

Author: Eric Engstfeld

Date: 3 de July, 2015

# SOAP Server

## Abstract

La finalidad de este documento consiste en exponer cómo está construido el componente SOAP Server, los pasos para su instalación, su adecuada configuración y detallar sus funcionalidades principales y características para una futura extensión y reutilización por parte de otros componentes. Al final del mismo se encontrará una sección de glosario explicando aquellos términos técnicos que ameritan aclaración y a su vez otra de contacto con los datos de sus desarrolladores en caso de dudas o sugerencias.

## Index

1. [Introducción](#)
2. [Librerías](#)
3. [Pre-Requisitos](#)
4. [Instalación](#)
5. [Consideraciones especiales](#)
  - [Instalación en ambiente productivo](#)
6. [Mantenimiento](#)
7. [Glosario](#)
8. [Contacto](#)

## Introducción

El SOAP Server junto al Gateway forman parte de la solución de adaptar una

aplicación web con un front y back de protocolo REST como comunicación a la arquitectura demandada por ANSES y la conexión al Director. En este caso, el SOAP Server, se encarga específicamente de recibir los request REST a los diferentes métodos expuestos y redirigirlos intactos a la aplicación BACKEND correspondiente. Los servicios que se exponen en la especificación (WSDL) son dinámicos y cargados en base al archivo de configuración detallado más adelante. Es decir, está preparado para exponer N servicios SOAP teniendo cada uno un servicio de BACKEND independiente para su redirección, esto es posible utilizando templating detallado también más adelante en la sección Mantenimiento.

## Librerías

El lenguaje utilizado es Python 2.7 conteniendo a su vez las siguientes librerías disponibles en su repositorio (<https://pypi.python.org/>):

- [Flask](#)
- [flask-spyne](#)
- [flask\\_cors](#)
- [httplib2](#)
- [spyne](#)
- [lxml](#)
- [jinja2](#)

## Pre-Requisitos

1. Para ambiente productivo:

- Python 2.7
- Apache2

2. Para ambiente de desarrollo:

- Python 2.7

## Instalación

Ingresa al directorio raíz del proyecto soap-server-dynamic y ejecutar el setup.py para instalar las librerías:

```
python setup.py build
python setup.py develop
```

Una vez instaladas las librerías podremos ejecutar el código, pero antes de ello debemos armar nuestro propio archivo de configuración para adaptarlo a nuestro entorno. Copiamos alguno de ejemplo desde el directorio /etc y lo editamos:

```
cp etc/soapserver.cfg /my/proyect/configuration/path/  
vim /my/proyect/configuration/path/soapserver.cfg
```

A continuación detallaremos cada una de las secciones del mismo para facilitar y hacer más amena esta tarea:

~~~~~ **Keys that will be inserted on template. Strings must have double quotes ("example") the others just one.**  
NAMESPACE="http://ansesarg.anses.gov.ar" -> Namespace a utilizar por los servicios  
SERVICE\_LOCATION="/soap/retrieve" -> Ubicación del servicio SOAP  
VALIDATOR="None" -> Validador del schema SOAP con el cual se comunicará.  
Por defecto es "None" SERVICES=[ -> Servicios que se deseen exponer en el WSDL, cada uno debe contener un nombre del metodo SOAP y la URL a invocar para redirigir el request  
REST { "function":"to\_json",  
"location":"http://localhost:4000/" }, { "function":"chicocana",  
"location":"http://localhost:4000/" } ]

## App configuration

PORT=4501 -> Puerto donde correrá la aplicación en caso de ejecutarse como standalone ~~~~~

Finalmente podremos correr nuestra aplicación exportando primero la variable de ambiente indicando la configuración a utilizar y luego ejecutando el componente:

```
export  
SOAPSERVER_SETTINGS=/my/proyect/configuration/path/soap  
server.cfg  
python src/runserver.py
```

Deberíamos ver el siguiente output:

```
* Running on http://0.0.0.0:4500/ (Press CTRL+C to  
quit)  
* Restarting with stat
```

Felicidades, el SOAP Server está corriendo! Puede visualizar el WSDL expuesto en <http://localhost:4500/soap/retrieve?wsdl>

## 5. Consideraciones especiales

En el siguiente apartado se encontrarán ciertas consideraciones especiales o problemas comunes a modo de consulta en caso de que se tope con alguna de ellas.

## 5.1 Instalación en ambiente productivo

En un ambiente productivo debe tenerse en cuenta que el componente debe correr en un WEB SERVER. En este caso tomaremos como ejemplo [Apache2](#). Dentro de la raíz del proyecto encontrará un archivo .wsgi listo para ser integrado y configurado en el mismo.

## 6. Mantenimiento

La funcionalidad principal de este componente reside en la utilización de templates a la hora de escribir cada método SOAP. Este template de métodos se puede cambiar y alterar en consecuencia todos los métodos SOAP expuestos. Para ello podremos editar el template ubicado en src/templates/runserver.py y adaptarlo a nuestro gusto. A continuación un ejemplo del mismo:

```
~~~~~ class MethodsExpose(spyne.Service):    target_namespace=
{{namespace}}    service_url_path    =    {{soap_location}}    in_protocol    =
Soap11(validator={{soap_validator}}) out_protocol = Soap11()
```

```
{% for service in services %}
@spyne.srpc(Unicode,Unicode,Unicode,Unicode,Unicode,Unicode,
_returns=Unicode)
def {{service.function}}(url_pattern, content_type, web_method,
params, raw, location):
    #try:
        app.logger.debug("Call service With: url_pattern:{0}
content_type:{1} web_method:{2} raw_data:{3} params:{4}".format(url_pattern,content_type,web_method,raw, params))

        h = http.Http(disable_ssl_certificate_validation=True)

        url = {{service.location}} + url_pattern
        app.logger.debug("Called URL "+ url)
        response, content = h.request(url, web_method.strip()), raw, headers=json.loads(params))
        app.logger.debug("Response is "+ str(response)+ " and content is "+ str(content))
        return content
    #except Exception,ex:
        #    app.logger.exception(ex)
        #    return { "success": False, "data":str(ex), "msg":"Ha ocurrido un error en el servidor SOAP" }
{% endfor %}

@spyne.srpc(_returns=Unicode)
def isalive():
    return "I'm Alive"
```

```
~~~~~
```

## 7. Glosario

- Gateway: El Gateway es otro componente dentro del mismo repositorio y solución que, junto al SOAP Server, proveen el módulo de seguridad completo para adaptar sus aplicaciones FRONT al esquema de seguridad de ANSES. Para más información consulte el README.md en la raíz de su directorio (source/gateway/)

## 8. Contacto

Si desea contactarse con el equipo por alguna duda o sugerencia puede realizarlo a través del siguiente e-mail: [eric@moorea.io](mailto:eric@moorea.io)