Wrapping LAPPS Services

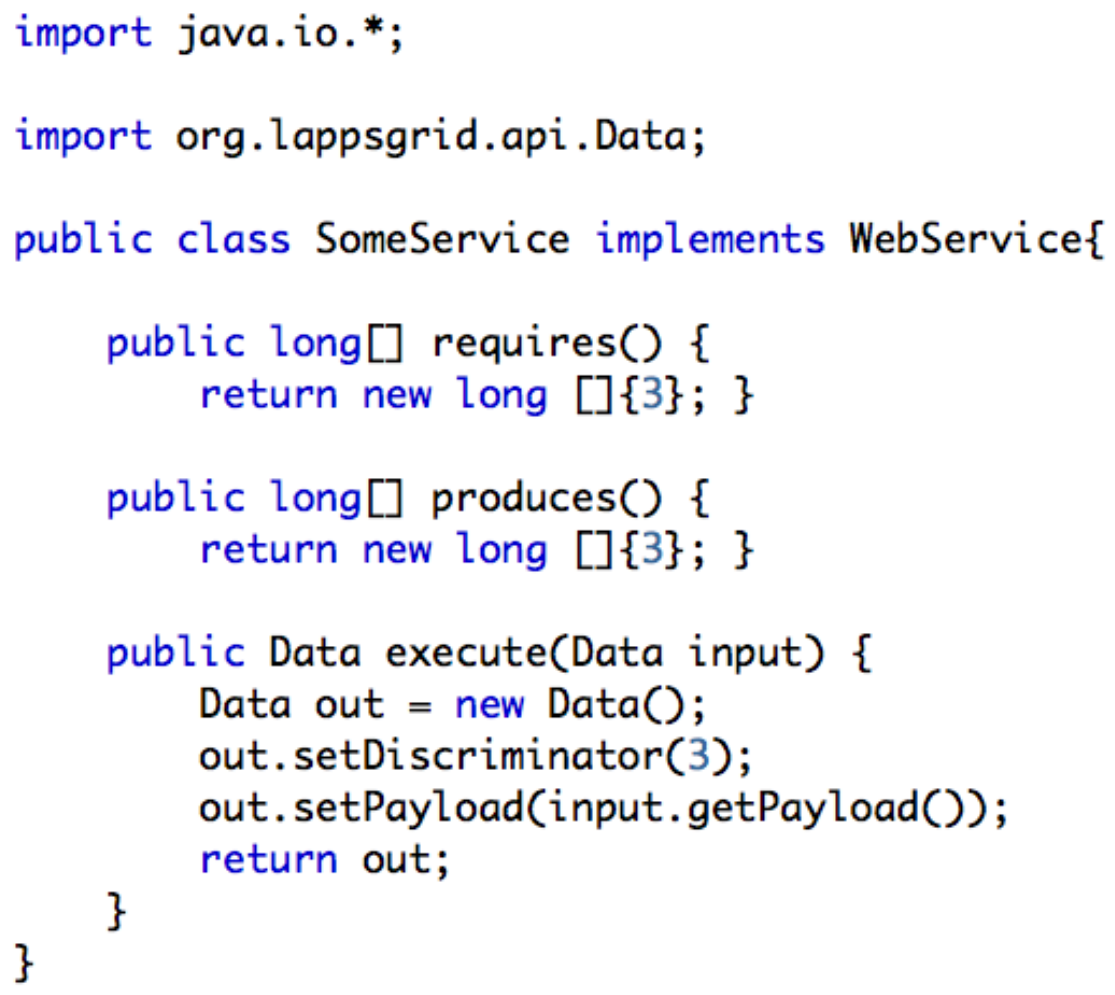
Wrapping a Service

- Preliminaries: Java, Maven and Emacs
- Background:
  - LAPPS API: consistent inteface
  - discriminators
  - JSON format
  - LAPPS Vocabulary
- Wrapping
  - web service sec
  - LAPPS services, various compliance levels
- Deplying and Registering
  - Service Grid and Composer

Wrapping a Service

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

Background: Consistent Interface



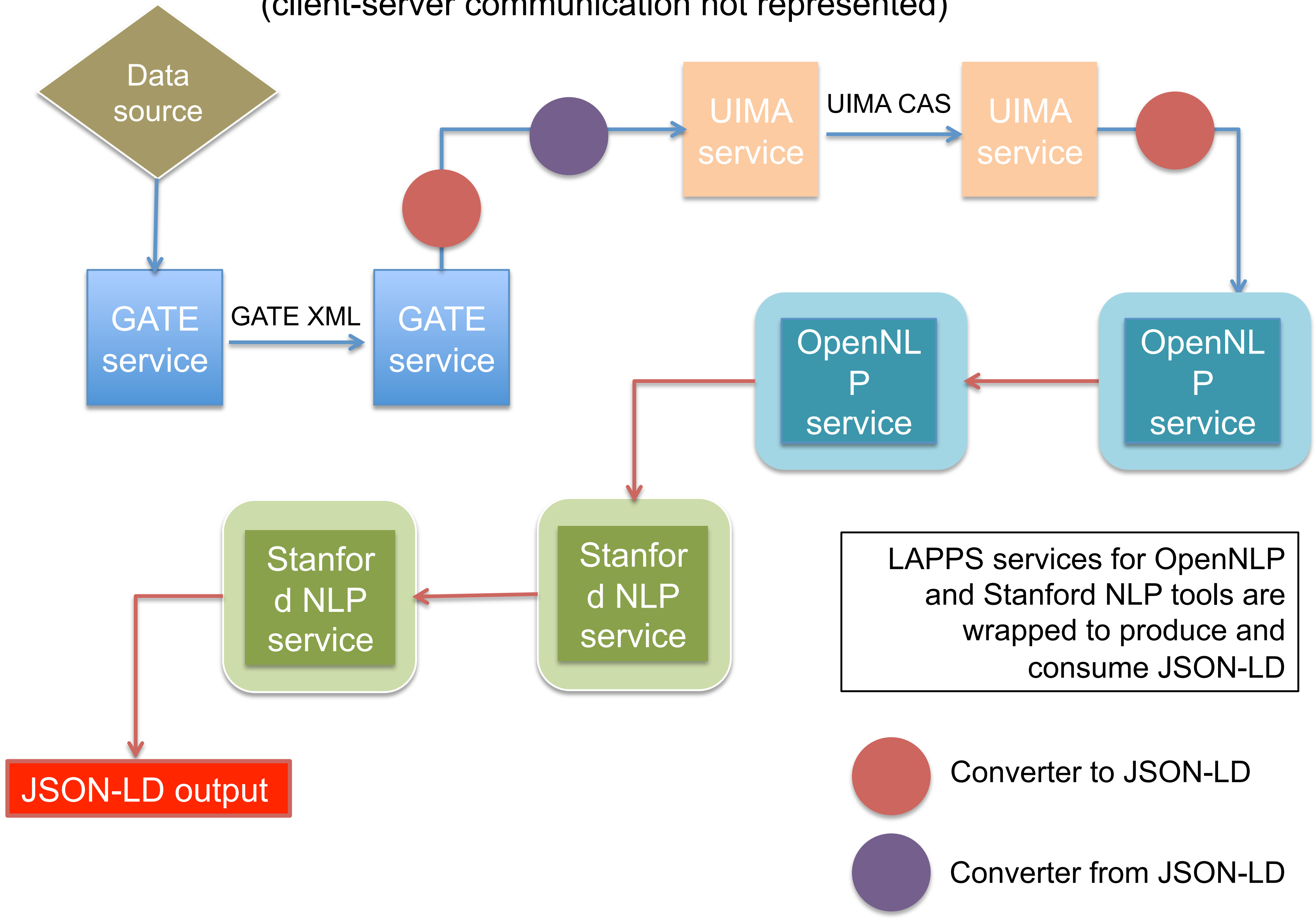
```java
import java.io.*;

import org.lappsgrid.api.Data;

public class SomeService implements WebService{

    public long[] requires() {
        return new long []{3}; }

    public long[] produces() {
        return new long []{3}; }

    public Data execute(Data input) {
        Data out = new Data();
        out.setDiscriminator(3);
        out.setPayload(input.getPayload());
        return out;
    }
}
```

Background: Discriminators

- Used to determine what components can be pipelined
  - Composer
  - Planner
- Types (hypothetical)
  - Identifier Discriminators
  - Format Discriminators
  - Content Discriminators

Logical flow

(client-server communication not represented)



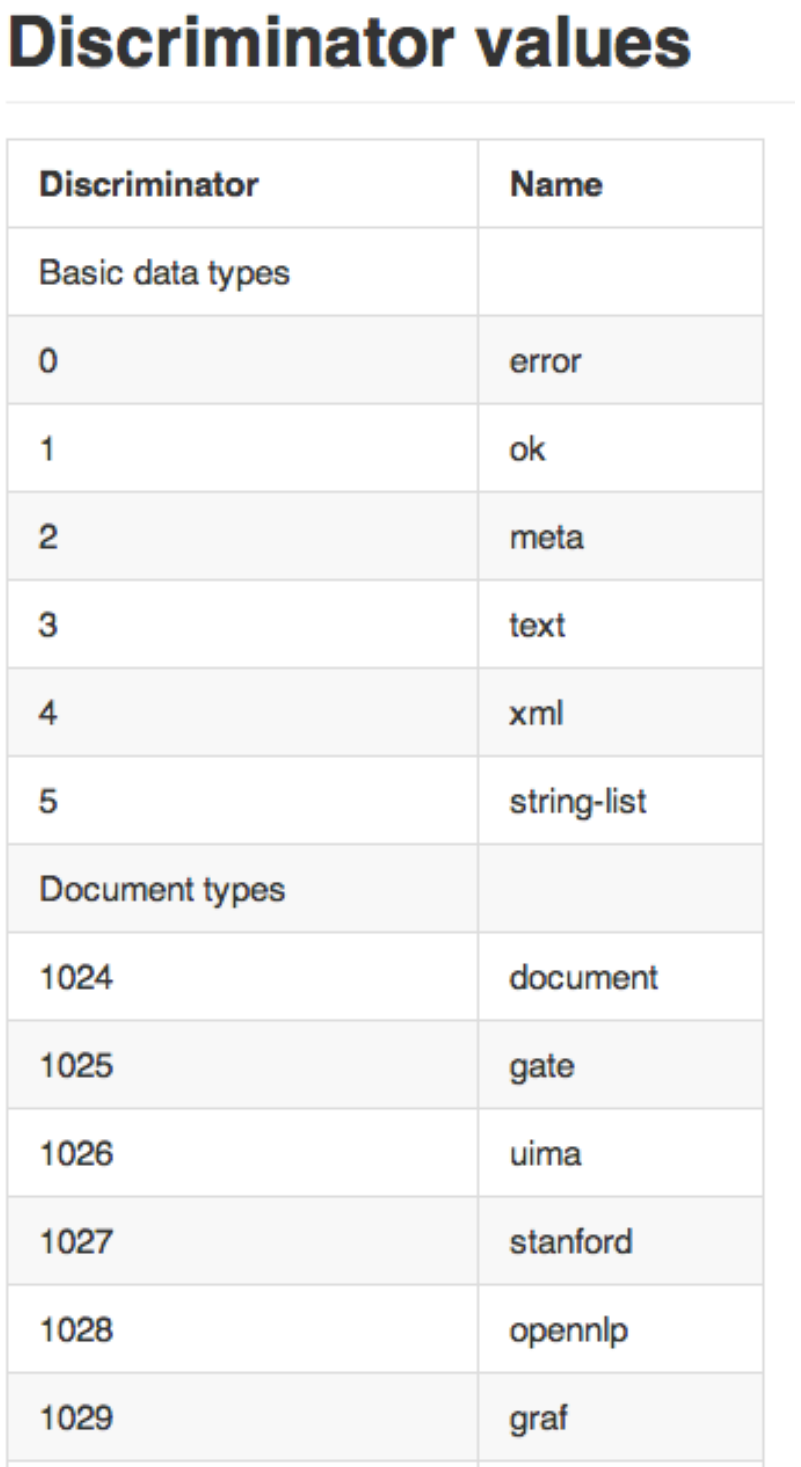
LAPPS services for OpenNLP and Stanford NLP tools are wrapped to produce and consume JSON-LD

Examples (hypothetical)

	requires()
OpenNLP.Splitter	text
OpenNLP.Tokenizer	text OR opennlp-splitter-output
OpenNLP.JsonTokenizer	json AND sentences
OpenNLP.Tagger	opennlp-tokenizer-output OR (text AND tokens AND otpl)
OpenNLP.JsonTagger	json AND tokens

	produces()
OpenNLP.Splitter	opennlp-splitter-output AND sentences
OpenNLP.Tokenizer	opennlp-tokenizer-output AND text AND tokens AND otpl
OpenNLP.JsonTokenizer	json AND tokens
OpenNLP.Tagger	opennlp-tagger-output AND text AND pos
OpenNLP.JsonTagger	json AND postags AND tagset:penn

- long requires()
- log produces()



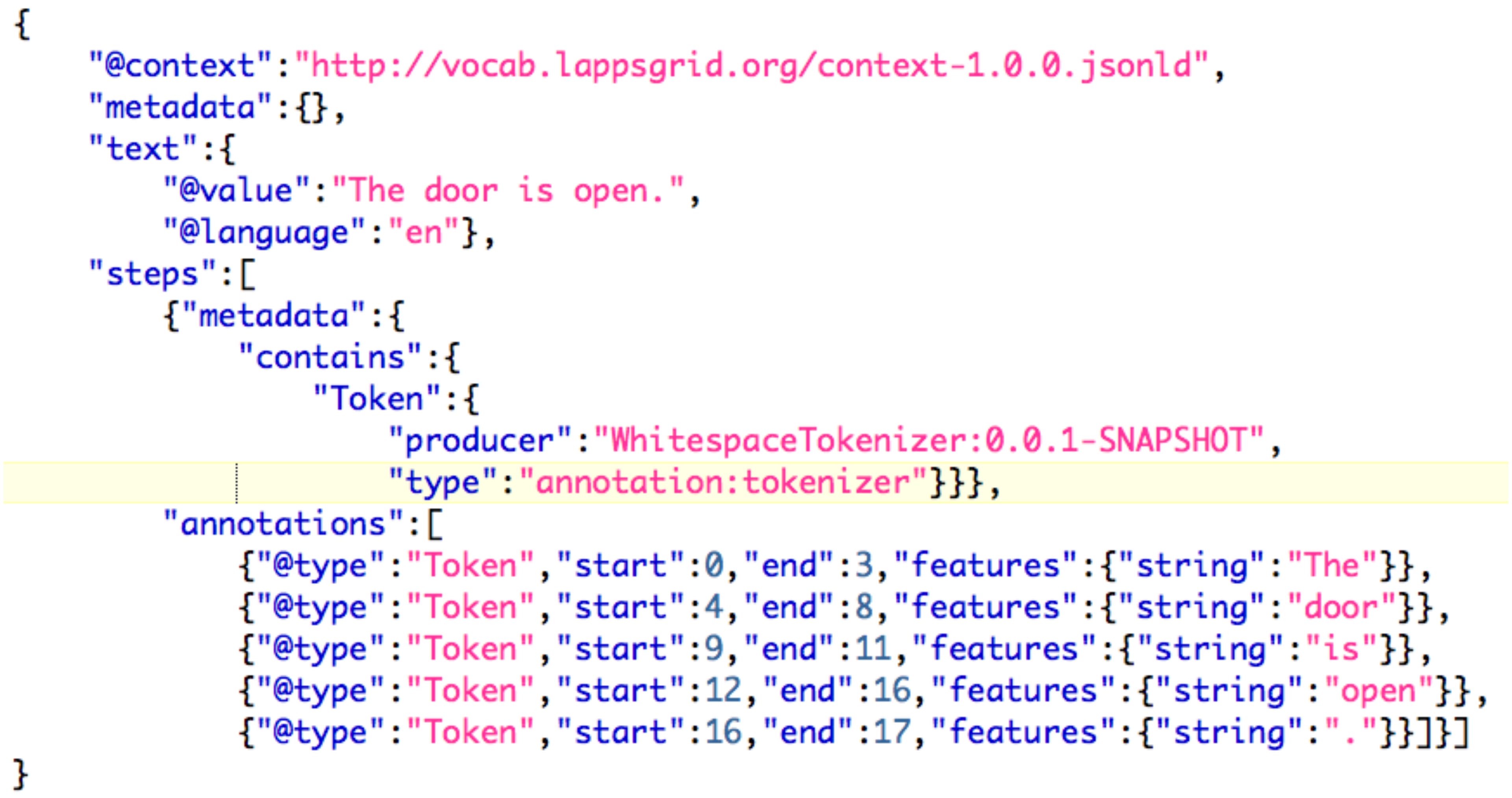
Discriminator values

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

Background: JSON

- Consistent syntax for intermediate data
- All annotations live as JSON objects inside of annotation lists in annotation steps
- Stand-off annotation

JSON – LAPPS Interchange Format



```
{
    "@context":"http://vocab.lappsgrid.org/context-1.0.0.jsonld",
    "metadata":{},
    "text":{
        "@value":"The door is open.",
        "@language":"en"},
    "steps":[
        {"metadata":{
            "contains":{
                "Token":{
                    "producer":"WhitespaceTokenizer:0.0.1-SNAPSHOT",
                    "type":"annotation:tokenizer"}}},
        "annotations":[
            {"@type":"Token","start":0,"end":3,"features":{"string":"The"}},
            {"@type":"Token","start":4,"end":8,"features":{"string":"door"}},
            {"@type":"Token","start":9,"end":11,"features":{"string":"is"}},
            {"@type":"Token","start":12,"end":16,"features":{"string":"open"}},
            {"@type":"Token","start":16,"end":17,"features":{"string":"."}}]}]
}
```

Background: LAPPS Repository

- **http://vocab.lappsgrid.org**

LAPPS Exchange Vocabulary Type Hierarchy

- Thing: alternateName
  - Annotation: producer, rules, id, start, end
    - NamedEntity
      - Date: dateType
      - Location: locType
      - Organization: orgType
      - Person: gender
    - Sentence: sentenceType
    - Token: pos, lemma, orth, tokenType, length
  - Document: source, encoding, language, id
    - TextDocument
    - AudioDocument

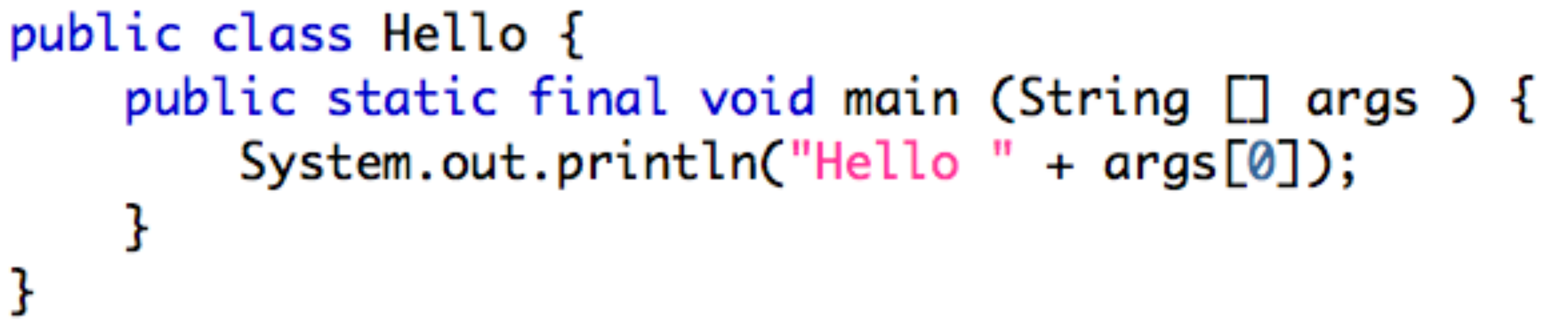
Thing>Annotation>Token

Definition	A string of one or more characters that serves as an indivisible unit for the purposes of morpho-syntactic labeling (part of speech tagging).
Producer type(s)	tokenizer, POStagger
similarTo	http://www.isocat.org/datcat/DC-1403
URI	http://vocab.lappsgrid.org/Token

Properties	Expected Type	Description	sameAs
Metadata (Common Properties) from Token			
posTagset	URI	The POS tagset used for morpho-syntactic tagging.	
Properties from Token			
pos	String or URI	Part-of-speech tag associated with the token.	
lemma	String or URI	The root (base) form associated with the token. URI may point to a lexicon entry.	
tokenType	String or URI	Sub-type such as word, punctuation, abbreviation, number, symbol, etc. Ideally a URI referencing a pre-defined descriptor.	
orth	String or URI	Orthographic properties of the token such as LowerCase, UpperCase, UpperInitial, etc. Ideally a URI referencing a pre-defined descriptor.	
length	Integer	Length of the token.	
Metadata (Common Properties) from Annotation			
producer	List of URI	The software that produced the annotations.	
rules	List of URI	The documentation for the rules that were used to identify the annotations.	
Properties from Annotation			
id	String	A unique identifier associated with the annotation.	
start	Integer	The starting offset (0-based) in the primary data.	
end	Integer	The ending offset (0-based) in the primary data.	
Properties from Thing			
alternateName	String	An alias for the item.	

Creating a Web Service

Take a simple program with just one class



```java
public class Hello {
    public static final void main (String [] args ) {
        System.out.println("Hello " + args[0]);
    }
}
```

Creating a Web Service

Make a simple web service with a class and an interface (in two source files)

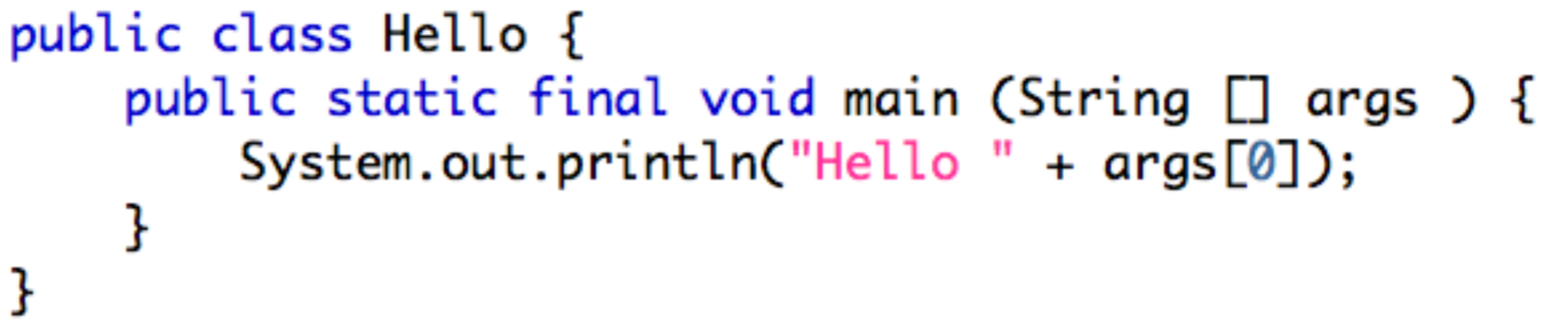
```java
public interface IHello {
    public String hello(String name);
}


public class Hello implements IHello {
    public String hello(String name) {
        return "Hello World! Hello " + name + "!";
    }
}
```

Creating a LAPPS Service (1)

- First level of compliance
- Use the standard LAPPS service interface

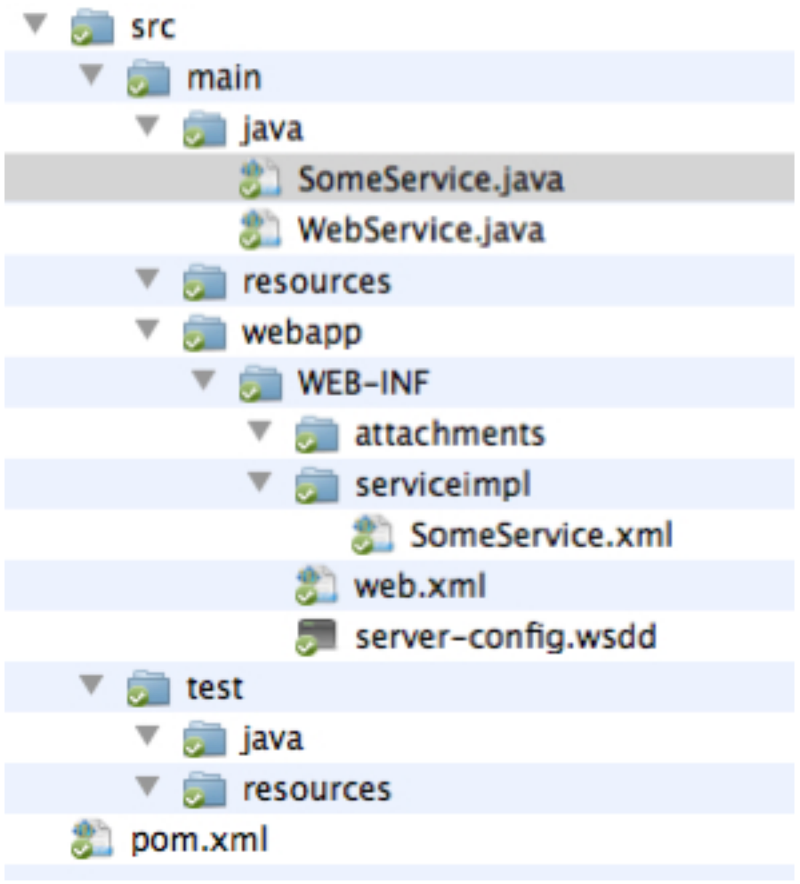
Source code to wrap

- Hello revisited



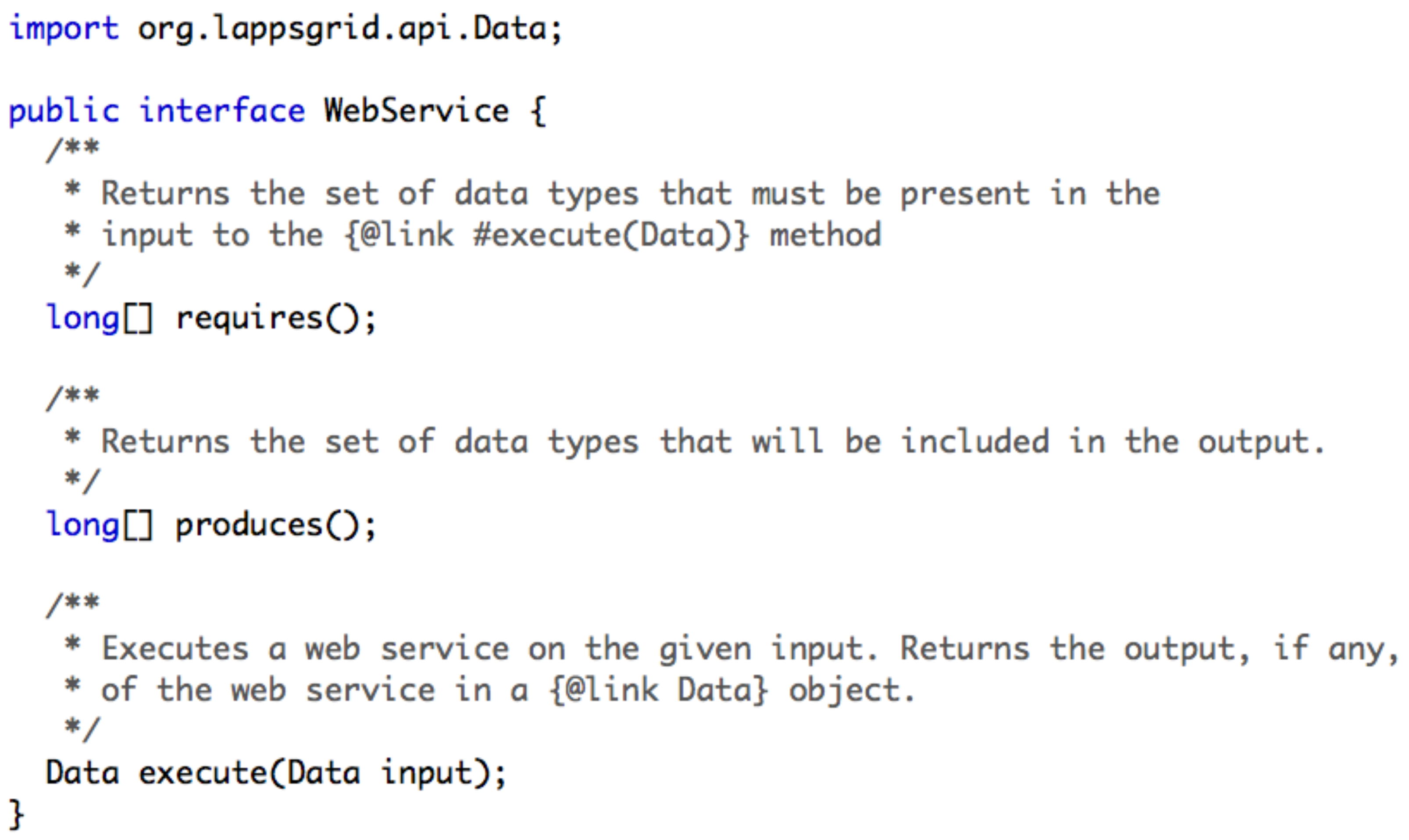
```java
public class Hello {
    public static final void main (String [] args ) {
        System.out.println("Hello " + args[0]);
    }
}
```

Project Template



```
src
    main
        java
            SomeService.java
            WebService.java
        resources
        webapp
            WEB-INF
                attachments
                serviceimpl
                    SomeService.xml
                web.xml
                server-config.wsdd
    test
        java
        resources
pom.xml
```

Web Service Interface



```java
import org.lappsgrid.api.Data;

public interface WebService {
  /**
   * Returns the set of data types that must be present in the
   * input to the {@link #execute(Data)} method
   */
  long[] requires();

  /**
   * Returns the set of data types that will be included in the output.
   */
  long[] produces();

  /**
   * Executes a web service on the given input. Returns the output, if any,
   * of the web service in a {@link Data} object.
   */
  Data execute(Data input);
}
```

The Standard Service Class

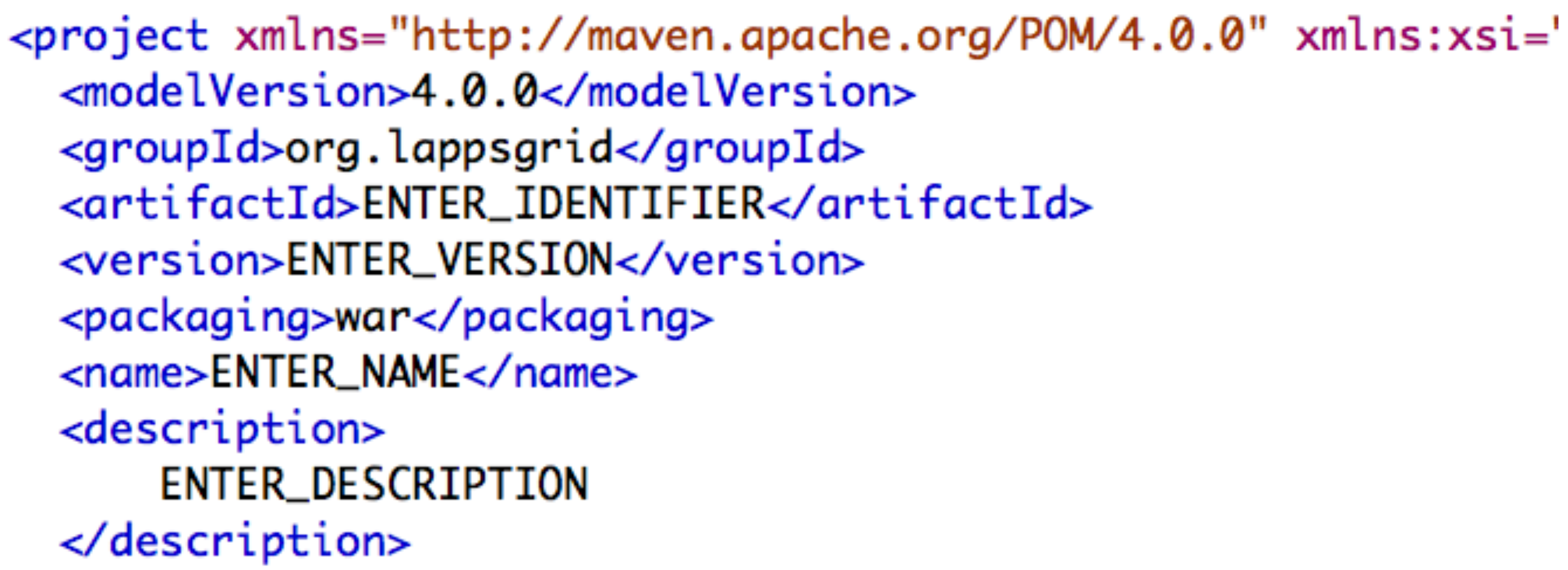
```
import java.io.*;

import org.lappsgrid.api.Data;

public class SomeService implements WebService{

    public long[] requires() {
        return new long []{3}; }

    public long[] produces() {
        return new long []{3}; }

    public Data execute(Data input) {
        Data out = new Data();
        out.setDiscriminator(3);
        out.setPayload(input.getPayload());
        return out;
    }
}
```

Hello Adapted

```java
import java.io.*;
import org.lappsgrid.api.Data;

public class Hello implements WebService {

    public static final void main (String [] args ) {
        System.out.println("Hello " + args[0]);
    }

    public long[] requires() {
        return new long []{3};
    }

    public long[] produces() {
        return new long []{3};
    }

    public Data execute(Data input) {
        Data out = new Data();
        out.setDiscriminator(3);
        out.setPayload("Hello " + input.getPayload());
        return out;
    }
}
```

Editing the POM file

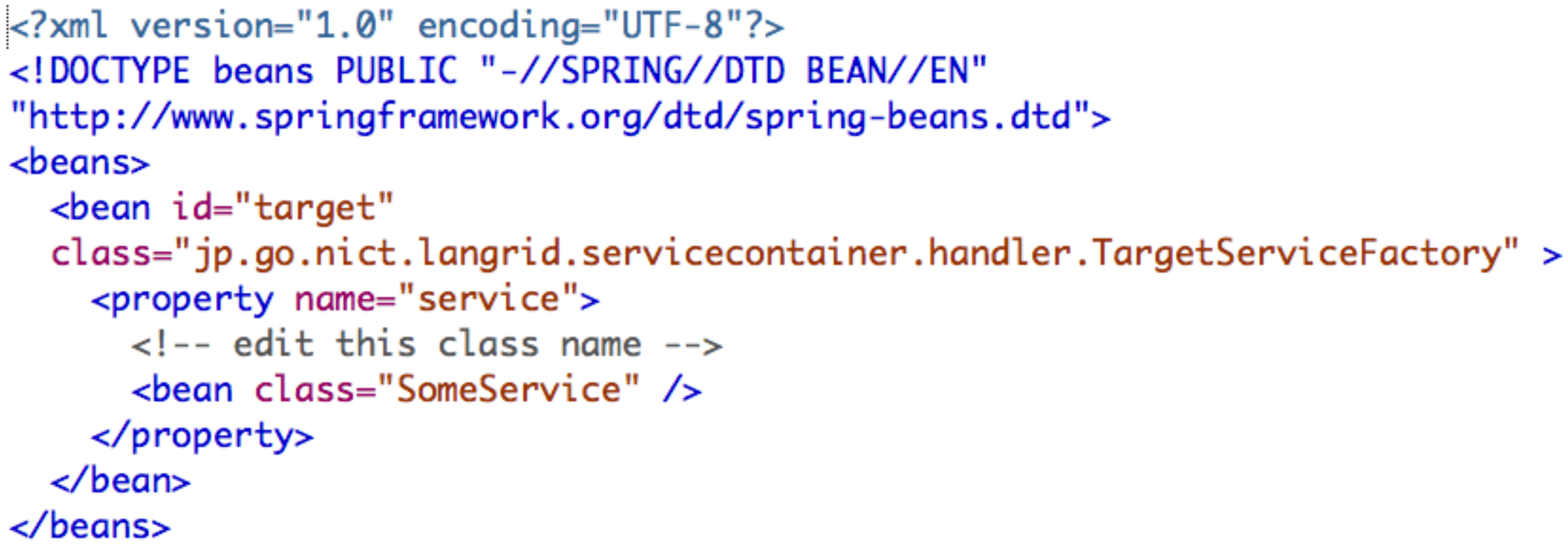
- POM: Project Object Model
- Maven’s way to declare elements of a project



```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi='
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.lappsgrid</groupId>
  <artifactId>ENTER_IDENTIFIER</artifactId>
  <version>ENTER_VERSION</version>
  <packaging>war</packaging>
  <name>ENTER_NAME</name>
  <description>
      ENTER_DESCRIPTION
  </description>
```

Define what class to use

- Rename SomeService.xml
  - src/main/webapp/WEB-INF/serviceimpl
- Define top-level class for service

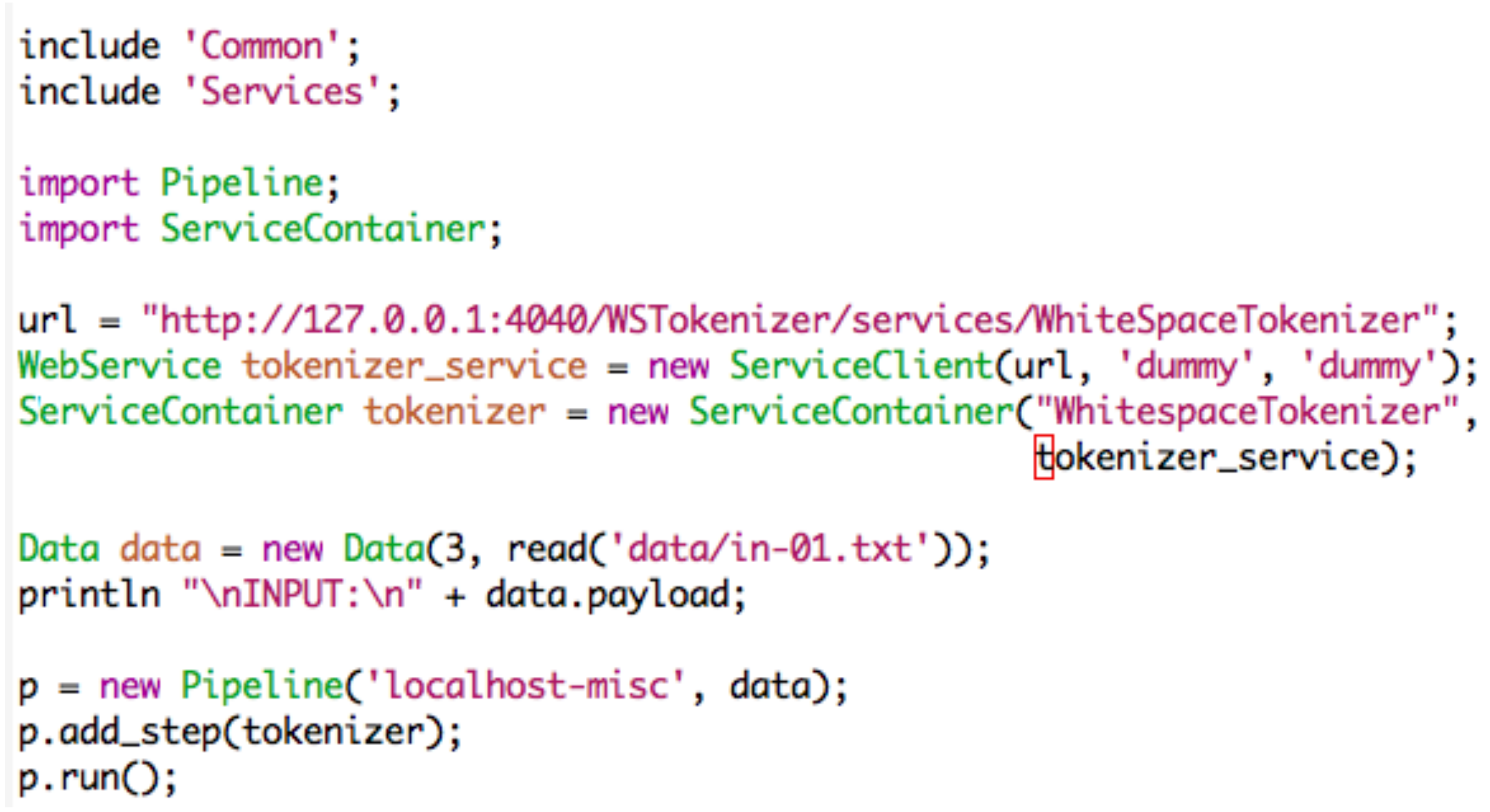


```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
"http://www.springframework.org/dtd/spring-beans.dtd">
<beans>
  <bean id="target"
  class="jp.go.nict.langrid.servicecontainer.handler.TargetServiceFactory" >
    <property name="service">
      <!-- edit this class name -->
      <bean class="SomeService" />
    </property>
  </bean>
</beans>
```

Maven

- mvn clean
- mvn compile
- mvn package
- mvn jetty:run
- (mvn clean compile package jetty:run)

LSD - LAPPS Services DSL

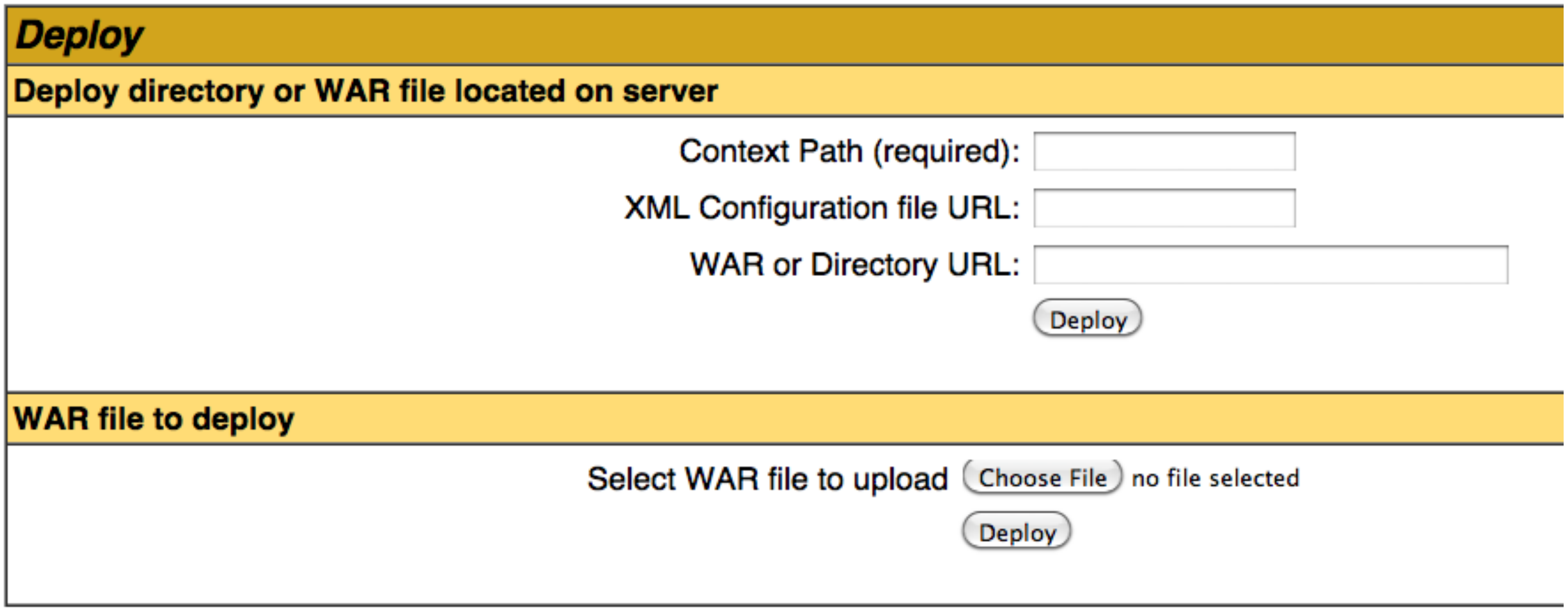


```
include 'Common';
include 'Services';

import Pipeline;
import ServiceContainer;

url = "http://127.0.0.1:4040/WSTokenizer/services/WhiteSpaceTokenizer";
WebService tokenizer_service = new ServiceClient(url, 'dummy', 'dummy');
ServiceContainer tokenizer = new ServiceContainer("WhitespaceTokenizer",
                                                 tokenizer_service);

Data data = new Data(3, read('data/in-01.txt'));
println "\nINPUT:\n" + data.payload;

p = new Pipeline('localhost-misc', data);
p.add_step(tokenizer);
p.run();
```

Deployment & Registration

- Deploy the war file to some server
  (we use Tomcat)
- Register the service with the LAPPS grid

Tomcat Manager



Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

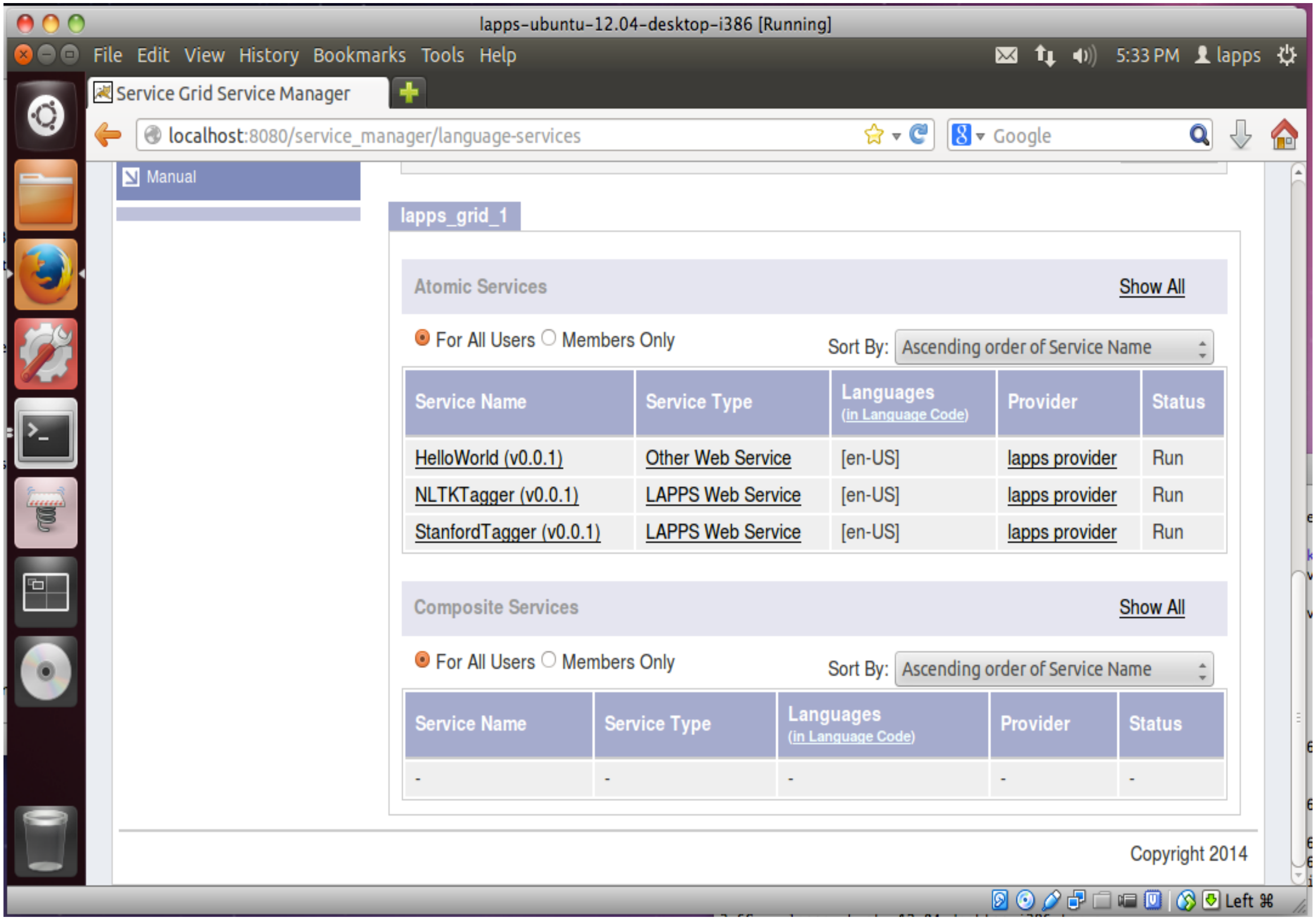
WAR or Directory URL:

Deploy

WAR file to deploy

Select WAR file to upload Choose File no file selected

Deploy



lapps-ubuntu-12.04-desktop-i386 [Running]
File Edit View History Bookmarks Tools Help
5:33 PM lapps
Service Grid Service Manager
localhost:8080/service_manager/language-services
Google
Manual
Left ⌘

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
-	-	-	-	-

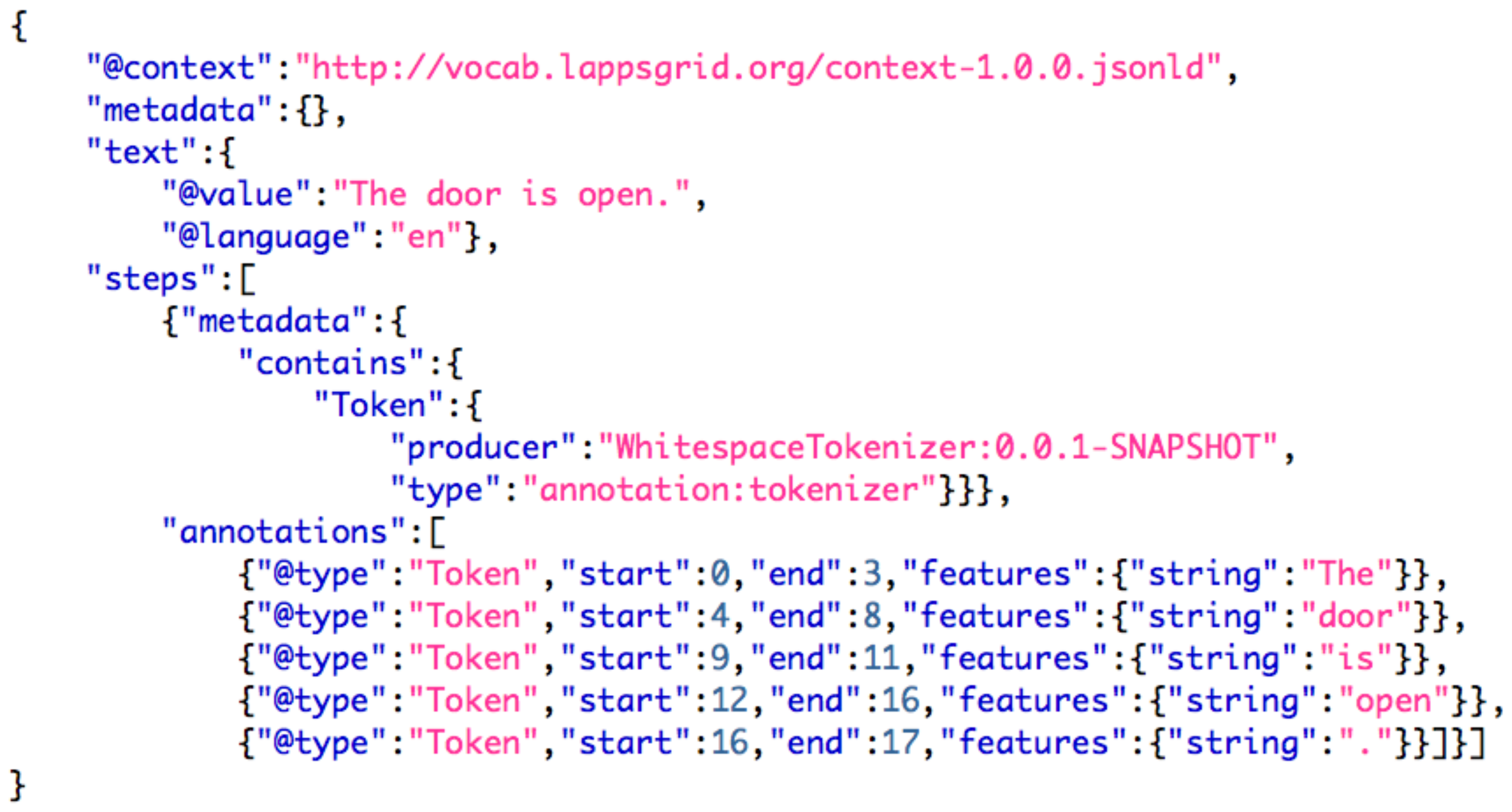
Copyright 2014

In real life now…

Creating a LAPPS Service (2)

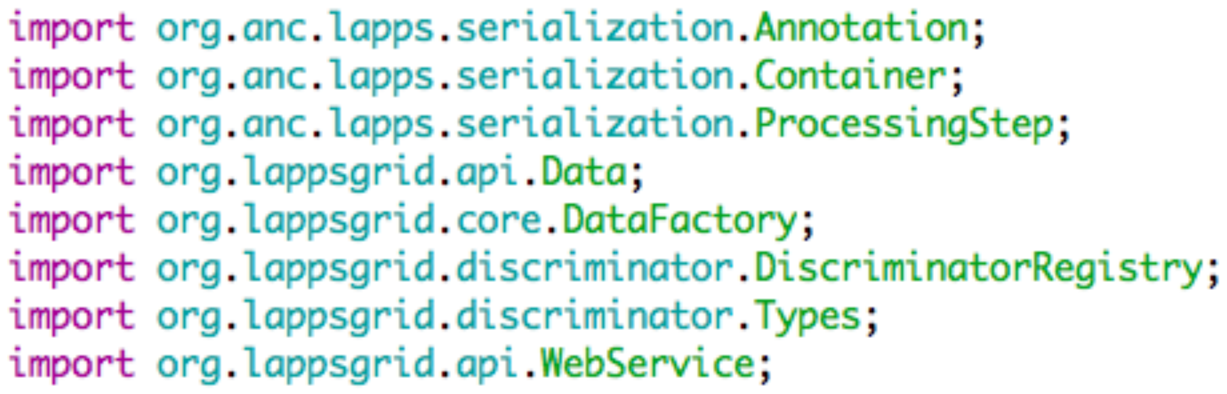
- Second level of compliance
- Create output in the JSON-LD based LAPPS Interchange Format (LIF)
- Stand-off annotation
- Whitespace Tokenizer

Output Generated



```
{
    "@context":"http://vocab.lappsgrid.org/context-1.0.0.jsonld",
    "metadata":{},
    "text":{
        "@value":"The door is open.",
        "@language":"en"},
    "steps":[
        {"metadata":{
            "contains":{
                "Token":{
                    "producer":"WhitespaceTokenizer:0.0.1-SNAPSHOT",
                    "type":"annotation:tokenizer"}}},
        "annotations":[
            {"@type":"Token","start":0,"end":3,"features":{"string":"The"}},
            {"@type":"Token","start":4,"end":8,"features":{"string":"door"}},
            {"@type":"Token","start":9,"end":11,"features":{"string":"is"}},
            {"@type":"Token","start":12,"end":16,"features":{"string":"open"}},
            {"@type":"Token","start":16,"end":17,"features":{"string":"."}}]}]
}
```

Import



```
import org.anc.lapps.serialization.Annotation;
import org.anc.lapps.serialization.Container;
import org.anc.lapps.serialization.ProcessingStep;
import org.lappsgrid.api.Data;
import org.lappsgrid.core.DataFactory;
import org.lappsgrid.discriminator.DiscriminatorRegistry;
import org.lappsgrid.discriminator.Types;
import org.lappsgrid.api.WebService;
```

Code Snippets

```
String text = data.getPayload();
Container container = new Container(false);
container.setText(text);
container.setLanguage("en");
ProcessingStep processingStep = container.newStep();
processingStep.addContains("Token",
                           this.getClass().getName() + ":"
                           + VERSION, "annotation:tokenizer");
```

```
Annotation ann = processingStep.newAnnotation("Token", start, end);
ann.addFeature("string", text.substring(start, end));
```

```
return DataFactory.json(container.toString());
```

The Full Code

In real life again…

Creating a LAPPS Service (3)

- Third level of compliance
- Use categories from the LAPPS vocabulary

JSON-LD

- LAPPS services are not required to exchange data in any particular format.
  - LAPPS GATE services exchange GATE XML
  - Must be prepared to deal with the consequences.
- JSON(-LD) is becoming more popular for data exchange on the web.
  - Good support across programming languages.
  - Recommended that if services do not use JSON-LD they provide a mapping from their format to JSON/JSON-LD
  - Ideally LAPPS services will exchange JSON-LD using a common vocabulary.

LEDS: LAPPS Exchange Data Structures

- Java/Groovy classes for serializing JSON
- Will be refactored soon but basic concepts will remain the same
- Three main classes
  - Container
  - ProcessingStep (View)
  - Annotation
- Other supporting classes for manipulating metadata
  - Contains, etc.

LEDS Classes

- Container
  - Map metadata
  - List<ProcessingStep> step
- ProcessingStep
  - Map metadata
  - List<Annotation> annotations
- Annotation
  - String id
  - String type
  - long start
  - long end
  - Map features
  - Map metadata

LAPPS Exchange Data Structures

- Available on the ANC’s Nexus repository
  - http://www.anc.org:8080/nexus

```
<groupId>org.anc.lapps</groupId>
<artifactId>serialization</artifactId>
<version>0.13.0</version>
```

- Will be refactored into the org.lappsgrid namespace

LAPPS Exchange Data Structures

- Provides simple round tripping between Java objects and their JSON-LD serialization
  - Uses Jackson for JSON serialization

```
Container container = new Container()
String json = container.toJson()
json = container.toPrettyJson()
…
Container = new Container(json)
```

LAPPS Exchange Data Structures

- Can link to a remote @context at http://vocab.lappsgrid.org/context-1.0.0.jsonld

```
Container container = new Container(false);
```

- Can include a local @context that can be manipulated at runtime

```
Container container = new Container();
Map context = new HashMap()
context.put("Token", "http://…");
Context.put("Sentence", "http://…")"
Container.setContext(context)
```

LAPPS Exchange Data Structures

```
{
  "@context" : {
    "Sentence" : "http://example.com/Sentence",
    "Token" : "http://example.com/Token"
  },
  "metadata" : { },
  "text" : { },
  "steps" : [ ]
}
```

Metadata

- Everything can contain metadata
- Services are free to use the metadata maps as needed.
  - LAPPS does not impose many restrictions on metadata
- Except for ProcessingStep (View)
  - Each step should have a *contains* map
  - Allows other processors to quickly find views they are interested in.

Metadata: contains

- Lists the annotation types in each ProcessingStep
  - Key is the annotation type (label)
  - Value is another map
    - *producer*: the name of the service that produced the annotation
    - *url*: the url of the service that produces the annotations
    - *type*: an IRI to a description of the annotation type
      - POS tag set
      - rules used for tokenization

Metadata: contains

```
Container container = new Container(false);
ProcessingStep step = container.newStep();
String producer= “com.example.Tokenizer”
String type = “tokenizer:example”
step.addContains(“Token”, url, type);
```

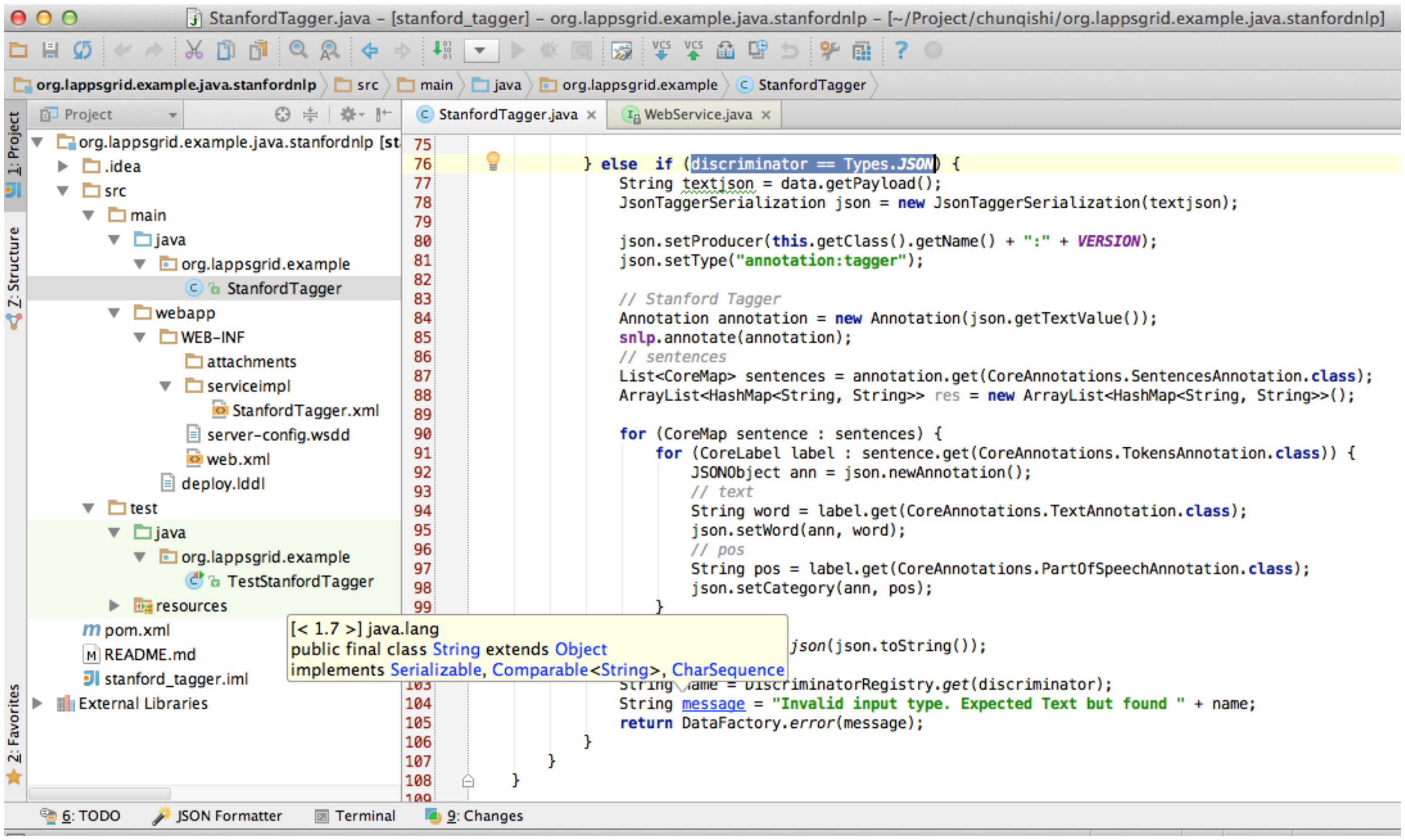
```
{
  "@context" : "http://vocab.lappsgrid.org/context-1.0.0.jsonld",
  "metadata" : { },
  "text" : { },
  "steps" : [ {
    "metadata" : {
      "contains" : {
        "Token" : {
          "producer" : "com.example.Tokenizer",
          "type" : "tokenization:example"
        }
      }
    },
    "annotations" : [ ]
  } ]
}
```

Metadata: contains

```
{
        "http://vocab.lappsgrid.org/metadata/contains": [
         {
           "http://vocab.lappsgrid.org/Token": [
             {
               "http://vocab.lappsgrid.org/metadata/producer": [
                 {
                   "@value": "com.example.Tokenizer"
                 }
               ],
               "http://vocab.lappsgrid.org/metadata/type": [
                 {
                   "@id": "http://vocab.lappsgrid.org/types/tokenization/example"
                 }
               ]
             }
           ]
         }]
}
```

More Wrapping Examples

Development Template



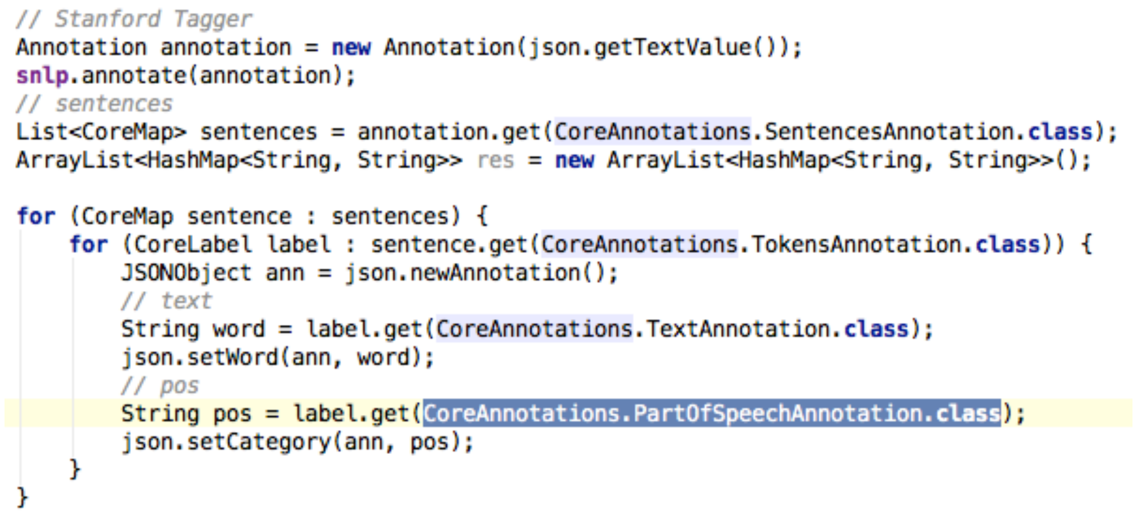
```
        } else  if (discriminator == Types.JSON) {
            String textjson = data.getPayload();
            JsonTaggerSerialization json = new JsonTaggerSerialization(textjson);

            json.setProducer(this.getClass().getName() + ":" + VERSION);
            json.setType("annotation:tagger");

            // 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);
                }
                                      json(json.toString());
            String  ame = DiscriminatorRegistry.get(discriminator);
            String message = "Invalid input type. Expected Text but found " + name;
            return DataFactory.error(message);
        }
    }
}
```

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

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

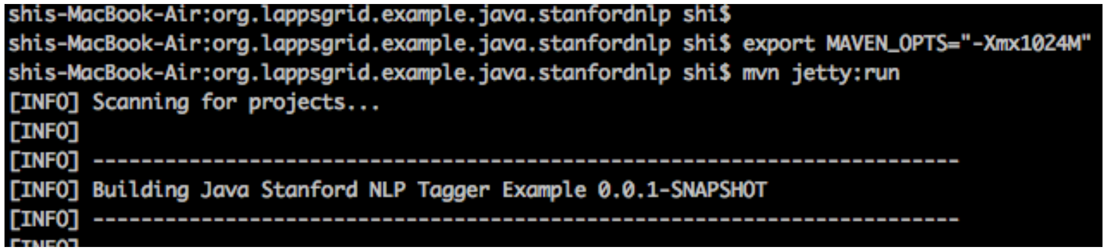
Stanford Tagger Wrapping

- 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);
    }
}
```

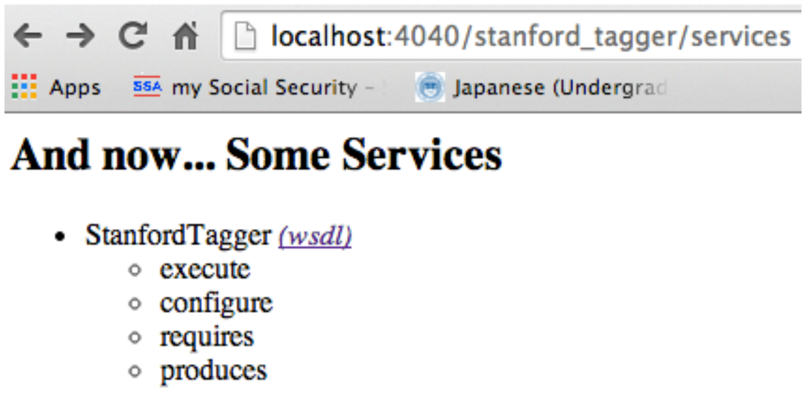
- Jetty Running



```
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] ------------------------------------------------------------------------
```

Stanford Tagger Testing

- Local Service

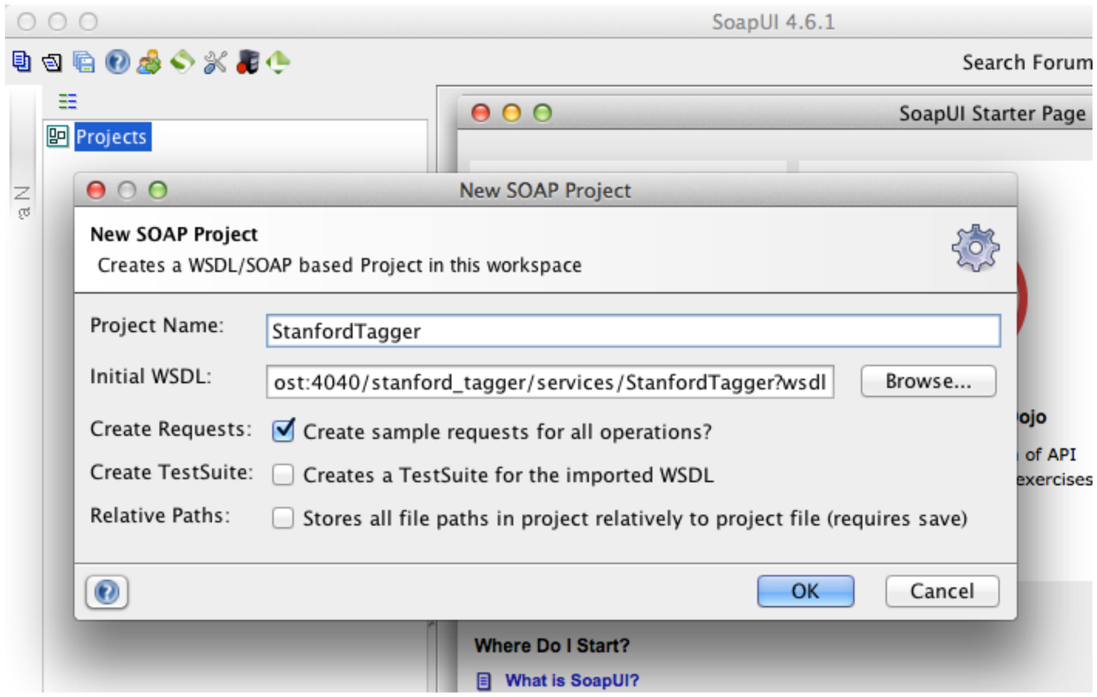


localhost:4040/stanford_tagger/services

And now... Some Services

- StanfordTagger *(wsdl)*
  - execute
  - configure
  - requires
  - produces

- SoapUI Testing



New SOAP Project

Creates a WSDL/SOAP based Project in this workspace

Project Name: StanfordTagger

Initial WSDL: ost:4040/stanford_tagger/services/StanfordTagger?wsdl

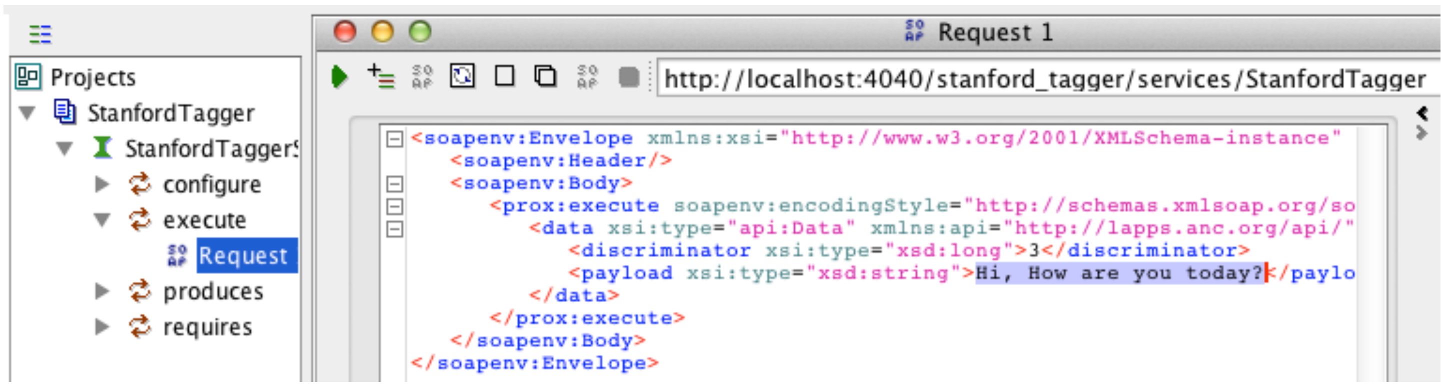
Create Requests: ☑ Create sample requests for all operations?

Create TestSuite: ☐ Creates a TestSuite for the imported WSDL

Relative Paths: ☐ Stores all file paths in project relatively to project file (requires save)

Stanford Tagger Testing Result

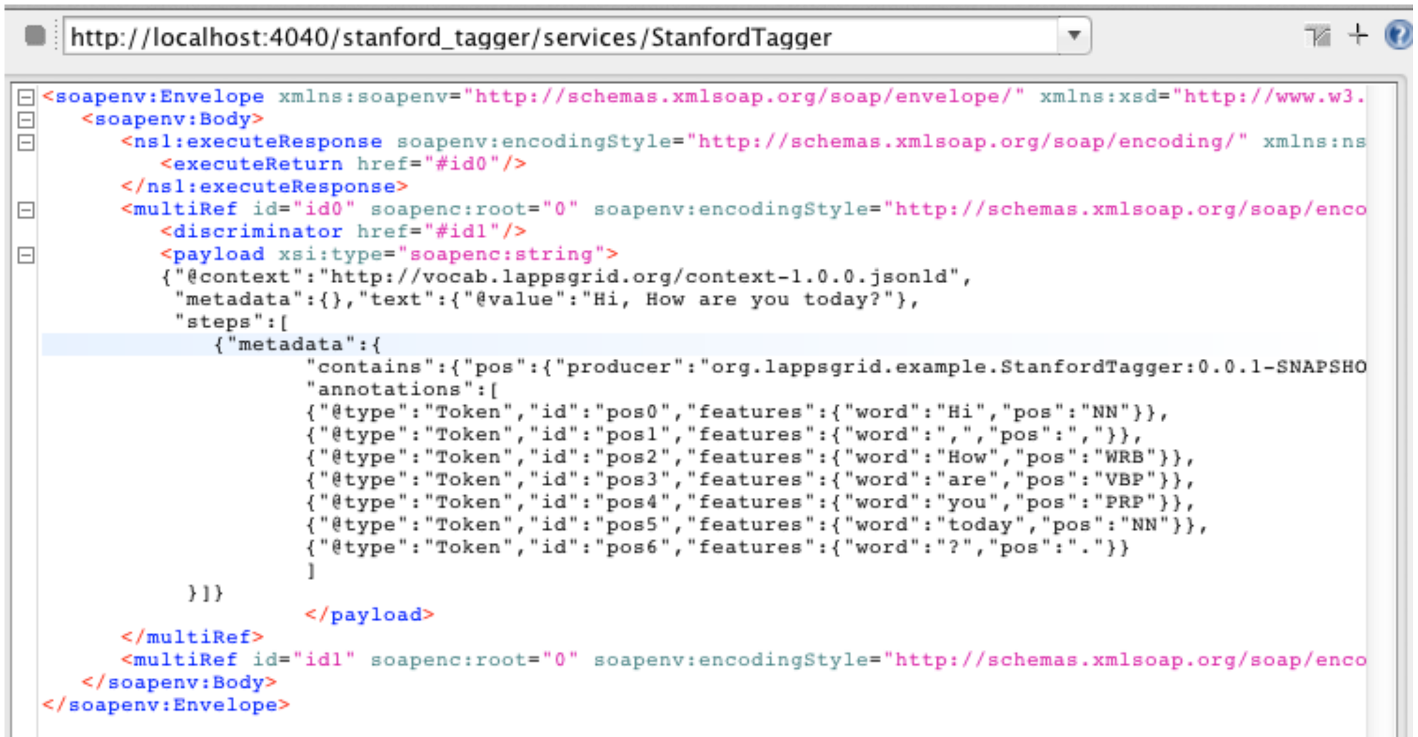
- Request



```
http://localhost:4040/stanford_tagger/services/StanfordTagger
```

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   <soapenv:Header/>
   <soapenv:Body>
      <prox:execute soapenv:encodingStyle="http://schemas.xmlsoap.org/so
         <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?</paylo
         </data>
      </prox:execute>
   </soapenv:Body>
</soapenv:Envelope>
```

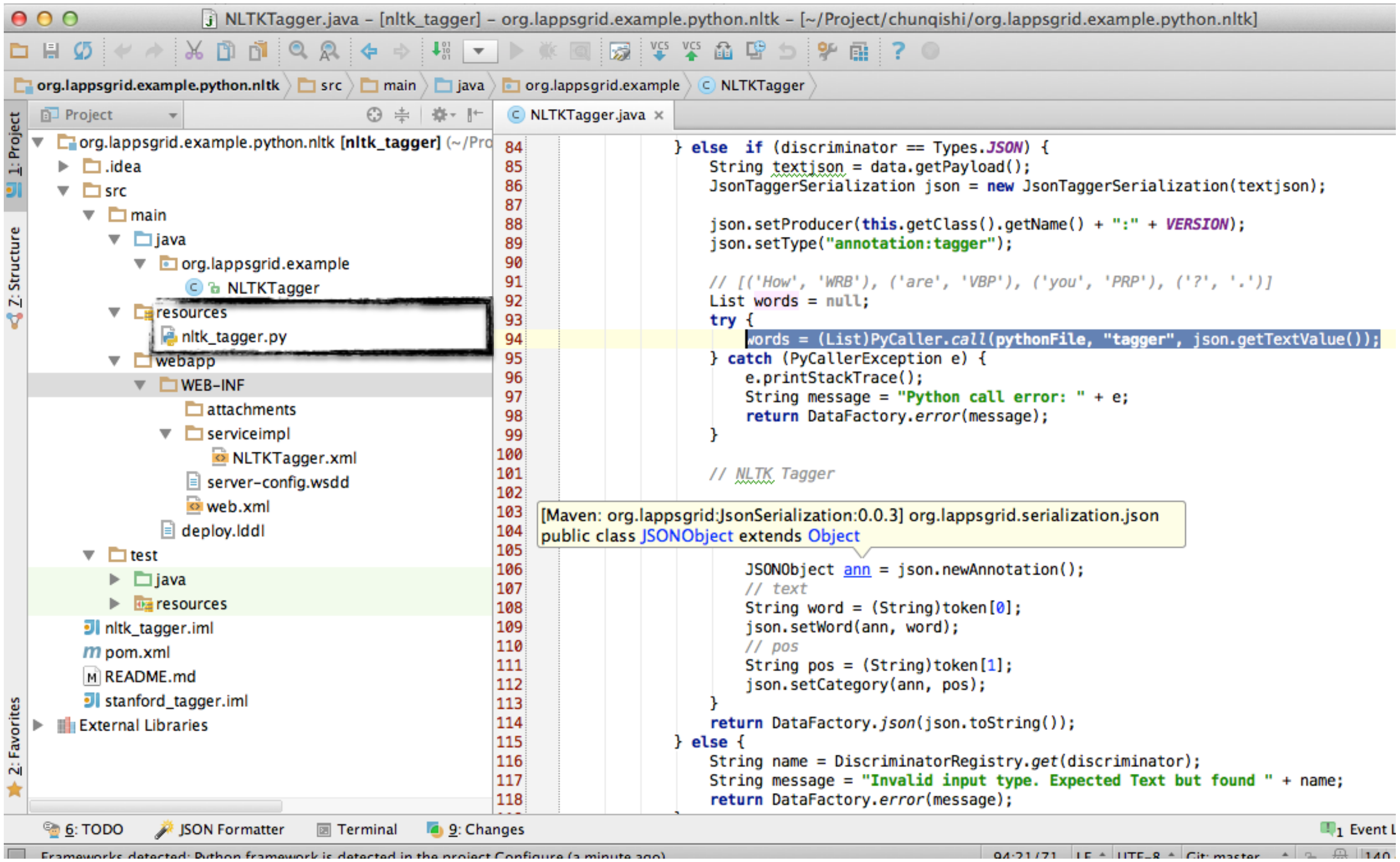
- Response



```
http://localhost:4040/stanford_tagger/services/StanfordTagger
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.
   <soapenv:Body>
      <ns1:executeResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:ns
         <executeReturn href="#id0"/>
      </ns1:executeResponse>
      <multiRef id="id0" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/enco
         <discriminator href="#id1"/>
         <payload xsi:type="soapenc:string">
         {"@context":"http://vocab.lappsgrid.org/context-1.0.0.jsonld",
          "metadata":{},"text":{"@value":"Hi, How are you today?"},
          "steps":[
            {"metadata":{
                   "contains":{"pos":{"producer":"org.lappsgrid.example.StanfordTagger:0.0.1-SNAPSHO
                   "annotations":[
                   {"@type":"Token","id":"pos0","features":{"word":"Hi","pos":"NN"}},
                   {"@type":"Token","id":"pos1","features":{"word":",","pos":","}},
                   {"@type":"Token","id":"pos2","features":{"word":"How","pos":"WRB"}},
                   {"@type":"Token","id":"pos3","features":{"word":"are","pos":"VBP"}},
                   {"@type":"Token","id":"pos4","features":{"word":"you","pos":"PRP"}},
                   {"@type":"Token","id":"pos5","features":{"word":"today","pos":"NN"}},
                   {"@type":"Token","id":"pos6","features":{"word":"?","pos":"."}}
                   ]
          }]}
                   </payload>
      </multiRef>
      <multiRef id="id1" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/enco
   </soapenv:Body>
</soapenv:Envelope>
```

Developing Template



NLTKTagger.java - [nltk_tagger] - org.lappsgrid.example.python.nltk - [~/Project/chunqishi/org.lappsgrid.example.python.nltk]

org.lappsgrid.example.python.nltk › src › main › java › org.lappsgrid.example › NLTKTagger

```
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
```

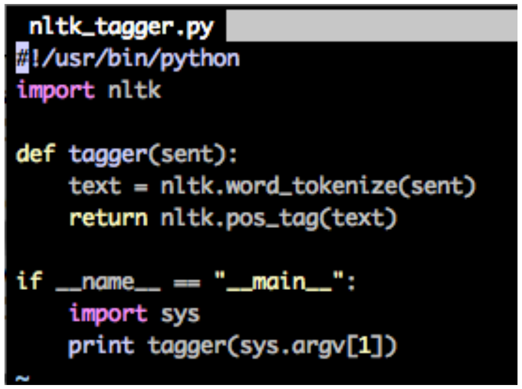
NLTKTagger.java

```java
        } else  if (discriminator == Types.JSON) {
            String textjson = data.getPayload();
            JsonTaggerSerialization json = new JsonTaggerSerialization(textjson);

            json.setProducer(this.getClass().getName() + ":" + VERSION);
            json.setType("annotation:tagger");

            // [('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);
            }

            // NLTK Tagger
[Maven: org.lappsgrid:JsonSerialization:0.0.3] org.lappsgrid.serialization.json
public class JSONObject extends Object

                JSONObject ann = json.newAnnotation();
                // text
                String word = (String)token[0];
                json.setWord(ann, word);
                // pos
                String pos = (String)token[1];
                json.setCategory(ann, pos);
            }
            return DataFactory.json(json.toString());
        } else {
            String name = DiscriminatorRegistry.get(discriminator);
            String message = "Invalid input type. Expected Text but found " + name;
            return DataFactory.error(message);
```

Frameworks detected: Python framework is detected in the project Configure (a minute ago)   94:21/71   LF   UTF-8   Git: master

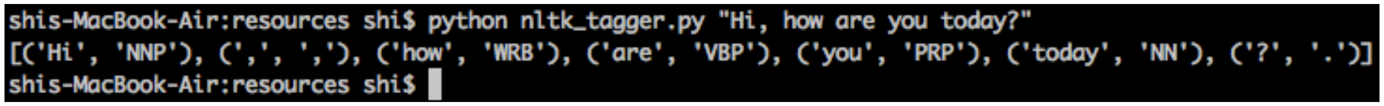
NLTK Python

- Python Program



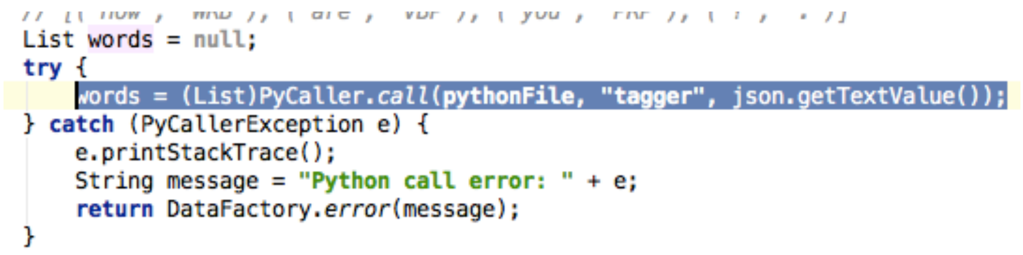
```
nltk_tagger.py
#!/usr/bin/python
import nltk

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



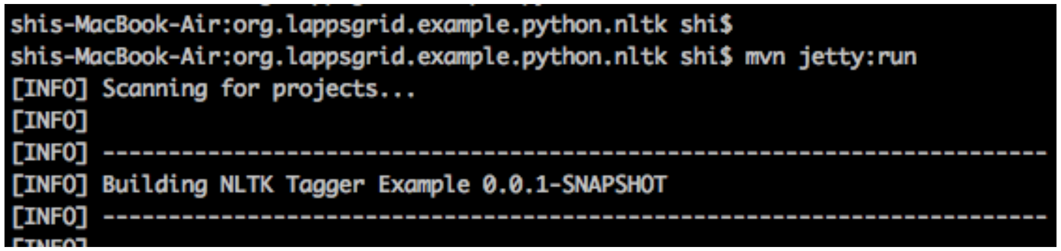
```
shis-MacBook-Air:resources shi$ python nltk_tagger.py "Hi, how are you today?"
[('Hi', 'NNP'), (',', ','), ('how', 'WRB'), ('are', 'VBP'), ('you', 'PRP'), ('today', 'NN'), ('?', '.')]
shis-MacBook-Air:resources shi$
```

- 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);
}
```

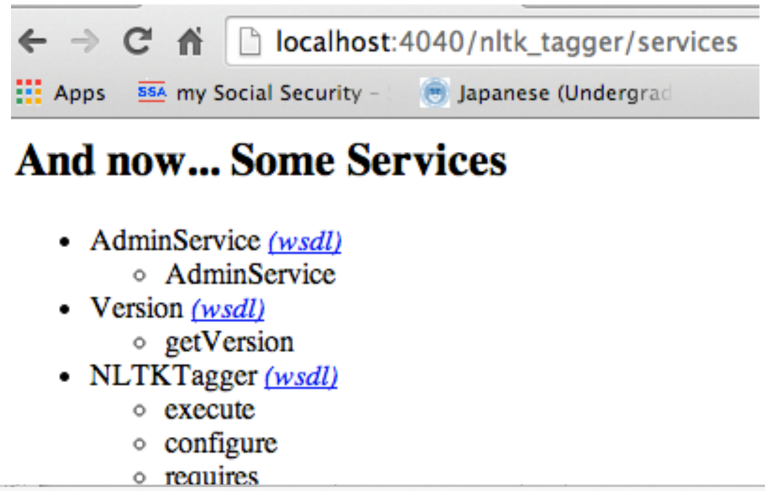
- Jetty Running



```
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



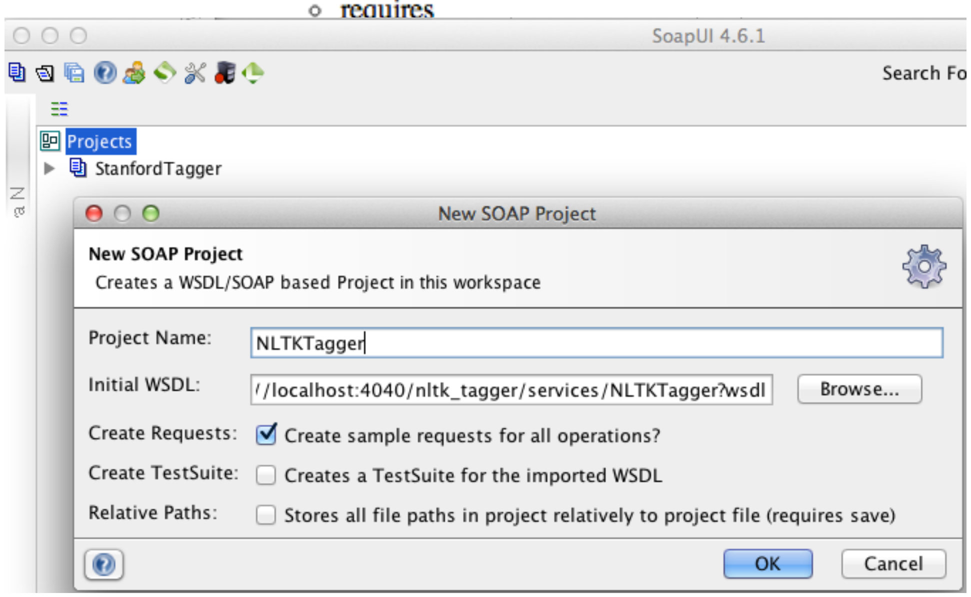
localhost:4040/nltk_tagger/services

Apps · my Social Security – · Japanese (Undergrad

And now... Some Services

- AdminService (wsdl)
  - AdminService
- Version (wsdl)
  - getVersion
- NLTKTagger (wsdl)
  - execute
  - configure
  - requires

- SoapUI Testing



SoapUI 4.6.1

Projects
StanfordTagger

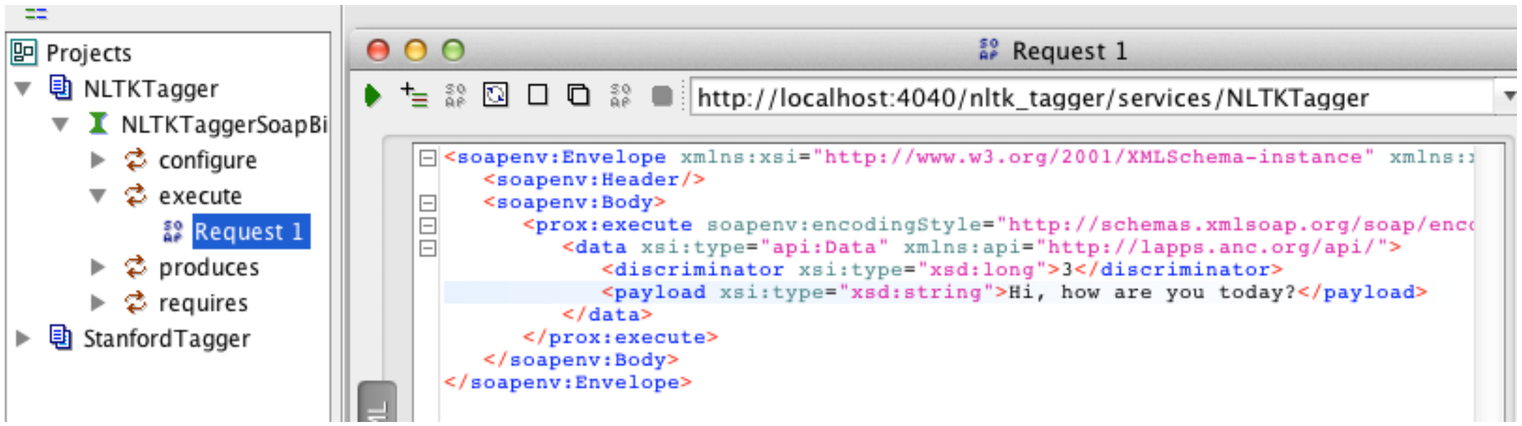
New SOAP Project

New SOAP Project
Creates a WSDL/SOAP based Project in this workspace

Project Name:	NLTKTagger
Initial WSDL:	//localhost:4040/nltk_tagger/services/NLTKTagger?wsdl
Create Requests:	[✓] Create sample requests for all operations?
Create TestSuite:	[ ] Creates a TestSuite for the imported WSDL
Relative Paths:	[ ] Stores all file paths in project relatively to project file (requires save)

NLTK Tagger Testing Result

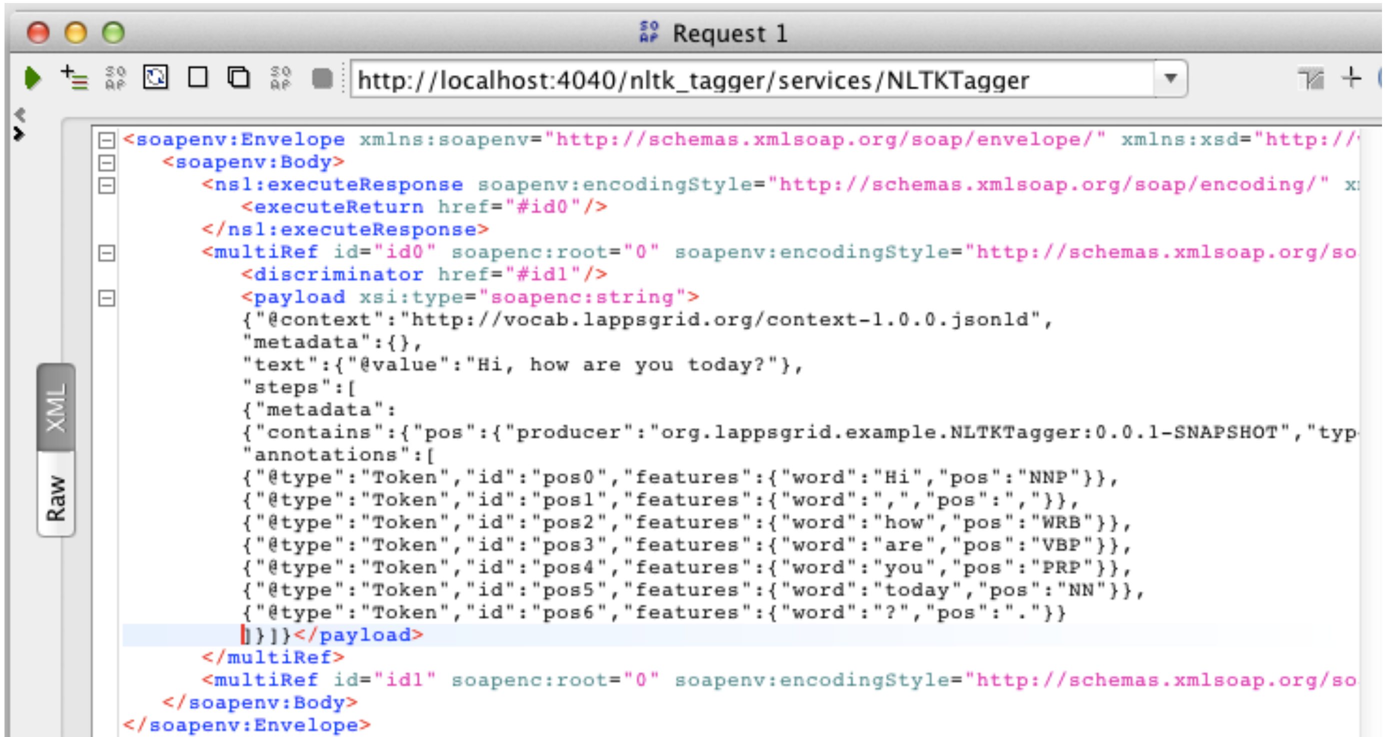
- Request



http://localhost:4040/nltk_tagger/services/NLTKTagger

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:x
   <soapenv:Header/>
   <soapenv:Body>
      <prox:execute soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/enco
         <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



http://localhost:4040/nltk_tagger/services/NLTKTagger

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://
   <soapenv:Body>
      <ns1:executeResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" x
         <executeReturn href="#id0"/>
      </ns1:executeResponse>
      <multiRef id="id0" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/so
         <discriminator href="#id1"/>
         <payload xsi:type="soapenc:string">
         {"@context":"http://vocab.lappsgrid.org/context-1.0.0.jsonld",
         "metadata":{},
         "text":{"@value":"Hi, how are you today?"},
         "steps":[
         {"metadata":
         {"contains":{"pos":{"producer":"org.lappsgrid.example.NLTKTagger:0.0.1-SNAPSHOT","typ
         "annotations":[
         {"@type":"Token","id":"pos0","features":{"word":"Hi","pos":"NNP"}},
         {"@type":"Token","id":"pos1","features":{"word":",","pos":","}},
         {"@type":"Token","id":"pos2","features":{"word":"how","pos":"WRB"}},
         {"@type":"Token","id":"pos3","features":{"word":"are","pos":"VBP"}},
         {"@type":"Token","id":"pos4","features":{"word":"you","pos":"PRP"}},
         {"@type":"Token","id":"pos5","features":{"word":"today","pos":"NN"}},
         {"@type":"Token","id":"pos6","features":{"word":"?","pos":"."}}
         ]}]}</payload>
      </multiRef>
      <multiRef id="id1" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/so
   </soapenv:Body>
</soapenv:Envelope>
```

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://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

Hands-On?

- Get Maven and Java
- lappsgrid.org