Report tzhong

In this PI1 I use the WhitespaceTokenizer example to get myself familiar with the workflow of using Maven and SoapUI(as well as Jetty) to develop webservice.

The WhitespaceTokenizer mainly contains two parts. The first part generates the metadata and conducts error checking. The second part is the "execute" part, which is the actual tokenizer, and it consists of a series of sections.

To be more specific, in the "execute" part, we first parse the input and check the discriminator, and return the unchanged string if an error is detected. Then we initialise the View object, extract the text and tokenise it, and at the same time update the view with annotations(each annotation corresponds to a token). Finally we create a DataContainer object to wrap our result and serialize it into json format.

To give an example about the output, we can use SoapUI to track and test our program. For example, given the following sentence

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSche
   <soapenv:Header/>
   <soapenv:Body>
      <prox:execute soapenv:encodingStyle="http://schemas.x
         <input xsi:type="xsd:string">
{
    "discriminator":"http://vocab.lappsgrid.org/ns/media/te
    "payload": "Hello world for PI1, I am tzhong."
}
         </input>
      </prox:execute>
   </soapenv:Body>
</soapenv:Envelope>
```

the output is

```
},
"annotations" : [ {
  "id" : "tok0",
  "start" : 0,
  "end" : 5,
  "@type" : "http://vocab.lappsgrid.org/Token",
  "features" : {
    "word" : "Hello"
  }
}, {
  "id" : "tok1",
  "start" : 6,
  "end" : 11,
  "@type" : "http://vocab.lappsgrid.org/Token",
  "features" : {
    "word" : "world"
  }
}, {
  "id" : "tok2",
  "start" : 12,
  "end" : 15,
  "@type" : "http://vocab.lappsgrid.org/Token",
  "features" : {
    "word" : "for"
  `
```

The tokenizer does tokenize the sentence and record the start and end index of each word(as sequence of characters).