Factura Electrónica: Ventajas Biblioteca DLL LibPyAfipWs

La librería <u>LibPyAfipWs</u> es una DLL escrita en C que permite acceder a los webservices de AFIP / ANMAT y en Windows y Linux, compatible con varios lenguajes (Visual Basic, Visual Fox Pro, Delphi, PHP, .Net, Java, C / C++, etc.) Al no ser un OCX ni objeto COM, es muy versátil (igualmente ver <u>OcxFacturaElectronica</u> y <u>PyAfipWs</u> para una alternativa más dinámica para lenguajes modernos).

La biblioteca puede usarse de varias formas que a continuación se detallan (acceso en tiempo de ejecución, compilación y enlace, declaración en VB y VFP, etc.)

Para más información sobre los métodos soportados, ver <u>LibPyAfipWs</u>. Para más información, ver <u>ManualPyAfipWs</u>, <u>FacturaElectronica</u> y <u>PyAfipWs</u>

Acceso en tiempo de ejecución

La librería se puede acceder desde C o lenguajes similares simplemente llamando a la función <u>LoadLibrary("ruta.dll")</u> y luego se accede a las funciones con <u>GetProcAddress</u>. Es importante notar que este método no necesita que se linkee a la librería en tiempo de ejecución, por lo que es muy flexible para desarrollo.

Ejemplo en lenguaje C estándard para crear un ticket de acceso:

```
#include <windows.h>
int main(int argc, char *argv[]) {
 BSTR tra, cms, ta, ret;
 void *wsfev1;
 bool ok;
 long nro;
 HINSTANCE hPyAfipWsDll;
  FARPROC lpFunc, lpFree;
  /* cargo la librería y obtengo la referencia (poner ruta completa) */
  hPyAfipWsDll = LoadLibrary("..\\LIBPYAFIPWS.DLL");
  if (hPyAfipWsDll != NULL) {
    /* obtengo los punteros a las funciones exportadas en la librería */
   lpFunc = GetProcAddress(hPyAfipWsDll , "WSAA_CreateTRA");
    lpFree = GetProcAddress(hPyAfipWsDll , "PYAFIPWS_Free");
    if (lpFunc != (FARPROC) NULL) {
        /* llamo al método de la DLL para crear el ticket de req. de acceso */
        tra = (*lpFunc)("wsfe", (long)3600);
        printf("TRA: %s\n", tra);
        /* libero la memoria alojada por el string devuelto */
        (*lpFree)(tra);
    }
  FreeLibrary (hPyAfipWsDll);
```

Ver <u>libpyafipws dll test.c</u> (ejemplo completo)

Compilación y enlace

La librería se puede usarse utilizando el encabezado libpyafipws.h que define las funciones, libpyafipws.lib para linkearlo. Ejemplo en lenguaje C estándard:

```
#include "libpyafipws.h"
int main(int argc, char *argv[]) {
   BSTR tra, cms, ta;
   void *wsfev1;

   /* Generar ticket de requerimiento de acceso */
   tra = WSAA_CreateTRA("wsfe", 999);
   printf("TRA:\n%s\n", tra);
   /* Firmar criptograficamente el mensaje */
   cms = WSAA_SignTRA((char*) tra, "reingart.crt", "reingart.key");
   printf("CMS:\n%s\n", cms);
   /* Llamar al webservice y obtener el ticket de acceso */
   ta = WSAA_LoginCMS((char*) cms);
   printf("TA:\n%s\n", ta);
```

Ver <u>factura electronica.c</u> (ejemplo completo)

C# (.Net) con InteropServices

Desde .NET se puede llamar a las funciones de la librería usando los servicios de interoperabilidad (DllImport en System.Runtime.InteropServices):

```
using System;
using System. Text;
using System.Runtime.InteropServices;
namespace ConsoleApplication1
    class Program
        [DllImport("F:\\LIBPYAFIPWS.DLL")]
        private static extern string WSAA_CreateTRA(
              string service,
              long ttl
        );
        static void Main(string[] args)
            string tra;
            tra = WSAA_CreateTRA("wsfe", 3600);
            Console.WriteLine("TRA = {0}", tra);
            Console.ReadLine();
        }
    }
```

Ver <u>factura electronica.cs</u> (ejemplo completo)

Nota: Para VB.NET se recomienda la interfaz por objetos (Component Object Model): <u>PyAfipWs</u>, ver <u>wsfev1.vb</u> (ejemplo completo)

Visual Basic, Visual Fox Pro y similares

En lenguajes modernos, se puede declarar la función externa, por ej. en VB (cambiar . . \ por la ruta completa, donde este la DLL y el resto de los archivos de instalación):

unction WSAA_CreateTRA Lib "..\libpyafipws.dll" (ByVal service As String, ByVal ttl As Long) As String'

```
unction WSAA_SignTRA Lib "..\libpyafipws.dll" (ByVal tra As String, ByVal cert As String, ByVal pk As String)
unction WSAA_LoginCMS Lib "..\libpyafipws.dll" (ByVal tra As String) As String
erar un Ticket de Requerimiento de Acceso (TRA) para WSFEv1
36000 ' tiempo de vida = 10hs hasta expiración
WSAA_CreateTRA("wsfe", ttl)
ecificar la ubicacion de los archivos certificado y clave privada
= CurDir() + "\"
tificado: certificado es el firmado por la AFIP
vePrivada: la clave privada usada para crear el certificado
ficado = "..\reingart.crt" ' certificado de prueba
Privada = "..\reingart.key" ' clave privada de prueba
erar el mensaje firmado (CMS)
WSAA_SignTRA(tra, Path + Certificado, Path + ClavePrivada)
Print cms
mar al webservice de autenticación:
WSAA_LoginCMS (cms)
rimir el ticket de acceso, ToKen y Sign de autorización
.Print ta
```

Ver <u>libpyafipws dll test.bas</u> (ejemplo completo)

Para Visual Fox Pro sería similar, declarando la función exportada por la DLL de la siguiente manera:

```
DECLARE STRING WSAA_CreateTRA IN ..\LIBPYAFIPWS STRING @ service, LONG @ ttl && Generar un Ticket de Requerimiento de Acceso (TRA) para WSFEv1 ttl = 36000 && tiempo de vida = 10hs hasta expiración tra = WSAA_CreateTRA("wsfe", ttl)
? tra
```

Ver <u>libpyafipws dll test.prg</u> (ejemplo completo)

Nota: Para VB y VFP se recomienda la interfaz por objetos (Component Object Model): PvAfipWs

Contacto

Para mayor información, consultar por mail a <u>facturaelectronica@sistemasagiles.com.ar</u> o telefónicamente al 15-3048-9211

Para soporte de la comunidad, revisar la <u>lista de temas</u> y/o <u>crear uno nuevo</u> Para novedades y consultas genereales, puede usar el <u>Foro Público</u>

PyAfipWs Copyright 2008, 2009, 2010, 2012, 2013 por MarianoReingart

Contacto 4