

# Contributing to the Lapps Grid Lapps Service Wrapping

Lapps Grid Group  
May 26, 2014

# Outline

Introduction

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

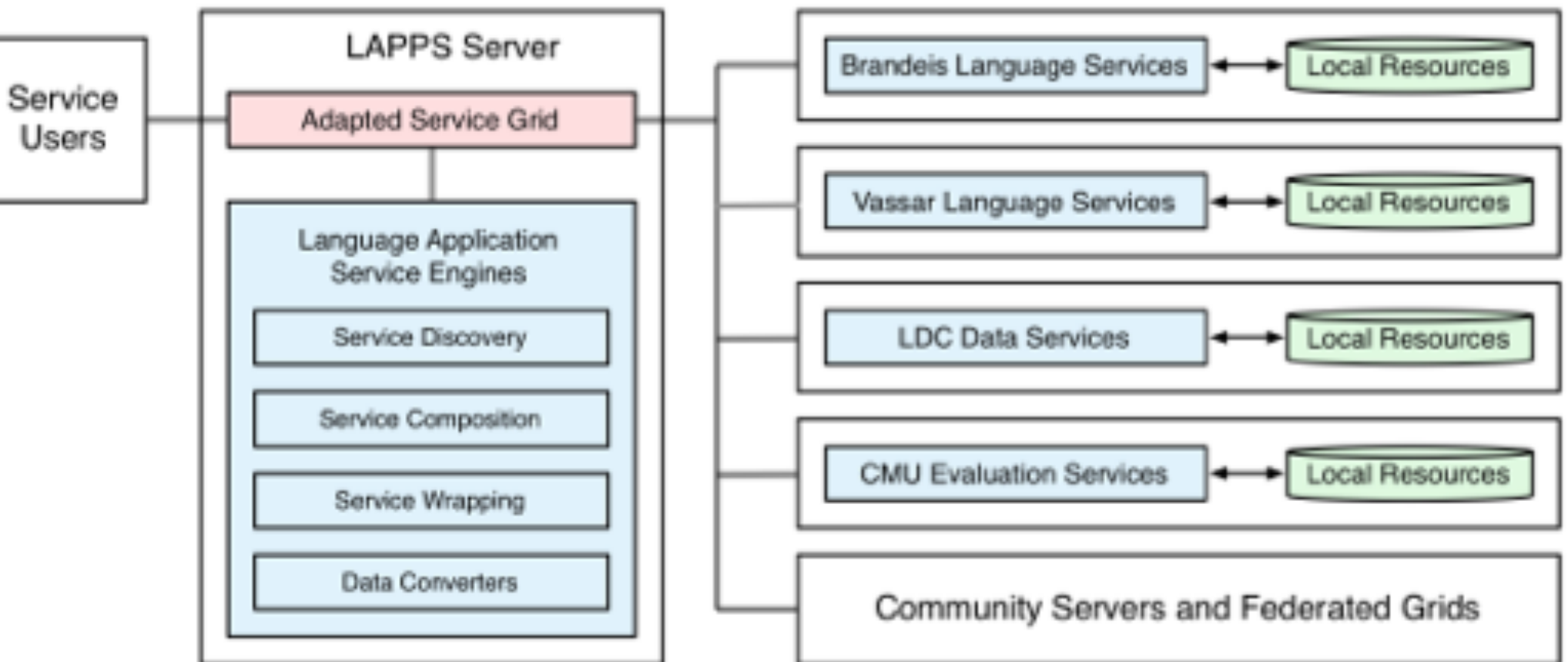
## Language Application (Lapps) Grid Project

- Language Service
- Lapps API Design

# Lapps Grid Architecture

Using Composite  
Lapps Services

Wrapping Atomic  
Lapps Services



# Lapps API Design

Consistent Interface

Discriminator

JSON Format

# Consistent 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      * <p/>
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 }
```

# Discriminator

g [] requires()  
g [] produces()

## Discriminator values

Discriminator	Name
Basic data types	
0	error
1	ok
2	meta
3	text
4	xml
5	string-list
Document types	
1024	document
1025	gate
1026	uima
1027	stanford
1028	opennlp
1029	graf
1030	ptb
1031	json
1032	json-ld

# JSON

```
{  "@context": "http://vocab.lappsgrid.org/context-1.0.0.jsonld",
  "metadata": {},
  "text": {      "@value": "Hi, how are you today?"      },
  "steps": [
    { "metadata": {
      "contains": {
        "Token": {
          "producer": "edu.brandeis.cs.lappsgrid.opennlp.Tokenizer:0.0.4",
          "type": "tokenizer:opennlp"          } } },
      "annotations": [
        { "@type": "Token", "id": "tok0", "start": 0, "end": 2,
          "features": { "word": "Hi" } },
        { "@type": "Token", "id": "tok1", "start": 2, "end": 3,
          "features": { "word": "," } },
        { "@type": "Token", "id": "tok2", "start": 4, "end": 7,
          "features": { "word": "how" } },
        { "@type": "Token", "id": "tok3", "start": 8, "end": 11,
          "features": { "word": "are" } },
        { "@type": "Token", "id": "tok4", "start": 12, "end": 15,
          "features": { "word": "you" } },
        { "@type": "Token", "id": "tok5", "start": 16, "end": 21,
          "features": { "word": "today" } },
        { "@type": "Token", "id": "tok6", "start": 21, "end": 22,
          "features": { "word": "?" } } ]
    }
  ]
}
```



# Contributing to Lapps Grid

## Wrapping Lapps Service

- NLP tools + Lapps API to atomic Lapps service

## Registering to Service Manager

- Atomic Lapps services become available for searching and compositing

# Service Wrapping Tutorial

Web Service: “Hello World!”

- “Hello World” Program (Java) —> WSDL

Lapps Service: “Stanford Tagger”

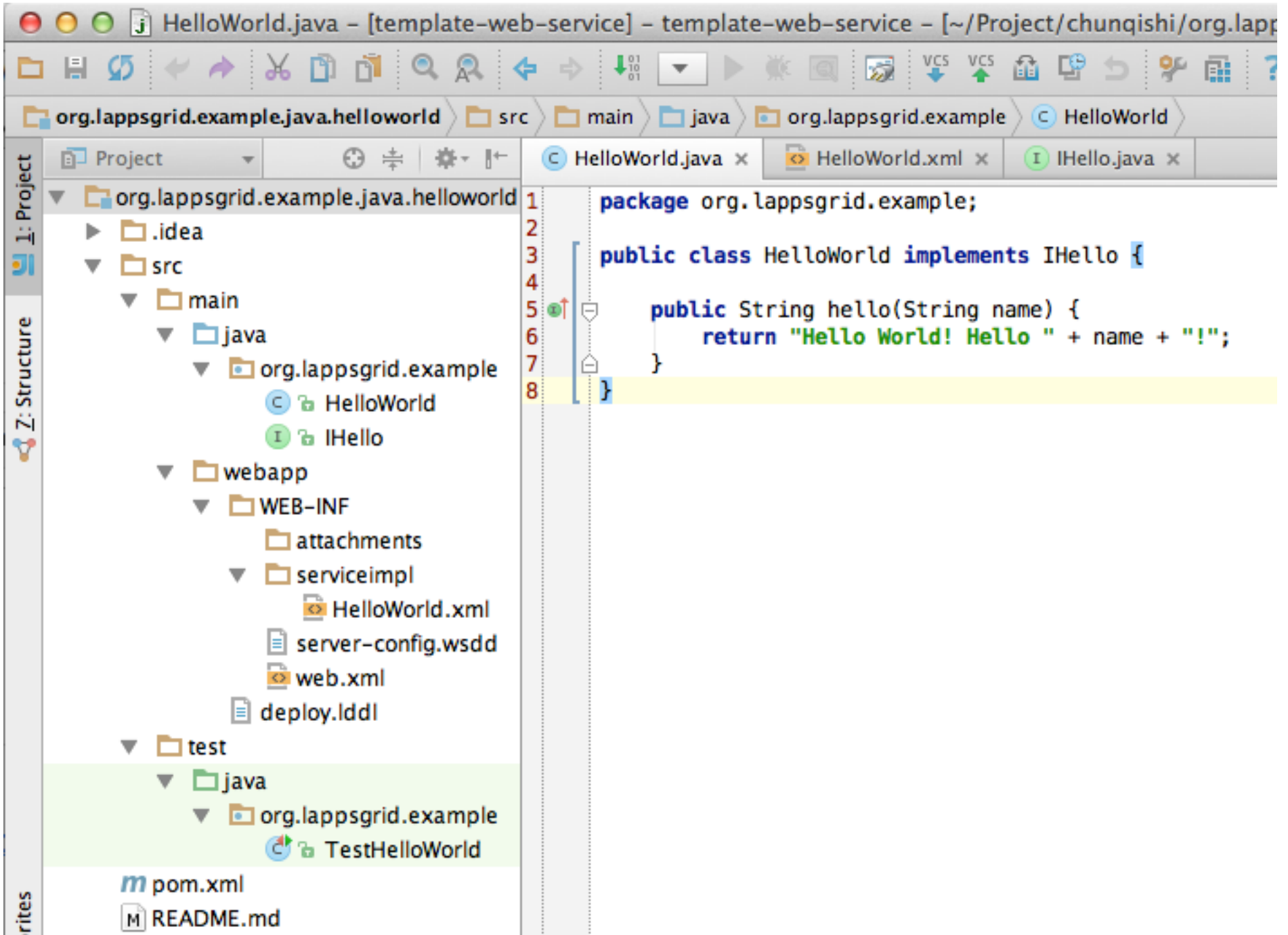
- Stanford Tagger (Java) + Lapps API —> WSDL

Lapps Service: “NLTK Tagger”

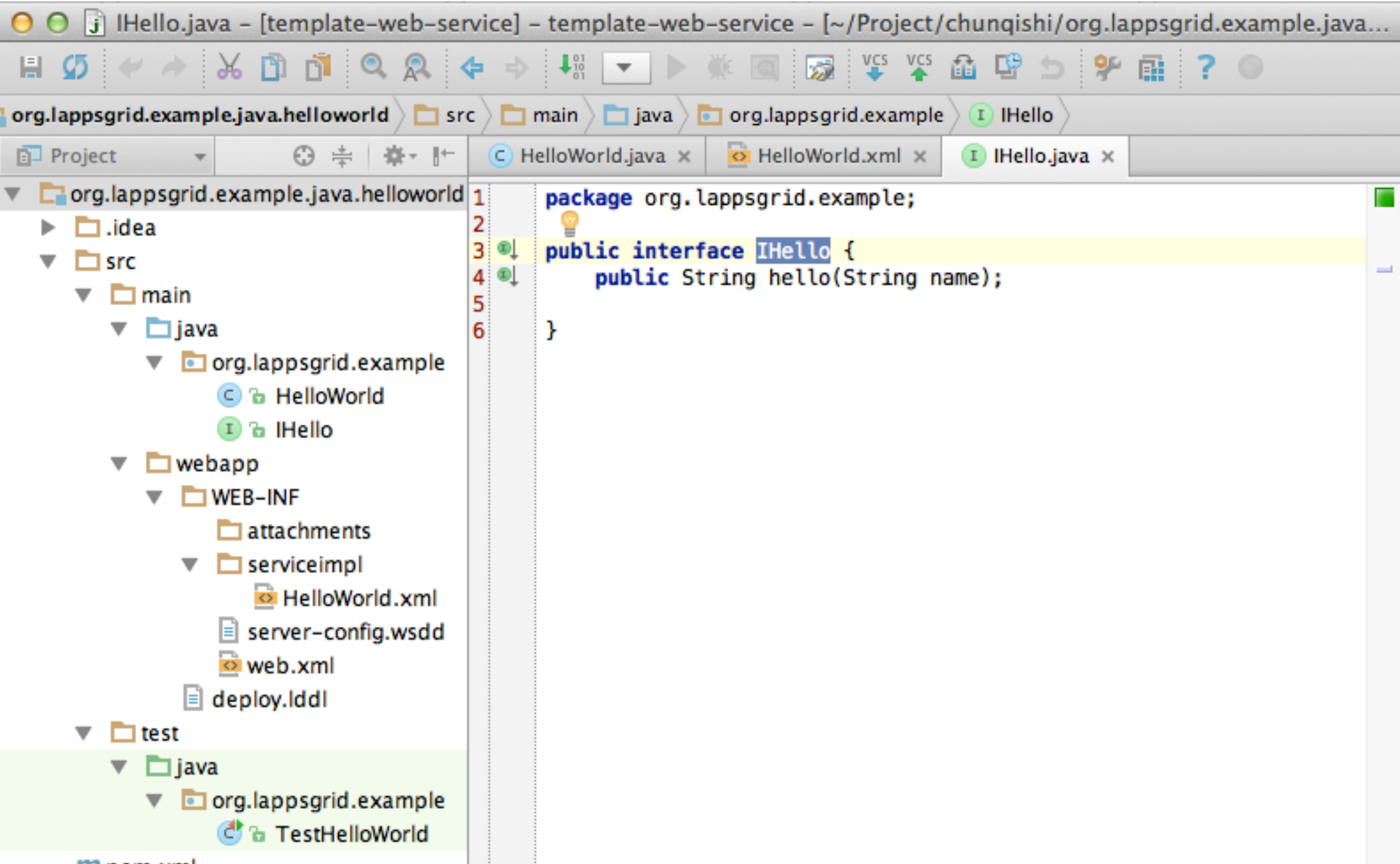
- NLTK Tagger (Python) + Lapps API —> WSDL

# Web Service Wrapping

# Hello World (Java)



# Interface Design



# Developing Template

Developing Template

• Maven for Dependency Library Management

Github Repository

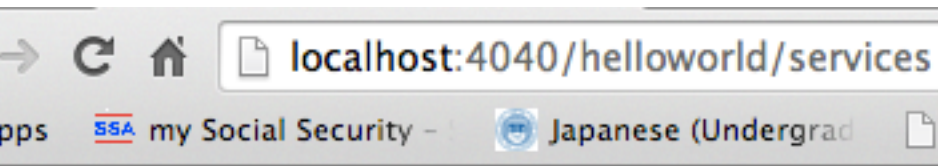
- <https://github.com/chunqishi/org.lappsgrid.example.java.helloworld>

Local Test

• Maven Compile/Package & Jetty Server based Testin

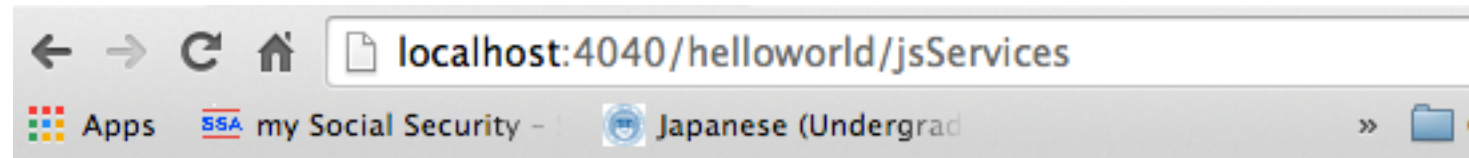
• Command: *mvn clean package jetty:run*

# 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		
--------	--	--

# 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 > org.lappsgrid.example.java.stanfordnlp [st]

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     json(json.toString());
102     String name = discriminatorRegistry.get(discriminator);
103     String message = "Invalid input type. Expected Text but found " + name;
104     return DataFactory.error(message);
105 }
106 }
107 }
108 }
109 }
110 }
```

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

# 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>>();

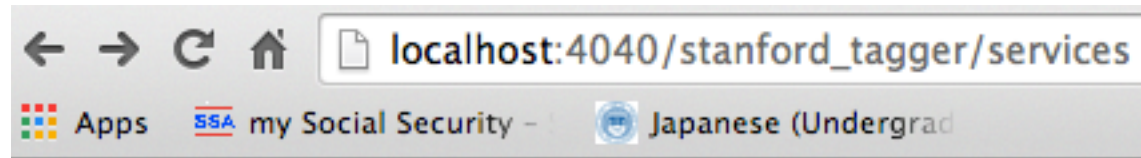
for (CoreMap sentence : sentences) {
    for (CoreLabel label : sentence.get(CoreAnnotations.TokensAnnotation.class)) {
        JSONObject ann = json.newAnnotation();
        // text
        String word = label.get(CoreAnnotations.TextAnnotation.class);
        json.setWord(ann, word);
        // pos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

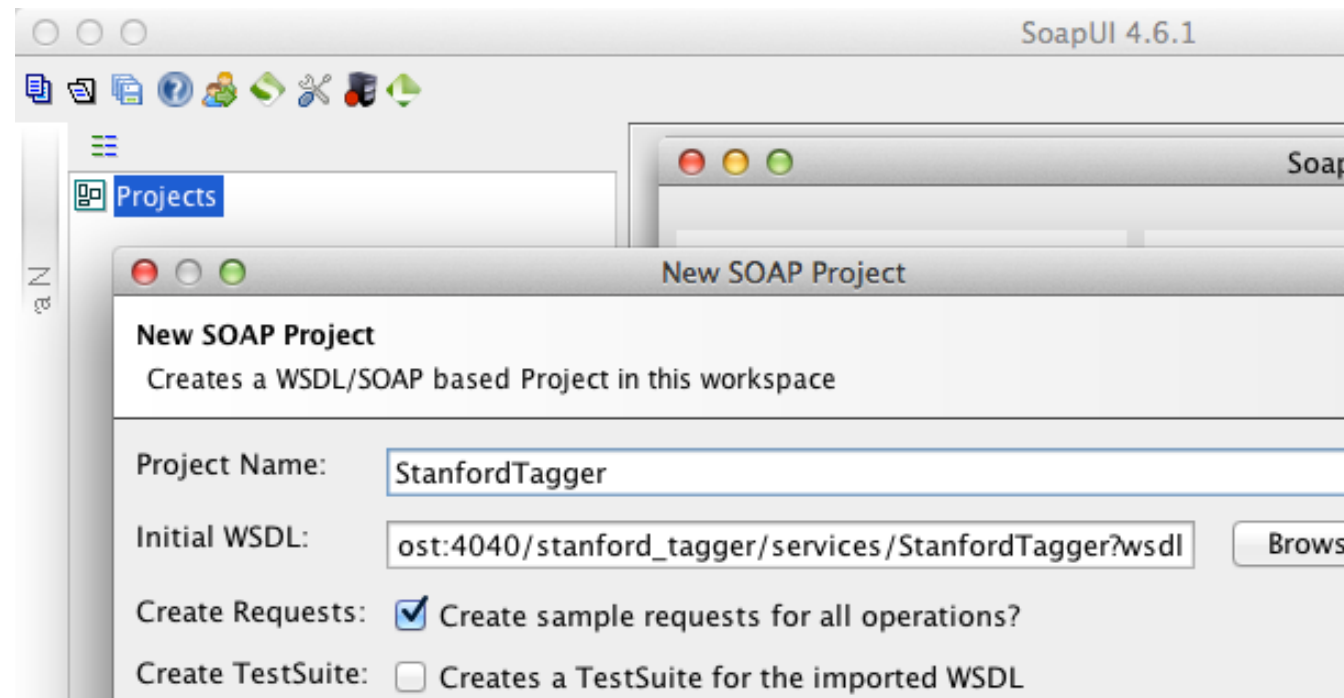
Local Service



## And now... Some Services

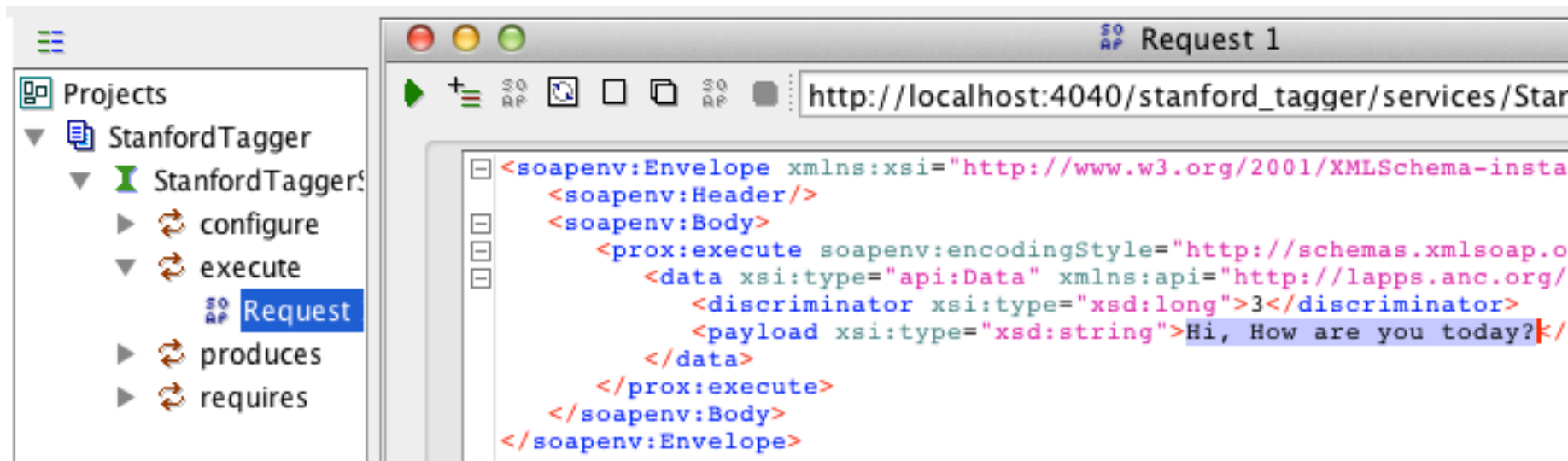
- StanfordTagger ([wsdl](#))
  - execute
  - configure
  - requires
  - produces

SoapUI Testing

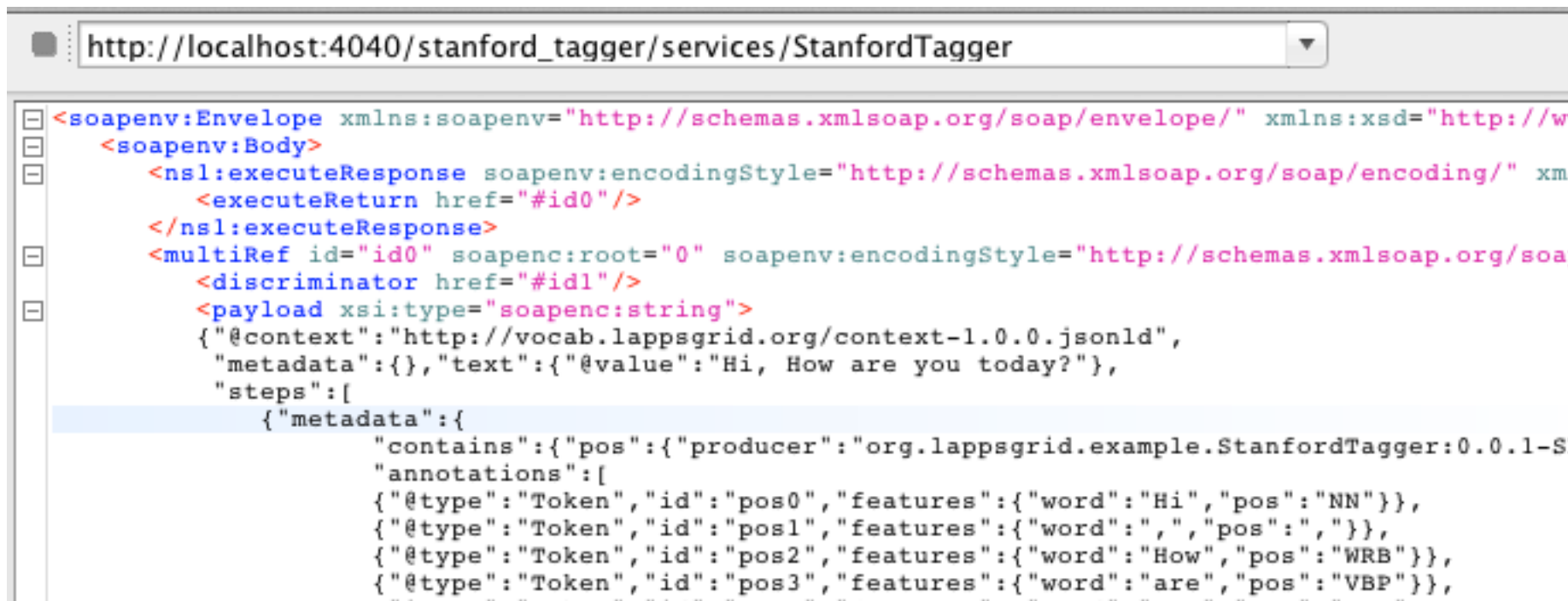


# Stanford Tagger Testing Result

Request



Response



# Lapps Service Wrapping (Python)



# Developing Template

The screenshot displays an IDE window titled "NLTKTagger.java - [nltk\_tagger] - org.lappsgrid.example.python.nltk - [~/Project/chunqishi/org.lappsgrid.example.python.nltk]". The left sidebar shows a project tree with the following structure:

- org.lappsgrid.example.python.nltk [nltk\_tagger] (~/Pro)
  - .idea
  - src
    - main
      - java
        - org.lappsgrid.example
          - NLTKTagger
        - resources
          - nltk\_tagger.py
        - webapp
          - WEB-INF
            - attachments
            - serviceimpl
              - NLTKTagger.xml
            - server-config.wsdd
            - web.xml
            - deploy.lddl
    - test
      - java
      - resources
  - nltk\_tagger.iml
  - pom.xml
  - README.md
  - stanford\_tagger.iml
  - external Libraries

The main editor shows the code for NLTKTagger.java, with line numbers 84 to 116 visible. The code includes JSON serialization and Python calls:

```
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.getTextVal
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);
```

A tooltip is visible over the code, showing the Maven dependency for `org.lappsgrid.serialization.json` and the public class `JSONObject` extending `Object`.

# NLTK Python

Python Program

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

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

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

Python Result

```
shis-MacBook-Air:resources shi$ python nlk_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"), ("today", "NN"), ("", "") }
List words = null;
try {
    words = (List)PyCaller.call(pythonFile, "tagger", json.getText())
} catch (PyCallerException e) {
    e.printStackTrace();
    String message = "Python call error: " + e;
    return DataFactory.error(message);
}
```

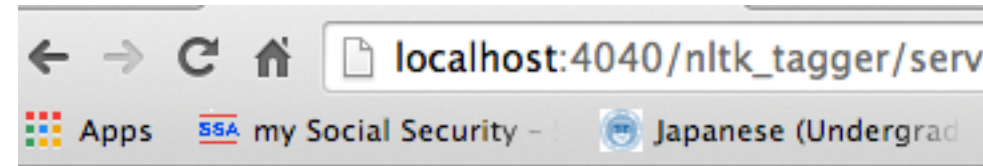
Jetty Running

```
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]
```

# NLTK Tagger Testing

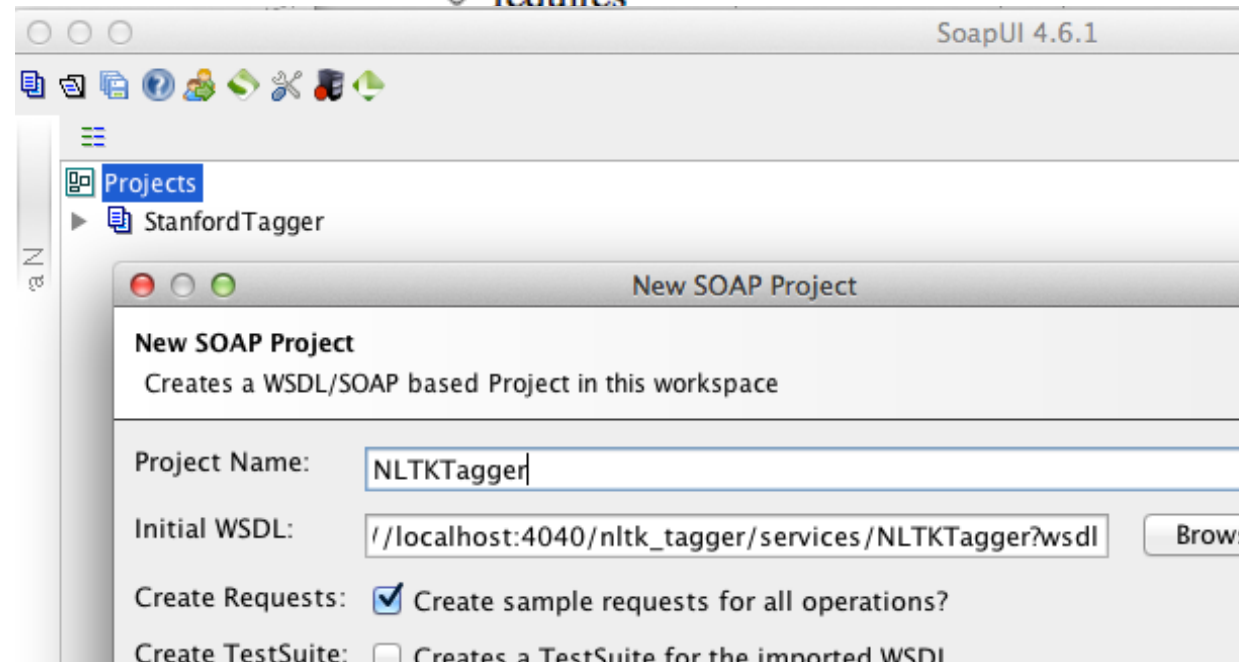
Local Service

SoapUI Testing



## And now... Some Services

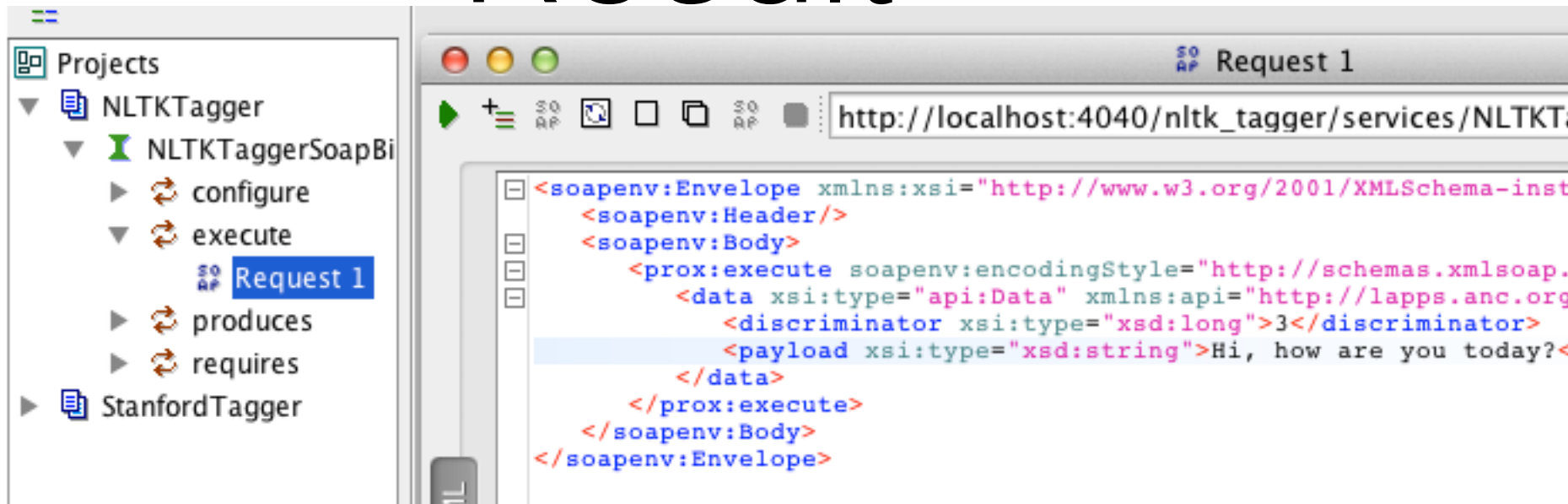
- AdminService ([wsdl](#))
  - AdminService
- Version ([wsdl](#))
  - getVersion
- NLTKTagger ([wsdl](#))
  - execute
  - configure
  - requires



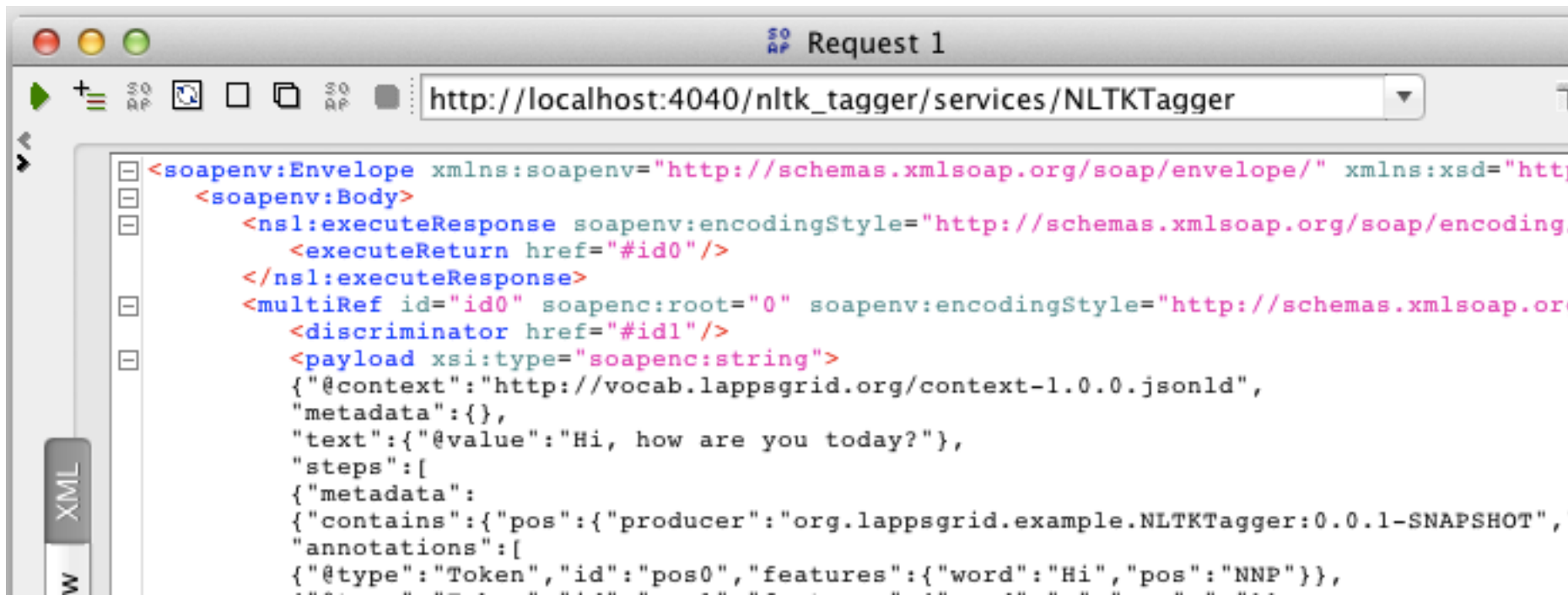


# NLTK Tagger Testing Result

uest



ponse



# Service Register

lapps-ubuntu-12.04-desktop-i386 [Running]

File Edit View History Bookmarks Tools Help

Service Grid Service Manager

localhost:8080/service\_manager/language-services

Manual

lapps\_grid\_1

Atomic Services [Show All](#)

☒ For All Users ☐ Members Only Sort By:

Service Name	Service Type	Languages (in Language Code)	Provider	Status
<a href="#">HelloWorld (v0.0.1)</a>	<a href="#">Other Web Service</a>	[en-US]	<a href="#">lapps provider</a>	Run
<a href="#">NLTKTagger (v0.0.1)</a>	<a href="#">LAPPS Web Service</a>	[en-US]	<a href="#">lapps provider</a>	Run
<a href="#">StanfordTagger (v0.0.1)</a>	<a href="#">LAPPS Web Service</a>	[en-US]	<a href="#">lapps provider</a>	Run

Composite Services [Show All](#)

☒ For All Users ☐ Members Only Sort By:

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

# Conclusion

Contributing to the Lapps Grid

Wrapping Lapps Service

- Java / Python Wrapping
- Templates from Github Repository

Registering into Service Manager

- Service Manager Installation Script

Developing Environment

- VirtualBox Image: Ubuntu

# Reference

API Docs: <http://www.anc.org/projects/lapps/api/project-info.html>

Service Templates:

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

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

<https://github.com/chunqishi/org.lappsgrid.example.python.nltk>

Service Managers:

[http://eldrad.cs-i.brandeis.edu/service\\_manager/language-services](http://eldrad.cs-i.brandeis.edu/service_manager/language-services)

[http://grid.anc.org:8080/service\\_manager/language-services](http://grid.anc.org:8080/service_manager/language-services)

VirtualBox Image:

<http://eldrad.cs-i.brandeis.edu/download/lapps-ubuntu-12.04-desktop-i386.tar.gz>