

Lapps Service Wrapping

Lapps Grid Group
May 26, 2014

Outline

- Overview
- From Software to Web Service
- From NLP Tool to Lapps Service
 - Java Example
 - Python Example
- Conclusion
- Reference

The Language Application Grid

- Availability & Interoperability of NLP Tools
 - Java, Python, tools
 - OpenNLP, Stanford NLP, Gate, NLTK
- Language Application (Lapps) Grid Project
 - Language Service
 - Lapps API Design

Lapps Grid Architecture

Lapps API Design

- Unique Interface
- Discriminator
- JSON Format

Interface (Java)

```
1 package org.lappsgrid.api;
2
3 import jp.go.nict.langrid.commons.rpc intf.Service;
4
5 @Service(namespace = "lapps:service")
6 public interface WebService {
7     /**
8      * Returns the set of data types that must be present in the
9      * input to the {@link #execute(Data)} method
10     */
11     long[] requires();
12
13     /**
14      * Returns the set of data types that will be included in the output.
15     */
16     long[] produces();
17
18     /**
19      * Executes a web service on the given input. Returns the output, if any,
20      * of the web service in a {@link Data} object.
21     */
22     Data execute(Data input);
23
24     /**
25      * Configures a DataSource.
26      * 

27      * Returns any errors in a {@link Data} object. Otherwise returns a Data
28      * object with the "ok" Discriminator type.
29      *
30      * @param config
31      * @return
32     */
33     Data configure(Data config);
34 }
35


```

Lapps Service Wrapping

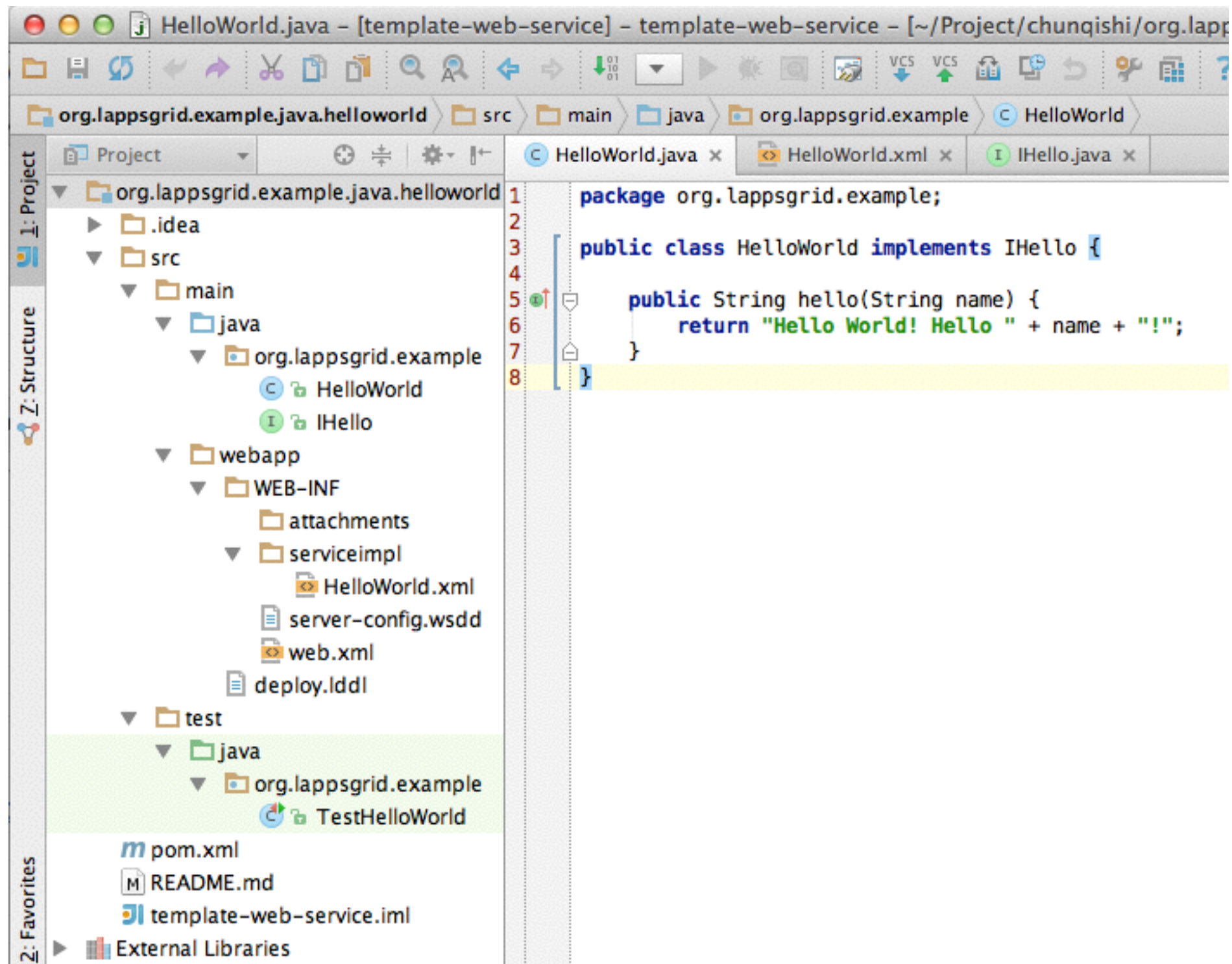
- Service Grid (Web Service) and Service Manager
 - Java Library to WSDL Web Service
 - Service Manager for Register Web Services
- Lapps Service
 - NLP library + Lapps API to Lapps Service
 - Lapps Grid for Register Lapps Service

Service Wrapping Tutorial

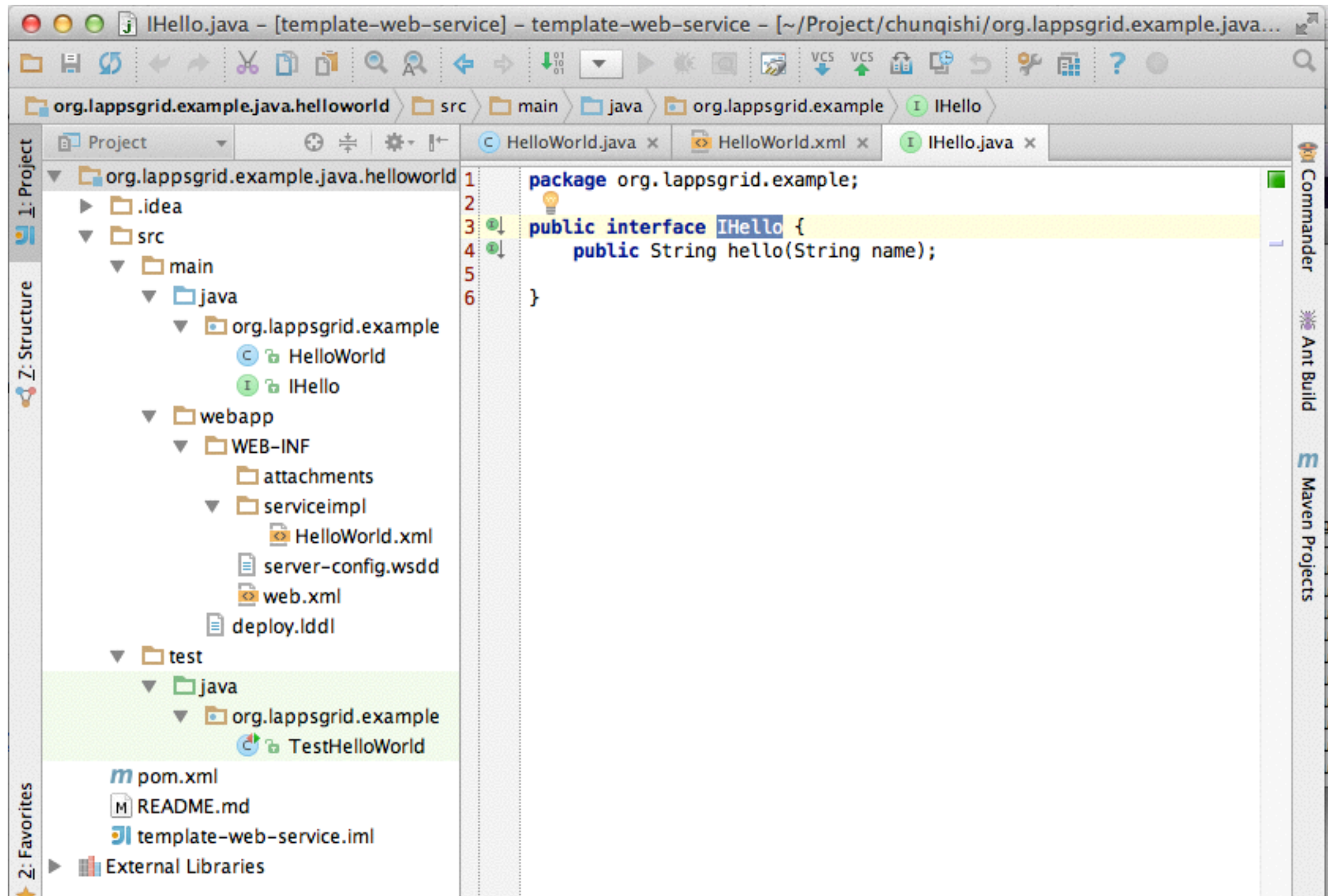
- Web Service: “Hello World!”
 - “Hello World” Program (Java) —> WSDL
- Lapps Service: “Stanford Tagger”
 - Stanford Tagger (Java) + Lapps API —> WSDL
- Lapps Service: “NLTK Tagger”
 - NLTK Tagger (Python) + Lapps API —> WSDL

Web Service Wrapping

Hello World (Java)



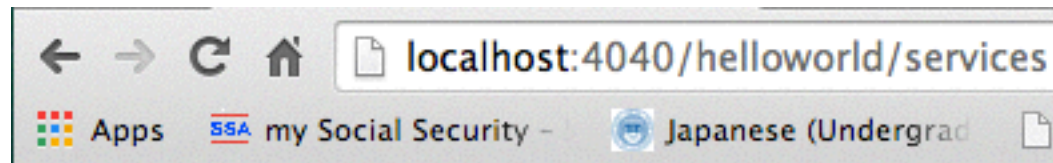
Interface Design



Developing Template

- Developing Template
 - Maven for Dependency Library Management
 - Github Repository
 - <https://github.com/chunqishi/org.lappsgrid.example.java.helloworld>
- Local Test
 - Maven Compile/Package & Jetty Server based Testing
 - Command: *mvn clean package jetty:run*

Web Service WSDL



And now... Some Services

- AdminService ([wsdl](#))
 - AdminService
- Version ([wsdl](#))
 - getVersion
- HelloWorld ([wsdl](#))
 - hello



And now... Some JsonRpc Services

- HelloWorld
 - interfaces
 - IHello
 - String hello(String) [sample] +

LREC [\[invoke\]](#) [\[clear\]](#)

Mon May 19 2014 17:00:09 GMT-0400 (EDT), 148msec. [request:

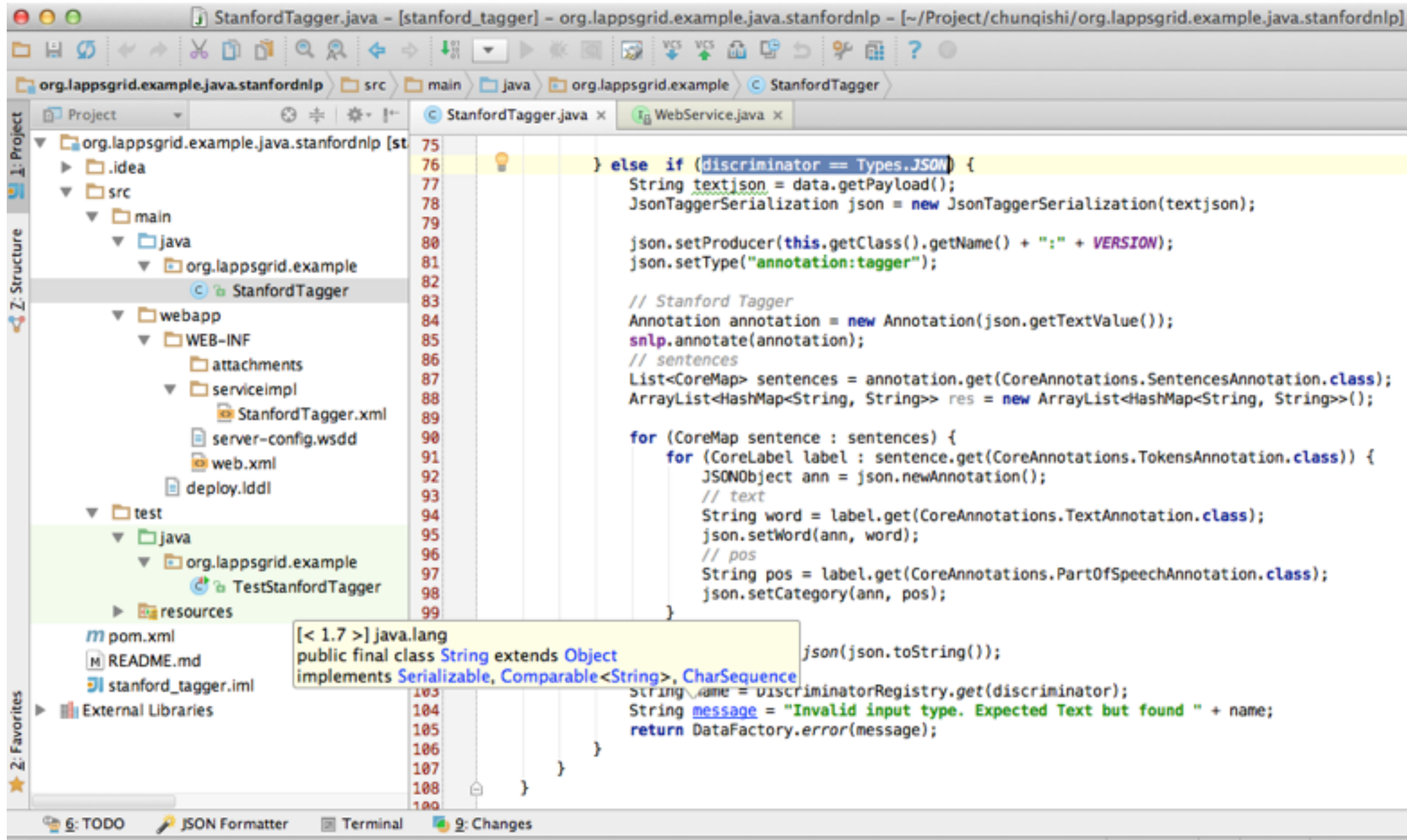
Object	
method	"hello"
params	Array(1)
	0 "LREC"

response:

Object	
error	NULL
headers	Array(0)
	[empty]

Lapps Service Wrapping (Java)

Developing Template



```
StanfordTagger.java - [stanford_tagger] - org.lappsgrid.example.java.stanfordnlp - [~/Project/chunqishi/org.lappsgrid.example.java.stanfordnlp]

org.lappsgrid.example.java.stanfordnlp
src
main
java
org.lappsgrid.example
StanfordTagger

Project Structure
org.lappsgrid.example.java.stanfordnlp [st]
  .idea
  src
    main
      java
        org.lappsgrid.example
          StanfordTagger
            webapp
              WEB-INF
                attachments
                serviceimpl
                  StanfordTagger.xml
                  server-config.wsdd
                  web.xml
                  deploy.ldapl
            test
              java
                org.lappsgrid.example
                  TestStanfordTagger
            resources
  pom.xml
  README.md
  stanford_tagger.iml
  External Libraries

StanfordTagger.java x WebService.java x

75
76 } else if (discriminator == Types.JSON) {
77     String textjson = data.getPayload();
78     JsonTaggerSerialization json = new JsonTaggerSerialization(textjson);
79
80     json.setProducer(this.getClass().getName() + ":" + VERSION);
81     json.setType("annotation:tagger");
82
83     // Stanford Tagger
84     Annotation annotation = new Annotation(json.getTextValue());
85     snlp.annotate(annotation);
86     // sentences
87     List<CoreMap> sentences = annotation.get(CoreAnnotations.SentencesAnnotation.class);
88     ArrayList<HashMap<String, String>> res = new ArrayList<HashMap<String, String>>();
89
90     for (CoreMap sentence : sentences) {
91         for (CoreLabel label : sentence.get(CoreAnnotations.TokensAnnotation.class)) {
92             JSONObject ann = json.newAnnotation();
93             // text
94             String word = label.get(CoreAnnotations.TextAnnotation.class);
95             json.setWord(ann, word);
96             // pos
97             String pos = label.get(CoreAnnotations.PartOfSpeechAnnotation.class);
98             json.setCategory(ann, pos);
99         }
100     }
101
102     json(json.toString());
103     String name = discriminatorRegistry.get(discriminator);
104     String message = "Invalid input type. Expected Text but found " + name;
105     return DataFactory.error(message);
106 }
107 }
108 }
109 }
110 }

[< 1.7 >] java.lang
public final class String extends Object
implements Serializable, Comparable<String>, CharSequence
```

<https://github.com/chunqishi/org.lappsgrid.example.java.stanfordnlp>

Stanford Tagger Wrapping

- Java Wrapping

```
// Stanford Tagger
Annotation annotation = new Annotation(json.getTextValue());
snlp.annotate(annotation);
// sentences
List<CoreMap> sentences = annotation.get(CoreAnnotations.SentencesAnnotation.class);
ArrayList<HashMap<String, String>> res = new ArrayList<HashMap<String, String>>();

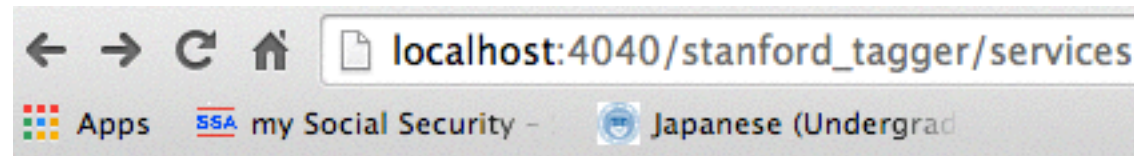
for (CoreMap sentence : sentences) {
    for (CoreLabel label : sentence.get(CoreAnnotations.TokensAnnotation.class)) {
        JSONObject ann = json.newAnnotation();
        // text
        String word = label.get(CoreAnnotations.TextAnnotation.class);
        json.setWord(ann, word);
        // pos
        String pos = label.get(CoreAnnotations.PartOfSpeechAnnotation.class);
        json.setCategory(ann, pos);
    }
}
```

- Jetty Running

```
shis-MacBook-Air:org.lappsgrid.example.java.stanfordnlp shi$
shis-MacBook-Air:org.lappsgrid.example.java.stanfordnlp shi$ export MAVEN_OPTS="-Xmx1024M"
shis-MacBook-Air:org.lappsgrid.example.java.stanfordnlp shi$ mvn jetty:run
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building Java Stanford NLP Tagger Example 0.0.1-SNAPSHOT
[INFO] -----
[INFO]
```


Stanford Tagger Testing

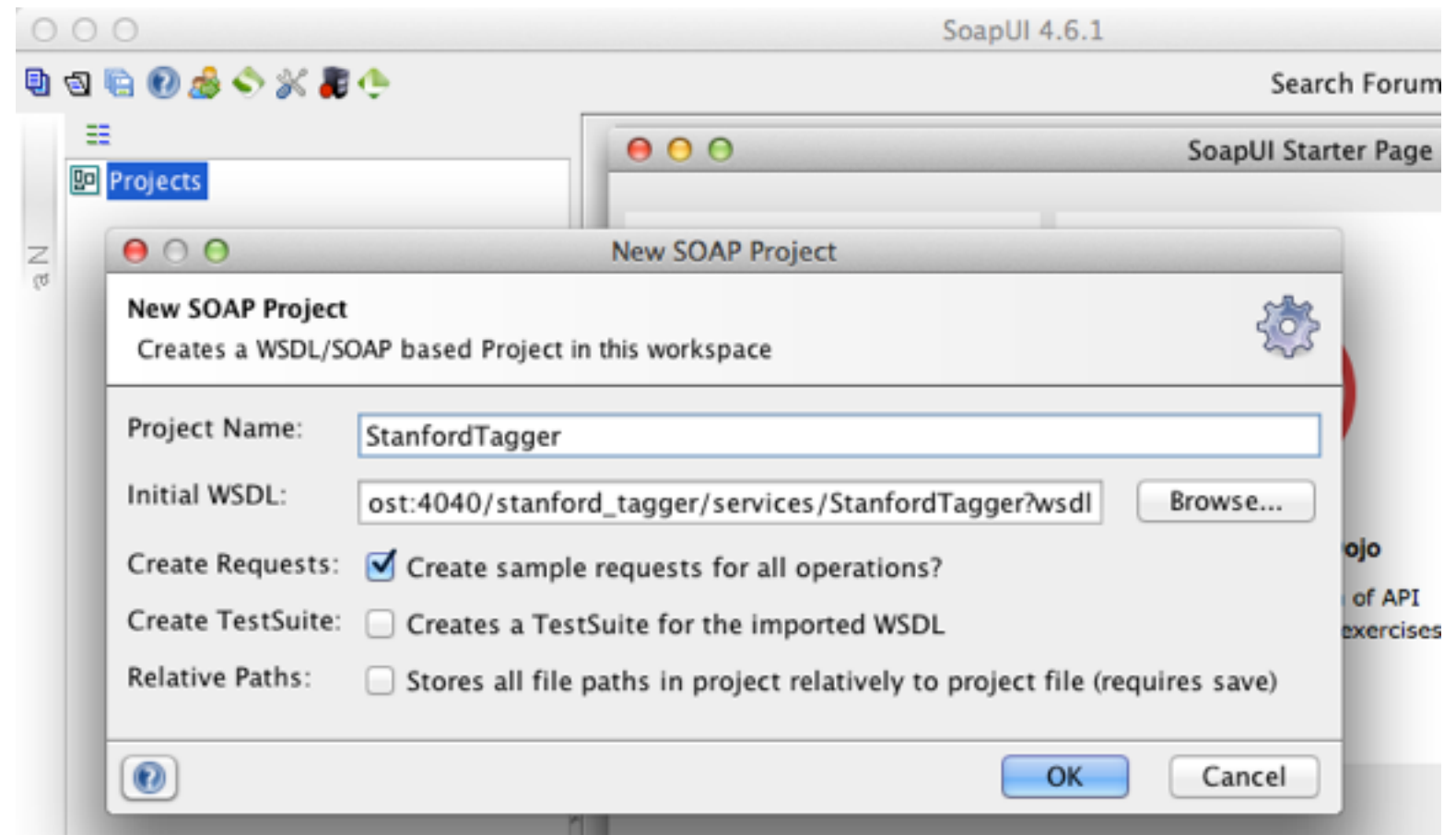
- Local Service



And now... Some Services

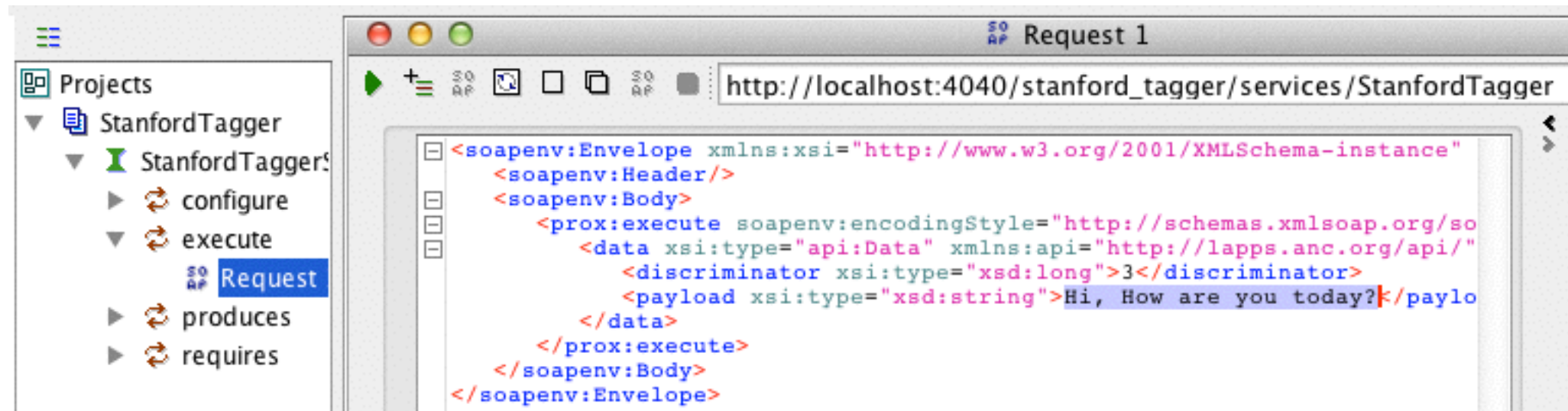
- StanfordTagger ([wsdl](#))
 - execute
 - configure
 - requires
 - produces

- SoapUI Testing



Stanford Tagger Testing Result

- Request



The screenshot shows a web browser window titled "Request 1" with the address bar displaying "http://localhost:4040/stanford_tagger/services/StanfordTagger". The main content area shows a SOAP request XML document. On the left, a "Projects" sidebar lists "StanfordTagger" with sub-items "configure", "execute", "Request" (highlighted), "produces", and "requires".

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <prox:execute soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:prox="http://lapps.anc.org/api/">
      <data xsi:type="api:Data" xmlns:api="http://lapps.anc.org/api/">
        <discriminator xsi:type="xsd:long">3</discriminator>
        <payload xsi:type="xsd:string">Hi, How are you today?</payload>
      </data>
    </prox:execute>
  </soapenv:Body>
</soapenv:Envelope>
```

- Response



The screenshot shows a web browser window with the address bar displaying "http://localhost:4040/stanford_tagger/services/StanfordTagger". The main content area shows a SOAP response XML document. The response contains an "executeResponse" element with a "return" element pointing to a "multiRef" element. The "multiRef" element contains a "payload" element with a JSON object representing the tagging result.

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://lapps.anc.org/api/"
  xmlns:ns2="http://vocab.lappsgrid.org/context-1.0.0.jsonld">
  <soapenv:Header/>
  <soapenv:Body>
    <ns1:executeResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="http://lapps.anc.org/api/">
      <executeReturn href="#id0"/>
    </ns1:executeResponse>
    <multiRef id="id0" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
      <discriminator href="#id1"/>
      <payload xsi:type="soapenc:string">
        {
          "@context": "http://vocab.lappsgrid.org/context-1.0.0.jsonld",
          "metadata": {
            "@value": "Hi, How are you today?",
            "steps": [
              {
                "metadata": {
                  "contains": {
                    "pos": {
                      "producer": "org.lappsgrid.example.StanfordTagger:0.0.1-SNAPSHOT"
                    }
                  }
                }
              }
            ]
          }
        }
      </payload>
    </multiRef>
  </soapenv:Body>
</soapenv:Envelope>
```

Lapps Service Wrapping (Python)

Developing Template

The screenshot displays an IDE window titled "NLTKTagger.java - [nltk_tagger] - org.lappsgrid.example.python.nltk". The left sidebar shows the project structure for "org.lappsgrid.example.python.nltk [nltk_tagger]". The "resources" folder contains the file "nltk_tagger.py", which is highlighted with a black rectangle. The main editor shows the Java code for "NLTKTagger.java". The code is a Java class that implements a tagger. It has a method "tag" that takes a "DataFactory" and a "Text" object as input. The code uses a "discriminator" to determine the input type. If it's "JSON", it uses "JsonTaggerSerialization" to parse the input and call a Python script "tagger" to tag the text. If it's "Text", it uses "TextTaggerSerialization" to parse the input and call a Python script "tagger" to tag the text. The code also includes a "main" method that runs the tagger on a sample input.

```
84 } else if (discriminator == Types.JSON) {
85     String textjson = data.getPayload();
86     JsonTaggerSerialization json = new JsonTaggerSerialization(textjson);
87
88     json.setProducer(this.getClass().getName() + ":" + VERSION);
89     json.setType("annotation:tagger");
90
91     // [('How', 'WRB'), ('are', 'VBP'), ('you', 'PRP'), ('?', '.')]
92     List words = null;
93     try {
94         words = (List)PyCaller.call(pythonFile, "tagger", json.getTextValue());
95     } catch (PyCallerException e) {
96         e.printStackTrace();
97         String message = "Python call error: " + e;
98         return DataFactory.error(message);
99     }
100
101     // NLTK Tagger
102
103     [Maven: org.lappsgrid:JsonSerialization:0.0.3] org.lappsgrid.serialization.json
104     public class JSONObject extends Object
105
106     JSONObject ann = json.newAnnotation();
107     // text
108     String word = (String)token[0];
109     json.setWord(ann, word);
110     // pos
111     String pos = (String)token[1];
112     json.setCategory(ann, pos);
113 }
114 return DataFactory.json(json.toString());
115 } else {
116     String name = DiscriminatorRegistry.get(discriminator);
117     String message = "Invalid input type. Expected Text but found " + name;
118     return DataFactory.error(message);
119 }
```

A tooltip is visible over the line 103, showing the Maven artifact: "[Maven: org.lappsgrid:JsonSerialization:0.0.3] org.lappsgrid.serialization.json public class JSONObject extends Object".

NLTK Python

- Python Program

```
nltk_tagger.py
#!/usr/bin/python
import nltk

def tagger(sent):
    text = nltk.word_tokenize(sent)
    return nltk.pos_tag(text)

if __name__ == "__main__":
    import sys
    print tagger(sys.argv[1])
~
```

- Python Result

```
shis-MacBook-Air:resources shi$ python nltk_tagger.py "Hi, how are you today?"
[('Hi', 'NNP'), (',', ','), ('how', 'WRB'), ('are', 'VBP'), ('you', 'PRP'), ('today', 'NN'), ('?', '.')]
shis-MacBook-Air:resources shi$
```

- Java Wrapping

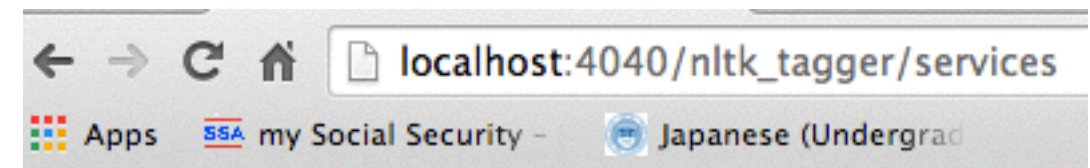
```
// [ how , WRB , ( are , VBP , ( you , PRP , ( , , . ) ]
List words = null;
try {
    words = (List)PyCaller.call(pythonFile, "tagger", json.getTextValue());
} catch (PyCallerException e) {
    e.printStackTrace();
    String message = "Python call error: " + e;
    return DataFactory.error(message);
}
```

- Jetty Running

```
shis-MacBook-Air:org.lappsgrid.example.python.nltk shi$
shis-MacBook-Air:org.lappsgrid.example.python.nltk shi$ mvn jetty:run
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building NLTK Tagger Example 0.0.1-SNAPSHOT
[INFO] -----
[INFO]
```

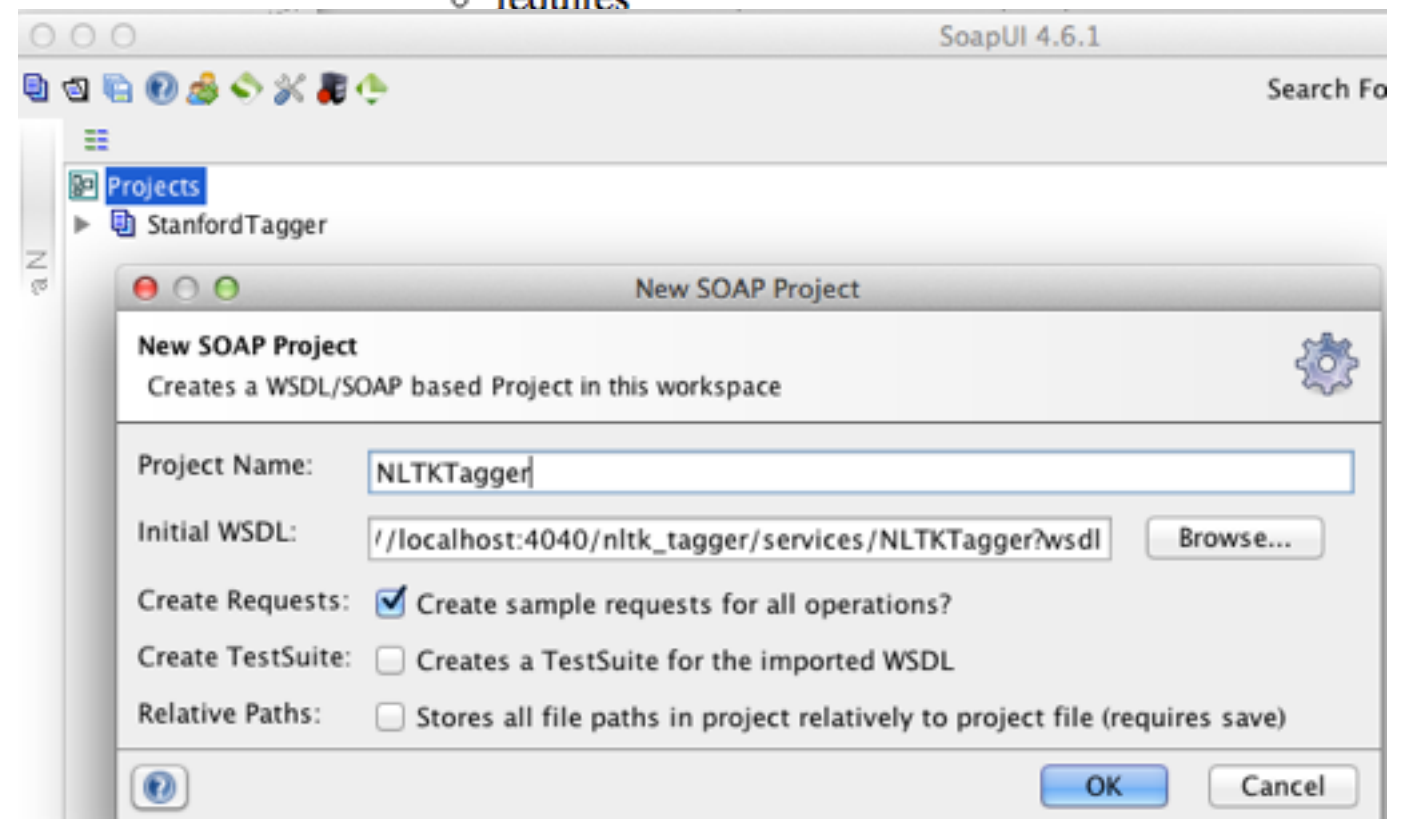
NLTK Tagger Testing

- Local Service
- SoapUI Testing



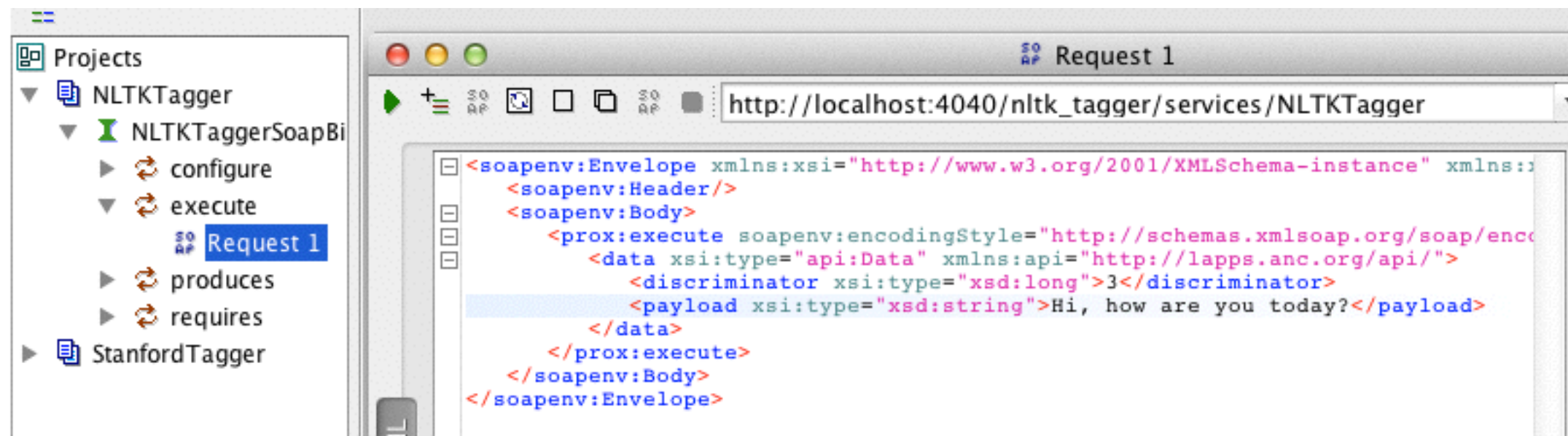
And now... Some Services

- AdminService ([wsdl](#))
 - AdminService
- Version ([wsdl](#))
 - getVersion
- NLTKTagger ([wsdl](#))
 - execute
 - configure
 - requires



NLTK Tagger Testing Result

- Request



- Response



Service Register

- Service Manager

The screenshot shows a web browser window titled "lapps-ubuntu-12.04-desktop-i386 [Running]" with the address bar displaying "localhost:8080/service_manager/language-services". The page is titled "Service Grid Service Manager" and features a sidebar with icons for various system utilities. The main content area is divided into two sections: "Atomic Services" and "Composite Services". Both sections have a "Show All" link and a filter for "For All Users" (selected) or "Members Only". The "Atomic Services" section is sorted by "Ascending order of Service Name" and displays a table with three services: HelloWorld (v0.0.1), NLTKTagger (v0.0.1), and StanfordTagger (v0.0.1). The "Composite Services" section is also sorted by "Ascending order of Service Name" but is currently empty.

lapps_grid_1

Atomic Services [Show All](#)

☒ For All Users ☐ Members Only Sort By: Ascending order of Service Name

Service Name	Service Type	Languages (In Language Code)	Provider	Status
HelloWorld (v0.0.1)	Other Web Service	[en-US]	lapps provider	Run
NLTKTagger (v0.0.1)	LAPPS Web Service	[en-US]	lapps provider	Run
StanfordTagger (v0.0.1)	LAPPS Web Service	[en-US]	lapps provider	Run

Composite Services [Show All](#)

☒ For All Users ☐ Members Only Sort By: Ascending order of Service Name

Service Name	Service Type	Languages (In Language Code)	Provider	Status
-	-	-	-	-

Copyright 2014

Conclusion

- Lapps Service Wrapping Template
 - Java / Python Wrapping
 - Github Repository
- Service Manager for Registering
 - Service Manager Installation Script
- Developing Environment
 - VirtualBox Image: Ubuntu

Reference