**JAX-WS**

3 Implementations

1. Sun RI
2. Apache Axis2
3. Apache CXF

1.Contract First Approach[wsdl – java , Bottom Up]

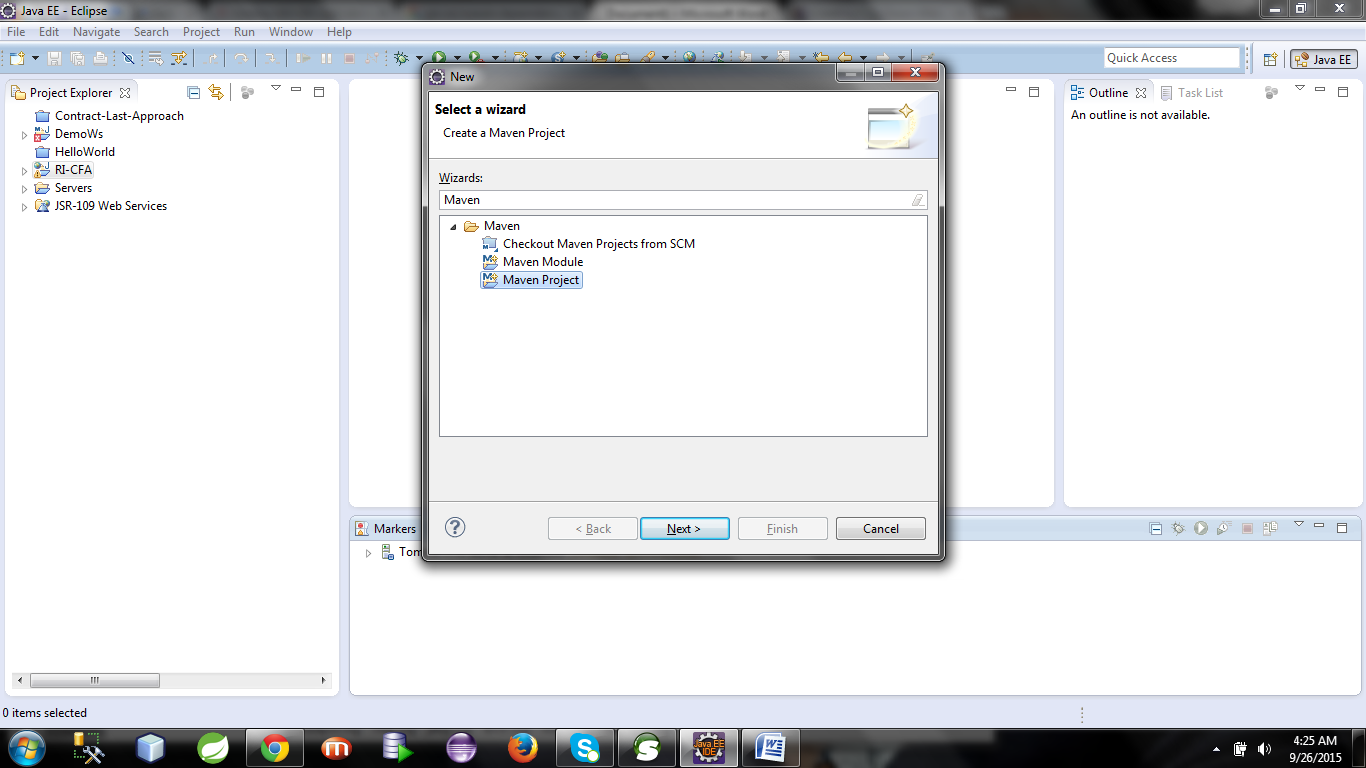
2. Contract Last Approach[ java –wsdl , Top Down]

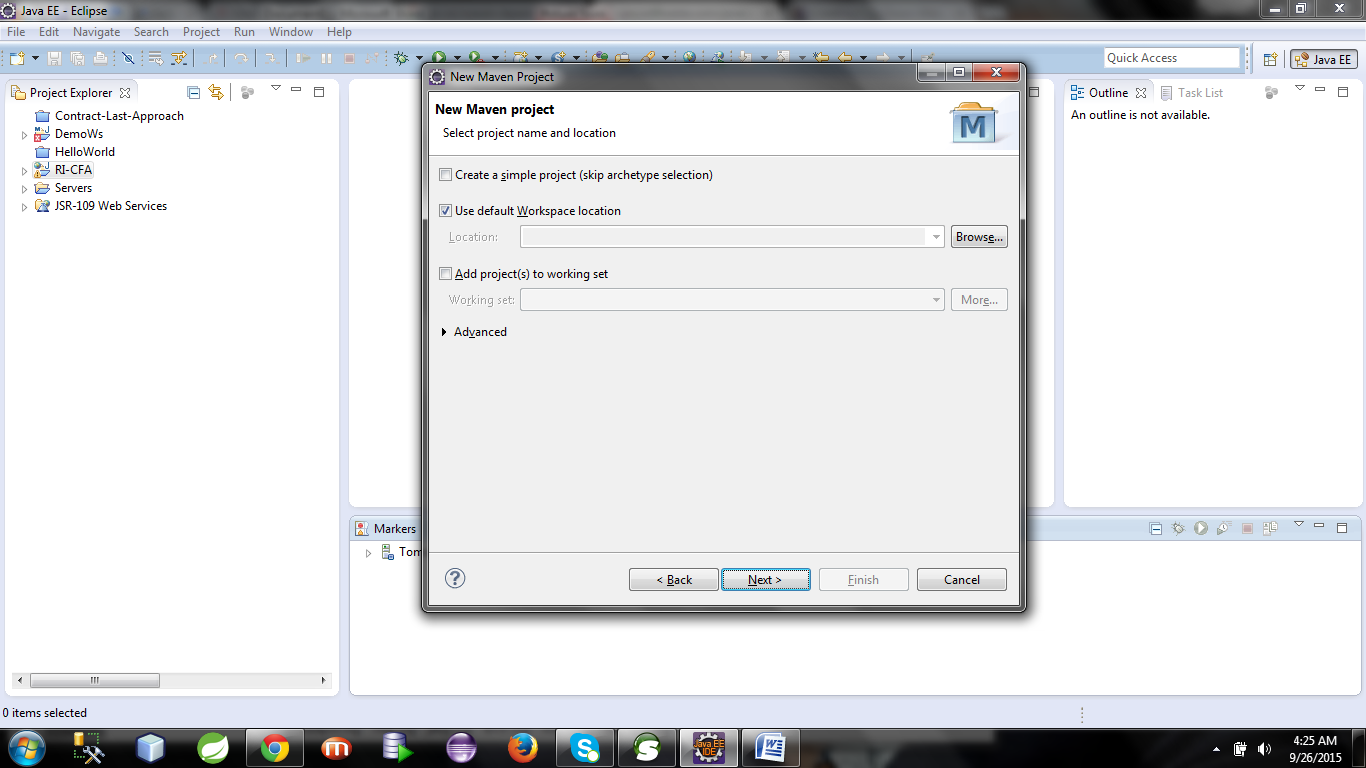
Tools Used: Maven For jar Dependency

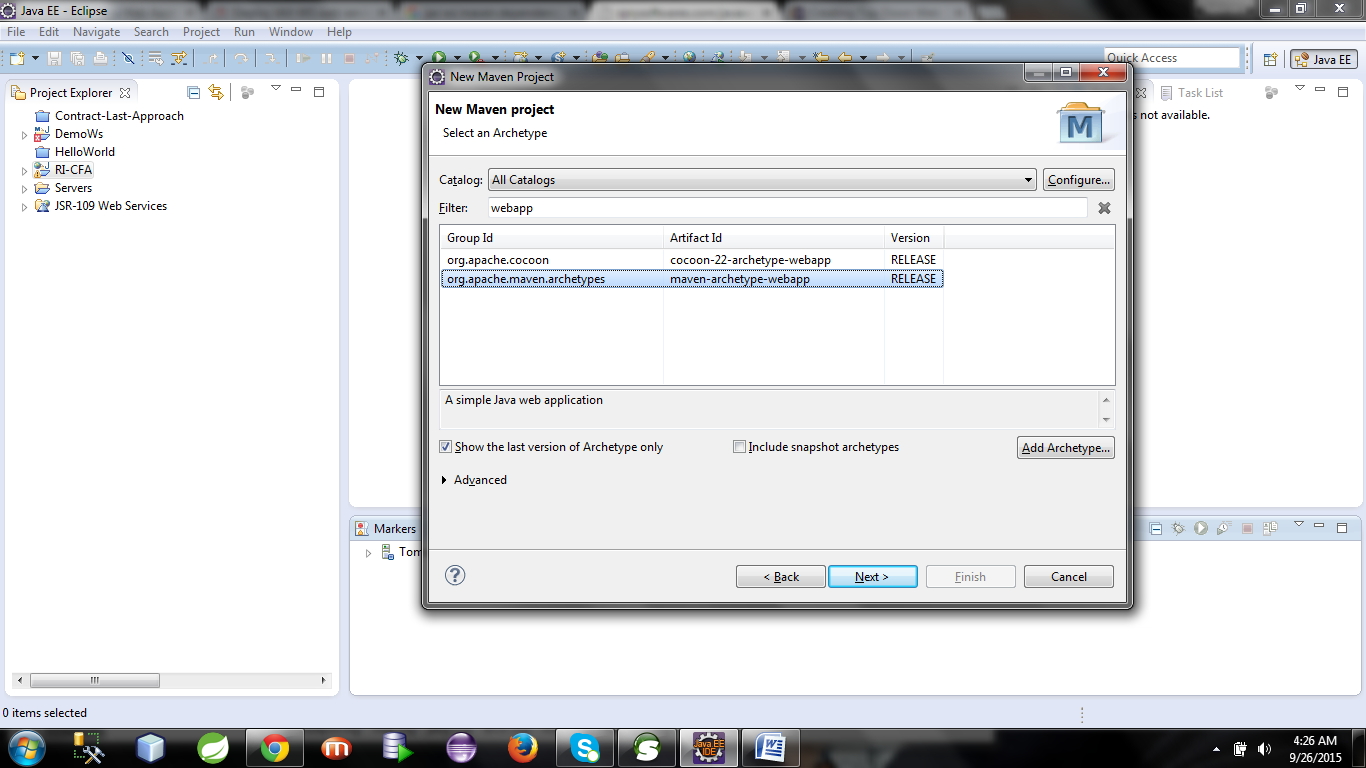
**RI Implementation**

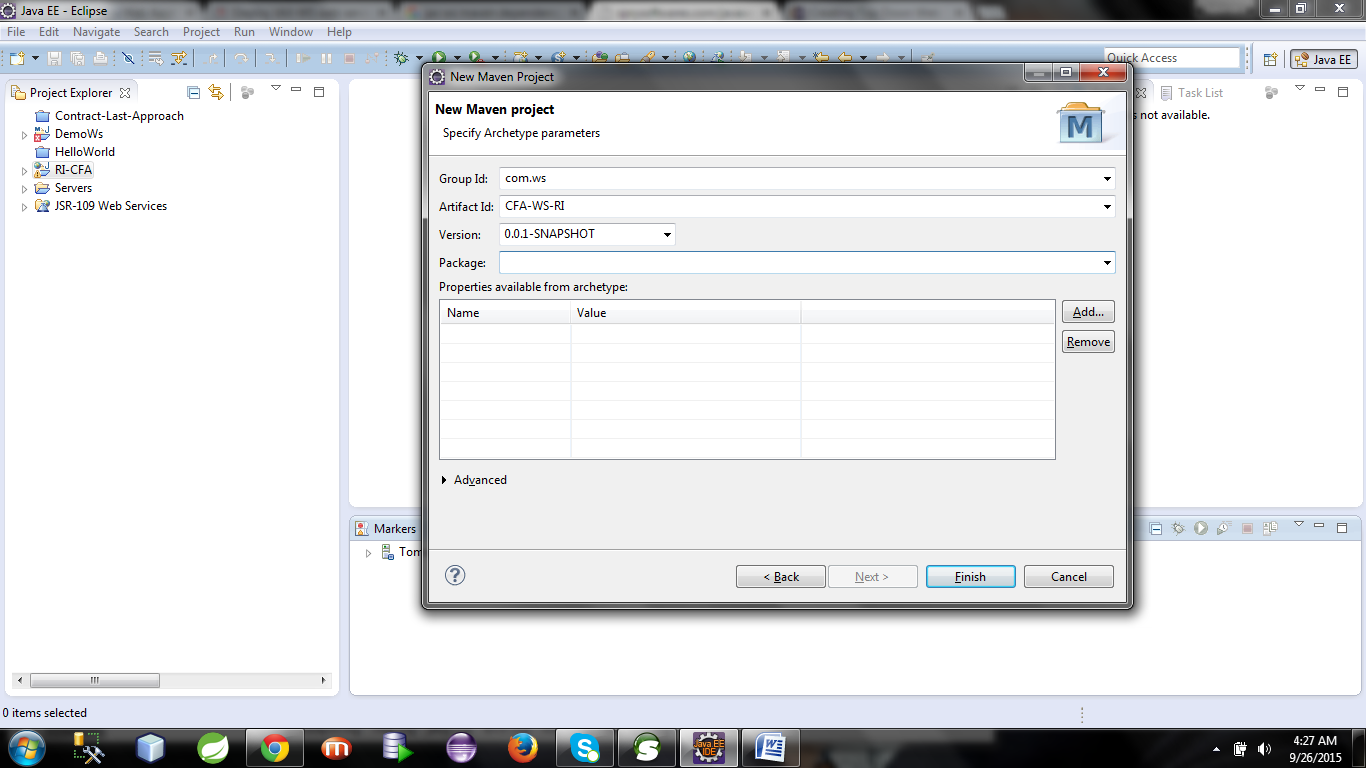
**Contract First Approach:**

Creating a Maven WebApp Project

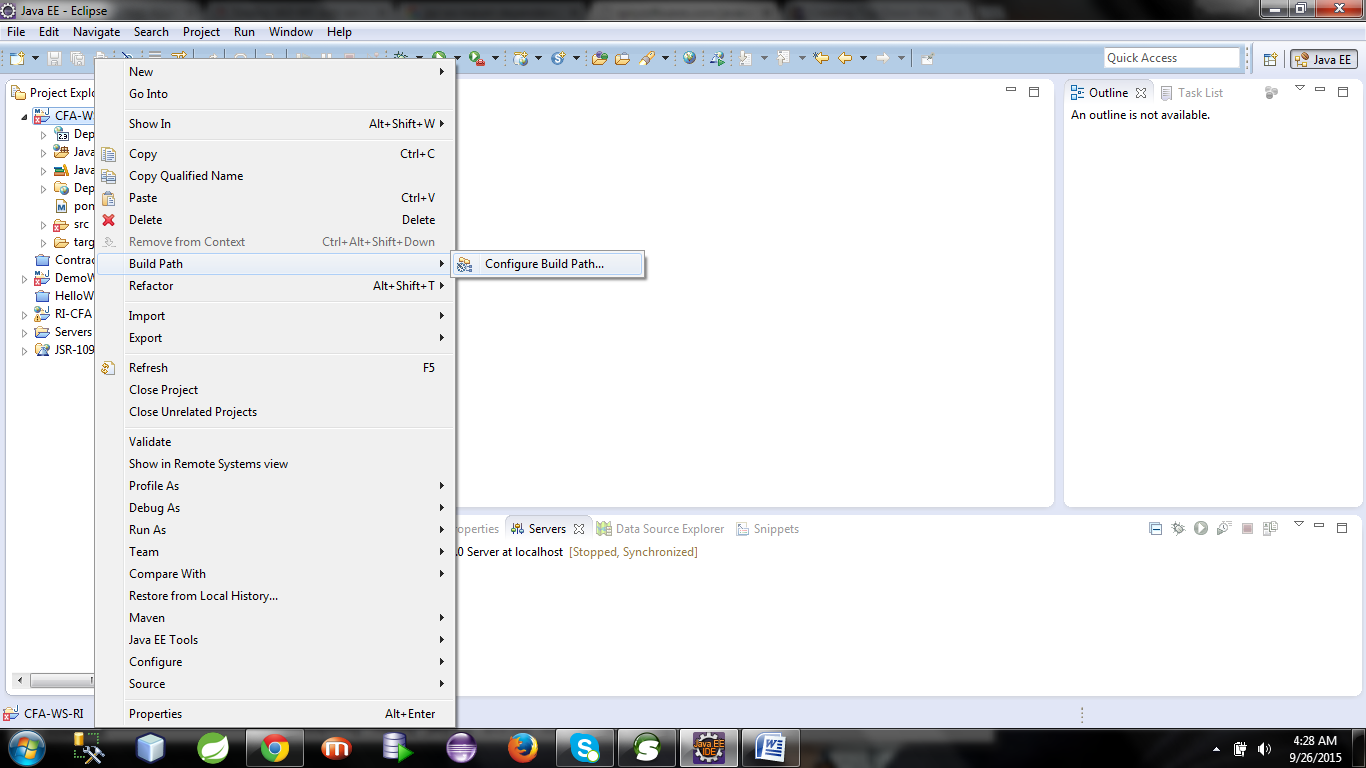
****

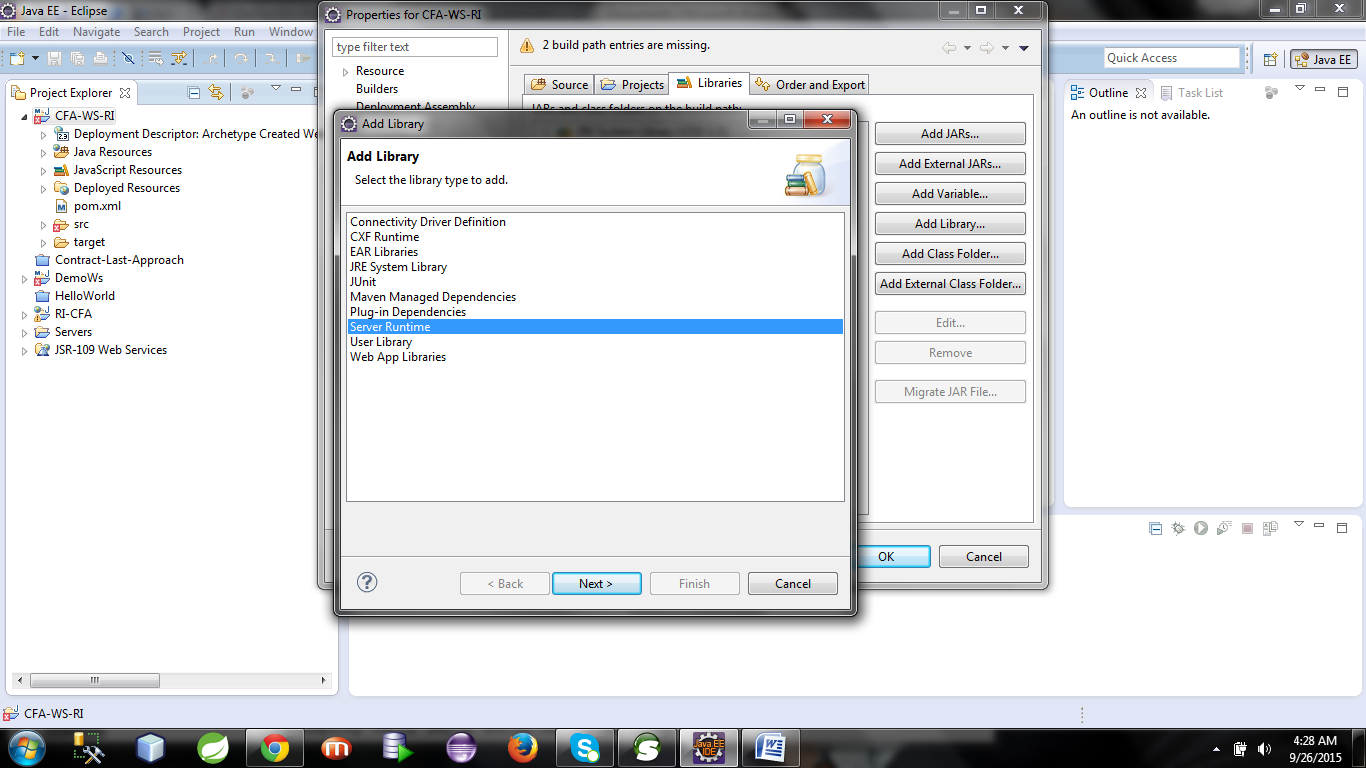
****

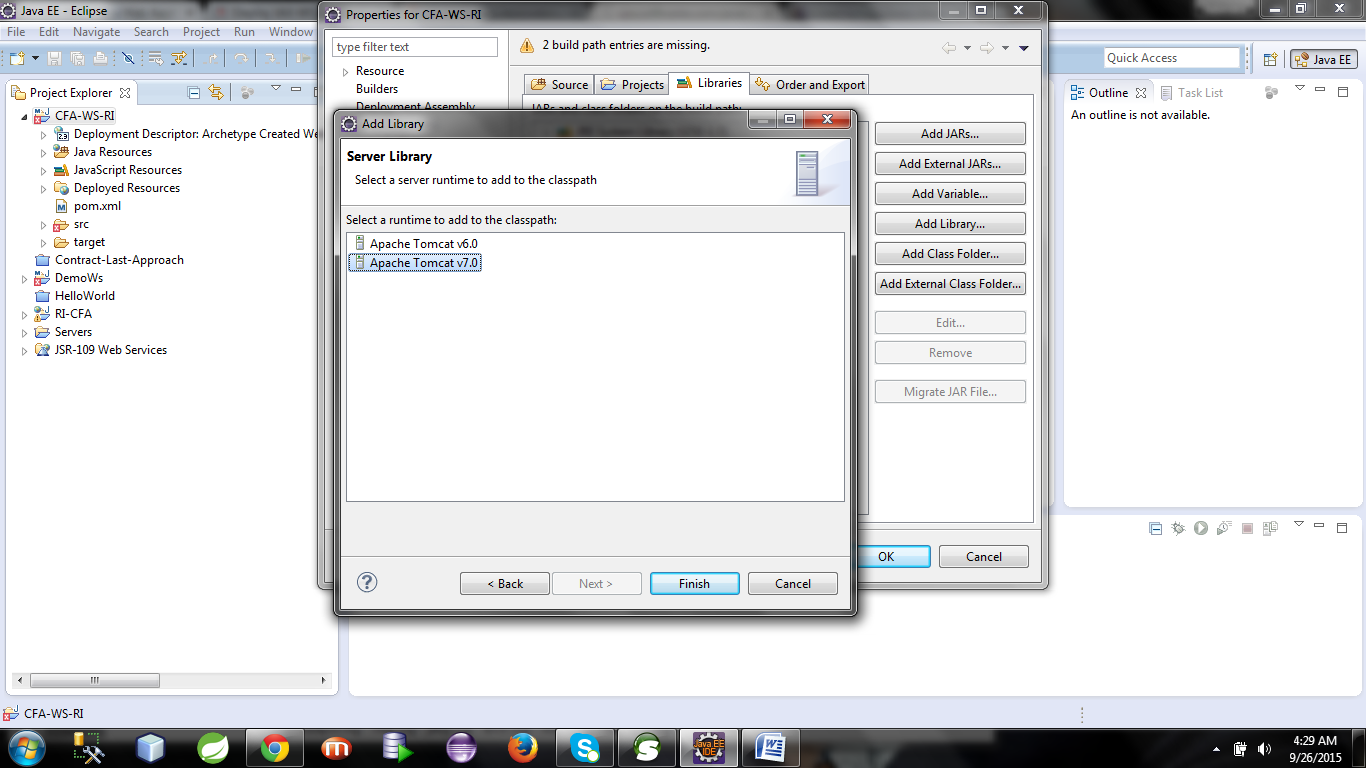
****

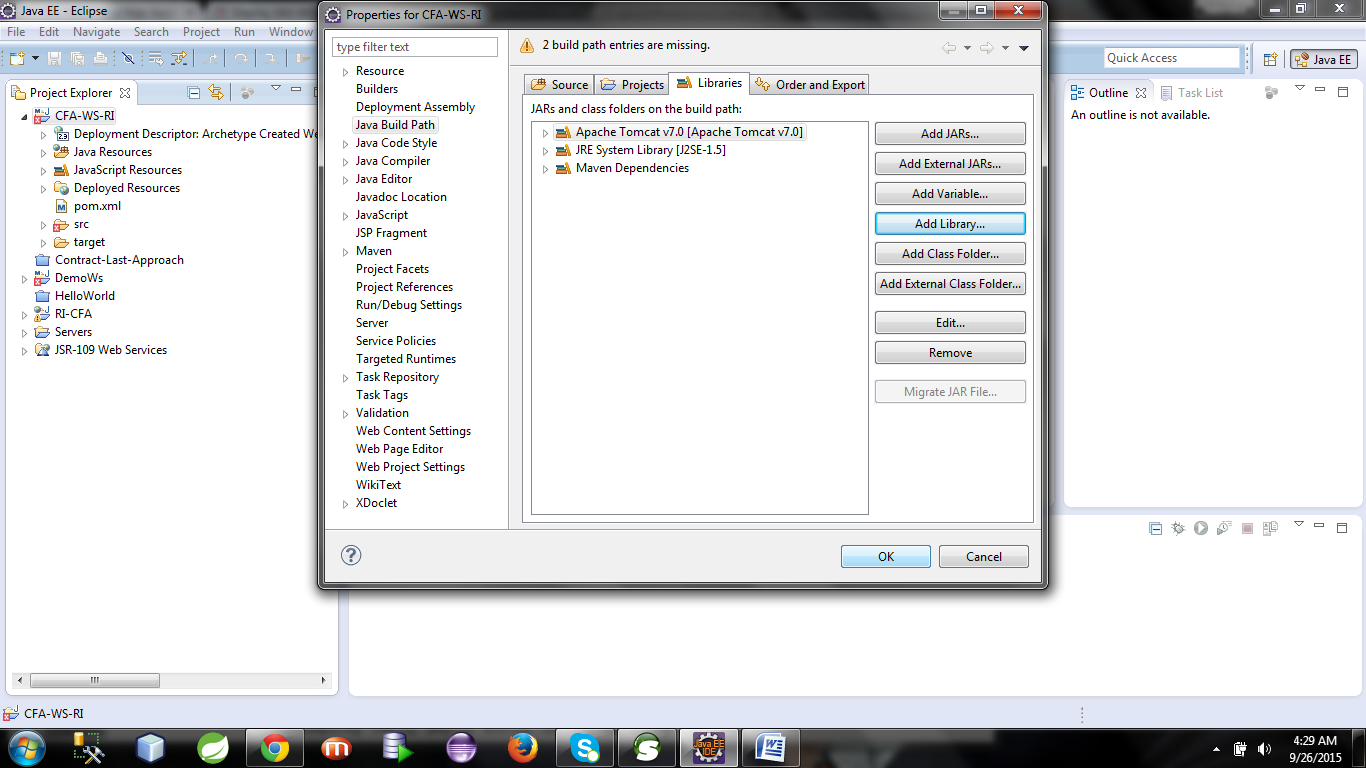
****

Configuring a Tomcat 7.0 Server

****

****

****

****

**Pom.xml**

Dependency jars for jax ws has been configured here

<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>

<finalName>CFA-WS-RI</finalName>

</build>

</project>

As we are working with Contract First Approach We must have a wsdl for our business application. Lets consider the below scenario.

Developing a provider using **contract first approach**

**public interface MedicineFinder {**

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

**}**

We have to develope a wsdl for this interface.  
Medicine list is a collection type.

**MedicineList=List<Medicine>//important note**

So we are going to build a wsdl now. Let the wsdl name is **apollopharmacy.wsdl.** Place this wsdl as part of 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"*>

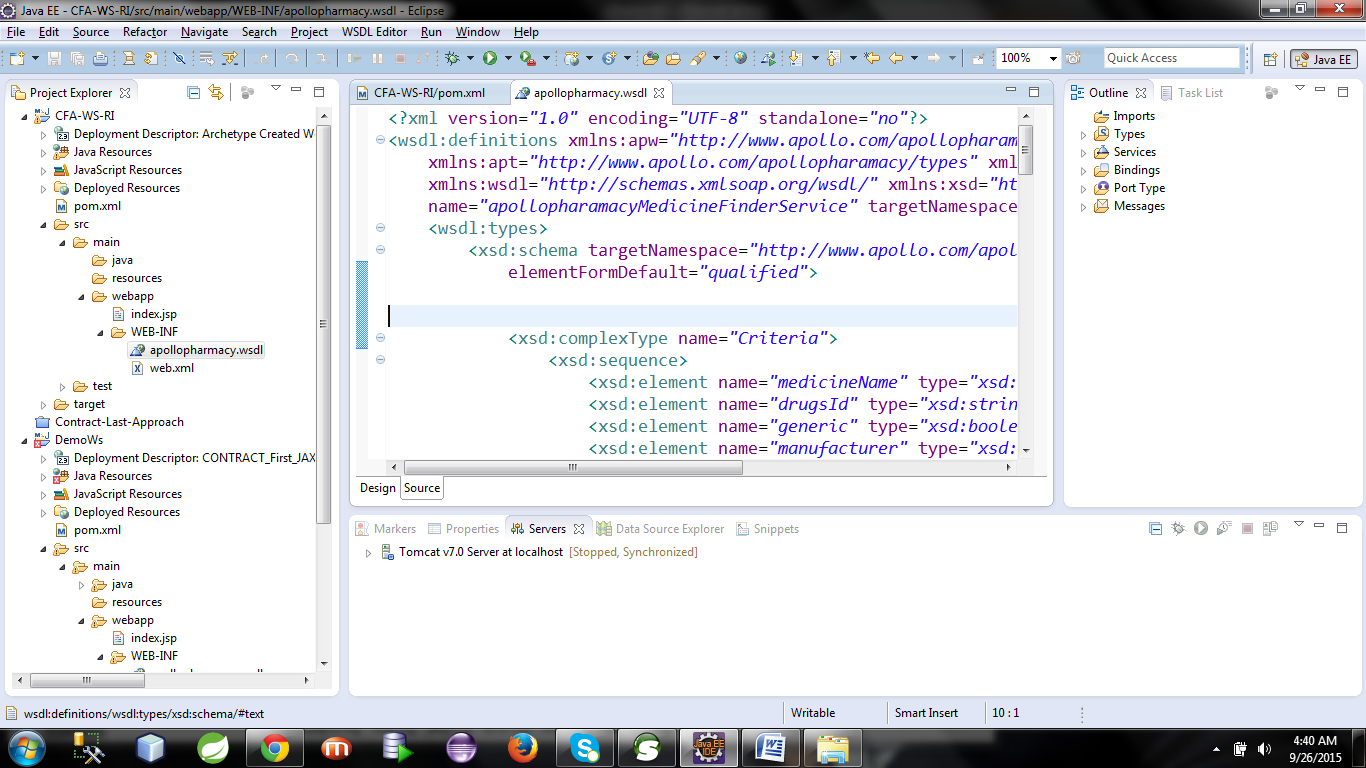
<wsdl:port binding=*"apw:apollopharamacySOAPBinding"* name=*"apollopharamacySOAPPort"*>

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

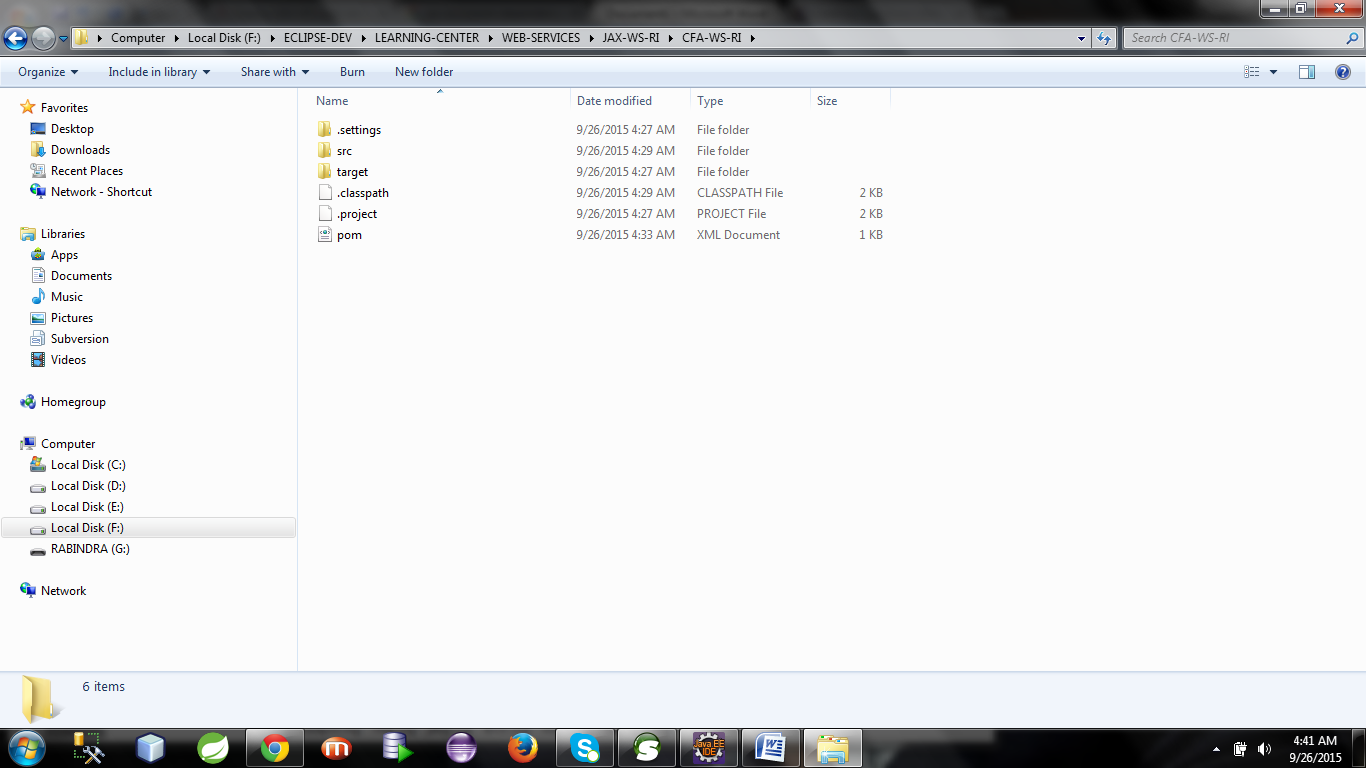
</wsdl:port>

</wsdl:service>

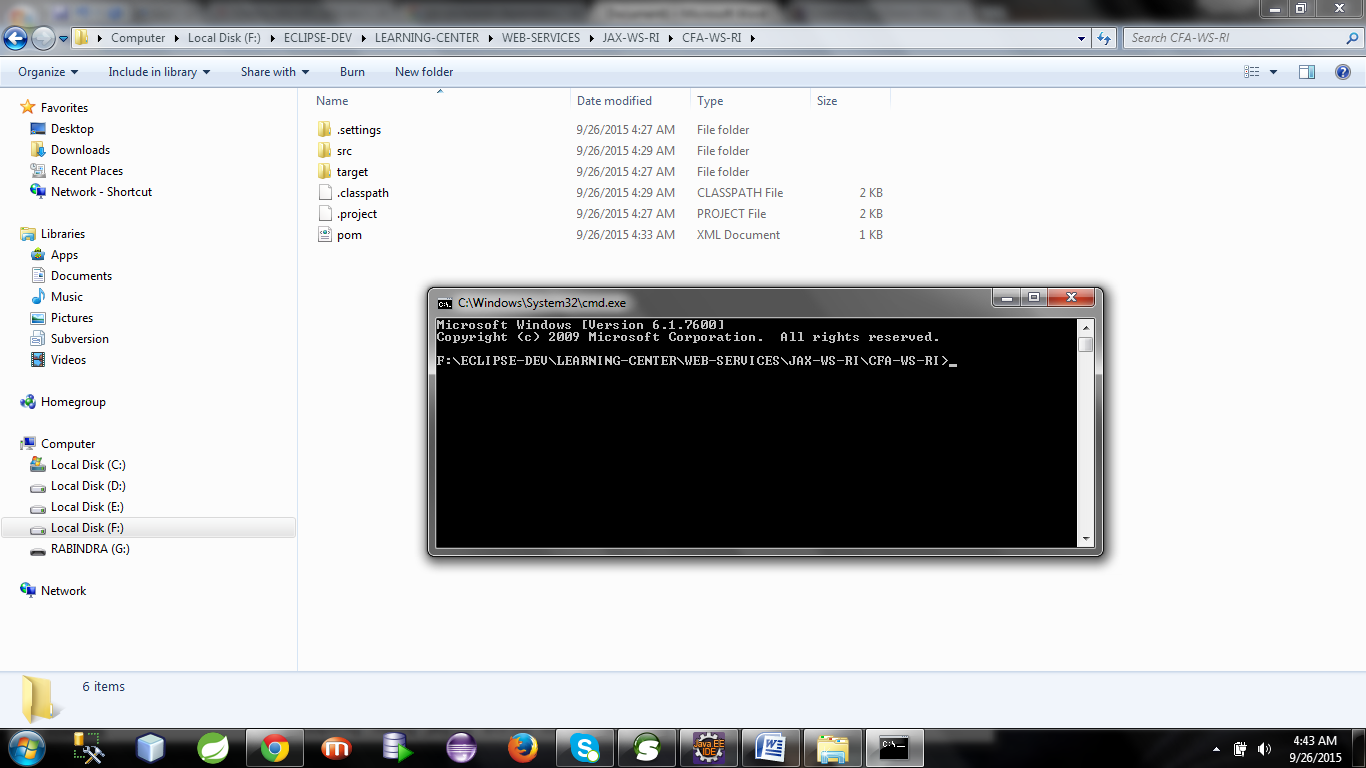
</wsdl:definitions>



Now move to the project root directory which is located inside your current workspace.



Open a command prompt in this location



**>set JAVA\_HOME=C:\Program Files\Java\jdk1.5.0\_10  
>set path=%JAVA\_HOME%\bin  
>set path=%path%;C:\Sun\jwsdp-2.0\jaxws\bin**

But here I don’t need to set first two commands because I have configured those as part of my environment variables.

So I only executed the third command only in the command prompt.

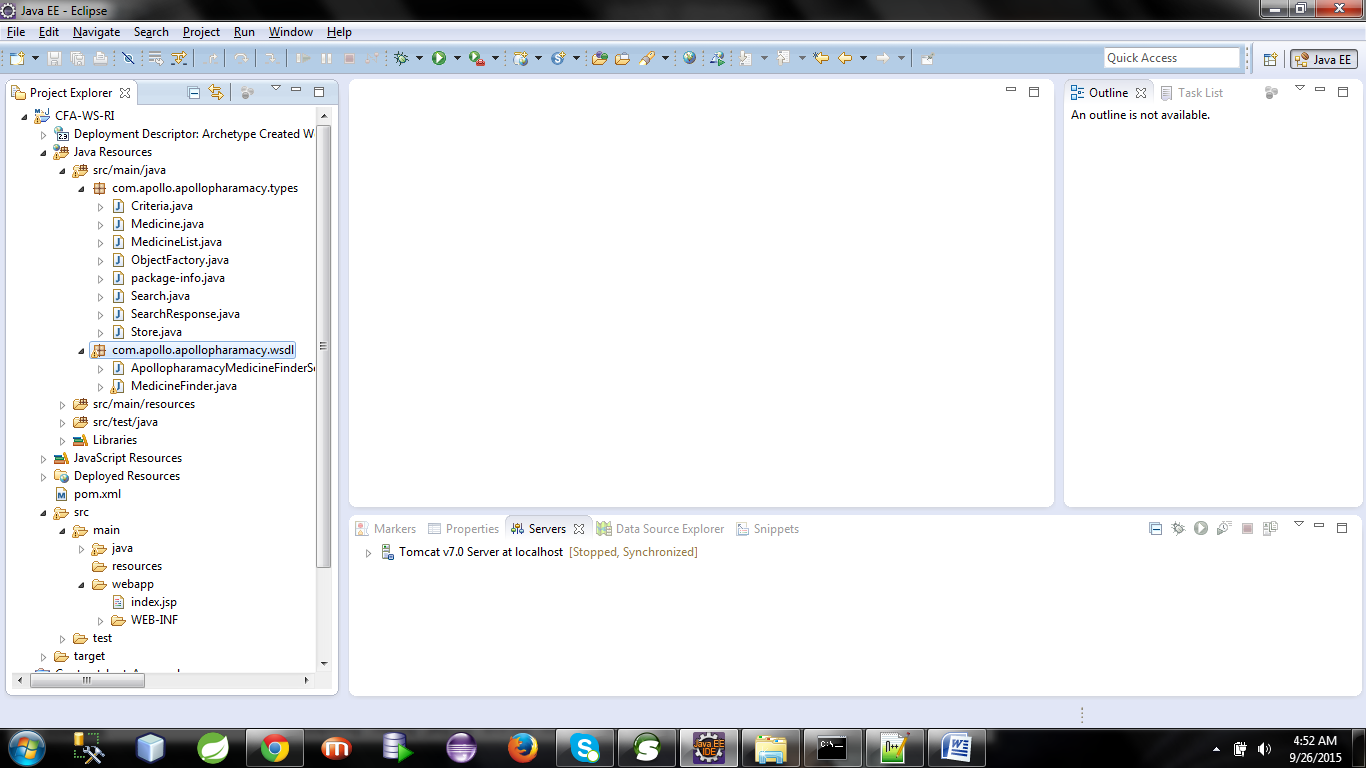
For contract first approach we have to run the **wsimport tool privided by jaxws ri implementation.**

Type the below command at the project root directory:

**>** **wsimport -d src\main\java -keep -verbose src\main\webapp\WEB-INF\apollopharmacy.wsdl**

**This will generate  
>all the complex type specific classes  
>sei interface with appropiate annotation  
>service class(to be used by consumer to acces the webservice)**

Now refresh your project to see the generated java classes.



The code for sei interface generated by the wsimport tool is given below.

**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

\* 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);

}

**create the implementation class for sei interface**

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

**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;

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

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

**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);

}

**return** medList;

}

}

Now write **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.wsdl.MedicineFinderImpl'*

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

</endpoints>

Configure the Listener and WsServlet as part of web.xml

<?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>

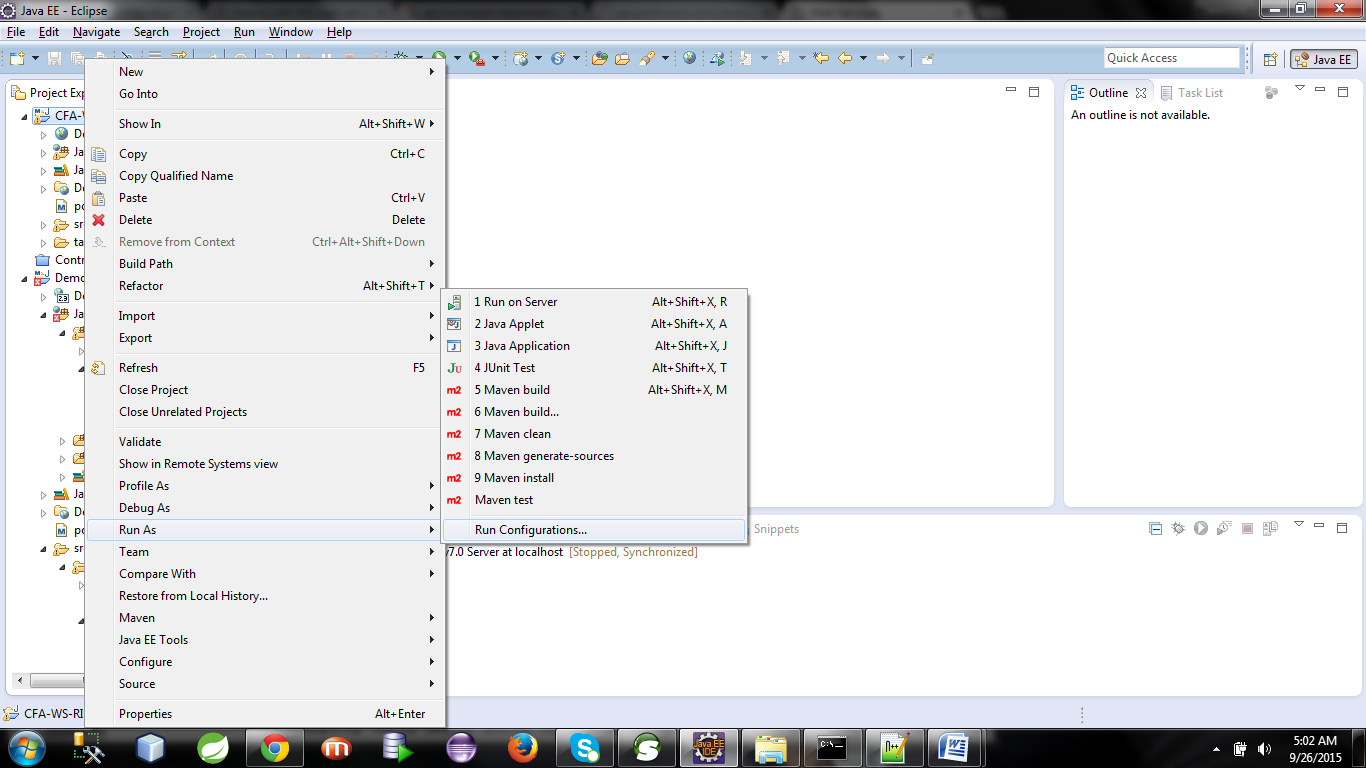
<url-pattern>/medicine</url-pattern>

</servlet-mapping>

</web-app>

Note that we can access our webservice with url-pattern as /medicine

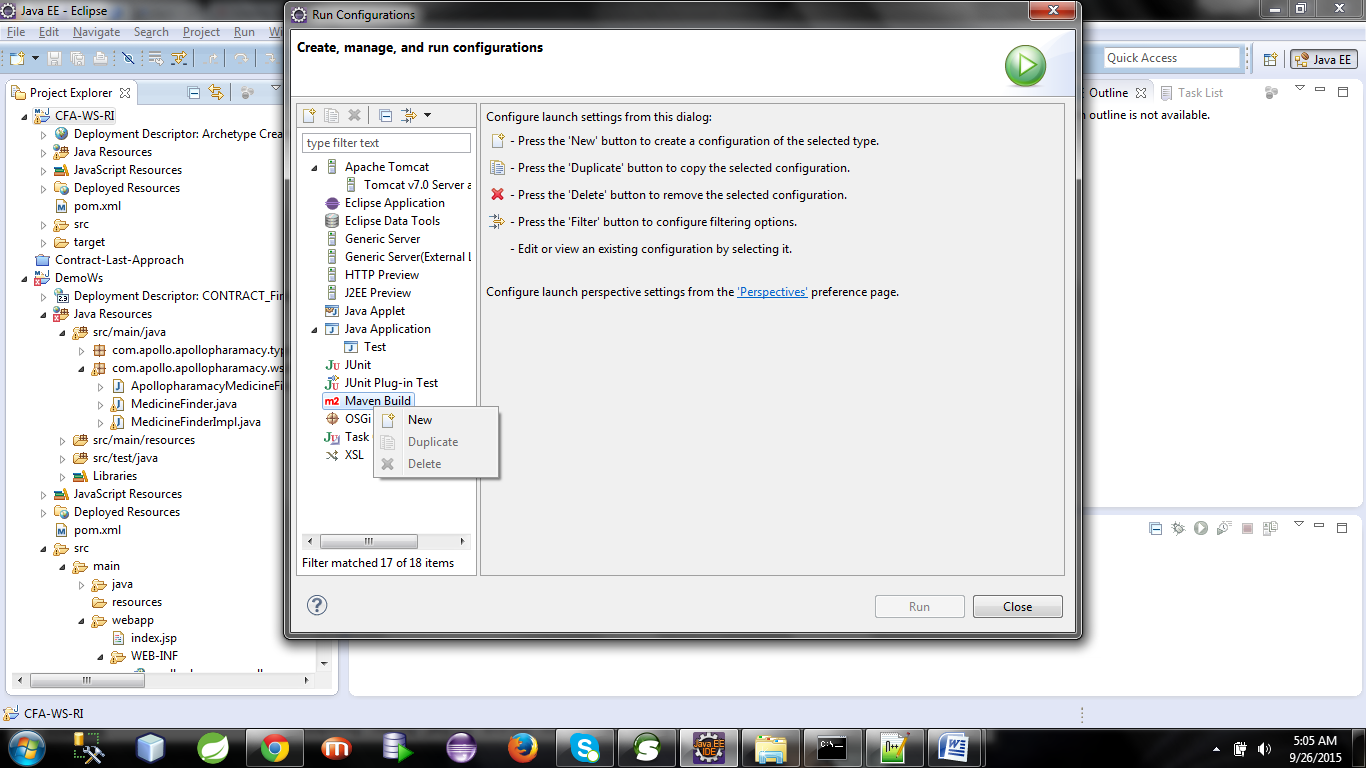
Now deploy the application as below.



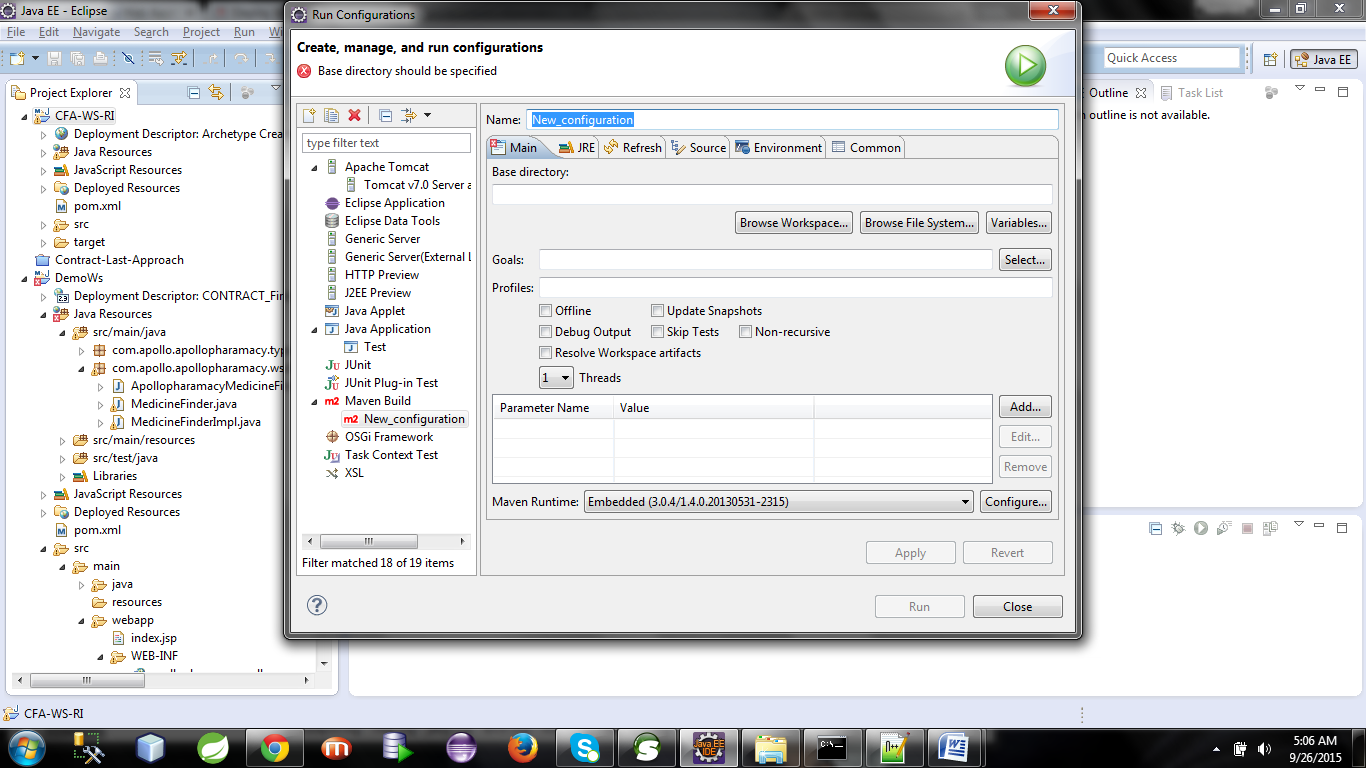
Right click on your project > Run as > Run Configuration

See the left panel for an item named as maven.

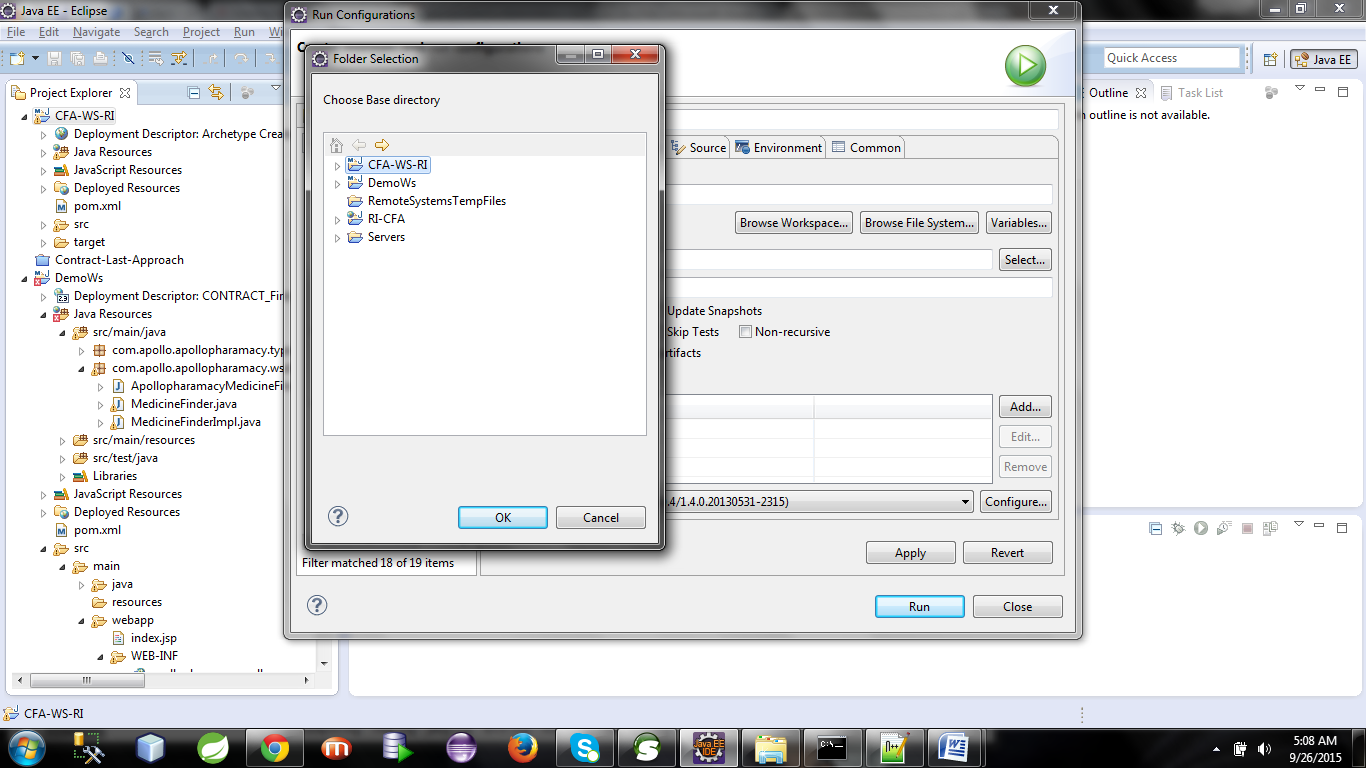
Right click on maven build > new



**After clicking on new the below wizard will open**



Now Browse your workspace to locate your project.

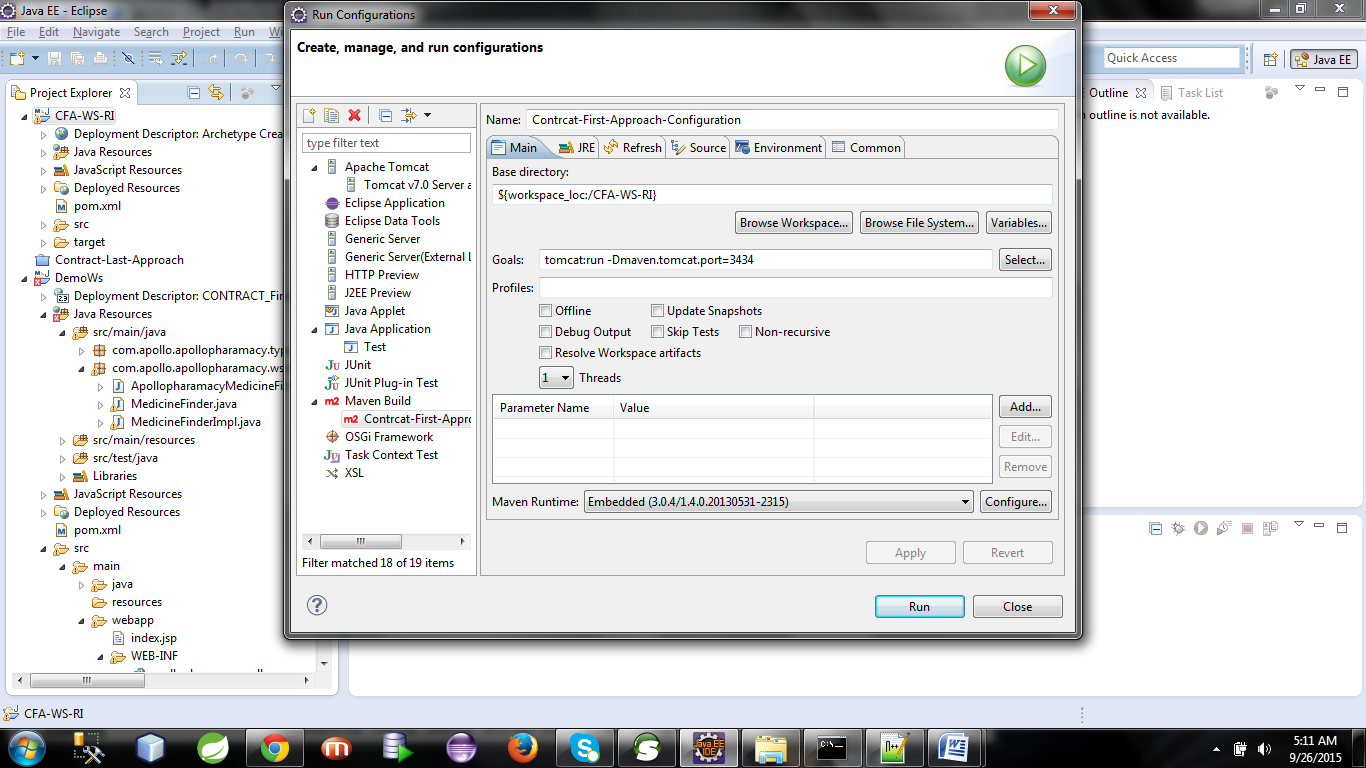


After selecting your project from workspace press ok

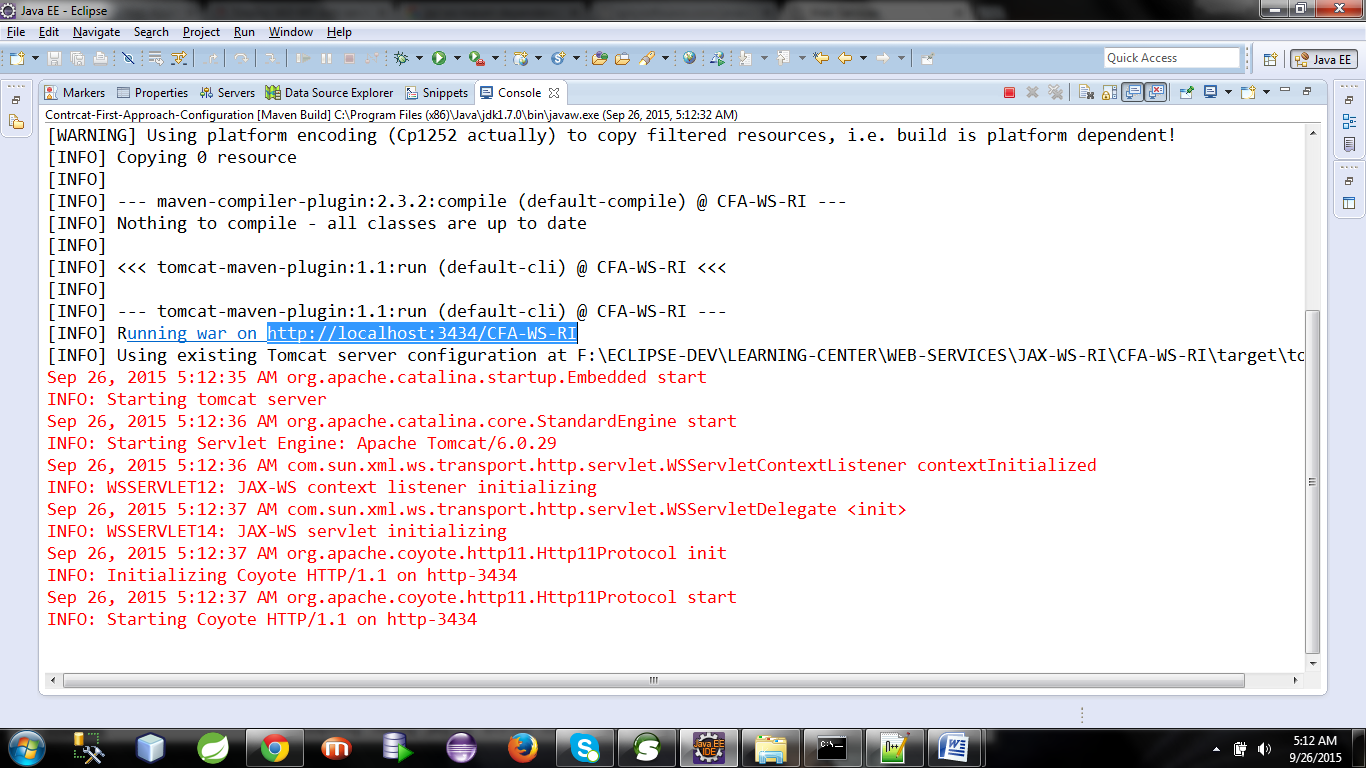
Name : Contrcat-First-Approach-Configuration

Goals: tomcat:run -Dmaven.tomcat.port=3434

Then press Apply and then Run

****

Now look at the console



From the console you can get the location where your application is running.

Copy that location and browse that in your browser [google,firefox,ie] by appending your webservice url-pattern.

Here our application url is <http://localhost:3434/CFA-WS-RI>

WebService url-pattern is /medicine

Hence we can locate our webserviec at

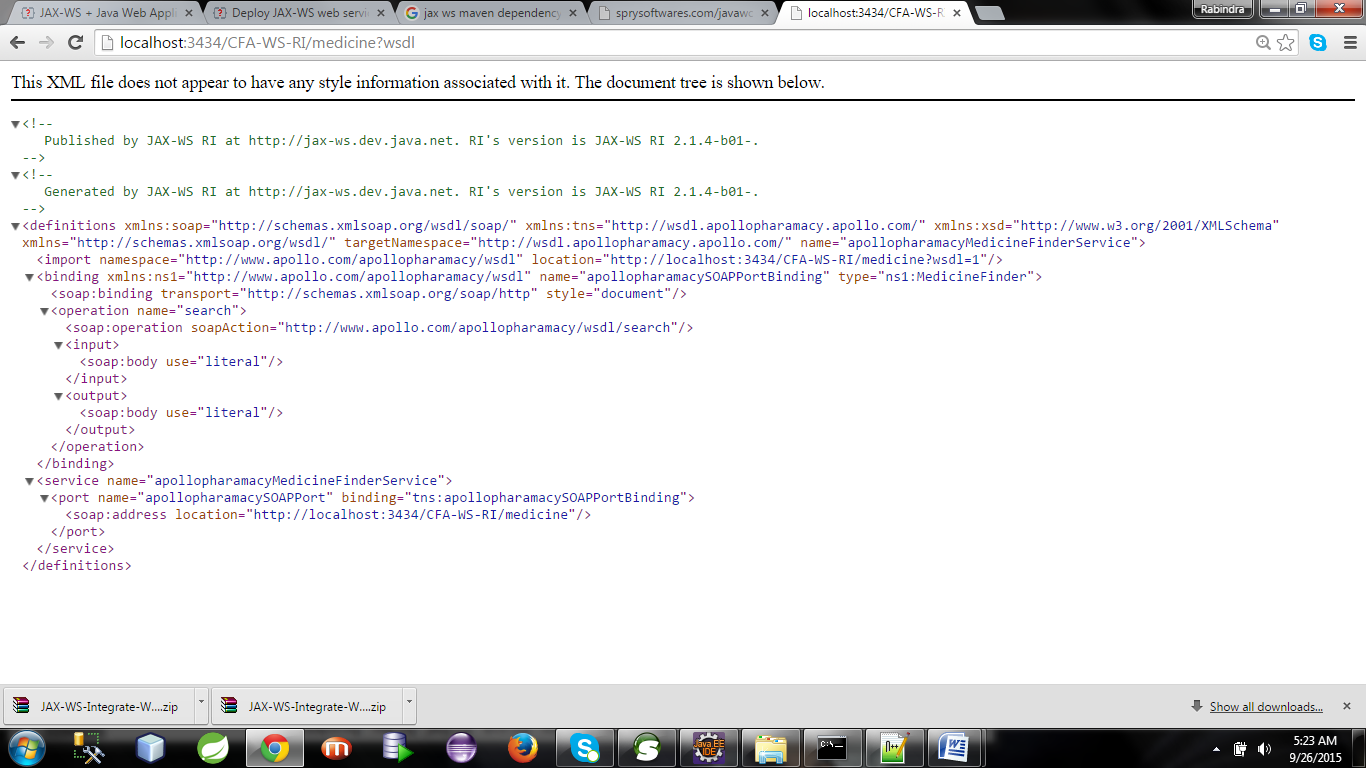
<http://localhost:3434/CFA-WS-RI>/medicine



Now we got the webservice dashboard.

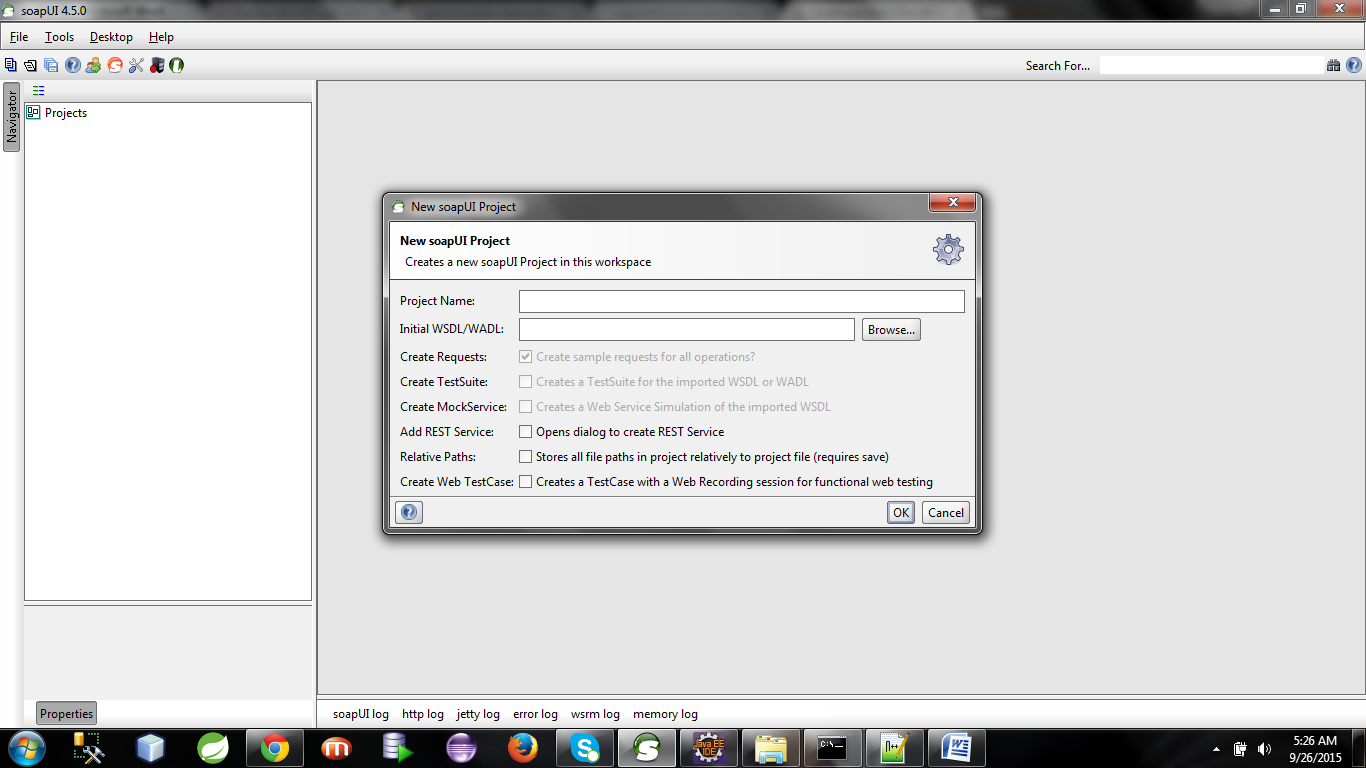
Click on the wsdl to get the wsdl url.

I got the wsdl url as : <http://localhost:3434/CFA-WS-RI/medicine?wsdl>

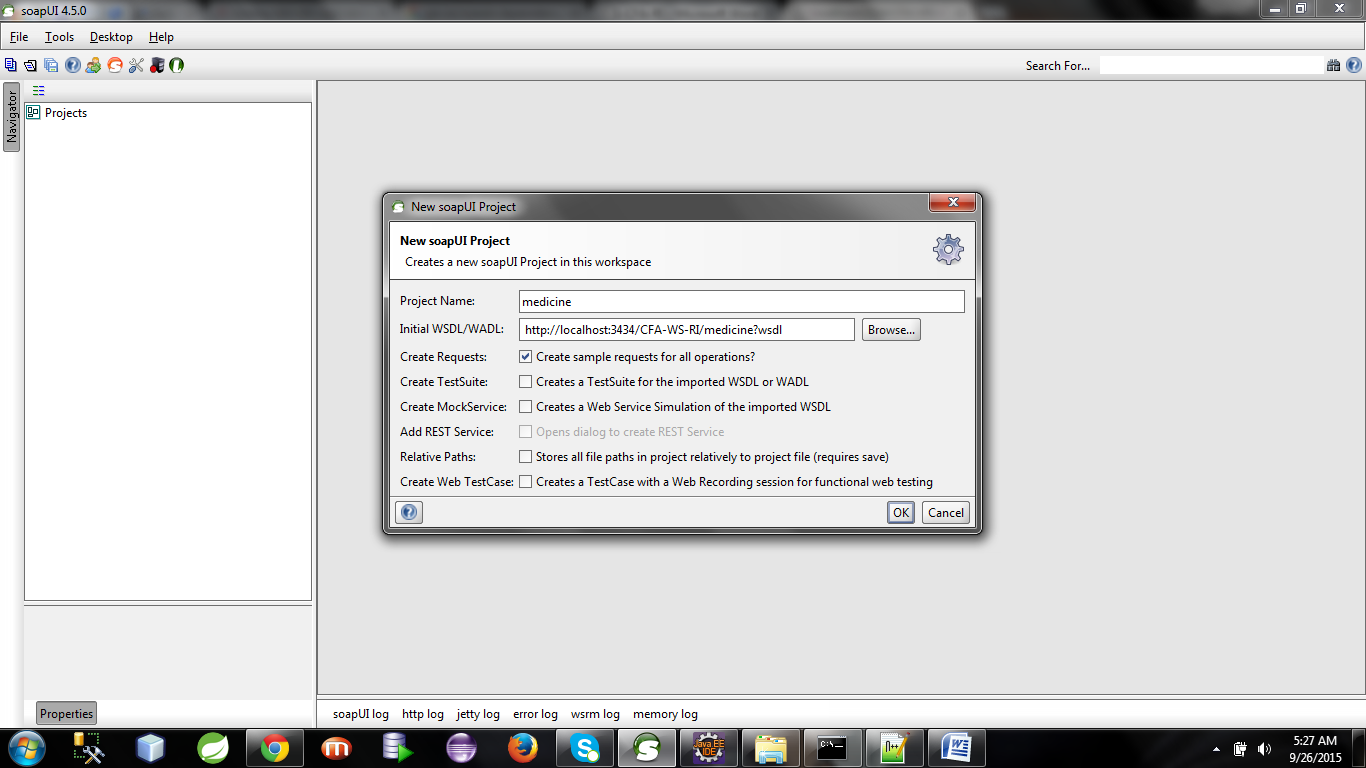


**Now take the wsdl url and open soap ui**

**Press ctrl+N**

****

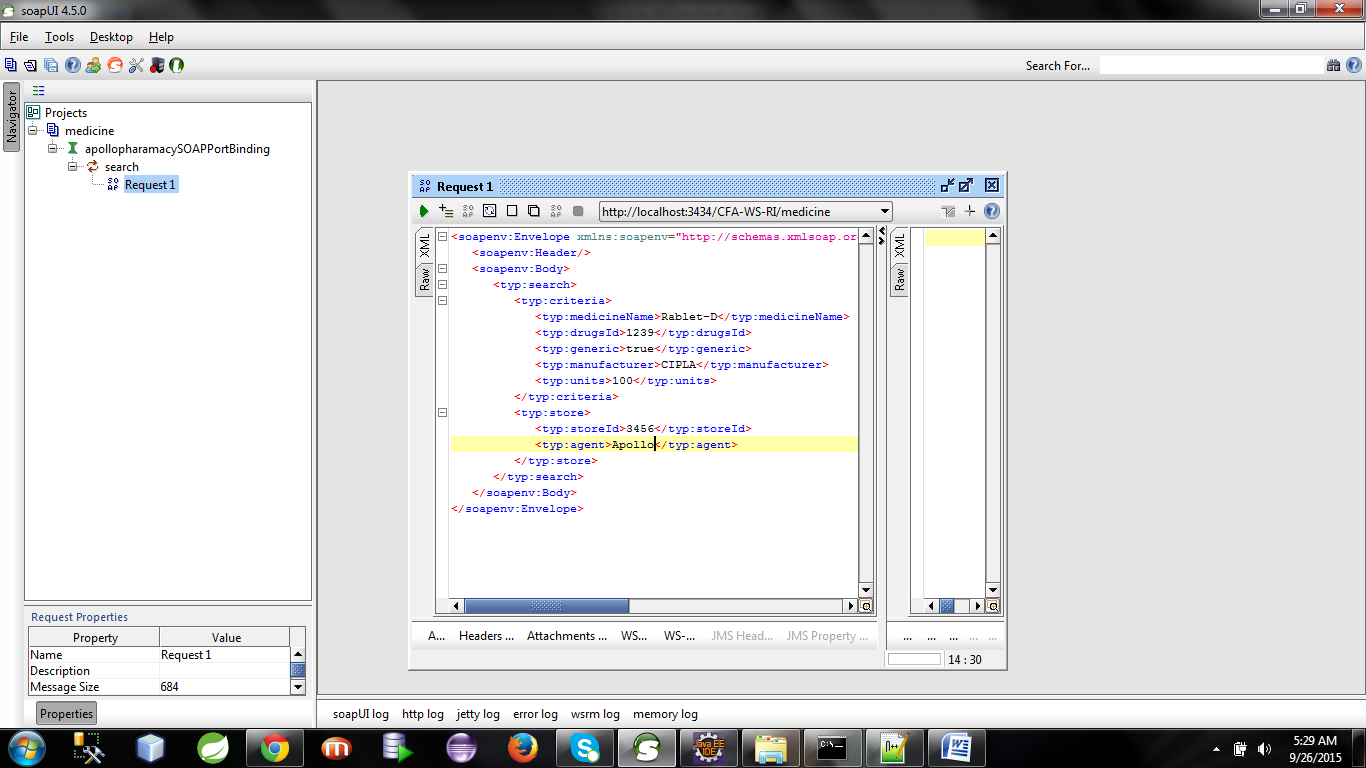
**Paste the wsdl url:**

****

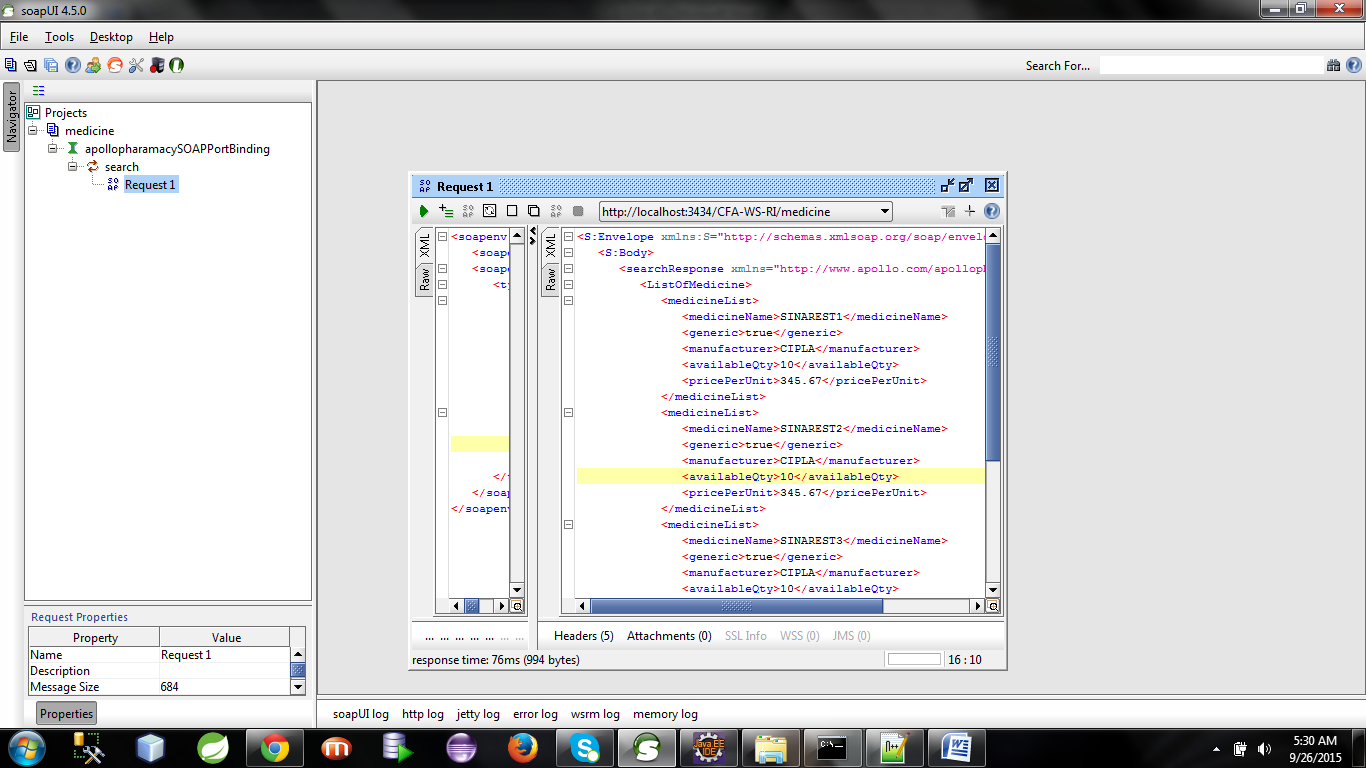
**Press Ok**

**And now test your application.**

**Request:**

****

**Press the green button to get Response**

****