# **Deploying a webservice to Tomcat**

**Tools:**

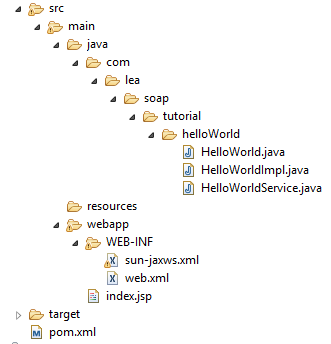
Java, jdk1.8.0\_101

Maven, apache-maven-3.3.9

Eclipse

SoapUI

**Directory Structure**



Step 1: Create Web service – annotate class with @WebService. This is the only annotation needed to identify it as a webservice

|  |
| --- |
| **package** com.lea.soap.tutorial.helloWorld;  **import** javax.jws.WebMethod;  **import** javax.jws.WebService;  /\*\*  \* Default style was Document  \* **@author** Lea  \*/  @WebService  **public** **interface** HelloWorld {    @WebMethod  **public** String greetings();  } |

|  |
| --- |
| package com.lea.soap.tutorial.helloWorld;  import javax.jws.WebMethod;  import javax.jws.WebService;  /\*\*  \* Mandatory: endpointInterface  \* @WebService is the annotation to define that the class is a webservice  \* @author Lea  \*  \*/  @WebService(endpointInterface = "com.lea.soap.tutorial.helloWorld.HelloWorld")  public class HelloWorldImpl implements HelloWorld {  private HelloWorldService helloWorldService;    public HelloWorldImpl (){  helloWorldService = new HelloWorldService();  }    @WebMethod  public String greetings() {  return helloWorldService.greetings();  }  } |

Specify value for endpointInterface

|  |
| --- |
| **package** com.lea.soap.tutorial.helloWorld;  **public** **class** HelloWorldService {    **public** String greetings(){  **return** "Hello World";  }    } |

Create new file called 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.lea.soap.tutorial.helloWorld.HelloWorldImpl"*  url-pattern=*"/helloWs"*/>  </endpoints> |

Open the web.xml and copy the following

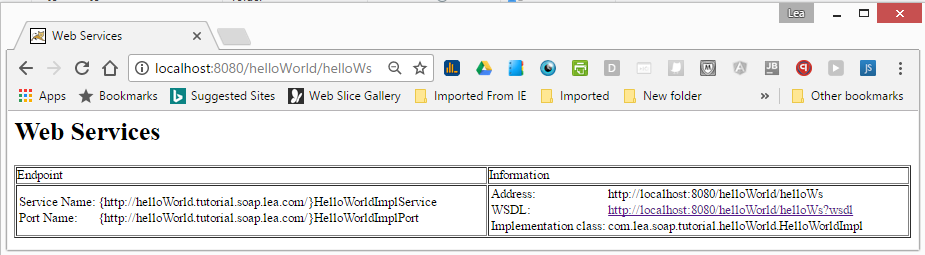
|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <!DOCTYPE web-app PUBLIC "-//Sun Microsystems,  Inc.//DTD Web Application 2.3//EN"  "http://java.sun.com/j2ee/dtds/web-app\_2\_3.dtd">  <web-app>  <listener>  <listener-class>  com.sun.xml.ws.transport.http.servlet.WSServletContextListener  </listener-class>  </listener>  <servlet>  <servlet-name>helloWs</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>helloWs</servlet-name>  <url-pattern>/\*</url-pattern>  </servlet-mapping>  <session-config>  <session-timeout>120</session-timeout>  </session-config>  </web-app> |

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.lea.soap.tutorial</groupId>  <artifactId>helloWorld</artifactId>  <packaging>war</packaging>  <version>0.0.1-SNAPSHOT</version>  <name>helloWorld Maven Webapp</name>  <url>http://maven.apache.org</url>  <properties>  <!-- Generic properties -->  <java.version>1.8</java.version>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  </properties>    <dependencies>  <!-- JAX-WS -->  <dependency>  <groupId>com.sun.xml.ws</groupId>  <artifactId>jaxws-rt</artifactId>  <version>2.2.3</version>  </dependency>  </dependencies>  <build>  <finalName>helloWorld</finalName>  </build>  </project> |

**Step 2:** Open command prompt, go to project directory and type : mvn clean install

Deploy the war file on Tomcat and type the URL: <http://localhost:8080/helloWorld/helloWs>



## Testing

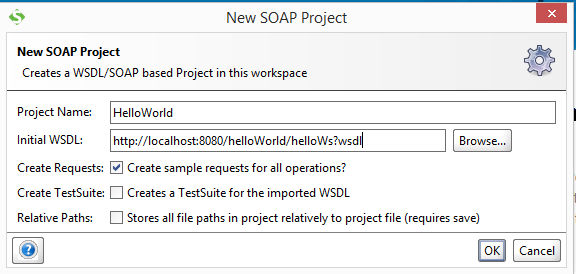
Option 1: Using java WS Client

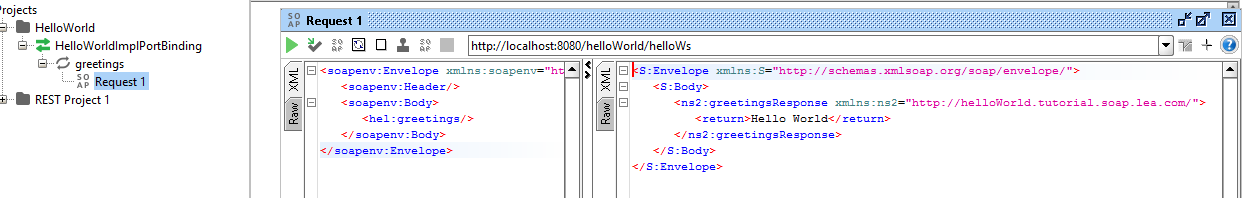
|  |
| --- |
| package com.lea.soap.tutorial.helloWorld;  import javax.jws.WebMethod;  import javax.jws.WebService;  @WebService  public interface HelloWorld {    @WebMethod  public String greetings();    } |

|  |
| --- |
| package com.lea.soap.tutorial.helloWorld;  import java.net.MalformedURLException;  import java.net.URL;  import javax.xml.namespace.QName;  import javax.xml.ws.Service;  public class HelloWorldClient {  public static void main(String[] args) throws MalformedURLException {  URL url = new URL("http://localhost:8080/helloWorld/helloWs?wsdl");    /\*\*  \* First argument is the backward naming convention of the package where we define the webservice  \* Example, the HelloWorldImpl is in package 'com.lea.soap.tutorial.helloWorld', this means  \*  \*/  QName qname = new QName("http://helloWorld.tutorial.soap.lea.com/", "HelloWorldImplService");  Service service = Service.create(url, qname);  HelloWorld hello = service.getPort(HelloWorld.class);  System.out.println(hello.greetings());  }  } |

Option 2: Open SoapUI and click New SOAP Project.

Type Project name, then enter the URL for Initial WSDL





Click the Request 1 on the left side, and it will automatically generate a request. Notice that the URL for SOAP was already specified and all we have to do is click the Run button.

That’s it! Happy Coding

Source Code:

