

Translator

How it works ?

Step 1- input word and languages

Start by imputing word you want to translate, from what language, and to witch language you want to translate

```
13
14 public class Translator {
15     static String rec;
16     String izvorniJezik;
17     String ciljaniJezik;
18     public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
19
20         Translator t = new Translator();
21
22         t.translator("Win", "srb", "eng");
23     }
24
25     public Translator() {
26     }
27 }
```

Step 2 – Incorrect input

If you input incorrect word, or an existing word

You will get error message

```
8 public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
9
10     Translator t = new Translator();
11
12     t.translator("Win", "srp", "eng");
13 }
14
15 public Translator() {
16 }
17
18
19 public void translator(String rec, String iJezik, String cJezik) throws ParserConfigurationException, SAXException, IOException
20 {
21     this.rec = rec;
22     this.izvorniJezik = iJezik;
23     this.ciljaniJezik = cJezik;
24
25     File inputFile = new File("episakReci.xml");
26     DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
27     DocumentBuilder dB = dbf.newDocumentBuilder();
28     Document doc = dB.parse(inputFile);
29     doc.getDocumentElement().normalize();
30 }
```

tput - Translator (run) X

```
run:
Proverite da li ste uneli ispravnu rec, ukoliko jeste onda rec ne postoji u našem rečniku
BUILD SUCCESSFUL (total time: 0 seconds)
```

Step 3- Correct input

If your input is correct, you will get your translated word

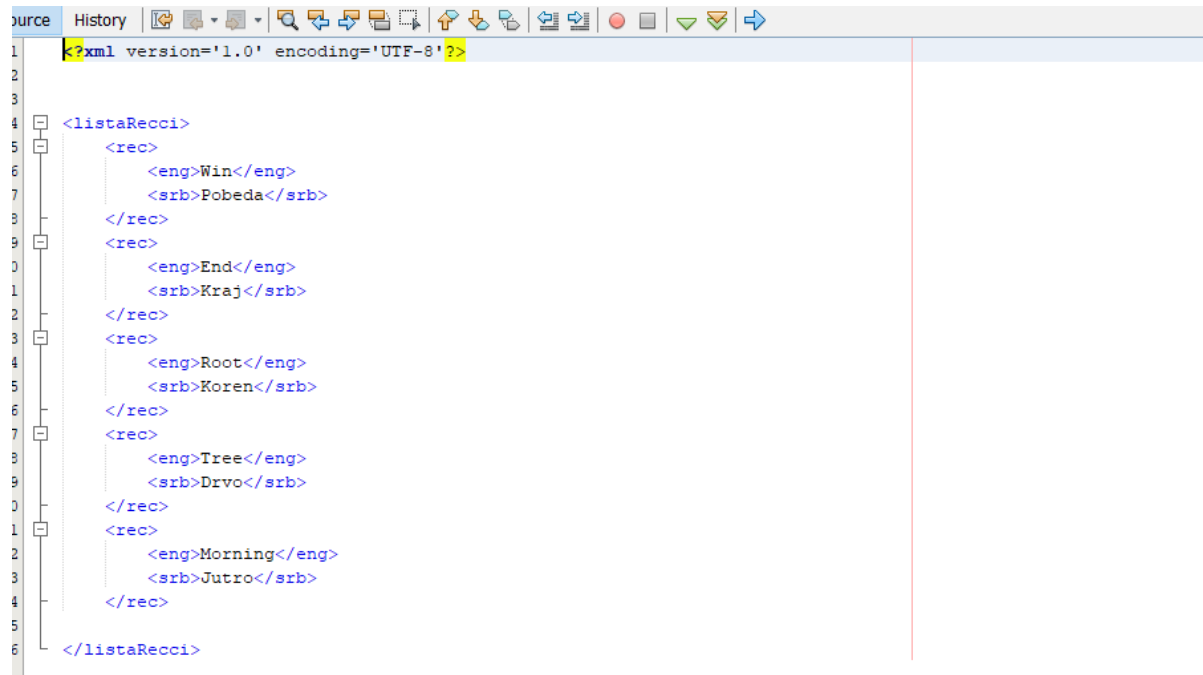
```
21  
22     t.translator("Win", "eng", "srb");  
23 }  
24  
25 public Translator() {  
26 }  
27  
28  
29 public void translator(String rec, String iJezik, String cJezik) throws ParserConfigurationException, SAXException,  
30 {  
31     this.rec = rec;  
32     this.izvorniJezik = iJezik;  
33     this.ciljaniJezik = cJezik;  
34  
35  
36     File inpuFile = new File("spisakReci.xml");  
37     DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();  
38     DocumentBuilder dB = dbf.newDocumentBuilder();  
39     Document doc = dB.parse(inpuFile);  
40     doc.getDocumentElement().normalize();
```

Output - Translator (run) X

```
run:  
English word: Win, on Serbian is word: Pobeda  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Step 4- List of words

Here's some words in our translator



The screenshot shows an XML editor with a toolbar at the top. The XML document is as follows:

```
<?xml version='1.0' encoding='UTF-8'?>

<listaRecci>
  <rec>
    <eng>Win</eng>
    <srb>Pobeda</srb>
  </rec>
  <rec>
    <eng>End</eng>
    <srb>Kraj</srb>
  </rec>
  <rec>
    <eng>Root</eng>
    <srb>Koren</srb>
  </rec>
  <rec>
    <eng>Tree</eng>
    <srb>Drvo</srb>
  </rec>
  <rec>
    <eng>Morning</eng>
    <srb>Jutro</srb>
  </rec>
</listaRecci>
```

The editor includes a left margin with line numbers 1 through 6. A tree view on the left side of the code area shows the hierarchical structure of the XML document, with expandable nodes for the root element and its children.