Deploy JAX-WS web services on Tomcat

Here's a guide to show you how to **deploy JAX-WS web services on Tomcat** servlet container. See following summary steps of a web service deployment.

1. Create a web service (of course).
2. Create a **sun-jaxws.xml**, defines web service implementation class.
3. Create a standard **web.xml**, defines `WSServletContextListener`, `WSServlet` and structure of a web project.
4. Build tool to generate WAR file.
5. Copy JAX-WS dependencies to "${Tomcat}/lib" folder.
6. Copy WAR to "${Tomcat}/webapp" folder.
7. Start It.

Directory structure of this example, so that you know where to put your files.

2. sun-jaxws.xml

Create a web service deployment descriptor, which is also known as **JAX-WS RI deployment descriptor** – sun-jaxws.xml.

File : sun-jaxws.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<endpoints
  xmlns="http://java.sun.com/xml/ns/jax-ws/ri/runtime"
  version="2.0">
  <endpoint
      name="HelloWorld"
      implementation="com.york.ws.HelloWorldImpl"
      url-pattern="/hello"/>
</endpoints>
```

When user access **/hello/** URL path, it will fire the declared web service, which is `HelloWorldImpl.java`.

3. web.xml

Create a standard web.xml **deployment descriptor** for the deployment. Defines `WSServletContextListener` as listener class, `WSServlet` as your hello servlet.

File : web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems,
Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/j2ee/dtds/web-app_2_3.dtd">

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

```
        <servlet-name>hello</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>hello</servlet-name>
        <url-pattern>/hello</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>120</session-timeout>
    </session-config>
</web-app>
```

4. WAR Content

Use Ant, Maven or JAR command to build a WAR file to include everything inside. The WAR content should look like this

```
WEB-INF/classes/com/york/ws/HelloWorld.class
WEB-INF/classes/com/york/ws/HelloWorldImpl.class
WEB-INF/web.xml
WEB-INF/sun-jaxws.xml
```