**Framework used:** JAX-RS(Jersey), JNI (Part of Java SDK/JDK)

**Versions used:**

Tomcat = 6  
IDE = Eclipse (Mars)

JRE = 1.8 (32 bit)

JDK = 1.8 (32 bit)

dll = (32 bit)  
MingW32  
DHC (chrome extension) is used to do HTTP calls

**Environment Variables:**

(note: Program Files (x86) for 32 bit)

* Add C:\Program Files (x86)\Java\jdk1.8.0\_111 (or similar) to a new variable JAVA\_HOME
* Add …\MingW\bin to PATH
* Add %JAVA\_HOME%\bin to PATH (it uses the first environment variable added)

**Eclipse configurations:**

Add Tomcat to Eclipse

1. Open **Window -> Preferences -> Server -> Installed Runtimes**to create a Tomcat installed runtime.
2. Click on **Add...** to open the **New Server Runtime** dialog, then select your runtime under **Apache**

Add libraries to Eclipse

The jar files for the frameworks can be added by Project => Properties => Java Build Path => Libraries => Add JAR ...

In the VM Options field (Run 🡪 Run configurations 🡪 Arguments 🡪 VM Arguments), add the following option, based on your library’s path: -Djava.library.path=<path\_to\_dll> (No “/” at the end)

Switching to the 32 bit JRE when executing works for me. Below are the steps on how to do this exactly in Eclipse:

* Go to *Run*
* Go to *Run configurations*
* Under the *JRE* tab you can switch to the 32 bit JRE

**Visual Studio Configurations**

Include directories in VS: ..\Java\jdk1.8.0\_111\include\win32 and ..\Java\jdk1.8.0\_111\include\ for JNI and also import the java header file by including the project folder and adding the file to the project.

**Used this tutorial to set up the Java web service**

<https://www.youtube.com/watch?v=EfEUDAHgrGQ>

Note:  
The web service in the repository can be used after pulling the code. The only thing that you need to do is add the .jar files in the project just as indicated in the video. You need to add all the .jar files in the jaxrs folders.

**Used this tutorial for JNI.(up until beginning of page 6)**

<http://www.cs.umanitoba.ca/~eclipse/8-JNI.pdf>

Note:

The program can be run with Eclipse instead of the command java …

Javah has to be called in the …/build/classes folder with the full package name, for example: com.webservice.<filename>

Run the C code commands from C project folder

Also use g++ as compiler for c++ code. (extension cpp instead of c if c++ is used)

put dll file in the project folder of the webservice

**READ ME:**

Git repository: <https://github.com/JackKLM/webservice>

Has 2 folders, 1 Java project, and 1 c++ project. Also contains the architecture images, and some zip files that you need.(not all of it)

The c++ project is used to generate the dll file. The Java project contains the web service which can call the dll file.

Note:

The Java project can be imported by going to File 🡪 Import 🡪 General 🡪 Existing Projects in Workspace

***Potential Pitfalls/problems***

* Tomcat on Windows/ COM object on Linux (normally Tomcat runs on Linux and COM object on Windows)
* COM object uses pointers (c language). In Java there are no pointers. Should run without problem if everything is calculated in the C++ code.
* Only tested locally. Might give some problems in live environment.