

Тема Web Services

Часть Знакомство с JAX-WS 2.1.

Автор OMENDBA (omendba@gmail.com)

2.01.2008

JAX-WS RI 2.1.3 – Фреймворк Веб-служб, который предоставляет инструменты и инфраструктуру для разработки решений для Веб-служб для конечных пользователей и middleware разработчиков.

Для работы данного ПО необходимо следующее ПО:

- JDK 5.0.2 или более поздний
- Ant 1.6.2 или более поздний
- Apache Tomcat 5.x

Установка на Apache Tomcat 5.5.17

1. Разархивировать файл JAXWS2.1.3-20071218.jar, запустив его из командной строки и согласиться с лицензией.

2. Установить переменные окружения:

- CATALINA_HOME=D:\PRO\apache-tomcat-5.5.17
- JAVA_HOME=D:\PRO\jdk1.6.0
- JAXWS_HOME=D:\jaxws-ri
- PATH=D:\PRO\jdk1.6.0\bin;C:\Program Files\netbeans-5.5\ide7\ant\bin
- JAVA_ENDORSED_DIRS=d:\TOMCAT_ENDORSED

3. Выполнить последовательность действий ниже:

```
C:\>ant -f %JAXWS_HOME%\tomcat.xml install
Buildfile: D:\jaxws-ri\tomcat.xml
update-catalina-props:
    [echo] Backing up D:\PRO\apache-tomcat-5.5.17/conf/catalina.properties...
    [copy] Copying 1 file to D:\PRO\apache-tomcat-5.5.17/conf
    [echo] Adding JAX-WS RI jars to shared.loader property in D:\PRO\apache-tomcat-5.5.17/conf/catalina.properties...
install:
    [echo] Installing JAX-WS RI 2.1.3 for D:\PRO\apache-tomcat-5.5.17 ...
    [mkdir] Created dir: D:\PRO\apache-tomcat-5.5.17\shared\lib
    [copy] Copying 19 files to D:\PRO\apache-tomcat-5.5.17\shared\lib
install-api-on-tomcat:
install-api-on-tomcat6:
install-api-on-tomcat5:
    [echo] Install endorsed apis - jaxws-api.jar and jaxb-api.jar on tomcat5
    [copy] Copying 2 files to D:\PRO\apache-tomcat-5.5.17\common\endorsed
jdk6-message:
    [echo] NOTE: JAX-WS RI 2.1.3 is being installed using JDK6.
    [echo] The file 'jaxws-api.jar and jaxb-api.jar' must be installed
    [echo] into $JAVA_HOME/lib/endorsed for JAX-WS RI 2.1.3 to prevent
    [echo] conflicts between the JAX-WS 2.0 classes
    [echo] bundled with JDK 6 and JAX-WS 2.1 bundled with JAX-WS RI 2.1.3 .
    [echo] You must have write-permissions on the JDK 6 directory.
    [echo] This is a concern when using a system-wide JDK installation.
    [echo] Invoke 'ant install-api' to install only
    [echo] jaxws-api.jar and jaxb-api.jar into the 'endorsed' directory.
BUILD SUCCESSFUL
Total time: 1 second
C:\>ant -f %JAXWS_HOME%\tomcat.xml install-api
Buildfile: D:\jaxws-ri\tomcat.xml
install-api:
    [copy] Copying 2 files to D:\PRO\jdk1.6.0\jre\lib\endorsed
BUILD SUCCESSFUL
Total time: 1 second
C:\> copy %JAXWS_HOME%\lib\jaxb-api.jar %JAVA_ENDORSED_DIRS%
Скопировано файлов: 1.
```

Первое приложение

Этот пример демонстрирует использование синхронного, асинхронного пула и асинхронные обратные вызовы процедуры, когда работает клиент веб-сервиса.

Пример собирает, развертывает простой веб-сервис.

- etc – конфигурационные файлы:
 - build.properties, deploy-targets.xml сценарий ant для внедрения конечной точки war-файла
 - sun-jaxws.xml дескриптор для внедрения для веб-контейнера
- src – исходные файлы
 - client/AddNumbersClient.java – приложение-клиент
 - server/AddNumberImpl.java – реализация сервера
- apt задача ant для компиляции кода сервера и генерирования JAX-WS артефактов
- wsimport задача ant для компиляции и генерирования WSDL-файлов
 - генерация SEI – AddNumbersPortType
сервис класс – AddNumbersService

Имя
c:\async\ (1)
<etc>
<src>
build.xml
c:\async\etc\ (4)
custom-client.xml
deploy-targets.xml
sun-jaxws.xml
web.xml
c:\async\src\ (0)
<async>
c:\async\src\async\ (0)
<client>
<server>
c:\async\src\async\client\ (1)
AddNumbersClient.java
c:\async\src\async\server\ (1)
AddNumberImpl.java

Внедрение сервера:

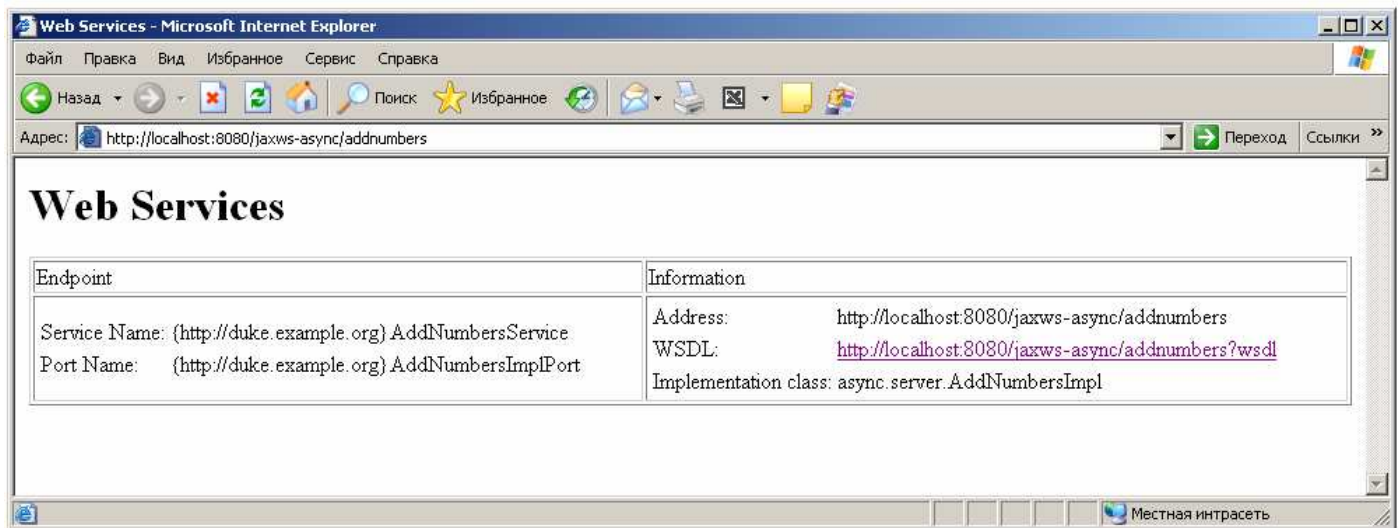
```
C:\>ant -f C:\async\build.xml server -Dtomcat=true
Buildfile: C:\async\build.xml
setup:
server:
clean:
[delete] Deleting directory C:\async\build
setup:
[mkdir] Created dir: C:\async\build
[mkdir] Created dir: C:\async\build\classes
```

```

[mkdir] Created dir: C:\async\build\war
build-server-java:
[apt] warning: Annotation types without processors: [javax.xml.bind.annotation.XmlRootElement, javax.xml.bind.annotation.XmlAccessorType, javax.xml.bind.annotation.XmlType, javax.xml.bind.annotation.XmlElement]
[apt] 1 warning
create-war:
[war] Building war: C:\async\build\war\jaxws-async.war
deploy-tomcat:
[copy] Copying 1 file to D:\PRO\apache-tomcat-5.5.17\webapps
deploy-appserver:
deploy:
BUILD SUCCESSFUL
Total time: 4 seconds

```

Для проверки функционирования сервера, зайдём по ссылке <http://localhost:8080/jaxws-async/addnumbers>.



Дескриптор имеет вид:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS
RI 2.1.3-b02-. --><!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's
version is JAX-WS RI 2.1.3-b02-. -->
<definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://duke.example.org"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    targetNamespace="http://duke.example.org"
    name="AddNumbersService">
    <types>
        <xsd:schema>
            <xsd:import namespace="http://duke.example.org"
                schemaLocation="http://localhost:8080/jaxws-async/addnumbers?xsd=1"/>
        </xsd:import>
        </xsd:schema>
    </types>
    <message name="addNumbers">
        <part name="parameters" element="tns:addNumbers"></part>
    </message>
    <message name="addNumbersResponse">
        <part name="parameters" element="tns:addNumbersResponse"></part>
    </message>
    <portType name="AddNumbersImpl">
        <operation name="addNumbers">
            <input message="tns:addNumbers"></input>
            <output message="tns:addNumbersResponse"></output>
        </operation>
    </portType>
    <binding name="AddNumbersImplPortBinding" type="tns:AddNumbersImpl">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
            style="document"></soap:binding>
        <operation name="addNumbers">

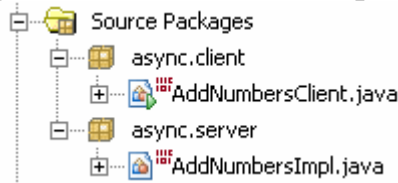
```

```

        <soap:operation soapAction=""></soap:operation>
        <input>
            <soap:body use="literal"></soap:body>
        </input>
        <output>
            <soap:body use="literal"></soap:body>
        </output>
    </operation>
</binding>
<service name="AddNumbersService">
    <port name="AddNumbersImplPort" binding="tns:AddNumbersImplPortBinding">
        <soap:address location="http://localhost:8080/jaxws-async/addnumbers"/>
    </port>
</service>
</definitions>

```

Теперь проверим работу клиента. Создадим проект в NetBeans 5.5.



```

package async.server;
import javax.jws.WebService;
@WebService(serviceName = "AddNumbersService",targetNamespace = "http://duke.example.org")
public class AddNumbersImpl {
    public int addNumbers(int number1, int number2) {
        System.out.println("Received Request!");
        return number1+number2;
    }
}

```

```

package async.client;
import javax.xml.ws.AsyncHandler;
import javax.xml.ws.Response;
import java.rmi.RemoteException;
import java.util.concurrent.ExecutionException;
import java.util.concurrent.Future;
public class AddNumbersClient {
    private AddNumbersImpl port;
    public AddNumbersClient() {
        port = new AddNumbersService().getAddNumbersImplPort();
    }
    public static void main(String[] args) {
        try {
            AddNumbersClient client = new AddNumbersClient();
            //invoke synchronous method
            client.invokeSynchronous();
            //invoke async polling method
            client.invokeAsyncPoll();
            //invoke async callback method
            client.invokeAsyncCallback();
        } catch (RemoteException e) {
            e.printStackTrace();
        } catch (InterruptedException e) {
            e.printStackTrace();
        } catch (ExecutionException e) {
            e.printStackTrace();
        }
    }
    private void invokeSynchronous() throws RemoteException {
        int number1 = 10;
        int number2 = 20;
        System.out.println("\nInvoking synchronous addNumber():");
        int result = port.addNumbers(number1, number2);
        System.out.printf("The result of adding %d and %d is %d.\n",
            number1, number2, result);
    }
    private void invokeAsyncPoll() throws InterruptedException, ExecutionException {
        int number1 = 10;
        int number2 = 20;
        System.out.println("\nInvoking Asynchronous Polling addNumbersAsync():");
        Response<AddNumbersResponse> resp = port.addNumbersAsync(number1, number2);
    }
}

```

```

        Thread.sleep(2000);
        AddNumbersResponse output = resp.get();
        System.out.printf("The result of adding %d and %d is %d.\n",
            number1, number2, output.getReturn());
    }
    private void invokeAsyncCallback() throws InterruptedException {
        int number1 = 10;
        int number2 = 20;
        System.out.println("\nInvoking Asynchronous Callback addNumbersAsync():");
        AddNumbersCallbackHandler callbackHandler = new AddNumbersCallbackHandler();
        Future<?> response = port.addNumbersAsync(number1, number2, callbackHandler);
        Thread.sleep(2000);
        AddNumbersResponse output = callbackHandler.getResponse();
        System.out.printf("The result of adding %d and %d is %d.\n",
            number1, number2, output.getReturn());
    }
    private class AddNumbersCallbackHandler implements AsyncHandler<AddNumbersResponse> {
        private AddNumbersResponse output;
        public void handleResponse(Response<AddNumbersResponse> response) {
            try {
                output = response.get();
            } catch (ExecutionException e) {
                e.printStackTrace();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        AddNumbersResponse getResponse() {
            return output;
        }
    }
}

```

init:

deps-jar:

Compiling 1 source file to D:\code\JavaApplication1\build\classes

compile-single:

run-single:

Invoking synchronous addNumber():

The result of adding 10 and 20 is 30.

Invoking Asynchronous Polling addNumbersAsync():

The result of adding 10 and 20 is 30.

Invoking Asynchronous Callback addNumbersAsync():

The result of adding 10 and 20 is 30.

BUILD SUCCESSFUL (total time: 7 seconds)

