PANACEA Soaplab tutorial

WP 3

Author	Marc Poch
Affiliation	UPF
Status	internal report
Version	3.0
Release	draft
Date	2012-05-05
History:	

Abstract: This document is a short tutorial for Soaplab, a Java tool to easily deploy command line tools as web services.

Contents

1. Introduction

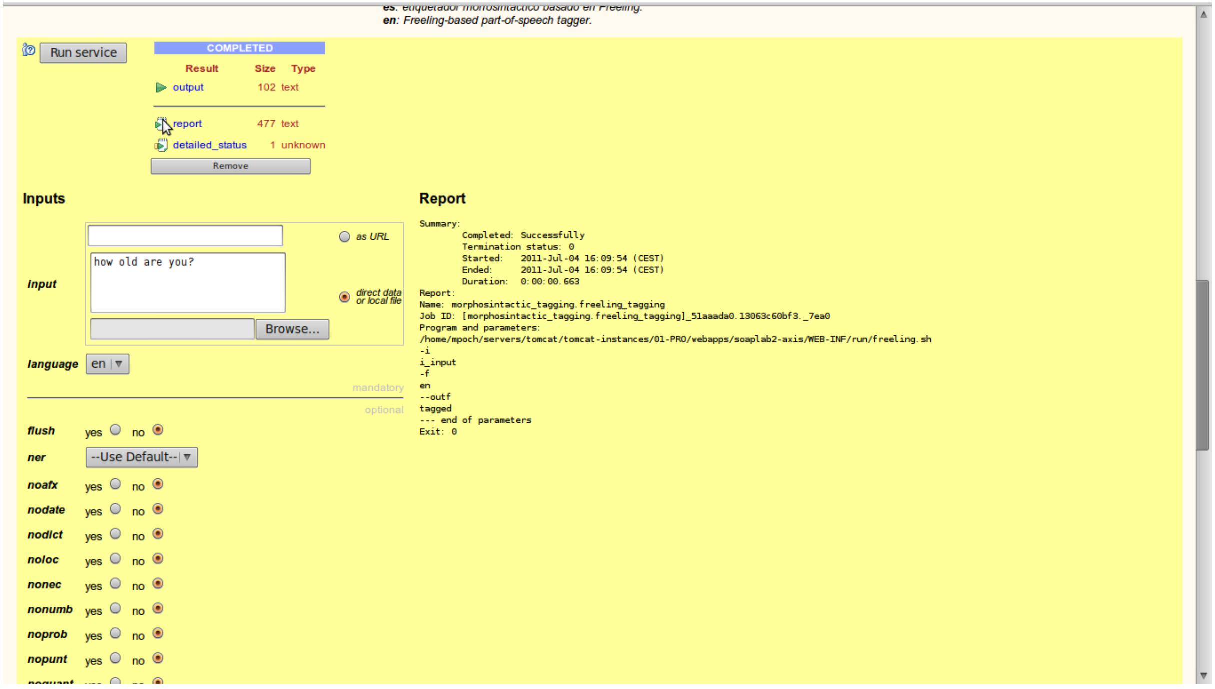
This document is a tutorial for PANACEA web service providers willing to use Soaplab to deploy their web services. It can be used as a guideline for users to access all of the necessary information and material to start working.

The tutorial is divided according to the three development phases of the project: each of them adding new functionalities, corrections or improvements.

2. Platform version 1

Soaplab is a software tool that automatically turns command line applications into web services. The service provider only needs to describe the command line tool and its parameters using metadata and Soaplab can deploy it as a web service. To do so it makes use of a servlet container such as Apache Tomcat.

Once deployed, Soaplab web services can be run using Taverna (http://www.taverna.org.uk), from some scripts (http://www.ebi.ac.uk/Tools/webservices/soaplab/guide) or from the automatically generated Soaplab web client called “Spinet” (Figure 1).



es: *etiquetador morfosintáctico basado en Freeling.*
en: *Freeling-based part-of-speech tagger.*

Run service

COMPLETED

Result	Size	Type
output	102	text
report	477	text
detailed_status	1	unknown

Remove

Inputs

input ○ *as URL* ● *direct data or local file*

```
how old are you?
```

Browse...

language en

mandatory

optional

flush yes ○ no ●

ner --Use Default--

noafx yes ○ no ●

nodate yes ○ no ●

nodict yes ○ no ●

noloc yes ○ no ●

nonec yes ○ no ●

nonumb yes ○ no ●

noprob yes ○ no ●

nopunt yes ○ no ●

Report

```
Summary:
        Completed:  Successfully
        Termination status:  0
        Started:    2011-Jul-04 16:09:54 (CEST)
        Ended:      2011-Jul-04 16:09:54 (CEST)
        Duration:   0:00:00.663
Report:
Name:  morphosintactic_tagging.freeling_tagging
Job ID:  [morphosintactic_tagging.freeling_tagging]_51aaada0.13063c60bf3._7ea0
Program and parameters:
/home/mpoch/servers/tomcat/tomcat-instances/01-PRO/webapps/soaplab2-axis/WEB-INF/run/freeling.sh
-i
i_input
-f
en
--outf
tagged
--- end of parameters
Exit:  0
```

Figure 1: Spinet web client to test and run Soaplab web services

2.1. Technical description summary

Name: Soaplab (Soaplab2)

Version: soaplab2-2.3.1

Servlet container: Apache Tomcat 6

Documentation:
http://soaplab.sourceforge.net/soaplab2
http://www.ebi.ac.uk/Tools/webservices/soaplab/guide

2.2. Describing your command line tool: Metadata

Each command line tool will be described using an ACD file. To build this file follow the instructions and examples from: http://soaplab.sourceforge.net/soaplab2/MetadataGuide.html.

Metadata files must be stored in **src/etc/acd/sowa** folder in your Soaplab2 installation folder (unless you configure Soaplab otherwise).

In PANACEA, we use **infile** parameters with "**comment: default_for_direct**". This way input data can be provided with a **URL** (link) or direct data (a stream of characters). Direct data in Spinet web client can be provided directly by typing in a text box or by uploading a file with "Browse..." button.

2.3. Deployment and configuration

Use *ant gen* to create metadata from ALL ACD files. In PANACEA we use **ant gensowa** only to generate XML from files in the SOWA directory (src/etc/acd/sowa)**.**

Soaplab can be deployed using two protocols. In PANACEA, we use **axis1** because the Taverna workflow editor is not 100% compatible with jaxws.

In PANACEA, we use the "typed interface" to deploy web services because more applications will be able to call these web services(http://soaplab.sourceforge.net/soaplab2/TypedServices.html). Set **typedinterface.enable = true** in your build.properties file and your web services will be deployed using the typed interface.

2.4. Test your web service: Spinet web client

When your web services are deployed in Tomcat you can test them using the Spinet web client. Soaplab automatically deploys this web form based site (Figure 1) where you can run your web services.

2.5. Clients for Soaplab

You can use the **Taverna** workflow editor (http://www.taverna.org.uk/), **Spinet** web client or **perl** programs such as: http://www.ebi.ac.uk/Tools/webservices/soaplab/guide. **More** information can be found in http://soaplab.sourceforge.net/soaplab2/ClientsOverview.html.

3. Platform version 2

News, bugs and improvements' comments for the 2nd version of the platform.

3.1. Bug: Tomcat 7 and Soaplab 2.3.1 Spinet (Solved)

There is a bug when using Soaplab 2.3.1 Spinet web client which makes it impossible to see the job's results when Soaplab is deployed on a Tomcat 7 server.

This problem can be solved by removing (or commenting) two lines from one Spinet file.

File: soaplab2-2.3.1/src/etc/webapps/results.jsp

```
//request = (HttpServletRequest)pageContext.getRequest();
//response = (HttpServletResponse)pageContext.getResponse();
```

3.2. Soaplab output size limit patch

http://myexperiment.elda.org/files/3

This Soaplab patch is used to improve performance and avoid some SOAP messaging errors when dealing with large data outputs using the operations we usually use with Soaplab in PANACEA.

Some Soaplab operations have a SOAP message to get results. For example:

```
<output>all the output is directly transferred here.</output>
<output_url>output_url</output_url>
```

As you can see, for small data this is OK. But for large data it is very inefficient. In fact, the data is transferred twice! Firstly, on the soap message and secondly, when the following web service uses the URL to download the file.

The patch allows the web service provider to put a size limit (in bytes) to outputs by using a property (default value is 1000000 bytes). The data can still be downloaded using output_url so workfows can be run normally:

```
results.sizelimit = 200000
morphosintactic_tagging.freeling_tagging.results.sizelimit = 100000
```

You can set the size limit for a particular web service or for all web services.

We tested this approach with some experiments which were previously not successful (large input files 2MB or more) and it worked. You only need to copy files and recompile web services.

3.3. Soaplab web services limits

Two different web service limitations have been developed: data size limit (explained on the previous section) and requests limits.

Web services can handle multiple requests at the same time allowing for parallelization and faster results. However, server machines have limited resources which must be fairly shared among users.

For second version of the Platform a maximum requests limit has been developed with some scripts. These scripts allow the service provider to configure the maximum number of parallel web services being executed on the server and a queue for waiting requests as well.

The scripts and detailed documentation can be found in (http://myexperiment.elda.org/files/4)

4. Platform version 3

News, bugs and improvements' comments for the third version of the platform.

4.1. Soaplab Version

We recommend the use of the latest Soaplab version: **2.3.2**

http://soaplab.sourceforge.net/soaplab2

4.2. Soaplab Spinet link patch

This patch is used to make links to Soaplab Spinet Web client directly open the desired Web Service form. This is done by adding some Javascript functions to the Soaplab Spinet web client.

http://myexperiment.elda.org/files/6