Servicios A122R - Ingreso de Comprobantes de Retenciones ARBA (Autenticación al servidor IDP de ARBA vía WSIDP)

- **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 para Servicio web para el ingreso de comprobantes de retención A-122R correspondiente a la Resolución Normativa N° 22/25 ARBA (Agencia de Recaudación Provincia de Buenos Aires).

Esta interfaz requiere un nuevo webservice de autenticación al servidor de ARBA de IDP, WSIDP.

Índice

Descargas

- Instalador: Instalador 1.01a para evaluación
- Ejemplo en VB: Ejemplo
- Código Fuente (Python):

URL

- Testing WSA122r: https://app2.test.arba.gov.ar/a122rSrv/api/external
- Producción WSA122r: https://app.arba.gov.ar/a122rSrv/api/external

Métodos WSA122R

- **`SetToken(token)`**: Asigna el valor del token conseguido despues de ejecutar el metodo **`ObtenerToken(cuit, cit, url, trace)`** en Métodos WSIDP. Ver ejemplo para el detalle de la asignación.
- **`Conectar(url, proxy, cacert, trace, testing)`**: los parámetros son similares a WSFEv1.Conectar (por el momento solo se usa url y trace para depuración).
- **`IniciarDj(cuit_agente, actividad_id, anio, mes, quincena)`**: Permite iniciar una DJ. La respuesta a este request, devolverá el ID de la DJ, la cual es necesaria para ir agregando comprobantes a la declaración jurada iniciada. La DJ iniciada pertenecerá, a la cuit del agente que se asignó en el campo ?cuitAgente?, debiéndose corresponder con el cuit del agente que generó el token.
- **`CrearComprobanteInterno(cuit_contribuyente, cuit_agente, sucursal, alicuota, base_imponible, importe_retencion,`**

razon_social_contribuyente, fecha_operacion, n_transaccion_agente):: Crea internamente un comprobante para luego poder darlo de alta, recibe los datos del comprobante a emitir. Ver ejemplo para el detalle de los parámetros.

- **AgregarDireccion(calle, numero, piso, departamento, codigo_postal, localidad, provincia)**: Agrega internamente una dirección al comprobante para luego poder autorizarla.
- **AltaComprobante(self, id_dj, cuit_agente)**: Permite ir incorporando comprobantes de retención A122R a la Declaración Jurada iniciada. Para poder relacionar el comprobante con la DJ, se debe tomar y conservar el ID de la Declaración Jurada. (El id del comprobante se guarda como **IdComprobante**)
- **BajaComprobante(id_comprobante)**: Permite dar de baja un comprobante, a partir del ID de comprobante obtenido en el servicio de alta.
- **ConsultarDj(cuit_agente, id_dj, actividad_id, anio, mes, quincena)**: Permite consultar las DDJJs por alguna de las siguientes opciones: A) Id de DJ y cuit agente, B) cuit agente, periodo y actividad.
- **ConsultarComprobante(cuit_agente, id_dj, anio, mes, quincena, cuit_contribuyente, estado)**: Permite consultar los comprobantes por alguna de las siguientes opciones, A) ID de DJ y cuit agente, B) Cuit agente, cuit contribuyente, periodo, estado (?ACTIVO?,?BAJA?,?TODOS?); siendo obligatorio el Cuit agente y periodo.
- **ConsultarComprobanteTxt(cuit_agente, id_dj, anio, mes, quincena, cuit_contribuyente, estado)**: Permite la descarga de los comprobantes de DJ filtrando por alguna de las siguientes opciones: A) Id de DJ y cuit agente, B) Cuit agente, cuit contribuyente, periodo y estado (?ACTIVO?,?BAJA?,?TODOS?); siendo obligatorio el Cuit agente y periodo.
- **ConsultarComprobantePDF(self, id_comprobante, archivo_destino)**: Devuelve archivo en formato PDF para el comprobante de retención solicitado, el archivo se guardara en la ruta de archivo_destino.

Métodos WSIDP

- **ObtenerToken(cuit, cit, client_id, secret, url)**: Devuelve el token que luego se va a asignar con el método de WSA122r **SetToken(token):**, El token expira cada 300s (5 minutos) la lógica de re-utilización esta implementada en el método. El token se guarda en un archivo .json en la carpeta de instalación de la librería dentro de cache, si la misma no existe crea una carpeta cache donde este ubicado el script.

Atributos WSA122r

- `Token` El token para autorizar las operaciones
- `Version` e `InstallDir` sirven para depuración de la interfaz.
- `Request` y `Response` los datos enviados al servidor de ARBA y respuesta JSON enviada por ARBA
- `Excepcion`, `Traceback` se completan en caso de error interno no esperado (por ej. falla de comunicación).
- `HttpCode` código http devuelvo por el servidor de ARBA
- `IdDj` Id que se obtienen al iniciar una dj
- `IdComprobante` Id que se obtienen al dar de alta un comprobante

Atributos WSIDP

- `Token` El token para autorizar las operaciones
- `Version` e `InstallDir` sirven para depuración de la interfaz.
- `Request` y `Response` los datos enviados al servidor de ARBA y respuesta JSON enviada por ARBA
- `Excepcion`, `Traceback` se completan en caso de error interno no esperado (por ej. falla de comunicación).
- `HttpCode` código http devuelvo por el servidor de ARBA

IMPORTANTE: para el manejo de errores, siempre se debe revisar el atributo `Excepcion`, si este no está en blanco, ha ocurrido un error no esperado y debe analizar el `Traceback` (traza) y volver a intentar. Siempre es útil almacenar los valores de `Response` como respaldo de la operación y para futura referencia o análisis.

Línea de Comandos

Para sistemas operativos Windows, UNIX/Linux y entornos legados, es posible operar el servicio ARBA A122R mediante línea de comandos utilizando el ejecutable **wsa122r.exe**.

La herramienta permite autenticar contra ARBA y realizar operaciones sobre Declaraciones Juradas y Comprobantes de Retención A122R.

Opcionalmente se puede especificar --test para operar en entorno de pruebas y --trace para imprimir por pantalla los datos enviados y recibidos.

Uso General

```
exe --cuit-auth [cuit del agente] --cit-auth [cit del agente] --client_id [client_id] --secret [secret] [opcion
```

Autenticación

Parámetros obligatorios para todas las operaciones:

```
--cuit-auth [cuit del agente] --cit-auth [cit del agente] --client_id [client_id] --secret [secret]
```

Entorno y Depuración

```
--test Usa el entorno de pruebas (homologación)
--trace Muestra por pantalla el request, response y traceback
```

Entrada y Salida de Datos

```
--cargar [archivo.json] Carga los datos desde un archivo JSON
--csv [cadena_csv] Carga los datos desde una línea CSV
--guardar Guarda la respuesta del servidor en un archivo JSON
--grabar Guarda los parámetros ingresados por consola
```

Nota: Las altas de comprobantes solo se pueden hacer por consola vía --cargar con un archivo JSON o vía --csv con una cadena csv. Todas las demás operaciones se pueden hacer vía consola o cadena csv.

Formato de la cadena csv

```
t_contribuyente,"tmp_sucursal",tmp_alicuota,tmp_base,tmp_imp_ret,"tmp_razon","fecha_operacion","dir_calle","di
```

Nota: Los valores que no se usen se deben enviar como 0 o "" en caso de ser string

Parámetros Comunes

```
--cuit [cuit del agente]
--actividad-id [id de actividad]
--anio [año]
--mes [mes]
--quincena [quincena]
--id-dj [id declaración jurada]
--id-cmp [id comprobante]
--estado [estado]
--cuit-contribuyente [cuit del contribuyente]
```

Operaciones Disponibles

Iniciar Declaración Jurada

```
cit del agente] --client_id [client_id] --secret [secret] --iniciar-dj --cuit [cuit del agente] --actividad-id
```

Inicia una Declaración Jurada A122R y devuelve el identificador **idDj**.

Alta de Comprobante

Desde archivo JSON:

```
--cuit-auth [cuit del agente] --cit-auth [cit del agente] --client_id [client_id] --secret [secret] --alta --ca
```

Desde línea CSV:

```
e --cuit-auth [cuit del agente] --client_id [client_id] --secret [secret] --cit-auth [cit del agente] --alta --
```

El comprobante queda asociado a la Declaración Jurada indicada.

Baja de Comprobante

```
-cuit-auth [cuit del agente] --cit-auth [cit del agente] --client_id [client_id] --secret [secret] --baja --id-
```

Consultar Declaración Jurada

Método A ? Por ID de DJ

```
 agente] --cit-auth [cit del agente] --client_id [client_id] --secret [secret] --consultar-dj --cuit [cuit del
```

Método B ? Por Período

```
nte] --client_id [client_id] --secret [secret] --consultar-dj [ruta y nombre al txt] --cuit [cuit del agente] -
```

Importante: Los métodos A y B son excluyentes y no deben combinarse.

Consultar Comprobantes

Por Declaración Jurada

```
 agente] --cit-auth [cit del agente] --client_id [client_id] --secret [secret] --consultar-cmp --cuit [cuit del
```

Por Período

```
 --cit-auth [cit del agente] --client_id [client_id] --secret [secret] --consultar-cmp --cuit [cuit del agente]
```

Importante: Los métodos A y B son excluyentes y no deben combinarse.

Consultar Comprobantes (TXT)

Por Declaración Jurada

```
th [cit del agente] --client_id [client_id] --secret [secret] --consultar-cmp-txt [(OPCIONAL)ruta + nombre.txt]
```

Por Período

```
del agente] --client_id [client_id] --secret [secret] --consultar-cmp-txt [(OPCIONAL)ruta + nombre.txt] --cuit
```

Importante: Los métodos A y B son excluyentes y no deben combinarse.

Consultar Comprobante (PDF)

```
gente] --cit-auth [cit del agente] --client_id [client_id] --secret [secret] --consultar-cmp-pdf [(OPCIONAL)rut
```

Ejemplo Intefase COM en VB

```
Sub Main()
    Dim a122r As Object, ok As Variant
    Dim idp As Object
    ' Crear la interfaz COM
    Set a122r = CreateObject("WSA122r")
    ' Crear objeto interface Web Service Autenticación
    Set idp = CreateObject("WSIDP")

    Debug.Print a122r.Version
    Debug.Print a122r.InstallDir
```

```
Debug.Print idp.Version
Debug.Print idp.InstallDir

' Solicitar datos a ARBA
cuit_agente = ""
cit = ""
client_id = ""
secret = ""

' Consigo token para autenticarme (solicitar URL del servidor IDP a ARBA)
token = idp.ObtenerToken(cuit_agente, cit, client_id, secret) ' Url por defecto homologacion

Debug.Print ">> Token: ", token

' Seteo el token
a122r.SetToken (token)

' Conecto al servidor de ARBA
ok = a122r.Conectar() ' Url por defecto homologacion

' Inicio una DJ
actividad_id = 6
anio = 2026
mes = 2
quincena = 1
' ok = a122r.IniciarDj(cuit_agente, actividad_id, anio, mes, quincena)
' Debug.Print ">> ID de DJ iniciada: ", a122r.IdDj

' si no quiero iniciar una nueva dj consulto el Id por período
consulta_dj = ""
consulta_dj = a122r.ConsultarDj(cuit_agente, 0, actividad_id, anio, mes, quincena)

Debug.Print ">> ID de DJ consultada: ", a122r.IdDj
Debug.Print ">> Datos de la DJ consultda: ", consulta_dj

' Creo un Comprobante Interno
cuit_contribuyente = "" ' Ingresar un cuit
sucursal = "00002"
alicuota = 1.75
base_imponible = 90000
importe_retencion = 1575
razon_social_contribuyente = "" ' Relacionado al cuit
fecha_operacion = "2026-02-03T00:00:00" ' Si no entra dentro del período de la DJ va a fallar
n_transaccion_agente = 0000

ok = a122r.CrearComprobanteInterno(cuit_contribuyente, cuit_agente, sucursal, _
alicuota, base_imponible, importe_retencion, razon_social_contribuyente, _
fecha_operacion, n_transaccion_agente)

' Agrego la direccion (datos relacionados al cuit)
calle = ""
numero = ""
piso = ""
depto = ""
codigo_postal = ""
localidad = ""
provincia = ""

ok = a122r.AgregarDireccion(calle, numero, piso, depto, codigo_postal, _
localidad, provincia)

' Doy de alta el comprobante
```

```
    ' ok = a122r.AltaComprobante(a122r.IdDj, cuit_agente)
    ' Debug.Print ">> Id Comprobante: ", a122r.IdComprobante

    ' Si no quiero dar de alta un nuevo comprobante consulto el Id por período
    ' y cuit del contribuyente
    estado = "TODOS"
    consulta_cmp = ""
    consulta_cmp = a122r.ConsultarComprobante(cuit_agente, 0, anio, mes, _
    quincena, cuit_contribuyente, estado)

    Debug.Print ">> Id Comprobante Consultado: ", a122r.IdComprobante

    ' Imprimo el comprobante en PDF
    ruta = a122r.InstallDir + "\cache\comp.pdf"
    id_comp = a122r.IdComprobante
    ok = a122r.ConsultarComprobantePdf(id_comp, ruta)

    If ok Then
            Debug.Print ">> PDF Generado En: ", ruta
    End If

    Debug.Print "------------ WSIDP DEBUG ------------"
    Debug.Print ">> Request IDP: ", idp.Request
    Debug.Print ">> Response IDP: ", idp.Response
    Debug.Print ">> Treaceback IDP: ", idp.Traceback
    Debug.Print ">> Excepcion IDP: ", idp.Excepcion
    Debug.Print "------------ WSA122R DEBUG ------------"
    Debug.Print ">> Request WSA122r: ", a122r.Request
    ' Debug.Print ">> Response WSA122r: ", a122r.Response
    Debug.Print ">> Traceback WSA122r: ", a122r.Traceback
    Debug.Print ">> Excepcion WSA122r: ", a122r.Excepcion
End Sub
```

Novedades

Se recuerda que esta disponible el grupo de noticias (http://groups.google.com.ar/group/pyafipws) donde se publicarán futuras novedades sobre PyAfipWS: servicios web de factura electrónica y sus interfases (se recomienda suscribirse)

Costos y Condiciones

Por soporte comercial consultar por mail a info@sistemasagiles.com.ar, r.castrogiovani@gmail.com (directo) o in.reingart@gmail.com (directo).

Más información en PyAfipWs (ver Costos y Condiciones del Soporte Comercial)

Mariano Reingart