Factura Electrónica con Python

- **Documentación Componentes y Herramientas Generales:**
  - PyAfipWs: Interfase simil OCX con otros lenguajes (VB, VFP, Cobol ...) Costos y Condiciones
  - Manual: Documentación, Información Importante, Certificados, Errores Frecuentes
  - Herramienta "universal": archivos de intercambio TXT Cobol, DBF dBase/FoxPro, JSON PHP/Java
  - PyFEPDF: Generador de Factura Electrónica en formato PDF
- **Factura Electrónica - Servicios Web AFIP:**
  - Mercado Interno: Factura Electrónica A/B/C/M WSFEv1+ (RG2485/3067/3668/3749/4004/4109/4367)
  - Matrix (codificación productos): Factura Electrónica A/B con detalle (RG2904/3536)
  - Bienes de Capital: Bonos Fiscales Electrónicos - Factura Electrónica A (RG2557)
  - Exportación: Factura Electrónica E Exportadores (RG2758 RG3689 RG4401)
  - Turismo: Comprobantes Factura Electrónica T WSCT CAE/CAEA (RG3971)
  - FE Cred (FCE): Factura Electrónica Crédito MiPyMEs (RG4367)
  - Código de Autorización Electrónico Anticipado CAEA
- **Agropecuario - Servicios web AFIP:**
  - Código Trazabilidad de Granos: Transporte de granos WSCTGv4 (RG2806 RG3113 RG3493)
  - Liquidación y Certificación de Granos: WSLPGv1.17 F. C1116 A / B / RT (RG3419 RG3690 RG3691)
  - Liquidación de Tabaco Verde: WSLTVv1.3
  - Liquidación Única Mensual Lechería: WSLUMv1.3
  - Liquidación Sector Pecuario: Hacienda, Compra directa, Carne WSLSPv1.3
  - Remito Electrónico Cárnico: WSRemCarne (RG4256/18 y RG4303/18)
  - Remito Electrónica Harinero: WSRemHarina - (RG 4519/19)
  - Remito Electrónico Azúcar y Derivados: WSRemAzucar - (RG 4519/19)
  - Carta de Porte: WSCPE - (RG 5017/21)
  - Carta de Porte Derivados Granarios: WSCPEv2 - (RG 5235/22) **Nuevo!**
- **Otros webservices y utilidades AFIP**
  - Constatación de Comprobantes CAI, CAE, CAEA (WSCDC)
  - Padron Contribuyentes AFIP: Constancia de Inscripción RG1817/2005 WS-SR-Padron
  - Consulta de Operaciones Cambiarias: Compra de Divisas (WSCOC)
  - SIRE_CertificadoRetencionElectronica: Certificado de retención electrónica del Impuesto al Valor Agregado (SIRE-WS) RG4523/19
- **Webservices provinciales: ARBA (Prov. Bs.As.), AGIP (C.A.B.A), API (Sta.Fe), DGR (Córdoba):**
  - Remito Electrónico: COT Código de Operaciones de Translado (ARBA, API, AGIP, DGR)
  - Ingresos Brutos: Consulta de alícuotas WS DFE IIBB ARBA
  - Retenciones A-122R: Ingreso de comprobantes de retenciones
- **SNT: Sistema Nacional de Trazabilidad ANMAT, SEDRONAR, SENASA**
  - Trazabilidad de Medicamentos: ANMAT Disposición 3683/2011
  - Trazabilidad de Productos Médicos: ANMAT Disposición 2303/2014 y 2175/14
  - Trazabilidad de Precursores Químicos: RENPRE SEDRONAR Resolución 900/12
  - Trazabilidad de Productos Fitosanitarios: SENASA Resolución 369/13
  - Trazabilidad de Productos Fitosanitarios: SENASA Resolución 369/21 (WS_INFO_EMPRESAS / WS_DATOS_GENERALES)

- **Aplicativos Genéricos y Herramientas Avanzadas:**
    - ♦ PyRece: Aplicativo visual simil SIAP - RECE (CSV, PDF, Email)
    - ♦ FE.py: Herramienta universal, unificada e integrada
    - ♦ FacturaLibre: Aplicacion online (web2py)
    - ♦ PyFactura Aplicativo visual y simple (gui2py) para CAE y PDF factura electrónica
    - ♦ LibPyAfipWs: Biblioteca DLL para lenguajes C / C++ y similares
    - ♦ Factura Electrónica en Python: Información Técnica (SOAP, XML, PDF, DBF, etc.)

Interfaz Python de software libre para Emisión y almacenamiento electrónico de comprobantes originales AFIP - Argentina.

RG 1956/05, 1361/02, 1345/02, 2265/07, 2289/07 y 2557/09

Índice

Descripción General

PyAfip.ws contienen módulos para acceder a los servicios webs de factura electrónica de la AFIP, basados en los ejemplos disponibles en el sitio de homologación:

- wsaa.py: interface con Web-Service Autenticación y Autorización (basado en wsaa-client.php)
- wsfe.py: interface con el Web-Service de Factura Electrónica (basado en wsfe-client.php)
- wsbfe.py: interface con el Web-Service de Factura Electrónica Bienes de Capital - Bono Fiscal Electrónico
- pyafipws.py: servidor COM para acceder a los servicios web desde otros lenguajes en windows (VB, VFP, Delphi, etc.)
- pyrece.py: aplicativo factura electrónica simil SIAP RECE (gui)
- rece.py: utilitario para factura electrónica por archivo de texto (consola)
- receb.py: utilitario para factura electrónica bienes de capital por archivo de texto (consola)
- rg1361.py: utilitario para almacenamiento de duplicados - SIAP SIRED (consola)

Son completamente funcionales (Fueron testeados en servidores de homologación), aunque wsfe.py es solo un ejemplo que se debe adaptar a cada caso particular. Para evitar adaptar el wsfe para cada caso, se intenta refactorizar estos ejemplos en algo más usable tanto desde python como desde linea de comando.

Observaciones

Para la traducción de los ejemplos PHP proporcinados por la AFIP, se intentó mantener la estructura de dicho código, a manera de ejemplo de traducción de php a py y para aseguramos cumplir el mismo método, facilitando la verificación y validación.

Cuando fue necesario, se agregaron funciones o objetos que emulaban a los de php, y se modificó en el caso en que las construcciones en python eran mejores.

XML

El ejemplo en php utiliza SimpleXMLElement, que es una herramienta para trabajar con XML de manera simple y orientada a objetos.

Esta herramienta se reimplementó en python encapsulando xml.dom.minidom (ver SimpleXmlElement en pyafip/ws/simplexml.py)

La principal diferencia es que no convierte los tipos (int, long, etc.). Siempre devuelve elementos xml (texuales), que hay que convertirlos explicitamente.

SOAP

Por el lado de web-services, se intento con SOAPpy, y en menor medida soap.py, y no se llegó a probar ZSI.

Aparentemente el WSAA es un webservice en java, donde no hubo problema en usar SOAPpy, pero el WSFE es .NET, donde SOAPpy no funciona por incompatibilidades en el manejo de XML.

Ante las incompatibilidades, se decidió hacer una implementación del cliente soap desde cero (ver SoapClient en pyafip/ws/soap.py), utilizando httplib2 para la conexión y SimpleXMLElement para el manejo del requerimiento y respuesta XML. Esto posibilitó armar los xml de manera compatible con el web service en .NET y comunicarse sin problemas.

Este cliente es simple y minimo, pero funcional. El unico inconveniente es que no parsea el wsdl, por lo que hay que extraer los datos del web service manualmente (SOAPAction y el espacio de nombres a utilizar). Tampoco se puede listar los métodos disponibles, pero esto no es problema ya que se puede leer el wsdl.

Al usar SimpleXmlELement, realiza la serialización simple convirtiendo a string, pero la desserialización debe ser hecha manualmente (conversión de tipos o creación de objetos).

Varios

Textos de Ancho Fijo

Para la interfaz de texto por línea de comando (consola), se desarrollaron funciones para facilitar el manejo de archivos de texto con campos de ancho fijo (formatos utilizados por ej. por COBOL y los aplicativos SIAP de la AFIP). Ver pyafip/ws/rece.py, pyafip/ws/receb.py

Interfaz por base de datos

Estamos desarrollando una herramienta generica para autorización y generación de facturas electrónicas mediante bases de datos, unificando los servicios web (WSFE, WSBFE y WSFEX).

Generación de PDF

Para crear las facturas electrónicas en formato PDF se utilizó la librería PyFpdf, mejorándola y adaptándola con los siguientes temas:

- Impresión de código de barras (ver Interleaved2of5 en pyfpdf/FPDF.py)

- Definición de campos por CSV, para poder modificar el diseño fácilmente (ver pyfpdf/ejemplos/form.py)
- Mejoras y correcciones menores

Interfaz Gráfica PythonCard

Existen algunos temas menores con los unicodes entre distintas plataformas (Windows y Linux)

Ficha técnica

- Requisitos:
  - Python24: no se testeo con versiones anteriores (utiliza xml.dom.minidom)
  - M2Crypto: para firma y encriptación
  - httplib2: para conexiones seguras
- Autores: MarianoReingart y MarceloAlaniz
- Licencia: LGPLv3 y GPLv3 (ver archivos fuente)
- Fuentes: pyafip/ws
- Repositorio SVN: http://www.sistemasagiles.com.ar/svn/pyafip/ws/
- Descargables multiplataforma:
  - Ver GitHub (actualizado) y GoogleCode (histórico)

Instalación

- En Debian GNU/Linux:

```
apt-get install python-httplib2 python-m2crypto
```

- En Windows:
  - Instalar Python 2.5.2
  - Instalar Win32OpenSSL 0.9.7m
  - Instalar M2Crypto 0.18.2 (0.19 no funciona)
  - Instalar httplib2. Descomprimir y ejecutar por línea de comando: `c:\python25\python.exe setup.py install`
  - Instalar Extensiones Win32 para interfase COM. Ejecutar `c:\python25\python.exe pyafipws.py --register` para registrar el Servidor COM y poder acceder desde otros lenguajes.
- Crear certificados con OpenSSL (ver PyAfipWs#Certificados)

Interfase con otros Lenguajes

Se ha desarrollado una interface COM autoinstalable para windows compatible con otros lenguajes (Visual Basic, Visual Fox Pro, Delphi, PHP, .Net, Java, etc.) y una interfase por archivo de texto simil SIAP/RECE para lenguajes que no soporten la creación de objetos COM (algunas versiones de COBOL y Fox Pro).

Más información en PyAfipWs

Aplicativo Ad-Hoc

Se ha desarrollado un aplicativo (ejecutable con interfase "visual") para windows/linux, que autoriza, genera

pdf y envia los mais con facturas electrónicas.

Más información en PyRece

Novedades

- Ver Grupo de Noticias en Google

Capacitación

- Ver Curso en la ACP

Se ofrece soporte técnico comercial (pago), consultar por desarrollos especiales, interfaces web, etc. a pyrece@sistemasagiles.com.ar o telefónicamente al 15-3048-9211

Por consultas gratuitas sobre el lenguaje python y demás, dirigirse a PyAr.

Para soporte sin cargo de la comunidad, revisar la lista de temas y/o crear uno nuevo. Por novedades y consultas genereales, puede usar el Google Groups (Foro Público). Código fuente en GitHub.

MarianoReingart