

Soap Client

Importante: estas herramientas se han empaquetado en una nueva biblioteca [PySimpleSOAP](#). Ver nueva página: [SoapClient](#) (incluyendo nuevas características y mejoras)

Soap Client es una implementación del cliente soap desde cero (ver [pyafip/ws/soap.py](#)), utilizando `httplib2` para la conexión y [SimpleXmlElement](#) para el manejo del requerimiento y respuesta XML.

Permite controlar los pedidos y armar los xml de manera compatible (por ej. con el web service en .NET y comunicarse sin problemas)

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.

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).

Forma de Uso

- Crear objeto SoapClient con los parámetros: `client = SoapClient(location, action, namespace, trace)`
 - ◆ location: ubicación (URL) definida en el WSDL
 - ◆ action: acción definida en el WSDL
 - ◆ namespace: espacio de nombre definido en el WSDL
 - ◆ trace: indica si se debe depurar (mostrar pedido y respuesta)
 - ◆ proxy: datos del servidor próximo (opcional)
 - ◆ exceptions: indica si se debe detectar y lanzar una excepción transmitida (SoapException)
- Llamar a los métodos del cliente (como si fueran métodos locales). Por ej., `client.loginCms(in0=cms)` llama al método remoto `loginCms` con el parámetro `in0`
- Analizar el resultado como si fuera un objeto (ver [SimpleXmlElement](#))
- Los atributos `xml_request` y `xml_response` tienen los datos XML enviados y recibidos (requerimiento y respuesta)

Ejemplo

```
>>> from soap import SoapClient, SoapFault
>>> client = SoapClient(
...     location = "https://wshomo.afip.gov.ar/wsfe/service.asmx",
...     action = 'http://ar.gov.afip.dif.facturaelectronica/',
...     namespace = "http://ar.gov.afip.dif.facturaelectronica/",
...     trace = True)
>>> results = client.FEDummy()
-----
POST https://wshomo.afip.gov.ar/wsfe/service.asmx
SOAPAction: "http://ar.gov.afip.dif.facturaelectronica/FEDummy"
Content-length: 329
Content-type: text/xml; charset="UTF-8"
```

```

<?xml version="1.0" encoding="utf8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Body>
  <FEDummy xmlns="http://ar.gov.afip.dif.facturaelectronica/">
  </FEDummy>
</soap:Body>
</soap:Envelope>

status: 200
content-length: 438
x-powered-by: ASP.NET
x-aspnet-version: 2.0.50727
server: Microsoft-IIS/6.0
cache-control: private, max-age=0
date: Wed, 30 Dec 2009 17:15:19 GMT
content-type: text/xml; charset=utf-8
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body>
<FEDummyResponse
xmlns="http://ar.gov.afip.dif.facturaelectronica/">
<FEDummyResult>
<appserver>OK</appserver><dbserver>OK</dbserver>
<authserver>OK</authserver>
</FEDummyResult>
</FEDummyResponse></soap:Body></soap:Envelope>
=====
>>> results.asXML()
'<?xml version="1.0" encoding="utf8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><soap:Body>
<FEDummyResponse
xmlns="http://ar.gov.afip.dif.facturaelectronica/">
<FEDummyResult>
<appserver>OK</appserver>
<dbserver>OK</dbserver>
<authserver>OK</authserver>
</FEDummyResult></FEDummyResponse>
</soap:Body>
</soap:Envelope>'
>>> results.appserver
'OK'
>>> results.dbserver
'OK'
>>> results.authserver
'OK'
>>>

```

Ejemplo sin mensajes ni depuración (para cortar y pegar):

```

from soap import SoapClient, SoapFault
client = SoapClient(
    location = "https://wshomo.afip.gov.ar/wsfe/service.asmx",
    action = 'http://ar.gov.afip.dif.facturaelectronica/',
    namespace = "http://ar.gov.afip.dif.facturaelectronica/",
    trace = True)
results = client.FEDummy()

```

```
print results.asXML()  
print results.appserver  
print results.dbserver  
print results.authserver
```