Manual de Integração AverbePorto
Porto Seguro
Manual de Envio de XML para Averbação

Descritivo do sistema AverbePorto

O sistema de Averbação Eletrônica da Porto Seguro, o AverbePorto, recebe os arquivos **XML** de MDF-e (58), CT-e (57), Minuta de CT-e (94), NF-e (55), NFC-e (65) e seus XMLs de evento de cancelamento, para automatizar o processo de Averbação de Seguros.

Os arquivos XML podem ser enviados manualmente via Interface Web ou automaticamente, via E-mail ou API (Webservice REST). Em todas as modalidades, os arquivos XML podem ser enviados 1 à 1 ou em lote, compactados em um arquivo ZIP.

Nas telas de **Busca de Arquivos/Exportar para relatório** ou **Relatório/Registro de Envios** é possível consultar o **Número de Averbação ANTT** para os documentos guardados. No tópico Consulta Número de Averbação ANTT, estão descritas as formas sistêmicas de consulta.

Nota - 04/2022
Aos usuários que já utilizam a API, observar que o endpoint correto deve iniciar com **apis** ou **api** - Por motivos de segurança, o acesso via **www** ou **wws** será desativado em breve.

Não utilize o usuário e senha de acesso web para acessar a API, pois o acesso à API com ela será desabilitado no futuro. Gere suas Credenciais de API no módulo **Cadastro do Usuário**.

Para aqueles que fazem averbação via e-mail para CNPJ@averbeporto.com.br, não enviar como Cópia Oculta **Cco (Bcc)**.

Sumário

Controle de Versões da API 2
Manual de Integração Online 2
Averbação via Interface Web 3
XML 3
ZIP 3
Averbação via API (Web Service tipo REST) 4
Averbação via E-mail 9
Consulta Número de Averbação ANTT 10
    Acessando a API (Web Service REST) 10
    Acessando a Página/URL de consulta 10
Parâmetros Adicionais de Averbação 12
Acesso via Postman 15
Sobre as senhas do AverbePorto 16
Exemplo em PHP (cURL) 17
Exemplo em Delphi (Indy) 21

Controle de Versões da API

Versão	Produção	Mandatório	Descrição
1.0	29/01/2015	--	- Versão inicial
2.0	29/08/2017	01/10/2017	- Versionamento de API (v=2) - Número de Averbação ANTT (MDF-e 3.0)

Manual de Integração Online

Google Docs: https://goo.gl/FWAvBE

Download: https://www.averbeporto.com.br/proxy/manual.php?format=pdf

- Formatos alternativos para download: docx, odt, rtf, pdf, txt, zip (html), epub, md

Averbação via Interface Web

1. Com seu navegador web, acessar o endereço https://www.averbeporto.com.br.
2. No canto superior direito, **faça login** com seu usuário e senha web.
3. Se exibido o captcha, responda-o (casos de excesso de tentativas de logins falhos).
4. No menu à esquerda, clicar em **Envio de Arquivo**.
5. Na tela, selecione seu arquivo XML ou ZIP e clique em **Enviar e Guardar**.
6. Nas telas de **Busca de Arquivos/Exportar para relatório** ou **Relatório/Registro de Envios** é possível consultar o **Número de Averbação ANTT** para os documentos guardados. [Sobre outras formas de consulta].

Para mais informações sobre o sistema web, consulte a **Ajuda** no menu à esquerda.

Os arquivos XML e ZIP devem seguir o padrão de codificação para tipos de mídias oficiais:

XML

- application/xml
- text/xml

ZIP

- application/zip
- Limite de 400 XML (acima disso serão ignorados).

Averbação via API (Web Service tipo REST)

Resumo: Acessar a API (RPC) via método POST para **Login** (mesmo Usuário do Sistema Web - que deve ser alterada caso tenha menos de 8 dígitos; Senha da API deve ser gerada no módulo **Cadastro do Usuário**) e receber o **JSON** de resposta (também contendo "ses") e o **cookie de sessão** (**portal[ses]** - HEADER HTTP - deve ser utilizado em múltiplos envios), acessar novamente para **Enviar o arquivo** e receber o **JSON** de resposta com o status e o Número de Averbação ANTT. O Sistema Web acessa esta mesma API, ou seja, todas as requisições e respostas podem ser observadas via Ferramenta do Desenvolvedor em seu Navegador Web.

1. Endereços da API *[1]:
   Seguro: https://apis.averbeporto.com.br/php/conn.php (:443 - TLS 1.0, 1.1, 1.2 e 1.3) *[2] - HTTP/3
   ~~**Plain**: http://api.averbeporto.com.br/php/conn.php (:80 - Plain text - HTTP/2)~~ - descontinuado

   *[1] NÃO utilizar **www**. ou **wws**. ou nenhum outro subdomínio para acesso à **API**, será bloqueado.
   *[2] É necessário definir um user-agent como "*Mozilla/5.0 Sistema SeuNome/1.0.*" para acessar as apis, isso impedirá que o proxy bloqueie seu programa com o erro 403/1010 ou semelhante. Cuidado com nomes proibidos: bad user_agents.regex.txt
   *[3] Chamar o endereço da API no navegador irá gerar uma mensagem de erro (item 7) pela absoluta falta de parâmetros na comunicação.

2. Para **login**, POST (application/x-www-form-urlencoded) os parâmetros:

```
{
        "mod": "login",
        "comp": 5,
        "user": "USUARIO_Credencial_do_Web_Service_API_60_caracteres",
        "pass": "SENHA_Credencial_do_Web_Service_API_64_caracteres",
        ["dump": [1,2]]    // Opcional.
}
```

   - dump: Pode ser utilizado em qualquer requisição à API para auxiliar na depuração. Exibe um "dump" de como estão chegando as variáveis enviadas ao sistema pelo usuário.
   dump=1: Adiciona a tag "dump" no JSON de resposta, após todo o processamento normal da requisição.
   dump=2: Exibe o JSON de resposta apenas com a tag "dump", antes de qualquer processamento, e aborta o processamento normal.

   Exemplo de conteúdo do dump::


```
{       ...
        "dump":{
        "userName":"12345678901234", "logon":"2024-09-25 15:34:09",
        "last_uploads":{"drn_d":"4","drn_h":"2","drn_m":"2","drn_s":"1","p_d":"0","p_h":
"0","p_m":"0","p_s":"0","tot_d":4,"tot_h":2,"tot_m":2,"tot_s":1},
        "POST":{"comp":"5", "mod":"Upload", "path":"\/www\/www\/eguarda\/php\/",
"dump":"1", "v":2, "ext":".php"},
        "COOKIE":{"portal":{"ses":"7af39d9fcb0e82d48036ef02891c34ae"}},
        "FILES":{"file":{"name":"AverbePorto_CTe352409123456789012345799900000009421955
741985.xml", "type":"text\/xml", "tmp_name":"\/tmp\/phpR3k8bk", "error":0, "size":2112}}
        }
}
```


   - last_uploads: Estatísticas de envio. Acima de 10 envios por segundo e abusos no envio de XML pré-existente geram um aumento progressivo de delay na conexão de até 3 segundos (critérios sujeitos a alterações). **drn_**: duplicado, rejeitado ou negado; **p_**: processado ok; **_dhms**: dia, hr, min, seg.
   - pass: Não utilize o usuário e senha web no acesso à API. Acessando o sistema Web, vá no módulo **Cadastro do Usuário** e gere suas credenciais de API, clicando no **X** para iniciar o processo. (Instruções Detalhadas)

3. JSON de falha no login, usuário ou senha inválidos, enviados de maneira incorreta, ou sessão de login expirada:


```
{
        "success": 1,
        "logout": 1
}
```


4. JSON de falha na autenticação, no login devido a ausência de usuário ou senha, em outros módulos devido a ausência do cookie de sessão (portal[ses]).


```
{
        "success": 1,
        "error": {
                "code": "01",
                "msg": "No login."
        }
}
```


5. JSON de restrição por captcha (casos de excesso de tentativas de logins falhos com o usuário):


```
{
        "success": 0,
        "error": {
          "code": "02",
          "msg": "Captcha required"
        },
        "captcha_url": "https://www.averbeporto.com.br",
        "captcha_html": "<!DOCTYPE html>\n<html>\n\n<head>\n
<title>Verificação</title>\n    <script src=\"https://www.google.com/recaptcha/api.js\" async
defer></script>\n</head>\n\n<body>\n    <form action=\"php/conn.php\"
method=\"POST\">\n        <div class=\"g-recaptcha\"
data-sitekey=\"6LfH6f8qAAAAAOM7Yz22LdkGnZhPT_nJiMmDv09q\"
data-action=\"LOGIN\"></div>\n        <br>\n        <input type=\"hidden\" name=\"user\"
value=\"USUARIO\">\n        <input type=\"hidden\" name=\"redirect_url\" value=\"\">\n
<input type=\"submit\" value=\"Enviar\">\n    </form>\n</body>\n\n</html>"
}
```


* O captcha é exigido por motivos de segurança quando um excesso de tentativas de login falhos é registrado para o usuário.
* Sua TI deve verificar imediatamente se as credenciais estão sendo utilizadas de maneira incorreta. Ou, se as credenciais foram vazadas, tanto a senha web quanto as credenciais de API devem ser trocadas imediatamente.
* O HTML com o captcha pode ser exibido dentro do sistema TMS para ser resolvido por um humano, ou o segurado pode ser direcionado para a URL do site para fazer login e resolver o captcha para desbloquear o acesso do usuário temporariamente.

6. JSON de login bem sucedido:


```
{
        "success": 1,
        "C": {
                "id": "00",
                "userName": "USUARIO",
                "name": "Usuario",
                "email": "usuario@dominio.com",
                "portal_groups_id": "00",
                "type": "U"
        },
        "S":[...]
}
```

7. Para **upload**, fazer o POST (multipart/form-data) do cookie [1] de sessão (**portal[ses]** - HEADER HTTP - recebido no login), do arquivo (**file**) [2] e dos parâmetros:

```
{
        "comp": 5,
        "mod": "Upload",
        "path": "eguarda/php/",
        "recipient": "",
        ["v": N]  //N = Versão da API. Parâmetro opcional.
}
```

Obs 1. Os COOKIES residem no cabeçalho da resposta HTTP e não no corpo como o JSON.
Obs 2. Enviar o arquivo junto aos parâmetros, mesmo como um parâmetro codificado em base64, não irá funcionar. O arquivo deve ser enviado da mesma maneira que um formulário HTML o faria, em multipart.
Obs 3. Quando geradas as Credenciais de API no módulo Cadastro do Usuário, o cookie de sessão passa a ter validade de 1 semana, ao contrário de 1 dia para as credenciais web comuns.

8. Opções do parâmetro recipient (opcional, utilizar somente se assim indicado): Parâmetro que indica o tipo do(s) arquivo(s) sendo enviado(s). Necessário apenas nos casos indicados pela Seguradora.

   Vazio = Automático (Recomendado na grande maioria dos casos)
   E = Embarcador Emitente (Em desuso)
   F = Fornecedor (Em desuso)
   T = Transportador (Em desuso)
   D = Duplo Ramo (As NF-e são averbadas 2 vezes, uma como T e outra como E)

9. JSON de retorno:

```
{
  "success": 1,
  "S": [
       "P": 1, // Processado (xml guardado com sucesso)
       "D": 0, // Duplicado (xml pré-existente)
       "R": 0, // Rejeitado (xml não parece ser do tipo certo)
       "N": 0 // Negado (Não é xml ou zip)
  ],
  "prot": "1234567890123", // Protocolos do XML guardados P
  "prot": ["1234567890123"], // Protocolos dos XMLs (ZIP) guardados P
  "error": [ // Mensagens de erro para os resultados R ou N
       "code": 00,
       "msg": "Mensagem de erro"
   ]
}
```

10. JSON de erro genérico, significa “Erro desconhecido” e, geralmente, é resultado de uma requisição mal formada ou efetuada via um método diferente de POST:

```
{
        "success": 0,
        "error": {
                "msg": "Ghs.loc.error"
        }
}
```

11. Note que, no JSON de resposta, um **“success”: 1** se refere ao sucesso na comunicação, não necessariamente a efetivação da requisição.

12. Em caso de falha no arquivo ZIP, deve-se consultar o **Relatório de Envios** no sistema web para verificar os arquivos que foram guardados.

13. Nas telas de **Busca de Arquivos/Exportar para relatório** ou **Relatório/Registro de Envios** é possível consultar o **Número de Averbação ANTT (Protocolo de Envio)** para os documentos guardados. [Sobre outras formas de consulta].

- O envio tem as mesmas características de um POST de formulário HTML (mídia **application/x-www-form-urlencoded** ou **multipart/form-data**).

- Exemplos CURL
  Dicas: -i mostra o response header; **-v** mostra toda a comunicação; **--tls-max 1.2** limita a versão do tls

  Login:
  curl 'https://apis.averbeporto.com.br/php/conn.php' --compressed -c cookie.txt -X POST -H 'Content-Type: application/x-www-form-urlencoded; charset=UTF-8' --data 'mod=login&comp=5&user=USUARIO&pass=SENHA'

  Upload:
  curl 'https://apis.averbeporto.com.br/php/conn.php' --compressed -X POST -H 'Cookie: portal[ses]=8c878b3ab2ba27bf7bc4d5d448b6489b' -H 'Content-Type: multipart/form-data' -F mod=Upload -F comp=5 -F path=eguarda/php/ -F file=@ARQUIVO.xml

  ps:
  USUARIO e SENHA devem ser substituídos pelo usuário e senha do Segurado (usuário e senha próprio para acesso à API do Sistema Web).
  O Cookie deve ser substituído pelo cookie retornado no login.
  O ARQUIVO.xml deve ser substituído por um documento XML (corretamente formatado) do Segurado.

Averbação via E-mail

O envio via E-mail tem limitações técnicas inerentes, tanto na velocidade que é possível enviar XMLs quanto na leitura deles pelo sistema AverbePorto. Se o seu sistema for enviar centenas de XMLs por dia, ou tem necessidade de receber o protocolo ANTT imediatamente após o envio, esse método não é recomendado e, nesse caso, **o melhor é realizar a integração via API**.

1. No caso de aprovação da Porto Seguro em aceitar as averbações via e-mail, uma conta de e-mail no formato CNPJ@averbeporto.com.br será disponibilizada.

2. Essa conta deve ser usada como destino apenas de mensagens com anexos de documentos eletrônicos no formato XML.

3. Para que o sistema seja capaz de averbar esse arquivo XML, ele deve estar anexo diretamente na mensagem ou anexo em formato ZIP (XML dentro do ZIP - ZIP dentro de ZIP será rejeitado, assim como qualquer outro formato diferente destes).

4. Limite de 400 XML (acima disso serão ignorados).

5. Nas telas de **Busca de Arquivos/Exportar para relatório** ou **Relatório/Registro de Envios** é possível consultar o **Número de Averbação ANTT (Protocolo de Envio)** para os documentos guardados. [Sobre outras formas de consulta].

6. **IMPORTANTE**: Ao enviar mensagem para para sua conta de e-mail CNPJ@averbeporto.com.br, não enviar como **Cópia Oculta Cco (Bcc)**. O destinatário deve, obrigatoriamente, estar em **Para (To)** ou **Com Cópia (Cc)**.

Obs. Caso já esteja utilizando a averbação via e-mail e integre seu sistema via API posteriormente, solicitamos que os envios via e-mail sejam imediatamente interrompidos, pois o envio dos XML em duplicidade pode causar inconvenientes em suas averbações.

Consulta Número de Averbação ANTT

Resumo: Acessar a API via método POST (assim como no módulo Upload) ou acessar a URL via método GET e receber a resposta da consulta.

O "Número de Averbação ANTT" (Protocolo de Envio), advindo de um CT-e ou NF-e, deverá necessariamente ser utilizado para a composição de um MDF-e que os carregue (a partir da versão 3.00 do MDF-e, de acordo com resolução ANTT 4799/2015).

Acessando a API (Web Service REST)

1. Endereço da API: https://apis.averbeporto.com.br/php/conn.php

2. Para **consulta**, fazer o post do cookie de sessão (**portal[ses]** recebido no login), e dos parâmetros:
   comp=5&mod=Protocolo&path=atwe/php/&chave[]=12345678901234567890123456789012345678901234&chave[]=22345678901234567890123456789012345678901234

   Neste exemplo estão sendo pesquisadas 2 chaves. Para realizar a pesquisa inversa, basta passar os protocolos nos parâmetros **protocolo[]=12345678901234567890123456789012345678901234567890** e omitir os parâmetros **chave[]**.

3. Parâmetros opcionais:

   out = "json", "xml" ou "csv" // Formato do retorno. Padrão é JSON (Na forma padrão da API)
   download = 0 ou 1 // Cabeçalho do retorno como display ou download para arquivo. Padrão é 0 (display)
   delim = , // Parâmetro para out=csv. Indica o delimitador de campo do CSV. Padrão é vírgula (,)

4. JSON de retorno:

```
{
        "success": 1,
        "S": [{
                "chave": "12345678901234567890123456789012345678901234",
                "protocolo": "12345678901234567890123456789012345678901234567890"
        }, {
                "chave": "22345678901234567890123456789012345678901234",
                "protocolo": "22345678901234567890123456789012345678901234567890"
        }]
}
```

Acessando a Página/URL de consulta

1. Página de consulta (**standalone**) - Gera URL GET:
   https://www.averbeporto.com.br/atwe/protocolo.html

2. Exemplo de acesso via GET:
   https://apis.averbeporto.com.br/atwe/php/Protocolo.php?out=json&download=0&chave[]=12345678901234567890123456789012345678901234

   Parâmetros opcionais:

   out = "json", "xml" ou "csv" // Formato do retorno. Padrão é JSON (Na forma padrão da API)
   download = 0 ou 1 // Cabeçalho do retorno como display ou download para arquivo. Padrão é 0 (display)
   delim = , // Parâmetro para out=csv. Indica o delimitador de campo do CSV. Padrão é vírgula (,)

3. Exemplo de retorno CSV:

   Chave,Protocolo
   12345678901234567890123456789012345678901234,123456789012345678901234567890123456789012345678901234567890

4. Exemplo de retorno XML:

```
<data>
 <item0>
  <chave>12345678901234567890123456789012345678901234</chave>
  <protocolo>123456789012345678901234567890123456789012345678901234567890</protocolo>
 </item0>
</data>
```

Parâmetros Adicionais de Averbação

1. Campo Texto de Observações

O Sistema pode também ler **informações adicionais de averbação** passadas de maneira especial na tag de **observações (xObs ou infCpl)** dos documentos eletrônicos.

Para **inserir informações adicionais** de averbação no campo de **Observações** de seus documentos eletrônicos, use o **Gerador de Parâmetros de Averbação** em https://www.averbeporto.com.br/atwe/obs.html.

Após gerar o código de informações, **basta copiá-lo e colá-lo** em seu sistema TMS, no campo de observações do documento eletrônico.

Exemplo de utilização das Observações:

{mudanca:"s",vcontainer:"1.23",viagem:"1",placa:"ABC1234",dtviagem:"2015-12-31"}

a. O JSON pode ou não ter aspas duplas nos nomes dos campos que podem ser maiúsculos ou minúsculos. Já os valores precisam ter aspas duplas, podendo não ter apenas em caso do valor ser número (int ou float).
b. Apenas 1 JSON por xObs, o primeiro. Outros serão ignorados.
c. Informações de parâmetros da obs.html no cabeçalho do código fonte da página obs.html.

2. Tags do Grupo ObsCont

Caso o Segurado prefira integrar em seu sistema TMS a geração automática dos Parâmetros Adicionais, estes devem ser inseridos no XML como tags adicionais do grupo ObsCont (ou ObsFisco) em compl.

ATENÇÃO: Caso existam parâmetros no xObs, todos os ObsCont serão ignorados.

Página 159 do manual do CT-e v3.00.
Página 225 do manual do NF-e v6.00.

Os atributos **xCampo** do XML levam os mesmos nomes dos campos adicionais que vão nas observações (acima), assim como as tags **xTexto** aceitam os valores no mesmo formato.

Exemplo de utilização no CT-e:

```
<compl>
  <ObsCont xCampo="dtviagem">
    <xTexto>2015-12-31</xTexto>
  </ObsCont>
  <ObsCont xCampo="placa">
    <xTexto>ABC1234</xTexto>
  </ObsCont>
</compl>
```

3. Tabela de parâmetros

Descrição	Nome xcampo	Caracteres	Ocorrência	Informações
Sem valor declarado de transporte aéreo	semvalaereo	1	*: 0-1	s- sim; n- não;
Operação de ocd	ocd	1	*: 0-1	1- OCD; 2- OCDI; 3- OCD Especial; 4- OCDI Especial;
Operação de redespacho - Início	cmunini	1-n	*: 0-1	Código IBGE de municípios
Operação de redespacho - Fim	cmunfim	1-n	*: 0-1	Código IBGE de municípios
Unidade Federativa da fronteira	fronteira	2	*: 0-1	UF - 2 letras
Valor de benefícios internos	vbenefint	1-11	*: 0-1	Ponto decimal (.)
Número da viagem	viagem	1-9	*: 0-1	Número inteiro
Placa do veículo	placa	7	*: 0-1	ABC1234 (3 letras e 4 números)
Data da viagem	dtviagem	10	*: 0-1	aaaa-mm-dd
S. Embarcador	embarcador	1	Transp: 0-1	s- sim; n- não;
Transporte de mudanças	mudanca	1	Transp: 0-1	s- sim; n- não;
Operação de subcontratação	tpserv	1	Transp: 0-1	Número inteiro (Vide manual CT-e Sefaz)
Valor de vasilharme	vvasilhame	1-11	Transp: 0-1	Ponto decimal (.)
Valor de container	vcontainer	1-11	Transp: 0-1	Ponto decimal (.)
Cobertura de apólice	cobertura	3	Emb: 0-1	Número inteiro
Coberturas adicionais	adicionais	3-n	Emb: 0-1	Alfa-numéricos separados por vírgula (,)
Cobre NF de entrada	cobnfentrada	1	Emb: 0-1	s- sim; n- não;
Valor do lucro esperado	lucro	1-11	Emb: 0-1	Ponto decimal (.)
Valor de beneficiamento	vbenef	1-11	Emb: 0-1	Ponto decimal (.)
Valor de despesas	vdespesas	1-11	Emb: 0-1	Ponto decimal (.)
Modalidade do frete	modfrete	1	Emb: 0-1	Número inteiro

Acesso via Postman

- Download do aplicativo para PC: https://www.getpostman.com/apps

- Documentação Postman da API AverbePorto (**recomendado**):

1. Descrição da API: https://documenter.getpostman.com/view/207913/RWgozeRg
   Execute clicando no botão laranja "Run in Postman" ou
2. **Collection:** Importe esta Collection pronta (Import > Link ou File) no aplicativo (Vídeo: https://youtu.be/M-qHvBhULes)
3. Na aba Collection do aplicativo, irá aparecer uma nova coleção chamada "www.AverbePorto.com.br"

- Gerar **exemplos** para outras linguagens na aplicação:

1. Clicar em "Code" (abaixo dos botões Send e Save)
2. Selecionar a linguagem no combo localizado no topo da janela "Generate Code Snippets"

- Passos para configuração manual (* é recomendado utilizar a collection pronta acima):

- **URL**

  POST: https://apis.averbeporto.com.br/php/conn.php
  - Mais sobre a URL em Manual de Integração - AverbePorto

- **Login**
  Passar os parâmetros de login para efetuá-lo e receber o Cookie de Sessão. O Postman utilizará o cookie automaticamente para a mesma URL.
  - Não utilize o usuário e senha comum para acesso web do usuário para acessar a API. Acessando o sistema Web, vá no módulo **Cadastro do Usuário** e gere suas credenciais de API, clicando no **X** para iniciar o processo. (Instruções Detalhadas)

Body (x-www-form-urlencoded):
mod: login
comp: 5
user: SEU_USUARIO
pass: SUA_SENHA

- **Envio de arquivo**
  Upload do arquivo XML ou ZIP.

Body (form-data):
mod: Upload
comp: 5
path: eguarda/php/
file: "Selecionar arquivo xml ou zip"

Headers:
"Deixar completamente vazio" (inclusive remover o Content-type)

Sobre as senhas do AverbePorto

- A senha comum do usuário, usada para fazer login no site do sistema AverbePorto, não deve ser mantida a senha inicial enviada na criação do usuário. Essa senha deve ser alterada para uma nova, escolhida pelo usuário, porém de acordo com as regras de segurança indicadas no campo de senha quando ela estiver sendo digitada. A alteração da senha comum pode ser efetuada no módulo **Cadastro de Usuário** no espaço **Alterar senha**.
- O usuário e senha fortes de API não são escolhidos pelo usuário, serão gerados automaticamente e devem ser anotados pois não serão exibidos novamente.
- Se for necessário ver a credencial novamente, outra deve ser gerada em seu lugar, e será exibida apenas uma vez, no momento da geração. Sendo assim, a antiga para de funcionar imediatamente e a nova credencial gerada passa a ser a funcional. Então, tenha certeza de que, se as credenciais de API forem re-generadas, elas serão imediatamente utilizadas em seus sistemas, pois as antigas deixam de funcionar no ato.
- A posterior alteração da senha comum não influencia as credenciais de API, que se mantêm funcionando.
- O token de login da senha web tem validade de 1 dia, o token de login da senha de API tem validade de 1 semana. Ou seja, o mesmo login pode interagir com o sistema por esse período, após, o login deve ser efetuado novamente.

Como gerar as Credenciais de API

1. Após trocar a senha web e refazer login no site utilizando essa nova senha, a geração das credenciais de API pode ser efetuada no módulo **Cadastro de Usuário** no espaço **Credencial do Web Service API**.

2. Para iniciar o processo, clique no botão "**X**" à direita do campo **Senha**. Esse botão inicia o processo para apagar qualquer senha previamente gerada e então gerar uma nova.
   Esse botão é um "X" para inibir uso acidental ou aleatório. Ele deve ser usado apenas por pessoal de T.I. e que entende o que está fazendo.

3. Na janela que se abrir, para confirmação do procedimento, entre com a senha web comum do usuário. A mesma senha utilizada para login no site do sistema, e clique em **OK**.

4. Pronto, as novas credenciais de API geradas estarão dentro dos campos (usuário e senha - 60 e 64 caracteres). Agora, copie e cole essas credenciais em um lugar seguro para usá-las em seus sistemas, elas não serão exibidas novamente.

Obs. As credenciais são guardadas criptografadas. As credenciais de API não poderão ser visualizadas por ninguém, se necessário elas podem apenas ser alteradas, repetindo os mesmos passos para gerá-las.

Exemplo em PHP (cURL)

```php
<?php
/**
* Open an url on https using curl and return content
* @param string url          The url to open
* @param string refer        Referer (optional)
* @param mixed usecookie     If true, cookie.txt    will be used as default, or the usecookie value.
* @return string
*/
function open_https_url($url,$refer = "", $usecookie = false) {
   if ($usecookie) {

      if (file_exists($usecookie)) {

         if (!is_writable($usecookie)) {

            return "Can't write to $usecookie cookie file, change file permission to 777 or remove read only for windows.";
         }
      } else {
         $usecookie = ($usecookie === true)? "cookie.txt" : $usecookie;


         if (!touch($usecookie)) {

            return "Can't write to $usecookie cookie file, change file permission to 777 or remove read only for windows.";
         }
      }

   }


   $ch = curl_init();
   curl_setopt($ch, CURLOPT_URL, $url);
   curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
   curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);
   curl_setopt($ch, CURLOPT_HEADER, 0);
   curl_setopt($ch, CURLOPT_USERAGENT, "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)");

   if ($usecookie) {
      curl_setopt($ch, CURLOPT_COOKIEJAR, $usecookie);

      curl_setopt($ch, CURLOPT_COOKIEFILE, $usecookie);
   }

   if ($refer != "") {
```

```
        curl_setopt($ch, CURLOPT_REFERER, $refer );

    }

    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    return $ch;
}

/**
 * Limpeza ao terminar de executar
 * Requer $ws
 */
function ws_shutdown(){
        global $ws;

        if (file_exists($ws['cookie'])) {
                unlink($ws['cookie']);
        }
}
register_shutdown_function('ws_shutdown');


/**
 * Ajax Request

 * Requer $ws Global Config (comp/path/cookie)
 * aPost (array) json params
 * sModule (string) mod (i.e. login/Upload/Retrieve)
 * $sConn (string) URI to connect
 */
function websysRequest($aPost, $sModule = 'login', $sConn =
'https://apis.averbeporto.com.br/php/conn.php') {
        global $ws;

        if (!isset($aPost['comp'])) { $aPost['comp'] = $ws['comp']; }
        if (!isset($aPost['path'])) { $aPost['path'] = $ws['path']; } elseif ($aPost['path'] == '') {
                unset($aPost['path']); }
        $aPost['mod'] = $sModule;

        $ch = open_https_url($sConn, '', $ws['cookie']);

        curl_setopt($ch, CURLOPT_POST, 1);
        curl_setopt($ch, CURLOPT_POSTFIELDS, $aPost);

        $res = curl_exec($ch);
        curl_close($ch);

        return $res;
}
?>
```

Exemplo de Uso:

```php
<?php
require_once('func.php');

// Exemplo Config $ws
$ws = array(
	'comp' => 5,
	'path' => 'eguarda/php/',
	'conn' => 'https://apis.averbeporto.com.br/php/conn.php',
	'cookie' => tempnam(sys_get_temp_dir(), 'ws_'),
	'logged' => ''
);

/**
* Envia arquivo
*
* @param {String} Conteudo do arquivo
* @param {Array} Usuario e senha. Ex.: array('user'=>'USUARIO', 'pass'=>'SENHA', 'path'=>'')
* @param {String} (optional) Remetente (em caso de email)
* @return {Array} Retorna resposta do webservice
*/
function sendFile($sFileContent, $aUser, $sRecipient = ''){
	global $ws;
	$file = tmpfile();
	fwrite($file, $sFileContent);
	rewind($file);
	$meta = stream_get_meta_data($file);
$mime = mime_content_type($meta['uri']);

	$post = array(
	'file' => (version_compare(PHP_VERSION, '5.5') >= 0)? new CURLFile($meta['uri'], $mime) :
'@'.$meta['uri'].';type='.$mime
	);

	if ($sRecipient) {
		$post['recipient'] = $sRecipient;
	}

	// Login
	if ($ws['logged'] != $aUser['user']) {
		$res = json_decode(websysRequest($aUser), true);
		if (isset($res['logout']) && $res['logout']) {
			//ws_log('MAIL2EG: ['.$aUser['user'].']: '.posix_getpid().': Falha do login. ');
		}
	} else {
		$res['success'] = $res['C'] = true;
	}
```

```
		// Upload
		if ($res['success'] && isset($res['C'])) {
			$ws['logged'] = $aUser['user'];
			$res = json_decode(websysRequest($post, 'Upload'), true);
		}

		fclose($file);
		return $res;
}


$aUser = array(
	'user' => 'USUARIO',
	'pass' => 'SENHA',
	'path' => ''
);
$sFileContent = file_get_contents('ARQUIVO.xml');

$res = sendFile($sFileContent, $aUser);

print_r($res);
?>
```

Exemplo em Delphi (Indy)

```
unit PortoSeguroIndy;

interface
uses
  System.SysUtils, System.Variants, System.Classes,
  IdHTTP, IdMultipartFormData, Vcl.Dialogs, IDAntiFreeze,
  IdCookieManager, IdURI, Data.DBXJSON, System.JSON,
  IdIOHandler, IdIOHandlerSocket, IdIOHandlerStack, IdSSL, IdSSLOpenSSL;
type
  TPortoSeguroIndy = class
  private
    FPostCookieStream: TIdMultiPartFormDataStream;
    FPostFileStream: TIdMultiPartFormDataStream;
    FRespCookieStream: TStringStream;
    FRespFileStream: TStringStream;
    FIdHTTP: TIdHTTP;
    FIdSSLIOHandlerSocketOpenSSL : TIdSSLIOHandlerSocketOpenSSL;
    FCookie: string;
    FCookieName: string;
    FFileName: string;
    FMensagemRetorno: string;
    FPassword: string;
    FResponse: string;
    FResponseCookie: TJSONValue;
    FResponseCookieStr: string;
    FResponseFile: TJSONValue;
    FResponseFileStr: string;
    FSucesso: boolean;
    FCookieText: string;
    FURL: string;
    FUsername: string;

    procedure SetPassword(const Value: string);
    procedure SetURL(const Value: string);
    procedure SetUsername(const Value: string);
  public
    constructor Create;
    destructor Destroy; override;

    function Upload(AFileName: string): boolean;

    property Cookie: string read FCookie;
    property CookieName: string read FCookieName;
    property CookieText: string read FCookieText;
    property MensagemRetorno: string read FMensagemRetorno;
    property Password: string read FPassword write SetPassword;
    property Response: string read FResponse;
    property ResponseCookie: TJSONValue read FResponseCookie;
```

```
    property ResponseCookieStr: string read FResponseCookieStr;
    property ResponseFile: TJSONValue read FResponseFile;
    property ResponseFileStr: string read FResponseFileStr;
    property Sucesso: boolean read FSucesso;
    property URL: string read FURL write SetURL;
    property Username: string read FUsername write SetUsername;
  end;

const
  AURI: string = 'https://apis.averbeporto.com.br';

implementation

uses
  HPorto;
{ TPortoSeguroIndy }
{$REGION 'Constructor / destructor methods'}

constructor TPortoSeguroIndy.Create;
begin
  FUsername := DmPorto.Parametros.Usuario;
  FPassword := DmPorto.Parametros.Senha;

  FPostCookieStream := TIdMultiPartFormDataStream.Create;
  FPostCookieStream.AddFormField('mod', 'login');
  FPostCookieStream.AddFormField('comp', '5');
  FPostCookieStream.AddFormField('dump', '1');

  FPostFileStream := TIdMultiPartFormDataStream.Create;
  FPostFileStream.AddFormField('mod', 'Upload');
  FPostFileStream.AddFormField('comp', '5');
  FPostFileStream.AddFormField('path', 'eguarda/php/');
  FPostFileStream.AddFormField('recipient', 'T');

  FRespCookieStream := TStringStream.Create('');
  FRespFileStream := TStringStream.Create('');

  FIdHTTP := TIdHTTP.Create(nil);
  FIdHTTP.Request.ContentType := FPostCookieStream.RequestContentType;
  FIdHTTP.Request.ContentLength := FPostFileStream.Size;
  FIdHTTP.Request.UserAgent := 'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:12.0)
Gecko/20100101 Firefox/12.0';

  FCookie := '';
  FCookieName := '';
  FResponseFile := TJSONValue.Create;
  FResponseCookie := TJSONValue.Create;
  FCookieText := '';
  FURL := AURI + '/php/conn.php';
  FSucesso := false;
end;
```

```
destructor TPortoSeguroIndy.Destroy;
begin
  FPostCookieStream.Free;
  FPostFileStream.Free;
  FRespCookieStream.Free;
  FRespFileStream.Free;
  FIdHTTP := nil;
 inherited;
end;
{$ENDREGION}

{$REGION 'getter / setter methods'}
procedure TPortoSeguroIndy.SetPassword(const Value: string);
begin
  FPassword := Value;
end;

procedure TPortoSeguroIndy.SetURL(const Value: string);
begin
  FURL := Value;
end;

procedure TPortoSeguroIndy.SetUsername(const Value: string);
begin
  FURL := Value;
end;
{$ENDREGION}

{$REGION 'Class methods'}
function TPortoSeguroIndy.Upload(AFileName: string): boolean;
var
  URI: TIdURI;
begin
  try
    if (FUsername = '') then
      raise Exception.Create('User Porto Seguro não pode estar em branco.');
    if (FPassword = '') then
      raise Exception.Create('Password Porto Seguro não pode estar em branco.');
    if (AFileName = '') then
      raise Exception.Create('Arquivo: ' + AFileName + ' não pode estar em branco.');
    if (not FileExists(AFileName)) then
      raise Exception.Create('Arquivo: ' + AFileName + ' não existe.');

    FFileName := AFileName;
    // Get Cookie
    // Usuário de API com 60 caracteres - gerar no módulo “Cadastro de Usuário”
    FPostCookieStream.AddFormField('user', FUsername);
    // Senha de API com 64 caracteres - gerar no módulo “Cadastro de Usuário”
    FPostCookieStream.AddFormField('pass', FPassword, 'utf-8').ContentTransfer := '8bit';
    FPostCookieStream.AddFormField('mod', 'login');
```

```
    FPostCookieStream.AddFormField('comp', '5');
    FIdHTTP.Post(FURL, FPostCookieStream, FRespCookieStream);

    FResponseCookieStr := FRespCookieStream.DataString;
    FResponseCookie := TJSONString.Create(FResponseCookieStr);
    FCookieName := FIdHTTP.CookieManager.CookieCollection.Cookies[0].CookieName;
    FCookieText := FIdHTTP.CookieManager.CookieCollection.Cookies[0].CookieText;
    FCookie := Copy(FCookieText, Pos('portal[ses]', FCookieText), Pos(';', FCookieText)-1);

    // Upload file
    FPostFileStream.AddFile('file', FFileName);

    FIdHTTP.AllowCookies := true;
    FIdHTTP.CookieManager := TIdCookieManager.Create;
    URI := TIdURI.Create(AURI);

    FIdHTTP.CookieManager.AddServerCookie(FCookie, URI);
    FIdHTTP.Request.ContentType:= 'multipart/form-data;charset=iso-8859-1;';
    FIdHTTP.Request.ContentLength := FPostFileStream.Size;
    FIdHTTP.Post(FURL, FPostFileStream, FRespFileStream);
    FResponseFileStr := FRespFileStream.DataString;
    FResponse := FRespFileStream.DataString;
    FResponseFile := TJSONString.Create(FResponse);

    FSucesso := ((Pos('"success":1', FResponse) > 0) And (Pos('"prot"', FResponse) > 0)) ;

    if (Pos('"P":1', FResponse) > 0) then
      FMensagemRetorno := 'Processado (xml guardado com sucesso)'
    else if (Pos('"D":1', FResponse) > 0) then
      FMensagemRetorno := 'Duplicado (xml préexistente)'
    else if (Pos('"R":1', FResponse) > 0) then
      FMensagemRetorno := 'xml não parece ser do tipo certo'
    else if (Pos('"N":1', FResponse) > 0) then
      FMensagemRetorno := 'Negado (Não é xml ou zip)'
    else
      FMensagemRetorno := 'Erro inesperado: ' + FResponse + '. Entre em contato com sua TI.';

    Result := FSucesso;

  except
    on e: Exception do
      FMensagemRetorno := 'API não respondeu em '  + FloatToStr(FIdHTTP.ReadTimeout / 1000)
+ ' segundos';
  end;
end;

end.
```