涛's profile   Johnnyのスペース   Photos   Blog   Lists   More ▾

⚙ Tools ▾   Help

Blog

< Previous   Next >

📄 Entries

📊 Summary

Listed by:

Date

February 2010

January 2010

December 2009

November 2009

October 2009

September 2009

August 2009

July 2009

June 2009

May 2009

April 2009

March 2009

February 2009

January 2009

December 2008

November 2008

October 2008

September 2008

August 2008

August 15

NetBean构建轻量级Java WebService应用

一直以来，WebService的创建和调用都是比较棘手的问题，尽管有很多用法，包括使用Axis，在Eclipse环境中创建和调用。但是繁琐的配置文件的生成，以及依赖于Axis和一些特定jar包文件下载和引用，无疑增加了构建应用的复杂度。NetBean很好的解决了这一问题，在NetBean6.0.1下创建WebService服务，通过控制台程序和web框架两种方式实现调用，显得简单而且实用。

NetBean下构建轻量级WebService应用,在环境上的支持,支持JDK 1.6,jar包文件集成JAX-WS 2.1,但同时需要加载引入另外一个支持2.1的API Jar包文件,即jaxb-api-2.1.jar,前置准备工作需要做如下操作:

1, 确认D:\JAVA\server\Apache Tomcat 6.0.14\endorsed的存在,该文件夹对应自己机器Apache 服务器的安装路径.没有则创建.

2,Copy D:\JAVA\IDE\NetBeans 6.0.1\java1\modules\ext\jaxws21\api 下所有文件到1对应的路径中,本路径对应在NetBean的集成开发环境路径.

3,将jaxb-api-2.1.jar copy到两个路径下覆盖,并改名为jaxb-api.jar.[在NetBean未启动下进行

环境准备结束,开始构建轻量级webservice应用.

A,同传统的web应用构建一样,新建web应用程序.命名为WebServiceTest.

B,右键单击项目,选择新建"web服务",创建webservice如下.

一个方法sayHello,参数name,折到源码部分如下:

```
package HelloWebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
/**
*
* @author Johnny
*/
@WebService()
public class HelloWebService {

    @WebMethod(operationName = "sayHello")    public String sayHello(@WebParam(name = "name")
    String name) {
        //TODO 在此处编写实现代码:
        return "hello " + name;
    }
}
```

NetBean自动在web.xml中加入如下:

```
<listener>
```

```xml
    <listener-class>com.sun.xml.ws.transport.http.servlet.WSServletContextListener</listener-class>
  </listener>
  <servlet>
    <servlet-name>HelloWebService</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>HelloWebService</servlet-name>
    <url-pattern>/HelloWebService</url-pattern>
  </servlet-mapping>
```

同时自动生成了sun-jaxws.xml.

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

至此可直接在Apache服务器上运行该webservice,服务构建成功.

现在实现在客户端通过程序和web页面两种方式去实现调用.

1,使用程序,新建控制台程序,命名testwebserviceApp

在该项目上右键点击,并选择添加web服务客户端.

程序中源包中main函数代码实现部分:

```java
public static void main(String[] args) {
    try
    {
        org.me.testwebservice.HelloWebServiceService service =
            new org.me.testwebservice.HelloWebServiceService();
        org.me.testwebservice.HelloWebService port =
            service.getHelloWebServicePort();
        System.out.println("====webservice recall start=======");
        String result = port.sayHello("Johnny");
        System.out.println(result);
        System.out.println("=====webservice recall End========");
    }
    catch(Exception ex)
    {
        ex.printStackTrace();
    }
}
```

至此,客户端程序直接在控制台中运行即可

2,在web程序中使用方式一样,具体加入webservice客户端服务同.

web页面调用的action中核心代码如下:

```
org.me.webserviceclient.HelloWebServiceService

                        ss = new org.me.webserviceclient.HelloWebServiceService();
        org.me.webserviceclient.HelloWebService port =
                ss.getHelloWebServicePort();
        String result = port.sayHello("Johnny");
        request.setAttribute("infor", "webservicetest: " + result);
```

JSP界面上调用,EL表达式完成参数引用.

```html
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h2>Infomation</h2>
        ${sessionScope.user}
        <br/>
        <font color="red" size="3">${infor}</font>
    </body>
</html>
```

3:24 PM | Blog it | Java

Comments

Trackbacks

The trackback URL for this entry is:
http://zhangtao8101.spaces.live.com/blog/cns!35DD684BD2C7CEC5!1011.trak
Weblogs that reference this entry

- None