



STADAIMPCAU

Guia de implementação WebServices V1.1

Documentação Técnica

Índice

1. Introdução	4
2. Estrutura do envio de dados à AT (SOAP)	5
3. SOAP:Header	6
3.1. Exemplo de SOAP:Header.....	11
3.2. Erro de Autenticação/Autorização	11
4. SOAP:Body	13
4.1. Serviço processaMensagemICAUM001A	14
4.1.1. Exemplo de SOAP:Body	15
4.2. Serviço processaMensagemICAUM007A	15
4.2.1. Exemplo de SOAP:Body	16
4.3. Serviço obterDocumentosPDFNaoEntregues.....	16
4.3.1. Exemplo de SOAP:Body	17
4.4. Serviço obterDocumentosPDFEntregues	18
4.4.1. Exemplo de SOAP:Body	19
4.5. Serviço obterMensagensNaoEntregues	20
4.5.1. Exemplo de SOAP:Body	21
4.6. Serviço obterMensagensEntregues	22
4.6.1. Exemplo de SOAP:Body	23
4.7. Serviço entregarDocumentos.....	24
4.7.1. Exemplo de SOAP:Body	24
5. Códigos de Resultado	26
6. Tipos de PDF/Mensagem	27
7. Assinatura certificado SSL (CSR)	28
7.1. Gerar um certificado SSL	29
7.2. Verificar conteúdo do CSR gerado.....	29
7.3. Integrar certificado com a chave privada	30
8. Endereços Úteis	31
8.1. Página de produtores de software	31
8.2. Gestão de subutilizadores no Portal das Finanças.....	31
8.3. WSDL do envio de dados à AT por Webservice	31
8.4. Endereços para envio de dados à AT por Webservice.....	31

DATA	AUTOR	VERSÃO	COMENTÁRIO
03-02-2021	AT	1.0	Primeira versão
17-03-2021	AT	1.1	Adicionado Código de Resultado 901. Alteração nos serviços obterDocumentosPDFEntregues e obterMensagensEntregues: número de ocorrências máximo limitada a 50 ficheiros por resposta. Acrescentado capítulo 3-2 – Erro de Autenticação/Autorização.

1. Introdução

O presente manual contém as especificações técnicas do canal Webservice que será disponibilizado aos Operadores Económicos no âmbito do novo sistema de Importação para a tramitação da Declaração Aduaneira para Remessas de Baixo Valor (DAIRBV).

Os destinatários deste manual são os Operadores Económicos autorizados pela AT a comunicar eletronicamente com as Estâncias Aduaneiras, utilizando o canal Webservice.

Para este canal de comunicação serão disponibilizados serviços para:

- processamento da mensagem ICAUM001A;
- processamento da mensagem ICAUM007A;
- obtenção dos documentos PDF gerados pelo sistema;
- obtenção das notificações/mensagens geradas pelo sistema;
- envio de documentos digitalizados.

A estrutura das mensagens ICAUM001A e ICAUM007A bem como das mensagens geradas pelo sistema para notificação dos Operadores Económicos no âmbito do processo de tramitação da declaração é apresentada no documento *STADAIMPCAU – Guia XML – Definições Gerais v1.3* pelo que não serão abordadas neste manual.

2. Estrutura do envio de dados à AT (SOAP)

Neste ponto descreve-se a metodologia do Webservice do STADAIMPCAU.

O Webservice é efetuado segundo o protocolo SOAP e é constituído por duas secções:

- **SOAP:Header:**
 - Esta secção inclui todos os campos de autenticação do utilizador que vai ser responsável pela invocação do Webservice;
 - O utilizador poderá ser um subutilizador do NIF do sujeito passivo com perfil DWS;
 - A criação de subutilizadores deve ser efetuada no Portal das Finanças utilizando a opção:
 - “Serviços tributários/Outros serviços/Gestão de utilizadores”.
- **SOAP:Body**
 - Esta secção contém os dados referentes ao sistema STADAIMPCAU, os quais se detalham no ponto SOAP:Body.

3. SOAP:Header

O desenho do Header tem como requisito garantir a confidencialidade dos dados de autenticação e a impossibilidade de reutilização dos mesmos em ataques Man-in-the-middle (MITM). Por este motivo, só serão aceites invocações que respeitem os procedimentos de encriptação.

O SOAP:Header é construído de acordo com o standard WS-Security, definido pela OASIS e recorrendo à definição do Username Token Profile 1.1, também definido pela mesma organização.

Na seguinte tabela, detalha-se a forma de construção de cada campo, de acordo com as necessidades de segurança específicas do sistema de autenticação do Portal das Finanças.

Parâmetro	Descrição	Obrigatório (S/N)	Tipo de Dados
H.1 – Utilizador (Username)	<p>Identificação do utilizador que vai submeter os dados, composto da seguinte forma e de acordo com a autenticação do Portal das Finanças:</p> <p><NIF>/<SUBUTILIZADOR></p> <p>Exemplos:</p> <ul style="list-style-type: none">• 111111111• 55555555/1 (subutilizador n.º 1)• 55555555/0002 (subutilizador n.º 2)• 55555555/1234 (subutilizador n.º 1234)	S	String

Parâmetro	Descrição	Obrigatório (S/N)	Tipo de Dados
H.2 – Password	<p>O campo Password deverá conter a senha do utilizador/subutilizador, a mesma que é utilizada para entrar no Portal das Finanças.</p> <p>Esta Password tem de ser cifrada através da chave simétrica do pedido (ver H.2 – Nonce) e codificado em Base64.</p> <p style="text-align: center;">Password</p> <p>$:= \text{Base64}(C_{K_S}^{\text{AES,ECB,PKCS5Padding}}(\text{Senha}))$</p> <p>SenhaPF := Senha do utilizador definido no campo</p> <p>H.1 – Username;</p> <p>$C_{K_S}^{\text{AES,ECB,PKCS5Padding}}$:= Função de cifra utilizando o algoritmo AES, Modelo ECB, PKCS5Padding e a chave simétrica do pedido (K_s).</p> <p>Base64 := Codificação em Base 64 do resultado.</p>	S	string (base64)

Parâmetro	Descrição	Obrigatório (S/N)	Tipo de Dados
H.3 – Nonce	<p>Chave simétrica gerada a cada pedido e para cifrar o conteúdo dos campos H.3 - Password e H.4 - Created.</p> <p>Cada invocação do Webservice deverá conter esta chave gerada aleatoriamente e a qual não pode ser repetida.</p> <p>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.</p> <p>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.</p> <p>O campo é construído de acordo com o seguinte procedimento:</p> <p>Nonce := Base64($C_{RSA, Kpub_{SA}}(K_s)$)</p> <p>K_s := array de bytes com a chave simétrica de 128 bits, produzida de acordo com a norma AES.</p> <p>$C_{RSA, Kpub_{SA}}$:= Função de cifra da chave simétrica com o algoritmo RSA utilizando a chave pública do sistema de autenticação ($Kpub_{SA}$).</p> <p>Base64 := Codificação em Base 64 do resultado.</p>	S	String (base64)

Parâmetro	Descrição	Obrigatório (S/N)	Tipo de Dados
H.4 – Data de sistema (Created)	<p>O campo Created deverá conter a data e hora de sistema da aplicação que está a invocar o webservice.</p> <p>Esta data é usada para validação temporal do pedido, pelo que é crucial que o sistema da aplicação cliente tenha o seu relógio certo.</p> <p>Sugere-se a sincronização com o Observatório Astronómico de Lisboa: http://www.oal.ul.pt/index.php?link=acerto</p> <p>A zona temporal deste campo deverá estar definida para UTC e formatado de acordo com a norma ISO 8601 tal como é definido pelo W3C: http://www.w3.org/QA/Tips/iso-date http://www.w3.org/TR/NOTE-datetime</p> <p>Exemplo: 2017-01-01T19:20:30.45Z</p> <p>Este campo é cifrado com a chave de pedido (KS) e codificada em Base 64.</p> <p>Created $:= \text{Base64}(C_{KS}^{\text{AES,ECB,PKCS5Padding}}(\text{Timestamp}))$</p> <p>Timestamp := data hora do sistema (UTC);</p> <p>$C_{KS}^{\text{AES,ECB,PKCS5Padding}}$:= Função de cifra utilizando o algoritmo AES, Modelo ECB, PKCS5Padding e a chave simétrica do pedido (Ks).</p> <p>Base64 := Codificação em Base 64 do resultado.</p>	S	string (base64)

3.1. Exemplo de SOAP:Header

Como resultado da aplicação das regras de construção anteriores será produzido um *header* de pedido SOAP tal como se apresenta no seguinte exemplo:

```
<S:Header>
  <wss:Security xmlns:wss="http://schemas.xmlsoap.org/ws/2002/12/secext">
    <wss:UsernameToken>
      <wss:Username>500000016/15</wss:Username>
      <wss:Password>ikCyRV+SWfvZ5c6Q0bhrBQ==</wss:Password>
      <wss:Nonce>
        fkAHne7cquxplmCfBC8EEc2vskyUyNofWi0ptlijYg4gYCxir++unzfPVPpusloEtmLkcZjf+E6T9/76ts
        CqdupUkxOhWtkRH5IrNwmfEW1ZGFQgYTF21iyKBRzMdsJMhhHrofYYV/YhSPdT4dIlgG0tk9Z73
        6jFuW061mP2TNqHcR/mQR0yW/AEOC6RPumqO8Oafc9/b4KFBSfbpY9HRzbD8bKiTo20n0Pta
        mZevCSVHht4yt/Xwgd+KV70WFzyesGVMOgFRTWZyXyXBVaBrkJ58b6PoJxADLcpWRnw5+YeOs
        3cPU2o1H/YgAam1QuEHioCT2YTdRt+9p6ARNEIFg==
      </wss:Nonce>
      <wss:Created>>YEWoloqIY5DOD11SeXz+0i4b/AJg1/RgNcOH0YpSxGk</wss:Created>
    </wss:UsernameToken>
  </wss:Security>
</S:Header>
```

3.2. Erro de Autenticação/Autorização

No caso de existirem erros de Autenticação ou de Autorização para utilização do Webservice será devolvido um SOAP Fault com a seguinte estrutura:

```
<xsl:template name="createSoapFault">
  <xsl:param name="errorMessage"/>
  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ws="https://www.
portaldasfinancas.gov.pt/">
    <soapenv:Header/>
    <soapenv:Body>
      <soapenv:Fault>
        <faultcode>{$errorCode}</faultcode>
```

```

<faultstring>
  <xsl:value-of select="$errorMessage"/>
</faultstring>
<detail>
  <transactionID><xsl:value-of select="$vExternalTransactionID" /></transactionID>
  <timestamp><xsl:value-of select="date:format-date(date:date-time(), 'yyyy-MM-dd
HH:mm:ss')"/></timestamp>
</detail>
</soapenv:Fault>
</soapenv:Body>
</soapenv:Envelope>
</xsl:template>

```

No *ErrorMessage* será devolvida a mensagem 'Erro de Autenticação/Autorização'. No entanto, o *ErrorCode* será específico para cada tipo de erro o que permitirá perceber se se trata de um erro relacionado com a validação da Autenticação ou um erro relacionado com a validação da Autorização:

ErrorCode	Descrição
<i>Erro na autenticação</i>	
1<nn>	Erro <nn> na validação da Autenticação
1888	Formato do Pedido XML Incorreto: SecurityHeader, ClientIP ou nomeTaxonomia vazios ou nulos
1999	Erro na invocação da Autenticação.
<i>Erro na autorização</i>	
2<nn>	Erro <nn> na validação da Autorização sendo que os códigos possíveis de serem apresentados são: <ul style="list-style-type: none"> • 101 – Utilizador sem Adesão • 102 – Sub-utilizador desconhecido
2888	Formato do Pedido XML Incorreto: Utilizador vazio ou nulo
2999	Erro na invocação da Autorização

4. SOAP:Body

Nesta secção são definidos os serviços para a criação, anulação ou alteração de declarações, bem como a obtenção de documentos ou mensagens.

Os serviços definidos para este Webservice são:

Método	Descrição
processaMensagemIM001A	<p>Serviço que permite enviar as mensagens:</p> <p>ICAUM001A</p> <ul style="list-style-type: none"> • 0110 - Criação da declaração • 0140 – Proposta de Alteração Antes da Autorização de Saída - Pedido do Operador • 0150 – Proposta de Alteração Após a Autorização de Saída - Pedido do Operador • 0160 – Resposta à Proposta de alteração antes da autorização de saída por Iniciativa da AT • 0190 - Resposta à Proposta de alteração após a autorização de saída por Iniciativa da AT
processaMensagemIM007A	<p>Serviço que permite enviar as mensagens:</p> <p>ICAUM007A</p> <ul style="list-style-type: none"> • 0710 - Anulação a Pedido do Operador
obterDocumentosPDFNaoEntregues	<p>Serviço que permite a um operador obter os documentos PDF que ainda não lhe foram entregues para uma determinada DAIRBV.</p> <p>Este serviço permite pesquisar por NRL e Tipo de PDF, até um máximo de 5 NRL.</p> <p>O serviço retorna ainda um indicador que diz se após esta resposta ainda há mais informação por transmitir (1).</p>
obterDocumentosPDFEntregues	<p>Serviço que permite a um operador obter os documentos PDF já entregues anteriormente para uma determinada DAIRBV.</p> <p>Este serviço permite pesquisar por NRL e Tipo de PDF, para um único NRL.</p> <p>O serviço retorna ainda um indicador que diz se após esta resposta ainda há mais informação por transmitir (2).</p>
obterMensagensNaoEntregues	<p>Serviço que permite a um operador obter as mensagens XML que ainda não lhe foram entregues para uma determinada DAIRBV.</p> <p>Este serviço permite pesquisar por NRL e Tipo de XML, até um máximo de 5 NRL.</p> <p>O serviço retorna ainda um indicador que diz se após esta resposta ainda há mais informação por transmitir (1).</p>

Método	Descrição
obterMensagensEntregues	<p>Serviço que permite a um operador obter as mensagens XML já entregues anteriormente para uma determinada DAIRBV.</p> <p>Este serviço permite pesquisar por NRL e Tipo de XML, para um único NRL.</p> <p>O serviço retorna ainda um indicador que diz se após esta resposta ainda há mais informação por transmitir (2).</p>
entregarDocumentos	<p>Serviço que permite a um operador enviar até um máximo de 20 documentos digitalizados para 1 NRL.</p> <p>Só serão aceites documentos no formato PDF ou PNG.</p>

- (1) Com o intuito de restringir a dimensão da resposta aos serviços **obterDocumentosPDFNaoEntregues** e **obterMensagensNaoEntregues**, a resposta a estes serviços está limitada a 10 PDFs ou 10 mensagens por DAIRBV (ou seja, por NRL), respetivamente. Assim, caso existam mais dados a transmitir isso será sinalizado na resposta através dum atributo **existeMaisInformacao** e nesse caso o Operador deverá invocar novamente o serviço para obter o resto da informação.
- (2) Com o intuito de restringir a dimensão da resposta aos serviços **obterDocumentosPDFEntregues** e **obterMensagensEntregues**, a resposta a estes serviços está limitada a 50 PDFs ou 50 mensagens, respetivamente. Assim, caso existam mais dados a transmitir isso será sinalizado na resposta através dum atributo **existeMaisInformacao** e nesse caso o Operador deverá invocar novamente o serviço para obter o resto da informação.

4.1. Serviço processaMensagemICAUM001A

Input

Parâmetros de entrada	Tipo de dados	Tamanho	Mín/Máx	Descrição
processaMensagemInput	Objeto		1..1	Objeto de input.
mensagem	byte[]		1..1	Mensagem ICAUM001A em base64.

Output

Parâmetros de saída	Tipo de dados	Mín/Máx	Descrição
resultadoProcessamento	Objeto	1..1	Resultado do processamento da mensagem.
codigoResultado	String	1..1	Código de resultado.
descricaoResultado	String	1..1	Descrição do resultado.
dataEnvio	String	1..1	Data e hora de envio da resposta.

4.1.1. Exemplo de SOAP:Body

Como resultado da aplicação do serviço **processaMensagemICAUM001A** será produzido um *body* de pedido SOAP tal como se apresenta no seguinte exemplo:

```
<soapenv:Body>
  <web:processaMensagemICAUM001A>
    <processaMensagemInput>
      <mensagem>Mensagem ICAUM001A em base64</mensagem>
    </processaMensagemInput>
  </web:processaMensagemICAUM001A>
</soapenv:Body>
```

A este pedido o sistema responde da seguinte forma:

```
<env:Body>
  <m: processaMensagemICAUM001A>
    <return>
      <resultadoProcessamento>
        <codigoResultado>0</codigoResultado>
        <descricaoResultado> Mensagem submetida com sucesso.</descricaoResultado>
      </resultadoProcessamento>
      <dataEnvio>2020-12-12 19:30</dataEnvio>
    </return>
  </m: processaMensagemICAUM001A>
</env:Body>
```

4.2. Serviço processaMensagemICAUM007A

Input

Parâmetros de entrada	Tipo de dados	Tamanho	Mín/Máx	Descrição
processaMensagemInput	Objeto		1..1	Objeto de input.
mensagem	byte[]		1..1	Mensagem ICAUM007A em base64.

Output

Parâmetros de saída	Tipo de dados	Mín/Máx	Descrição
resultadoProcessamento	Objeto	1..1	Resultado do processamento da mensagem.
codigoResultado	String	1..1	Código de resultado.

descricaoResultado	String	1..1	Descrição do resultado.
dataEnvio	String	1..1	Data e hora de envio da resposta.

4.2.1. Exemplo de SOAP:Body

Como resultado da aplicação do serviço **processaMensagemICAUM007A** será produzido um *body* de pedido SOAP tal como se apresenta no seguinte exemplo:

```
<soapenv:Body>
  <web:processaMensagemICAUM007A>
    <processaMensagemInput>
      <mensagem>Mensagem ICAUM007A em base64</mensagem>
    </processaMensagemInput>
  </web:processaMensagemICAUM007A>
</soapenv:Body>
```

A este pedido o sistema responde da seguinte forma:

```
<env:Body>
  <m: processaMensagemICAUM007A>
    <return>
      <resultadoProcessamento>
        <codigoResultado>0</codigoResultado>
        <descricaoResultado> Mensagem submetida com sucesso.</descricaoResultado>
      </resultadoProcessamento>
      <dataEnvio>2020-12-12 19:30</dataEnvio>
    </return>
  </m: processaMensagemICAUM007A>
</env:Body>
```

4.3. Serviço obterDocumentosPDFNaoEntregues

Input

Parâmetros de entrada	Tipo de dados	Tamanho	Mín/Máx	Descrição
obterDocumentosInput	Objeto		1..1	Objeto de Input.
listaNRLs	ArrayList		1..5	Lista de Números de Referência Local.
numeroReferenciaLocal	String	22	1..1	Número de Referência Local da declaração.
tipo	String	3	0..1	Tipo de PDF.

Output

Parâmetros de saída	Tipo de dados	Mín/Máx	Descrição
ResultadoProcessamento	Objeto	1..1	Resultado do processamento da mensagem.
codigoResultado	String	1..1	Código de resultado.
descricaoResultado	String	1..1	Descrição do resultado.
declaracao	Objeto	0..5	
numeroReferenciaLocal	String	1..1	Número de referência local que identifica a declaração.
codigoResultado	String	0..1	Código de resultado do NRL (caso existam erros).
descricaoResultado	String	0..1	Descrição do resultado.
ficheiroResposta	Objeto	0..10	Lista de resultados dos documentos PDF não entregues.
nomeFicheiro	String	1..1	Nome do ficheiro.
ficheiro	String	1..1	Documento em base64.
existeMaisInformacao	String	1..1	Indicador da existência de mais PDFs. Se existirem mais PDFs será devolvido true , caso contrário será devolvido false .
dataEnvio	String	1..1	Data e hora de envio da resposta.

4.3.1. Exemplo de SOAP:Body

Como resultado da aplicação do serviço **obterDocumentosPDFNaoEntregues** será produzido um *body* de pedido SOAP tal como se apresenta no seguinte exemplo:

```
<soapenv:Body>
  <web:obterDocumentosPDFNaoEntregues>
    <obterDocumentosInput>
      <listaNRLs>
        <numeroReferenciaLocal>20PT999999999000000001</numeroReferenciaLocal>
        <numeroReferenciaLocal>20PT999999999000000002</numeroReferenciaLocal>
        <numeroReferenciaLocal>20PT999999999000000003</numeroReferenciaLocal>
        <numeroReferenciaLocal>20PT999999999000000004</numeroReferenciaLocal>
        <numeroReferenciaLocal>20PT999999999000000005</numeroReferenciaLocal>
      </listaNRLs>
      <tipo>VAL</tipo>
    </obterDocumentosInput>
  </web:obterDocumentosPDFNaoEntregues>
</soapenv:Body>
```

Como resultado desta invocação será devolvido o seguinte:

```
<env:Body>
  <m: obterDocumentosPDFNaoEntregues>
```

```

<return>
  <resultadoProcessamento>
    <codigoResultado>0</codigoResultado>
    <descricaoResultado> Mensagem processada com sucesso.</descricaoResultado>
  </resultadoProcessamento>
  <declaracao>
    <numeroReferenciaLocal>20PT999999999000000001</numeroReferenciaLocal>
    <codigoResultado>100</codigoResultado>
    <descricaoResultado>Dados encontrados.</descricaoResultado>
    <ficheiroResposta>
      <nomeFicheiro> Documento A </nomeFicheiro>
      <ficheiro> Documento em base64</ficheiro>
    </ficheiroResposta>
    <ficheiroResposta>
      <nomeFicheiro> Documento B </nomeFicheiro>
      <ficheiro> Documento em base64</ficheiro>
    </ficheiroResposta>
    <existeMaisInformacao>true</existeMaisInformacao>
  </declaracao>
  <declaracao>
    <numeroReferenciaLocal>20PT999999999000000002</numeroReferenciaLocal>
    <codigoResultado>102</codigoResultado>
    <descricaoResultado>Número de Referência Local inexistente.</descricaoResultado>
    <existeMaisInformacao>false</existeMaisInformacao>
  </declaracao>
  <dataEnvio>2020-12-12 19:30</dataEnvio>
</return>
</m: obterDocumentosPDFNaoEntregues>
</env:Body>

```

4.4. Serviço obterDocumentosPDFEntregues

Input

Parâmetros de entrada	Tipo de dados	Tamanho	Mín/Máx	Descrição
obterDocumentosInput	Objeto		1..1	Objeto de Input.
numeroReferenciaLocal	String	22	1..1	Número de Referência Local da declaração.
tipo	String	3	0..1	Tipo de PDF.

Output

Parâmetros de saída	Tipo de dados	Mín/Máx	Descrição
---------------------	---------------	---------	-----------

resultadoProcessamento	Objeto	1..1	Resultado do processamento da mensagem.
codigoResultado	String	1..1	Código de resultado.
descricaoResultado	String	1..1	Descrição do resultado.
declaracao	Objeto	0..1	
numeroReferenciaLocal	String	1..1	Número de referência local que identifica a declaração
codigoResultado	String	0..1	Código de resultado do NRL (caso existam erros).
descricaoResultado	String	0..1	Descrição do resultado.
ficheiroResposta	Objeto	0..50	Lista de resultados dos documentos PDF entregues.
nomeFicheiro	String	1..1	Nome do ficheiro.
ficheiro	String	1..1	Documento em base64.
existeMaisInformacao	String	1..1	Indicador da existência de mais PDFs. Se existirem mais PDFs será devolvido true , caso contrário será devolvido false .
dataEnvio	String	1..1	Data e hora de envio da resposta.

4.4.1. Exemplo de SOAP:Body

Como resultado da aplicação do serviço **obterDocumentosPDFEntregues** será produzido um *body* de pedido SOAP tal como se apresenta no seguinte exemplo:

```
<soapenv:Body>
  <web:obterDocumentosPDFEntregues>
    <obterDocumentosInput>
      <numeroReferenciaLocal>20PT9999999900000001</numeroReferenciaLocal>
      <tipo>VAL</tipo>
    </obterDocumentosInput>
  </web:obterDocumentosPDFEntregues>
</soapenv:Body>
```

A este pedido o sistema responde da seguinte forma:

```
<env:Body>
  <m: obterDocumentosPDFEntregues>
    <return>
      <resultadoProcessamento>
        <codigoResultado>0</codigoResultado>
        <descricaoResultado> Mensagem processada com sucesso.</descricaoResultado>
      </resultadoProcessamento>
      <declaracao>
        <numeroReferenciaLocal>20PT9999999900000001</numeroReferenciaLocal>
        <codigoResultado>100</codigoResultado>
        <descricaoResultado> Dados encontrados.</descricaoResultado>
      </declaracao>
    </return>
  </m: obterDocumentosPDFEntregues>
</env:Body>
```

```

    <ficheiroResposta>
      <nomeFicheiro> Documento A </nomeFicheiro>
      <ficheiro> Documento em base64</ficheiro>
    </ficheiroResposta>
  <ficheiroResposta>
    <nomeFicheiro> Documento B </nomeFicheiro>
    <ficheiro> Documento em base64</ficheiro>
  </ficheiroResposta>
</declaracao>
<dataEnvio>2020-12-12 19:30</dataEnvio>
</return>
</m: obterDocumentosPDFEntregues>
</env:Body>

```

4.5. Serviço obterMensagensNaoEntregues

Input

Parâmetros de entrada	Tipo de dados	Tamanho	Mín/Máx	Descrição
obterMensagensInput	Objeto		1..1	Objeto de Input.
listaNRLs	ArrayList		1..5	Lista de Números de Referência Local.
numeroReferenciaLocal	String	22	1..1	Número de Referência Local da declaração.
tipo	String	3	0..1	Tipo de Mensagem.

Output

Parâmetros de saída	Tipo de dados	Mín/Máx	Descrição
resultadoProcessamento	Objeto	1..1	Resultado do processamento da mensagem.
codigoResultado	String	1..1	Código de resultado.
descricaoResultado	String	1..1	Descrição do resultado.
declaracao	Objeto	0..5	
numeroReferenciaLocal	String	1..1	Número de referência local que identifica a declaração
codigoResultado	String	0..1	Código do resultado para o NRL (caso existam erros).
descricaoResultado	String	0..1	Descrição do resultado.
ficheiroResposta	Objeto	0..10	Lista de resultados das mensagens XML não entregues.
nomeFicheiro	String	1..1	Nome do ficheiro.
ficheiro	String	1..1	Documento em base64.
existeMaisInformacao	String	1..1	Indicador da existência de mais mensagens XML. Se existirem mais mensagens XML será devolvido true , caso contrário será devolvido false .
dataEnvio	String	1..1	Data e hora de envio da resposta.

4.5.1. Exemplo de SOAP:Body

Como resultado da aplicação do serviço **obterMensagensNaoEntregues** será produzido um *body* de pedido SOAP tal como se apresenta no seguinte exemplo:

```
<soapenv:Body>
  <web:obterMensagensNaoEntregues>
    <obterMensagensInput>
      <listaNRLs>
        <numeroReferenciaLocal>20PT999999999000000001</numeroReferenciaLocal>
        <numeroReferenciaLocal>20PT999999999000000002</numeroReferenciaLocal>
        <numeroReferenciaLocal>20PT999999999000000003</numeroReferenciaLocal>
        <numeroReferenciaLocal>20PT999999999000000004</numeroReferenciaLocal>
        <numeroReferenciaLocal>20PT999999999000000005</numeroReferenciaLocal>
      </listaNRLs>
      <tipo>VAL</tipo>
    </obterMensagensInput>
  </web:obterMensagensNaoEntregues>
</soapenv:Body>
```

A este pedido o sistema responde da seguinte forma:

```
<env:Body>
  <m: obterMensagensNaoEntregues>
    <return>
      <resultadoProcessamento>
        <codigoResultado>0</codigoResultado>
        <descricaoResultado> Mensagem processada com sucesso.</descricaoResultado>
      </resultadoProcessamento>
      <declaracao>
        <numeroReferenciaLocal>20PT999999999000000001</numeroReferenciaLocal>
        <codigoResultado>100</codigoResultado>
        <descricaoResultado> Dados encontrados.</descricaoResultado>
        <ficheiroResposta>
          <nomeFicheiro> Documento A </nomeFicheiro>
          <ficheiro> Documento em base64</ficheiro>
        </ficheiroResposta>
        <ficheiroResposta>
          <nomeFicheiro> Documento B </nomeFicheiro>
          <ficheiro> Documento em base64</ficheiro>
        </ficheiroResposta>
        <existeMaisInformacao>true</existeMaisInformacao>
      </declaracao>
      <declaracao>
        <numeroReferenciaLocal>20PT999999999000000002</numeroReferenciaLocal>
```

```

<codigoResultado>100</codigoResultado>
<descricaoResultado> Dados encontrados.</descricaoResultado>
<ficheiroResposta>
  <nomeFicheiro> Documento C </nomeFicheiro>
  <ficheiro> Documento em base64</ficheiro>
</ficheiroResposta>
<existeMaisInformacao>true</existeMaisInformacao>
</declaracao>
<dataEnvio>2020-12-12 19:30</dataEnvio>
</return>
</m: obterMensagensNaoEntregues>
</env:Body>

```

4.6. Serviço obterMensagensEntregues

Input

Parâmetros de entrada	Tipo de dados	Tamanho	Mín/Máx	Descrição
obterMensagensInput	Objeto		1..1	Objeto de Input.
numeroReferencialLocal	String	22	1..1	Número de Referência Local da declaração.
tipo	String	3	0..1	Tipo de Mensagem.

Output

Parâmetros de saída	Tipo de dados	Mín/Máx	Descrição
ResultadoProcessamento	Objeto	1..1	Resultado do processamento da mensagem.
codigoResultado	String	1..1	Código de resultado.
descricaoResultado	String	1..1	Descrição do resultado.
declaracao	Objeto	0..1	
numeroReferencialLocal	String	1..1	Número de referência local que identifica a declaração
codigoResultado	String	0..1	Código do resultado para o NRL (caso existam erros).
descricaoResultado	String	0..1	Descrição do resultado.
ficheiroResposta	Objeto	0..50	Lista de resultados das mensagens XML entregues.
nomeFicheiro	String	1..1	Nome do ficheiro.
ficheiro	String	1..1	Documento em base64.
existeMaisInformacao	String	1..1	Indicador da existência de mais mensagens XML. Se existirem mais mensagens XML será devolvido true , caso contrário será devolvido false .
dataEnvio	String	1..1	Data e hora de envio da resposta.

4.6.1. Exemplo de SOAP:Body

Como resultado da aplicação do serviço **obterMensagensEntregues** será produzido um *body* de pedido SOAP tal como se apresenta no seguinte exemplo:

```
<soapenv:Body>
  <web:obterMensagensEntregues>
    <obterMensagensInput>
      <numeroReferenciaLocal>20PT99999999000000001</numeroReferenciaLocal>
      <tipo>VAL</tipo>
    </obterMensagensInput>
  </web:obterMensagensEntregues>
</soapenv:Body>
```

A este pedido o sistema responde da seguinte forma:

```
<env:Body>
  <m: obterMensagensEntregues>
    <return>
      <resultadoProcessamento>
        <codigoResultado>0</codigoResultado>
        <descricaoResultado> Mensagem processada com sucesso.</descricaoResultado>
      </resultadoProcessamento>
      <declaracao>
        <numeroReferenciaLocal>20PT99999999000000001</numeroReferenciaLocal>
        <codigoResultado>100</codigoResultado>
        <descricaoResultado> Dados encontrados.</descricaoResultado>
        <ficheiroResposta>
          <nomeFicheiro> Documento A </nomeFicheiro>
          <ficheiro> Documento em base64</ficheiro>
        </ficheiroResposta>
        <ficheiroResposta>
          <nomeFicheiro> Documento B </nomeFicheiro>
          <ficheiro> Documento em base64</ficheiro>
        </ficheiroResposta>
      </declaracao>
      <dataEnvio>2020-12-12 19:30</dataEnvio>
    </return>
  </m: obterMensagensEntregues>
</env:Body>
```

4.7. Serviço entregarDocumentos

Input

Parâmetros de entrada	Tipo de dados	Tamanho	Mín/Máx	Descrição
identificacaoDeclaracao	Objeto		1..1	Identificação da declaração.
numeroReferenciaLocal	String	22	1..1	Número de Referência Local da declaração.
mrn	String	20	1..1	MRN da declaração.
numProvisorio	String	15	1..1	Número Provisório da declaração.
documento	Objeto		1..20	Lista de documentos a enviar.
codigoDocumento	String	15	1..1	Código do documento.
numAdicao	Short	1	1..1	Número de Adição. Se se tratar de um documento declarado ao nível da declaração então deve ser indicado '0'.
nomeFicheiro	String	22	1..1	Nome do ficheiro a enviar.
conteudoFicheiro	byte[]		1..1	Documento em base64.

Output

Parâmetros de saída	Tipo de dados	Mín/Máx	Descrição
resultadoProcessamento	Objeto	1..1	Resultado do processamento da mensagem.
codigoResultado	String	1..1	Código de resultado.
descricaoResultado	String	1..1	Descrição do resultado.
documento	Objeto	1..n	Lista dos documentos enviados com o resultado do processamento de cada documento.
codigoDocumento	String	1..1	Código do documento com erro.
nomeFicheiro	String	1..1	Nome do ficheiro com erro.
codigoResultado	String	1..1	Código do resultado para o documento.
descricaoResultado	String	1..1	Descrição do resultado.
dataEnvio	String	1..1	Data e hora de envio da resposta.

4.7.1. Exemplo de SOAP:Body

Como resultado da aplicação do serviço **entregarDocumentos** será produzido um *body* de pedido SOAP tal como se apresenta no seguinte exemplo:

```
<soapenv:Body>
  <web:entregarDocumentos>
    < identificacaoDeclaracao >
      <numeroReferenciaLocal>20PT123456987963258741</numeroReferenciaLocal>
      <mrn>20PT1234567889123R1</mrn>
```



```
<numeroProvisorio>899/2020</ numeroProvisorio>
</identificacaoDeclaracao>
<documento>
  <codigoDocumento>N380</tipoDocumento>
  <numAdicao>3</ numAdicao>
  <nomeFicheiro>documento_digitalizado1.pdf</nomeFicheiro>
  <conteudoFicheiro> Documento em base64</conteudoFicheiro>
</documento>
<documento>
  <codigoDocumento>N750</tipoDocumento>
  <numAdicao>C</ numAdicao>
  <nomeFicheiro>documento_digitalizado2.doc</nomeFicheiro>
  <conteudoFicheiro> Documento em base64</conteudoFicheiro>
</documento>
</web:entregarDocumentos>
</soapenv:Body>
```

Como resultado desta invocação será devolvido o seguinte:

```
<env:Body>
  <m: entregarDocumentos>
    <return>
      <resultadoProcessamento>
        <codigoResultado>0</codigoResultado>
        <descricaoResultado> Mensagem processada com sucesso.</descricaoResultado>
      </resultadoProcessamento>
      <documento>
        <codigoDocumento>N380</codigoDocumento>
        <nomeFicheiro> documento_digitalizado1.pdf </nomeFicheiro>
        <codigoResultado>200</codigoResultado>
        <descricaoResultado>Documento submetido com sucesso.</descricaoResultado>
      </documento>
      <documento>
        <codigoDocumento>N750</codigoDocumento>
        <nomeFicheiro> documento_digitalizado2.doc </nomeFicheiro>
        <codigoResultado>201</codigoResultado>
        <descricaoResultado>Formato do documento não suportado.</descricaoResultado>
      </documento>
      <dataEnvio>2020-12-12 19:30</dataEnvio>
    </return>
  </m:entregarDocumentos>
</env:Body>
```

5. Códigos de Resultado

A cada invocação, o sistema poderá responder com as mensagens abaixo descritas:

Código	Descrição
Códigos de Resultado genéricos	
0	Mensagem processada com sucesso. / Mensagem submetida com sucesso.
1	Ultrapassado o limite máximo de documentos por pedido.
2	Mensagem inválida. A estrutura da mensagem não corresponde à Mensagem ICAUM001A ou ICAUM007A.
900	Ocorreu um erro interno. Por favor contacte a AT indicando o identificador: <identificador>.
901	Estrutura do pedido inválida.
Códigos de Resultado específicos – obter PDFs/Mensagens	
100	Dados encontrados.
101	Não foram encontrados resultados para a operação pretendida.
102	Número de Referência Local inexistente.
Códigos de Resultado específicos – enviar Documentos	
200	Documento submetido com sucesso.
201	Formato do documento não suportado.
202	Tamanho máximo ultrapassado.
203	Número da Adição inexistente.
204	Código de documento inválido.

6. Tipos de PDF/Mensagem

Nos serviços *obterDocumentosPDFNaoEntregues*, *obterDocumentosPDFEntregues*, *obterMensagensNaoEntregues* e *obterMensagensEntregues*, pode ser especificado o tipo de PDF ou mensagem que se pretende obter. A lista de valores possíveis encontra-se abaixo descrita:

Código	Descrição
VAL	Devolve os PDFs/mensagens geradas durante a fase de Validação da declaração.
ACE	Devolve os PDFs/mensagens geradas na Aceitação da declaração.
CTR	Devolve os PDFs/mensagens geradas no âmbito de um processo de Controlo da declaração.
RLQ	Devolve os PDFs/mensagens associadas ao registo e notificação da dívida.
AUT	Devolve os PDFs/mensagens geradas no âmbito da Autorização de Saída.
ALT	Devolve os PDFs/mensagens geradas no âmbito de um processo de alteração da declaração.
ANU	Devolve os PDFs/mensagens geradas no âmbito de um processo de anulação da declaração.

7. Assinatura certificado SSL (CSR)

A invocação dos serviços web pressupõe um processo de autenticação mediante a validação da chave privada da aplicação, do conhecimento exclusivo da entidade aderente, sendo a respetiva chave pública comunicada e assinada pela AT.

O certificado a ser utilizado na operação é assinado pela AT, a pedido da entidade aderente.

Para este efeito, a entidade aderente deve efetuar um pedido de certificado (CSR – *Certificate Signing Request*).

O CSR é um pequeno ficheiro de texto cifrado que contém o certificado SSL e toda a informação necessária para que a AT possa assinar e devolver o certificado assinado digitalmente, para que possa ser utilizado no processo de autenticação na invocação do serviço web.

Os procedimentos para geração do CSR são simples, mas variam de acordo com a tecnologia web utilizada pela entidade aderente, razão pela qual devem ser consultados os respetivos manuais de apoio de cada ferramenta.

A informação que o CSR deve conter não pode ultrapassar os tamanhos máximos, conforme a descrição seguinte:

Campo CSR	Descrição	Tamanho Máximo
C = Country	O código ISO de 2 letras referente ao local da sede. Por exemplo, no caso de Portugal é “PT”.	2 (chars)
ST = Province, Region, County or State	Distrito da sede.	32 (chars)
L = Town/City	Local da sede.	32 (chars)
CN = Common Name	Neste campo deve ser indicado o número de identificação fiscal da entidade aderente.	9 (chars)
O = Business Name / Organisation	Designação legal da empresa.	180 (chars)
OU = Department Name / Organisational Unit	Departamento para contacto.	180 (chars)
E = An email address	O endereço de correio eletrónico para contacto, geralmente do responsável pela emissão do CSR ou do departamento de informática. Tem que ser um endereço de email válido.	80 (chars)
Key bit length	Chave pública do certificado SSL tem de ser gerada com 2048 bits.	2048 (bits)

A utilização de caracteres especiais (e.g., portugueses, línguas latinas, etc.) não é aceite em nenhum dos campos acima indicados, uma vez que a utilização desses caracteres vai invalidar a assinatura digital do certificado SSL.

Como resultado deste processo, a AT procederá à assinatura do certificado e remete em resposta ao pedido o certificado assinado para integração na chave privada da entidade aderente.

O certificado terá a validade de 12 meses a contar da data da assinatura.

7.1. Gerar um certificado SSL

Um certificado SSL é uma chave RSA composta por duas partes: chave privada e chave pública.

Como a chave privada deve ser apenas do conhecimento da entidade aderente, a emissão da mesma tem sempre de ser efetuada pelo próprio, em computador próprio, e nunca num *site* ou serviço web que encontre para o efeito.

Existem diversas ferramentas para geração de certificados SSL, proprietárias e Opensource. A AT utiliza a ferramenta OpenSSL, que é a ferramenta Opensource de referência, livre de custos de utilização.

Para gerar um certificado SSL, cada entidade aderente deve fazê-lo no seu próprio computador, utilizando o seguinte comando:

```
openssl req -new -subj "/C=PT/ST=Distrito da Sede/L=Local da Sede/O=Empresa /OU=Departamento de Informatica/CN=555555555/emailAddress=informatica@empresa.pt" -newkey rsa:2048 -nodes -out 555555555.csr -keyout 555555555.key
```

Cada entidade aderente deve substituir a informação específica no comando anterior pelos seus dados, uma vez que os apresentados são apenas exemplificativos, e não deve alterar a informação indicada a **Bold**.

Como resultado, do comando anterior será gerado o certificado SSL e serão produzidos dois ficheiros:

- 555555555.csr – Ficheiro com o pedido CSR a enviar à AT;
- 555555555.key – Ficheiro com a chave privada gerada.

7.2. Verificar conteúdo do CSR gerado

Antes de enviar o CSR para assinatura digita pela AT, pode e deve ser verificado o conteúdo do ficheiro para garantir que toda a informação está como pretendido. Para tal, deve ser usado o seguinte comando:

```
openssl req -text -noout -in 555555555.csr
```

Onde cada entidade aderente deve substituir os parâmetros que não estão a **Bold** pelos nomes dos ficheiros corretos.

7.3. Integrar certificado com a chave privada

Depois de receber o certificado SSL assinado pela chave digital da AT, é necessário integrar esse certificado com a chave privada gerada no passo anterior (555555555.key). Para tal, deve ser usado o seguinte comando:

```
openssl pkcs12 -export -in 555555555.crt -inkey 555555555.key -out 555555555.pfx
```

Onde cada entidade aderente deve substituir os parâmetros que não estão a **Bold** pelos nomes dos ficheiros corretos.

Como resultado, o certificado SSL assinado pela AT é integrado com a chave privada e gravada com uma *password* de acesso que cada entidade aderente deve definir na execução do comando.

8. Endereços Úteis

8.1. Página de produtores de software

<https://www.portaldasfinancas.gov.pt/pt/external/factemipf/painellInicialProdSoftware.action>

8.2. Gestão de subutilizadores no Portal das Finanças

<https://www.acesso.gov.pt/gestaoDeUtilizadores/consulta?partID=PFIN>

8.3. WSDL do envio de dados à AT por Webservice

<https://servicos.portaldasfinancas.gov.pt:806>

8.4. Endereços para envio de dados à AT por Webservice

Ambiente de testes:

<https://servicos.portaldasfinancas.gov.pt:806>

Ambiente de produção:

A disponibilizar.