**JAX-WS**

**CONTRACT FIRST APPROACH [RI IMPLEMENTATION]**

* **MAVEN**
* **ANT SCRIPT TO RUN WSIMPORT TOOL**

Lets create a webapp project using maven. Put the below wsdl as part of webapp/WEB-INF directory

**Apollopharmacy.wsdl**

<?xml version=*"1.0"* encoding=*"UTF-8"* standalone=*"no"*?>

<wsdl:definitions xmlns:apw=*"http://www.apollo.com/apollopharamacy/wsdl"*

xmlns:apt=*"http://www.apollo.com/apollopharamacy/types"* xmlns:soap=*"http://schemas.xmlsoap.org/wsdl/soap/"*

xmlns:wsdl=*"http://schemas.xmlsoap.org/wsdl/"* xmlns:xsd=*"http://www.w3.org/2001/XMLSchema"*

name=*"apollopharamacyMedicineFinderService"* targetNamespace=*"http://www.apollo.com/apollopharamacy/wsdl"*>

<wsdl:types>

<xsd:schema targetNamespace=*"http://www.apollo.com/apollopharamacy/types"*

elementFormDefault=*"qualified"*>

<xsd:complexType name=*"Criteria"*>

<xsd:sequence>

<xsd:element name=*"medicineName"* type=*"xsd:string"* />

<xsd:element name=*"drugsId"* type=*"xsd:string"* />

<xsd:element name=*"generic"* type=*"xsd:boolean"* />

<xsd:element name=*"manufacturer"* type=*"xsd:string"* />

<xsd:element name=*"units"* type=*"xsd:int"* />

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name=*"Store"*>

<xsd:sequence>

<xsd:element name=*"storeId"* type=*"xsd:string"* />

<xsd:element name=*"agent"* type=*"xsd:string"* />

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name=*"Medicine"*>

<xsd:sequence>

<xsd:element name=*"medicineName"* type=*"xsd:string"* />

<xsd:element name=*"generic"* type=*"xsd:boolean"* />

<xsd:element name=*"manufacturer"* type=*"xsd:string"* />

<xsd:element name=*"availableQty"* type=*"xsd:int"* />

<xsd:element name=*"pricePerUnit"* type=*"xsd:float"* />

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name=*"MedicineList"*>

<xsd:sequence>

<xsd:element name=*"medicineList"* type=*"apt:Medicine"*

minOccurs=*"0"* maxOccurs=*"unbounded"*></xsd:element>

</xsd:sequence>

</xsd:complexType>

<xsd:element name=*"search"*>

<xsd:complexType>

<xsd:sequence>

<xsd:element name=*"criteria"* type=*"apt:Criteria"*></xsd:element>

<xsd:element name=*"store"* type=*"apt:Store"*></xsd:element>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name=*"searchResponse"*>

<xsd:complexType>

<xsd:sequence>

<xsd:element name=*"ListOfMedicine"* type=*"apt:MedicineList"*></xsd:element>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

</wsdl:types>

<wsdl:message name=*"MedicineFinder\_SearchRequest"*>

<wsdl:part element=*"apt:search"* name=*"in"* />

</wsdl:message>

<wsdl:message name=*"MedicineFinder\_SearchResponse"*>

<wsdl:part element=*"apt:searchResponse"* name=*"out"* />

</wsdl:message>

<wsdl:portType name=*"MedicineFinder"*>

<wsdl:operation name=*"search"*>

<wsdl:input message=*"apw:MedicineFinder\_SearchRequest"* />

<wsdl:output message=*"apw:MedicineFinder\_SearchResponse"* />

</wsdl:operation>

</wsdl:portType>

<wsdl:binding name=*"apollopharamacySOAPBinding"* type=*"apw:MedicineFinder"*>

<soap:binding style=*"document"*

transport=*"http://schemas.xmlsoap.org/soap/http"* />

<wsdl:operation name=*"search"*>

<soap:operation soapAction=*"http://www.apollo.com/apollopharamacy/wsdl/search"* />

<wsdl:input>

<soap:body use=*"literal"* />

</wsdl:input>

<wsdl:output>

<soap:body use=*"literal"* />

</wsdl:output>

</wsdl:operation>

</wsdl:binding>

<wsdl:service name=*"apollopharamacyMedicineFinderService"*>

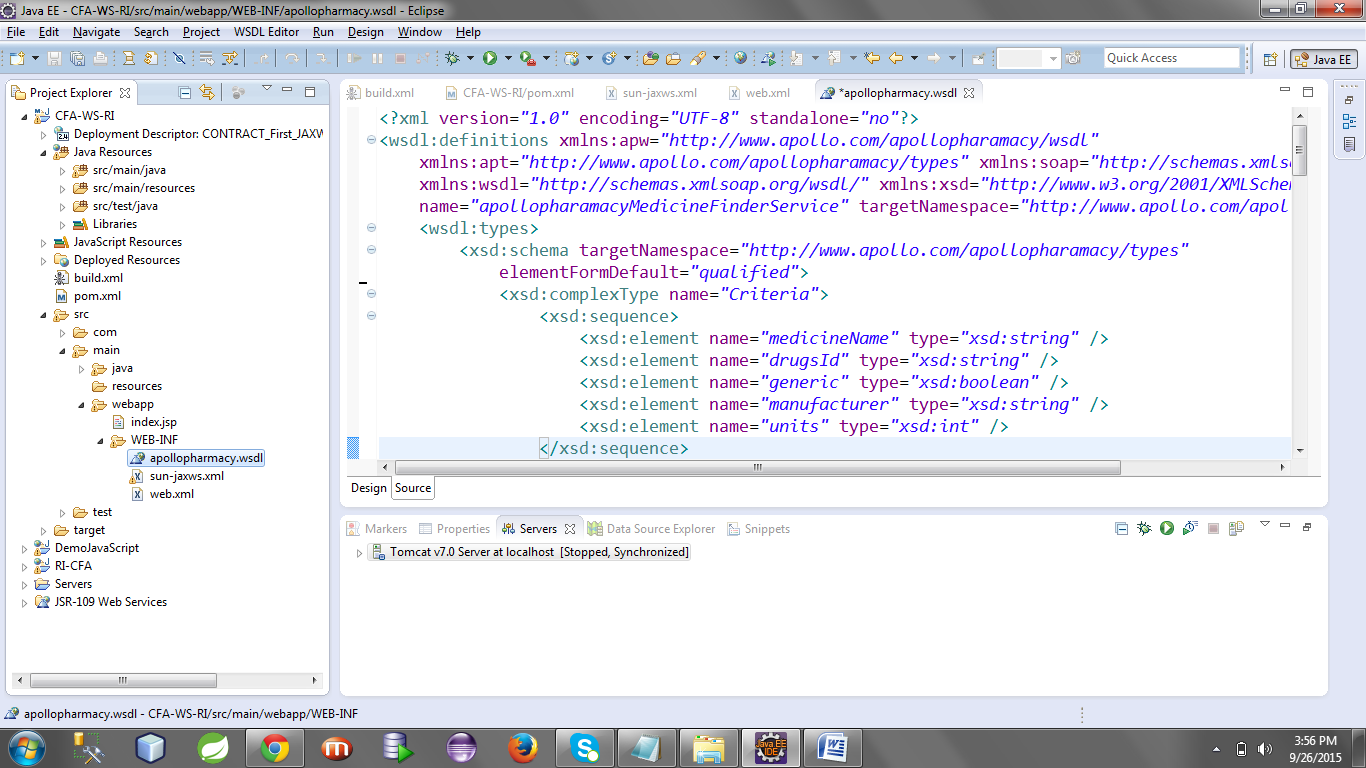
<wsdl:port binding=*"apw:apollopharamacySOAPBinding"* name=*"apollopharamacySOAPPort"*>

<soap:address location=*"http://www.apollo.com/"* />

</wsdl:port>

</wsdl:service>

</wsdl:definitions>

****

Now lets have a build.xml file as part of root directory of project. This file contains ant scripts to run the wsimport tool.

**build.xml**

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE project>

<project name="generate-server" default="main" >

<property name="javahome"

value="C:\Program Files\Java\jdk1.6.0\_32" />

<property name="wsdl.location"

value="src/main/webapp/WEB-INF/apollopharmacy.wsdl" />

<target name="main">

<exec executable="${javahome}\bin\wsimport.exe">

<arg line="${wsdl.location} -s src/main/java -Xdebug -verbose -Xnocompile" />

</exec>

</target>

</project>

Below is the pom.xml for jax-ws jar dependency and to run the ant script.

**pom.xml**

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.ws</groupId>

<artifactId>CFA-WS-RI</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>CFA-WS-RI Maven Webapp</name>

<url>http://maven.apache.org</url>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>com.sun.xml.ws</groupId>

<artifactId>jaxws-rt</artifactId>

<version>2.1.4</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-antrun-plugin</artifactId>

<version>1.7</version>

<executions>

<execution>

<id>default-cli</id>

<phase>package</phase>

<configuration>

<target>

<ant antfile=*"build.xml"* />

</target>

</configuration>

<goals>

<goal>run</goal>

</goals>

</execution>

</executions>

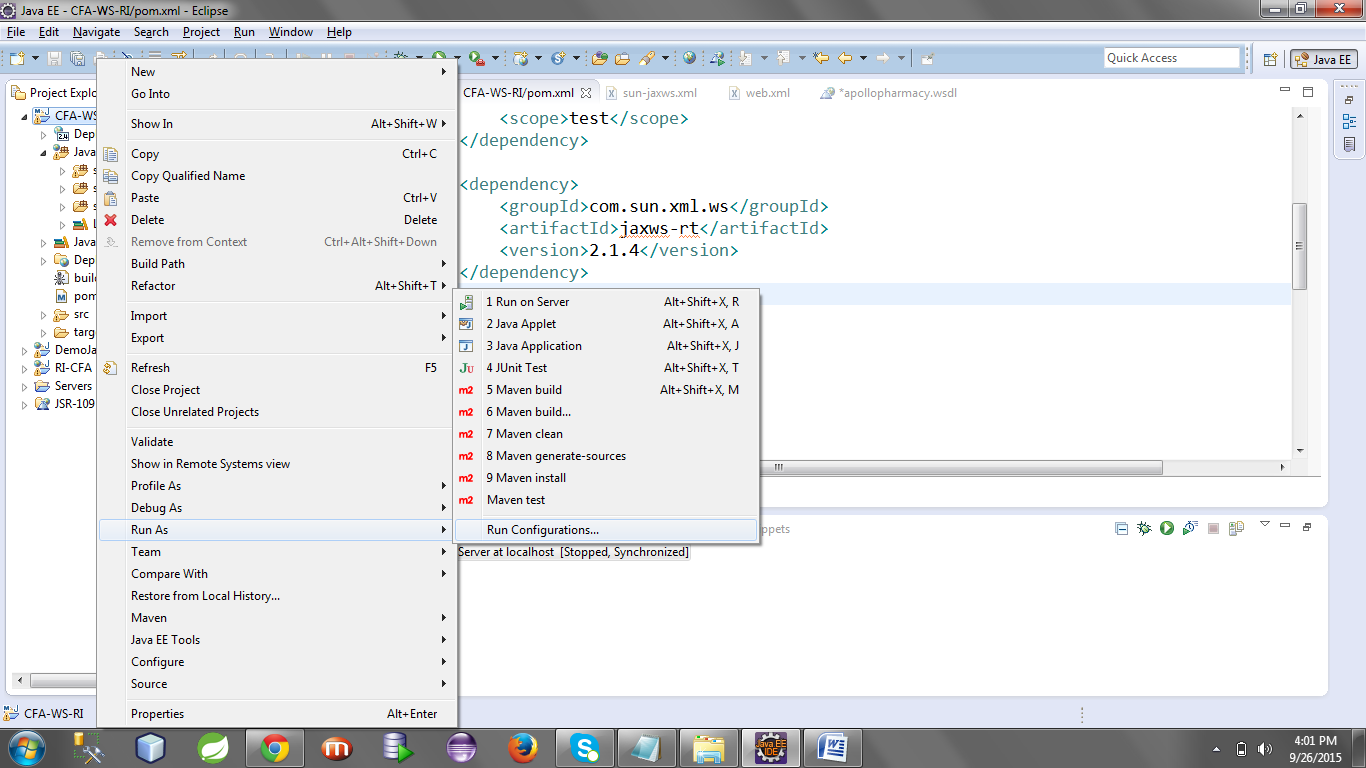
</plugin>

</plugins>

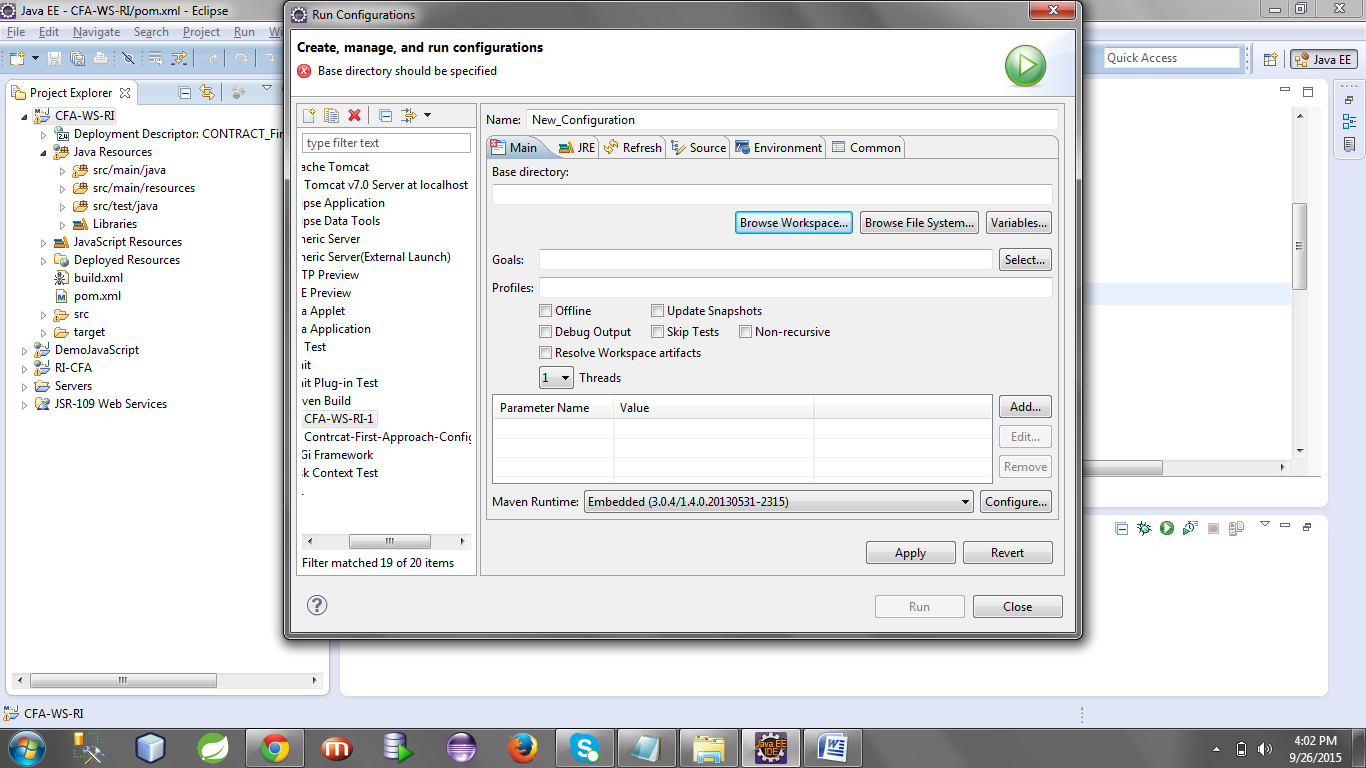
</build>

</project>

**Now right click on your project > Run as > run configuration**

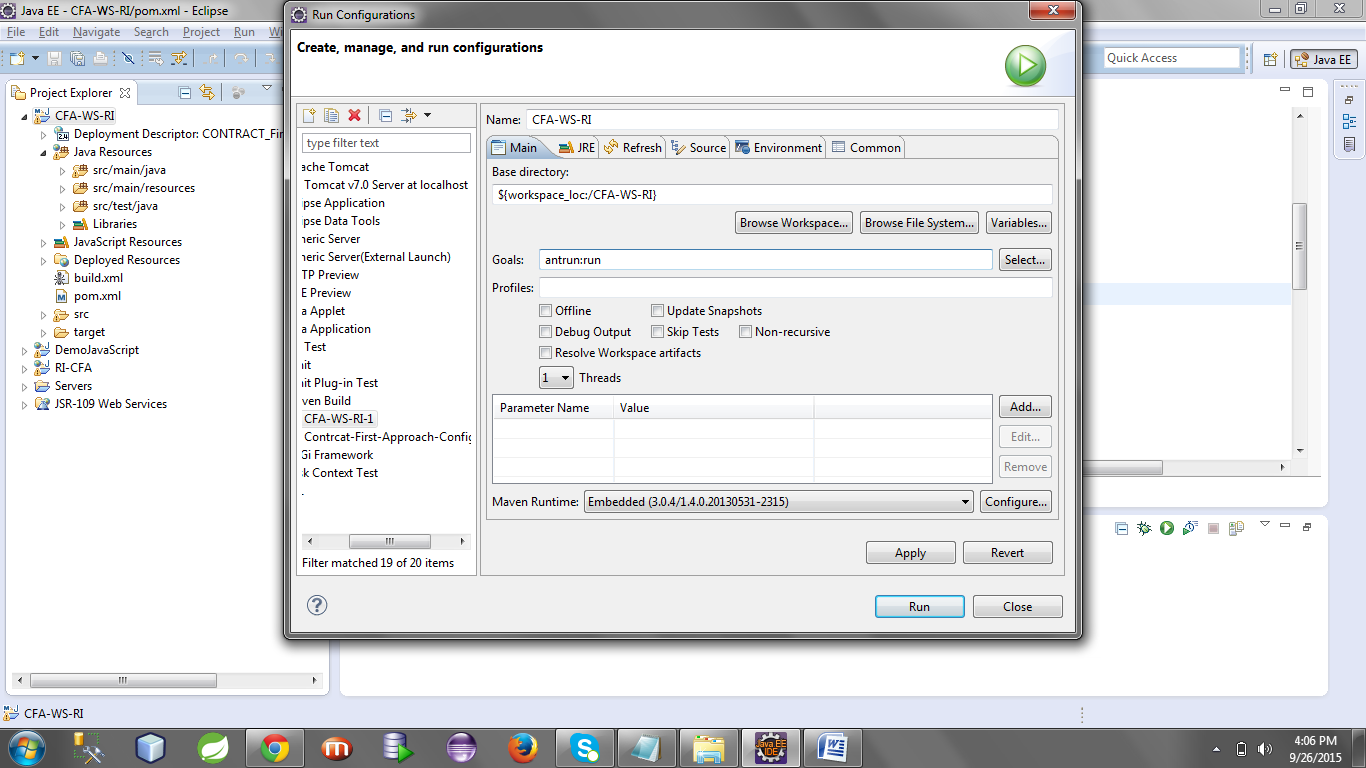
****

**Once you click run configuration it will open the below wizard**

****

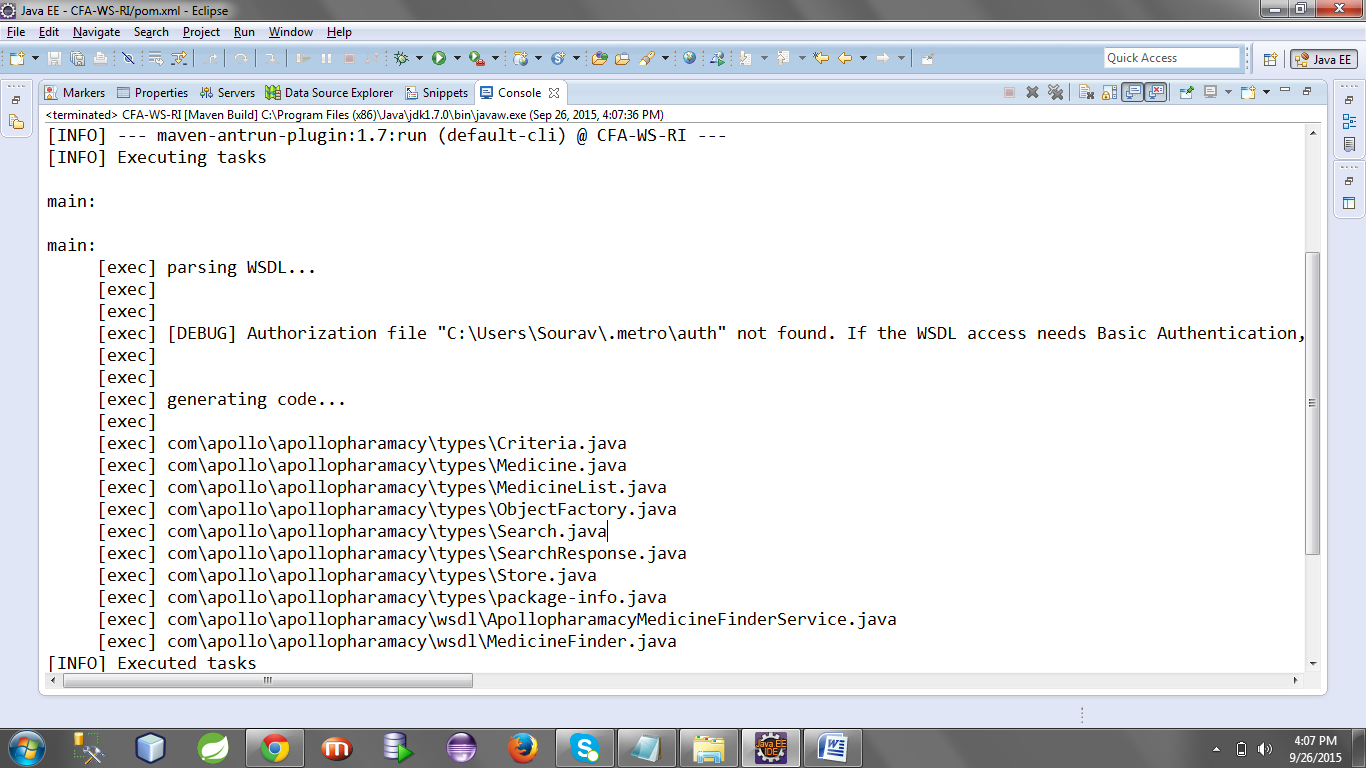
**3 things we have to do here now.**

1. **Give a name: CFA-WS-RI**
2. **Browse Your Project From workspace**
3. **Goals: antrun:run**

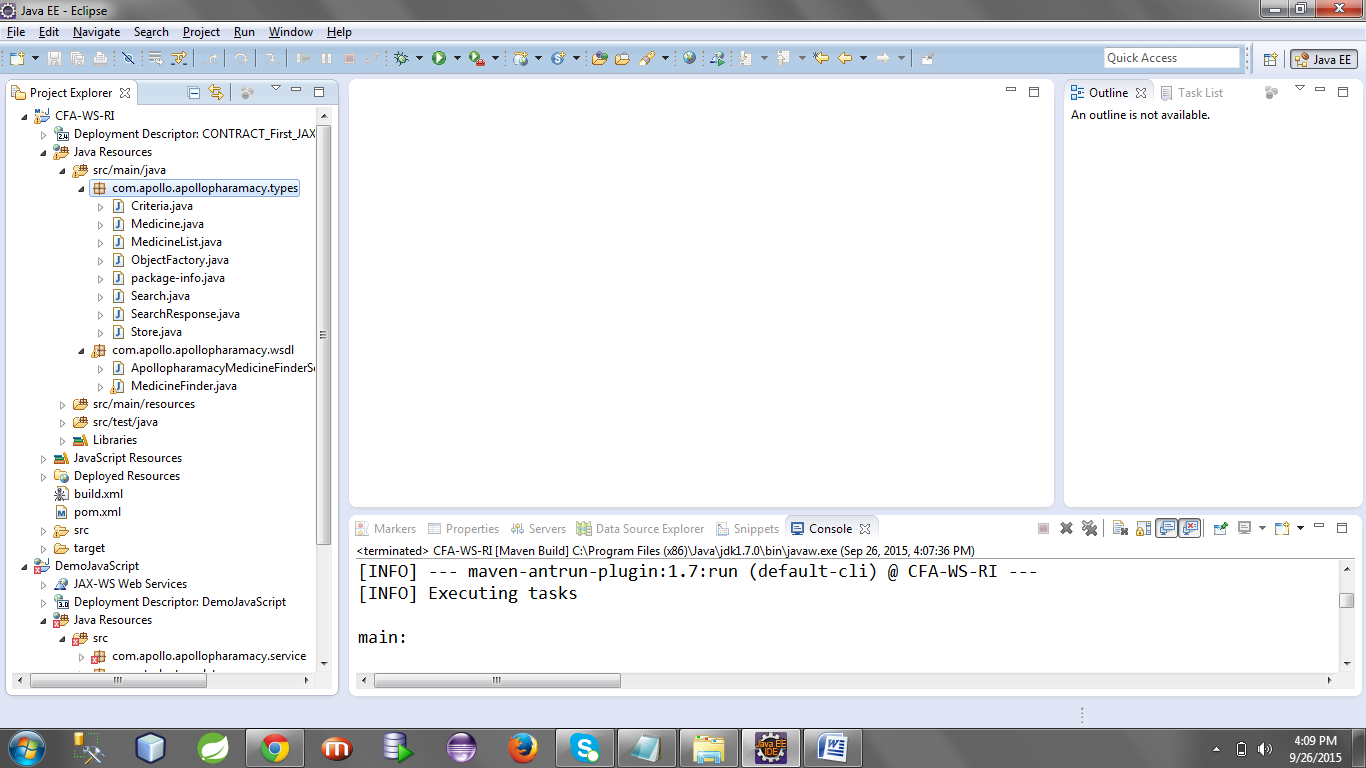
****

**Now click on Apply , then Run**

**Lets watch the console now**

****

**Now refresh your project.**

****

**Now create a implementation class for your sei interface so generated. Note that MedicineFinder is your sei interface and below is its generated code.**

**MedicineFinder.java**

**package** com.apollo.apollopharamacy.wsdl;

**import** javax.jws.WebMethod;

**import** javax.jws.WebParam;

**import** javax.jws.WebResult;

**import** javax.jws.WebService;

**import** javax.xml.bind.annotation.XmlSeeAlso;

**import** javax.xml.ws.RequestWrapper;

**import** javax.xml.ws.ResponseWrapper;

**import** com.apollo.apollopharamacy.types.Criteria;

**import** com.apollo.apollopharamacy.types.MedicineList;

**import** com.apollo.apollopharamacy.types.ObjectFactory;

**import** com.apollo.apollopharamacy.types.Store;

/\*\*

\* This class was generated by the JAX-WS RI.

\* JAX-WS RI 2.1.6 in JDK 6

\* Generated source version: 2.1

\*

\*/

@WebService(name = "MedicineFinder", targetNamespace = "http://www.apollo.com/apollopharamacy/wsdl")

@XmlSeeAlso({

ObjectFactory.**class**

})

**public** **interface** MedicineFinder {

/\*\*

\*

\* **@param** store

\* **@param** criteria

\* **@return**

\* returns com.apollo.apollopharamacy.types.MedicineList

\*/

@WebMethod(action = "http://www.apollo.com/apollopharamacy/wsdl/search")

@WebResult(name = "ListOfMedicine", targetNamespace = "http://www.apollo.com/apollopharamacy/types")

@RequestWrapper(localName = "search", targetNamespace = "http://www.apollo.com/apollopharamacy/types", className = "com.apollo.apollopharamacy.types.Search")

@ResponseWrapper(localName = "searchResponse", targetNamespace = "http://www.apollo.com/apollopharamacy/types", className = "com.apollo.apollopharamacy.types.SearchResponse")

**public** MedicineList search(

@WebParam(name = "criteria", targetNamespace = "http://www.apollo.com/apollopharamacy/types")

Criteria criteria,

@WebParam(name = "store", targetNamespace = "http://www.apollo.com/apollopharamacy/types")

Store store);

}

**MedicineFinderImpl.java**

**package** com.apollo.apollopharamacy.service;

**import** javax.jws.WebService;

**import** com.apollo.apollopharamacy.types.Criteria;

**import** com.apollo.apollopharamacy.types.Medicine;

**import** com.apollo.apollopharamacy.types.MedicineList;

**import** com.apollo.apollopharamacy.types.Store;

**import** com.apollo.apollopharamacy.wsdl.MedicineFinder;

@WebService(endpointInterface="com.apollo.apollopharamacy.wsdl.MedicineFinder",serviceName="apollopharamacyMedicineFinderService",portName="apollopharamacySOAPPort")

**public** **class** MedicineFinderImpl **implements** MedicineFinder {

// @Override

**public** MedicineList search(Criteria criteria, Store store) {

// **TODO** Auto-generated method stub

MedicineList medList=**null**;

Medicine med=**null**;

medList=**new** MedicineList();

**for**(**int** i=1;i<5;i++)

{

med=**new** Medicine();

med.setAvailableQty(10);

med.setGeneric(**true**);

med.setManufacturer("CIPLA");

med.setMedicineName("SINAREST"+i);

med.setPricePerUnit(345.67f);

medList.getMedicineList().add(med);

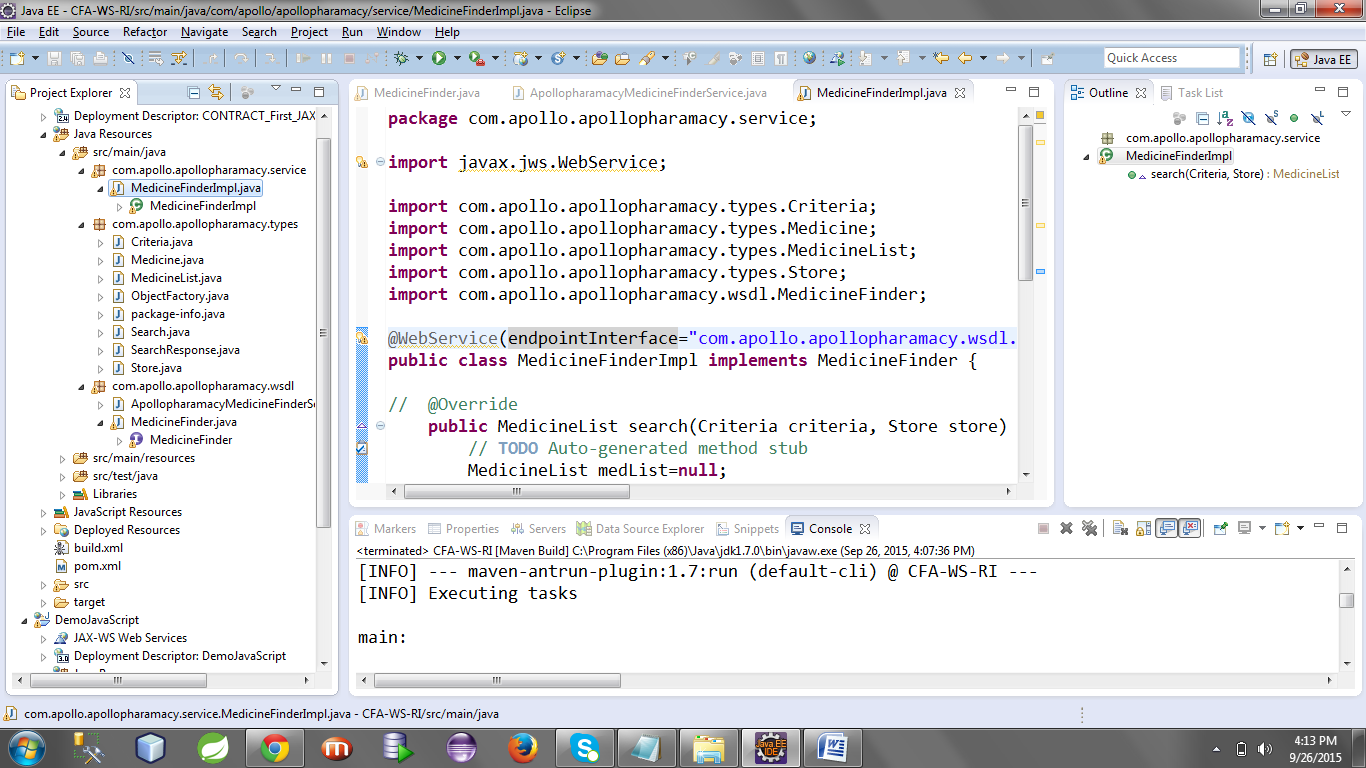
}

System.*out*.println("jklk");

**return** medList;

}

}

****

**Now set up 2 files sun-jaxws.xml and web.xml as below.**

**sun-jaxws.xml [Keep it inside WEB-INF]**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!--

The contents of this file are subject to the terms

of the Common Development and Distribution License

(the "License"). You may not use this file except

in compliance with the License.

You can obtain a copy of the license at

https://jwsdp.dev.java.net/CDDLv1.0.html

See the License for the specific language governing

permissions and limitations under the License.

When distributing Covered Code, include this CDDL

HEADER in each file and include the License file at

https://jwsdp.dev.java.net/CDDLv1.0.html If applicable,

add the following below this CDDL HEADER, with the

fields enclosed by brackets "[]" replaced with your

own identifying information: Portions Copyright [yyyy]

[name of copyright owner]

-->

<endpoints xmlns=*'http://java.sun.com/xml/ns/jax-ws/ri/runtime'* version=*'2.0'*>

<endpoint

name=*'ApolloService'*

implementation=*'com.apollo.apollopharamacy.service.MedicineFinderImpl'*

url-pattern=*'/medicine'*/>

</endpoints>

**Note that our webservice endpoint url-pattern is /medicine**

**Web.xml [keep it inside WEB-INF]**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/javaee"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"* id=*"WebApp\_ID"* version=*"3.0"*>

<display-name>CONTRACT\_First\_JAXWS\_OxforBookLibraryService</display-name>

<listener>

<listener-class>com.sun.xml.ws.transport.http.servlet.WSServletContextListener</listener-class>

</listener>

<servlet>

<servlet-name>ApolloService</servlet-name>

<servlet-class>com.sun.xml.ws.transport.http.servlet.WSServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>ApolloService</servlet-name>

<url-pattern>/medicine</url-pattern>

</servlet-mapping>

</web-app>

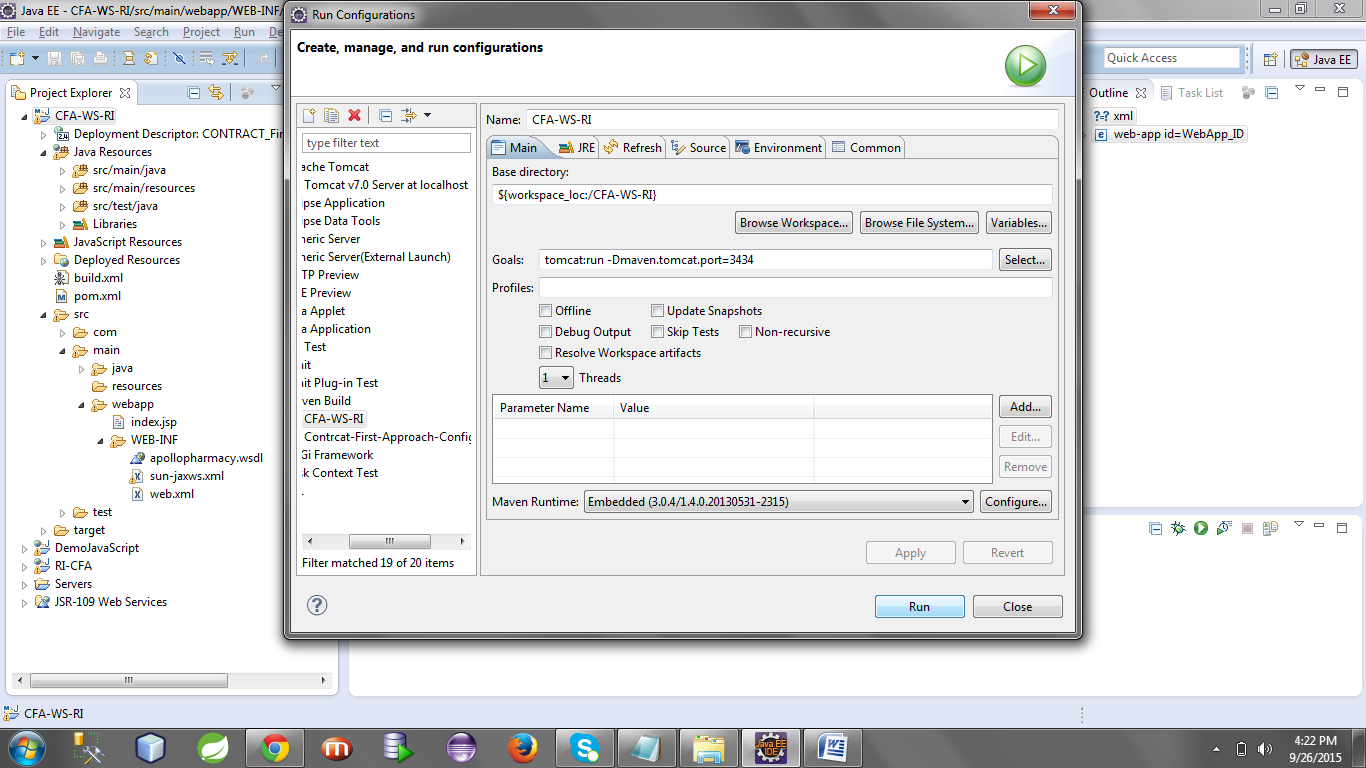
**Now its time to deploy our webservice into server.**

**Right click on your project > Run as > run configuration.**

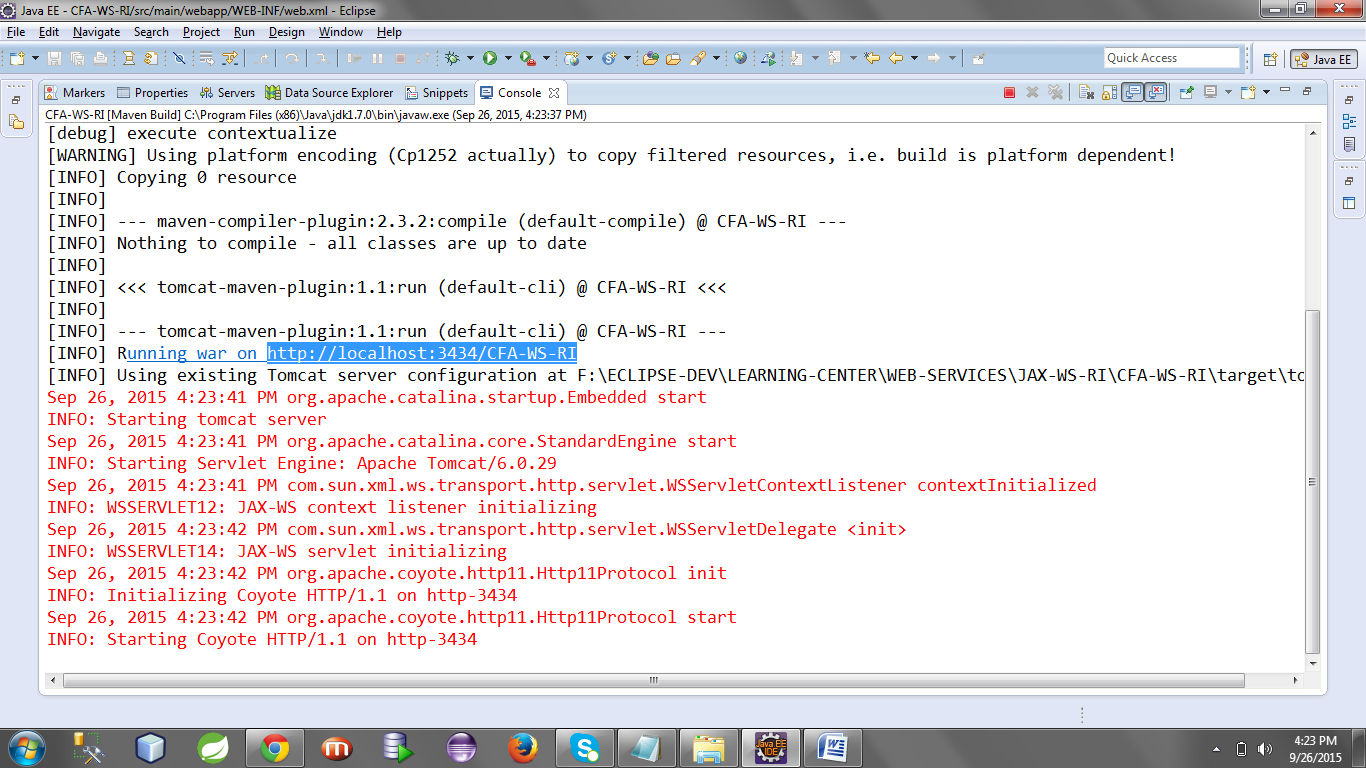
**Here we have to set the goal only.**

**Goal: tomcat:run –Dmaven.tomcat.port=3434**

**Now click on Apply and then Run**

****

**Now look at the console and get the war location.**

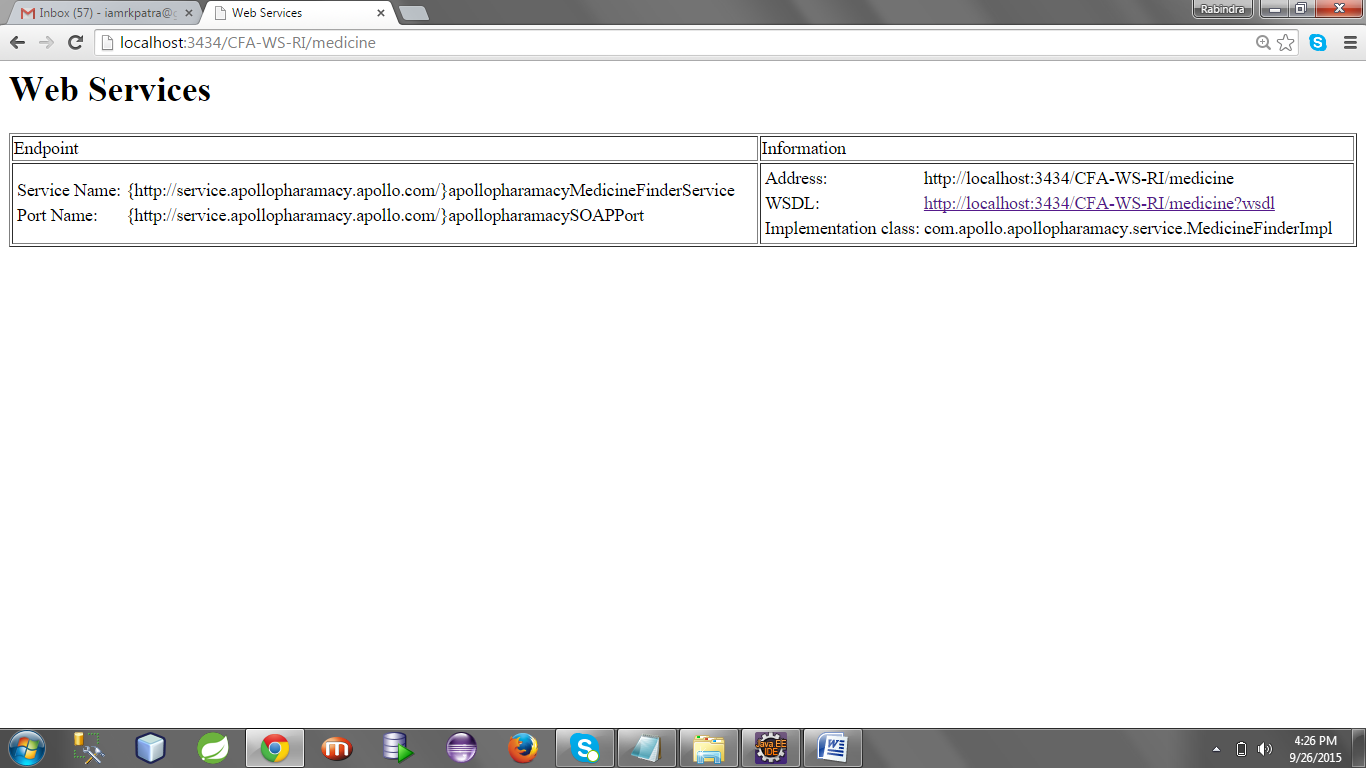
****

**Our webapp is running at :** <http://localhost:3434/CFA-WS-RI>

**Our webservice endpoint is :** **/medicine**

**Hence we can discover our webservice at:** <http://localhost:3434/CFA-WS-RI/medicine>

**Copy this URL and browse through chrome and it will show you the web service dashboard.**



**Now click on the wsdl to get the wsdl url as below**

[**http://localhost:3434/CFA-WS-RI/medicine?wsdl**](http://localhost:3434/CFA-WS-RI/medicine?wsdl)

****

**Now you have the wsdl url. You can easily test this webservice by using soap UI.**