psdocvault – API propio vs Webservice nativo (con ejemplos de SUBIDA de PDF)

Este documento explica cómo subir un PDF cifrado al módulo psdocvault usando el API propio del módulo y, opcionalmente, cómo exponerlo también por el Webservice nativo de PrestaShop. Se incluyen ejemplos en Visual Basic (VB.NET), PHP, JavaScript, C++, Velneo y otros (cURL, PowerShell).

1) Especificación del API propio (endpoint de subida)

URL: POST https://tu-dominio.com/module/psdocvault/api

Parámetros (multipart/form-data):

- op: upload (o upload_bo)
- token: token de seguridad configurado en el módulo
- id_customer: ID del cliente (obligatorio)
- id_order: ID del pedido (opcional)
- doc_type: invoice | delivery | refund
- file: fichero PDF (application/pdf)

Respuesta (JSON):

```
{
  "ok": true,
  "uuid": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "download_url": "https://tu-dominio.com/module/psdocvault/download?f=UUID"
}
```

2) Ejemplo genérico (cURL)

```
curl -X POST "https://tu-dominio.com/module/psdocvault/api" \
  -F "op=upload" \
  -F "token=MI_TOKEN_SEGURO" \
  -F "id_customer=12" \
  -F "id_order=345" \
  -F "doc_type=invoice" \
  -F "file=@/ruta/a/factura.pdf"
```

3) PHP (cURLFile)

```
<?php
$ch = curl_init("https://tu-dominio.com/module/psdocvault/api");
$post = [
  'op'          => 'upload',
  'token'       => 'MI_TOKEN_SEGURO',
  'id_customer' => 12,
  'id_order'    => 345,
  'doc_type'    => 'invoice',
  'file'        => new CURLFile('/ruta/a/factura.pdf', 'application/pdf', 'factura.pdf')
];
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $post);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($ch);
if ($response === false) {
  die('Error: '.curl_error($ch));
}
curl_close($ch);
echo $response;
```

4) JavaScript (fetch + FormData)

```
const fd = new FormData();
```

```
fd.append("op", "upload");
fd.append("token", "MI_TOKEN_SEGURO");
fd.append("id_customer", "12");
fd.append("id_order", "345");     // opcional
fd.append("doc_type", "invoice"); // invoice | delivery | refund
fd.append("file", document.querySelector("#fileInput").files[0]);

fetch("https://tu-dominio.com/module/psdocvault/api", {
  method: "POST",
  body: fd,
  credentials: "include"
})
.then(r => r.json())
.then(data => console.log(data))
.catch(err => console.error(err));
```

5) Visual Basic (VB.NET – multipart/form-data)

```
Imports System.Net
Imports System.IO

Module Module1
  Sub Main()
    Dim boundary As String = "----psdocvault-" & DateTime.Now.Ticks.ToString()
    Dim request As HttpWebRequest = CType(WebRequest.Create("https://tu-dominio.com/module/psdocvault/api
    request.Method = "POST"
    request.ContentType = "multipart/form-data; boundary=" & boundary

    Dim mem As New MemoryStream()
    Dim writer As New StreamWriter(mem)

    Sub AddField(name As String, value As String)
      writer.Write("--" & boundary & vbCrLf)
      writer.Write($"Content-Disposition: form-data; name=""{name}""" & vbCrLf & vbCrLf)
      writer.Write(value & vbCrLf)
    End Sub

    ' Campos
    AddField("op", "upload")
    AddField("token", "MI_TOKEN_SEGURO")
    AddField("id_customer", "12")
    AddField("id_order", "345")
    AddField("doc_type", "invoice")

    ' Fichero
    writer.Write("--" & boundary & vbCrLf)
    writer.Write("Content-Disposition: form-data; name=""file""; filename=""factura.pdf""" & vbCrLf)
    writer.Write("Content-Type: application/pdf" & vbCrLf & vbCrLf)
    writer.Flush()
    Dim fileBytes = File.ReadAllBytes("C:\\ruta\\factura.pdf")
    mem.Write(fileBytes, 0, fileBytes.Length)
    writer.Write(vbCrLf)

    ' Cierre
    writer.Write("--" & boundary & "--" & vbCrLf)
    writer.Flush()
    request.ContentLength = mem.Length
    Dim reqStream = request.GetRequestStream()
    mem.Position = 0
    mem.WriteTo(reqStream)
    reqStream.Close()

    Dim resp = CType(request.GetResponse(), HttpWebResponse)
    Dim sr As New StreamReader(resp.GetResponseStream())
    Console.WriteLine(sr.ReadToEnd())
  End Sub
End Module
```

6) C++ (libcurl – multipart)

```
#include <curl/curl.h>
#include <string>
#include <iostream>

size_t onWrite(char* ptr, size_t size, size_t nmemb, void* userdata) {
  std::string* s = static_cast<std::string*>(userdata);
  s->append(ptr, size * nmemb);
  return size * nmemb;
}
```

```
int main() {
  CURL* curl = curl_easy_init();
  if (!curl) return 1;

  curl_mime* mime = curl_mime_init(curl);
  curl_mimepart* part;
  std::string response;

  part = curl_mime_addpart(mime);
  curl_mime_name(part, "op");
  curl_mime_data(part, "upload", CURL_ZERO_TERMINATED);

  part = curl_mime_addpart(mime);
  curl_mime_name(part, "token");
  curl_mime_data(part, "MI_TOKEN_SEGURO", CURL_ZERO_TERMINATED);

  part = curl_mime_addpart(mime);
  curl_mime_name(part, "id_customer");
  curl_mime_data(part, "12", CURL_ZERO_TERMINATED);

  part = curl_mime_addpart(mime);
  curl_mime_name(part, "id_order");
  curl_mime_data(part, "345", CURL_ZERO_TERMINATED);

  part = curl_mime_addpart(mime);
  curl_mime_name(part, "doc_type");
  curl_mime_data(part, "invoice", CURL_ZERO_TERMINATED);

  part = curl_mime_addpart(mime);
  curl_mime_name(part, "file");
  curl_mime_filedata(part, "factura.pdf");

  curl_easy_setopt(curl, CURLOPT_URL, "https://tu-dominio.com/module/psdocvault/api");
  curl_easy_setopt(curl, CURLOPT_MIMEPOST, mime);
  curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, onWrite);
  curl_easy_setopt(curl, CURLOPT_WRITEDATA, &response);

  CURLcode res = curl_easy_perform(curl);
  std::cout << response << std::endl;

  curl_mime_free(mime);
  curl_easy_cleanup(curl);
  return res == CURLE_OK ? 0 : 1;
}
```

7) Velneo (pseudo – multipart)

```
// Consumir REST POST multipart para subir PDF a psdocvault
SET URL        = "https://tu-dominio.com/module/psdocvault/api"
SET MULTIPART  = new Multipart()
CALL MULTIPART.addField("op", "upload")
CALL MULTIPART.addField("token", "MI_TOKEN_SEGURO")
CALL MULTIPART.addField("id_customer", "12")
CALL MULTIPART.addField("id_order", "345")
CALL MULTIPART.addField("doc_type", "invoice")
CALL MULTIPART.addFile("file", "factura.pdf", "application/pdf")
CALL REST_POST(URL, MULTIPART, RESPONSE)
IF RESPONSE.ok
  TRACE(RESPONSE.data)
ENDIF
```

8) PowerShell (Invoke-WebRequest)

```
$form = @{
  op          = 'upload'
  token       = 'MI_TOKEN_SEGURO'
  id_customer = '12'
  id_order    = '345'
  doc_type    = 'invoice'
  file        = Get-Item 'C:\ruta\factura.pdf'
}
Invoke-WebRequest -Uri 'https://tu-dominio.com/module/psdocvault/api' -Method Post -Form $form
```

9) Webservice nativo de PrestaShop (opcional)

Si registras un recurso propio (p. ej., psdocvault_files) mediante el hook addWebserviceResources, podrás listar metadatos via /api. La descarga del PDF permanece en el endpoint del módulo y exige login del cliente.

```
# GET metadatos (ejemplo)
curl -X GET "https://tu-dominio.com/api/psdocvault_files?filter[id_customer]=[12]&ws_key=APIKEY"

# POST con base64 (si implementas specific_management para subida)
curl -X POST "https://tu-dominio.com/api/psdocvault_files?ws_key=APIKEY" \
  -H "Content-Type: application/json" \
  -d '{
        "id_customer": 12,
        "id_order": 345,
        "doc_type": "invoice",
        "filename": "factura.pdf",
        "content_base64": "<BASE64_DEL_PDF>"
      }
```

Notas de seguridad

- El fichero se cifra con AES-256-GCM usando la clave única del cliente.
- La descarga solo es posible si el cliente ha iniciado sesión (FrontController download).
- Guarda el token en un lugar seguro (Configuration) y rótalo periódicamente.