the
sender</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element maxOccurs="unbounded" name="messages"
type="xs:base64Binary">
                                <xs:annotation>
                                        <xs:documentation>Messages content encoded in
Base64</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="submitMessagesResponse">
                <xs:sequence>
                        <xs:element maxOccurs="unbounded" name="return" type="xs:string">
                                <xs:annotation>
                                        <xs:documentation>Response state for each message
received</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:complexType>

        <!-- retrieveNewOutputMessagesIds -->

        <xs:element name="retrieveNewOutputMessagesIdsRequest"
type="ns:retrieveNewOutputMessagesIdsRequest"/>
```

```
        <xs:element name="retrieveNewOutputMessagesIdsResponse"
type="ns:retrieveNewOutputMessagesIdsResponse"/>

        <xs:complexType name="retrieveNewOutputMessagesIdsRequest">
                <xs:sequence>
                        <xs:element name="receiver" type="ns:EoriType">
                                <xs:annotation>
                                        <xs:documentation>EORI number of the
receiver</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="retrieveNewOutputMessagesIdsResponse">
                <xs:sequence>
                        <xs:element maxOccurs="unbounded" minOccurs="0" name="return"
type="ns:MessageIdType">
                                <xs:annotation>
                                        <xs:documentation>Messages Ids</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:complexType>

        <!-- retrieveOutputMessagesByIds -->

        <xs:element name="retrieveOutputMessagesByIdsRequest"
type="ns:retrieveOutputMessagesByIdsRequest"/>
        <xs:element name="retrieveOutputMessagesByIdsResponse"
type="ns:retrieveOutputMessagesByIdsResponse"/>

        <xs:complexType name="retrieveOutputMessagesByIdsRequest">
                <xs:sequence>
                        <xs:element name="receiver" type="ns:EoriType">
                                <xs:annotation>
                                        <xs:documentation>EORI number of the
receiver</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element maxOccurs="unbounded" name="outputMessagesIds"
type="ns:MessageIdType">
                                <xs:annotation>
                                        <xs:documentation>Messages Ids</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="retrieveOutputMessagesByIdsResponse">
                <xs:sequence>
                        <xs:element maxOccurs="unbounded" minOccurs="0" name="return"
type="xs:base64Binary">
                                <xs:annotation>
                                        <xs:documentation>Messages content encoded in
Base64</xs:documentation>
                                </xs:annotation>
```

```
                    </xs:element>
                </xs:sequence>
            </xs:complexType>

            <!-- Types -->

            <xs:simpleType name="EoriType">
                <xs:annotation>
                    <xs:documentation>EORI number (can include organic user)</xs:documentation>
                </xs:annotation>
                <xs:restriction base="xs:string">
                <xs:maxLength value="22"/>
                <xs:pattern value="[A-Z]{2}.{1,15}(\/\d{1,4})?"/>
            </xs:restriction>
            </xs:simpleType>
            <xs:simpleType name="MessageIdType">
                <xs:annotation>
                    <xs:documentation>Message identification number</xs:documentation>
                </xs:annotation>
                <xs:restriction base="xs:string">
                <xs:maxLength value="36"/>
                <xs:pattern value=".{1,36}"/>
            </xs:restriction>
            </xs:simpleType>

</xs:schema>
  </wsdl:types>
  <wsdl:message name="submitMessagesRequest">
    <wsdl:part element="tns:submitMessagesRequest" name="submitMessagesRequest">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="retrieveNewOutputMessagesIdsRequest">
    <wsdl:part                                    element="tns:retrieveNewOutputMessagesIdsRequest"
name="retrieveNewOutputMessagesIdsRequest">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="submitMessagesResponse">
    <wsdl:part element="tns:submitMessagesResponse" name="submitMessagesResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="retrieveNewOutputMessagesIdsResponse">
    <wsdl:part                                   element="tns:retrieveNewOutputMessagesIdsResponse"
name="retrieveNewOutputMessagesIdsResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="retrieveOutputMessagesByIdsResponse">
    <wsdl:part                                    element="tns:retrieveOutputMessagesByIdsResponse"
name="retrieveOutputMessagesByIdsResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="retrieveOutputMessagesByIdsRequest">
    <wsdl:part                                     element="tns:retrieveOutputMessagesByIdsRequest"
name="retrieveOutputMessagesByIdsRequest">
    </wsdl:part>
  </wsdl:message>
```

```
 <wsdl:portType name="WSPostalTradersPort">
  <wsdl:operation name="submitMessages">
   <wsdl:input message="tns:submitMessagesRequest" name="submitMessagesRequest">
  </wsdl:input>
   <wsdl:output message="tns:submitMessagesResponse" name="submitMessagesResponse">
  </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="retrieveNewOutputMessagesIds">
   <wsdl:input                                    message="tns:retrieveNewOutputMessagesIdsRequest"
name="retrieveNewOutputMessagesIdsRequest">
  </wsdl:input>
   <wsdl:output                                  message="tns:retrieveNewOutputMessagesIdsResponse"
name="retrieveNewOutputMessagesIdsResponse">
  </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="retrieveOutputMessagesByIds">
   <wsdl:input                                     message="tns:retrieveOutputMessagesByIdsRequest"
name="retrieveOutputMessagesByIdsRequest">
  </wsdl:input>
   <wsdl:output                                   message="tns:retrieveOutputMessagesByIdsResponse"
name="retrieveOutputMessagesByIdsResponse">
  </wsdl:output>
  </wsdl:operation>
 </wsdl:portType>
 <wsdl:binding name="WSPostalTradersPortSoap12" type="tns:WSPostalTradersPort">
  <soap12:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="submitMessages">
   <soap12:operation soapAction=""/>
   <wsdl:input name="submitMessagesRequest">
    <soap12:body use="literal"/>
   </wsdl:input>
   <wsdl:output name="submitMessagesResponse">
    <soap12:body use="literal"/>
   </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="retrieveNewOutputMessagesIds">
   <soap12:operation soapAction=""/>
   <wsdl:input name="retrieveNewOutputMessagesIdsRequest">
    <soap12:body use="literal"/>
   </wsdl:input>
   <wsdl:output name="retrieveNewOutputMessagesIdsResponse">
    <soap12:body use="literal"/>
   </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="retrieveOutputMessagesByIds">
   <soap12:operation soapAction=""/>
   <wsdl:input name="retrieveOutputMessagesByIdsRequest">
    <soap12:body use="literal"/>
   </wsdl:input>
   <wsdl:output name="retrieveOutputMessagesByIdsResponse">
    <soap12:body use="literal"/>
   </wsdl:output>
  </wsdl:operation>
 </wsdl:binding>
 <wsdl:service name="WSPostalTradersPortService">
```

```
    <wsdl:port binding="tns:WSPostalTradersPortSoap12" name="WSPostalTradersPortSoap12">
      <soap12:address location=""/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```