

Übung „Netzbasierte Informationssysteme“ WS 2008/2009

Übungsblatt 4

Ausgabe am 18.11.2008

Abgabe bis spätestens 2.12.2008, 16.00 Uhr

Hinweise:

Bezüglich der Source-Code-Organisation in Eclipse gelten folgende Konventionen: statische Ressourcen wie z.B. Simple Java Web Services (.jws) werden in /web angelegt. Zu kompilierende Sourcen wie beispielsweise WebServices (.java) werden ganz normal in /src angelegt. Deployment Deskriptoren werden in einem Ordner /deploy verwaltet. Textuelle Lösungen zu den Aufgaben werden ganz normal in /docs abgelegt.

Wie gewohnt sind sämtliche bei der Bearbeitung dieses Übungsblatts von Ihnen generierten oder geänderten Dateien per eMail abzugeben!

Das Deployment Ihrer Web Services unter Ant ist nach Anpassung des bisherigen *Build Scripts*, insbesondere nach der Integration von Axis Ant Tasks (<axis-admin>) möglich. Sie finden ein neues Basisskript auf der Vorlesungswebseite. Ant-Unterstützung für Axis finden Sie unter: <http://ws.apache.org/axis/java/ant/ant.html>.

Axis Information:

1. Installieren sie Axis in ihrem Tomcat: <http://ws.apache.org/axis/>
2. Informieren Sie sich über die Apache SOAP Engine *Axis*. Lesen Sie zu diesem Zweck zumindest die *Documentation for Axis Users* unter <http://ws.apache.org/axis/java/>.
3. Validieren Sie ihre Axis Engine:

- Öffnen Sie die Startseite Ihrer Axis Engine <http://<Ihre Tomcat>/axis>
- Validieren Sie Axis (Link *Validate*). Stellen Sie sicher, dass keine Bibliotheken fehlen und Axis fehlerfrei läuft.
- Lassen Sie sich alle bereits eingestellten Webservices anzeigen (Link *View*)
- Testen Sie den lokalen Endpunkt über den *EchoHeader.jws* Webservice (Link *Call local endpoint* oder <http://<Ihre IP>:<Ihr Port>/axis/EchoHeaders.jws?method=list>)

Testen Sie den SOAP Endpunkt. Starten Sie hierfür den *Version* Webservice unter

<http://<Tomcat>/axis/services/Version?method=getVersion>

Als Ergebnis sollten Sie ein XML-Dokument (SOAP) mit Angaben über die Axis-Version erhalten.

Aufgabe 13: JWS (3 Punkte)

1. Schreiben Sie ein *HelloWorld* Webservice mit einer Methode *sayHello()*, die einen beliebigen Namen *Name* als Input-Parameter (*firstName*) annimmt und *Hello Name* zurückgibt. Speichern Sie das Programm im Axis Context Root Ihrer Tomcat-Instanz unter als *HelloWorld.jws*. Testen Sie Ihren Webservice. Initiieren Sie über Ihren Webbrowser folgenden Request: <http://<tomcat>/axis/HelloWorld.jws?method=sayHello&firstName=Adrian>
2. Schreiben Sie einen Java-SOAP-Client *SayHelloClient*, der Ihren *HelloWorld* Service nutzt und die Antwort des Services über *stdout* ausgibt.

Aufgabe 14: WSDD (7 Punkte)

1. Schreiben sie nun anstatt des simple Web-Services *HelloWorld* aus Aufgabe 13 einen funktional entsprechenden Webservice (*HelloWorld2*). Beachten Sie hierbei, dass Sie nicht auf das automatische Hot-Deployment (JWS) von Axis zurückgreifen, sondern den Web Service als kompilierte Klasse über den zu erstellenden Web Service Deployment Descriptor *deployHelloWorld2.wsdd* mit Hilfe des Axis-AdminClient der Apache Software Foundation© deployen müssen.

```
java org.apache.axis.client.AdminClient -l \
http://<Tomcat>/axis/servlet/AxisServlet <wsdd-file>
```

Hinweis: Alternativ können Sie für das Deployment auch ein <axis-admin> task verwenden.

2. Erstellen Sie einen Client *SayHelloClient2*, der den *HelloService HelloWorld2* nutzt.

3. Schreiben Sie ein *Undeployment-Descriptor* `undeployHelloWorld2.wsdd` und undeploieren sie den *HelloWorld2* Service.
4. Welche *Scopes* eines Service können im WSDD angegeben werden und was bewirken sie (Antwort bitte in `scopes.txt`)?

Aufgabe 15: Handler und Chains (5 Punkte)

- Welche drei Ketten beim Nachrichtenaustausch zwischen Client und Server werden in Axis unterschieden?
- Wie ist der Ablauf?
- Was ist ein `MessageContext`?
- Was sind Handler und welche Rolle spielen sie?

(Antwort in `chains_and_handlers.txt`)

Schreiben Sie einen Handler, der alle an den HelloWorld-Service übergebenen Namen und das aktuelle Datum sowie die aktuelle Uhrzeit in einer Datei mitschreibt. Implementieren sie dazu zuerst die Handler-Klasse *HelloWorldLogger* und fügen sie diese dann in den WSDD des *HelloWorld*-Services ein.

Aufgabe 16: GlobalWeatherClient (15 Punkte)

In dieser Aufgabe sollen Sie auf einen bereits deployten Webservice „GlobalWeather“ zugreifen und die Ergebnisse präsentieren. Rufen sie dazu zuerst die bereitgestellte `.wsdl` – Datei des Ziel-Web-Services ab: <http://www.webservices.net/globalweather.asmx?WSDL>. Nutzen Sie dann das Tool `WSDL2Java` und lassen Sie sich die nötigen Klassen für einen Client automatisch generieren. Implementieren Sie dann einen Client `GlobalWeatherClient.java` der die Methode `getWeather` des Webservices aufruft. Die Syntax des Aufrufs können Sie sich aus den erzeugten Klassen ableiten. Testen Sie Ihren Client – wie ist das Wetter gerade in Berlin? Lassen Sie sich die Rückgabe des Clients auf der Console ausgeben. Speichern Sie zusätzlich das Ergebnis des Aufrufs in einem Textfile `BerlinWeather.txt` in `/docs`. Erleichtern Sie Nutzern nun den Aufruf des Services – implementieren und deployen Sie ein Servlet `GlobalWeatherWizard`, das ein Formular enthält in dem man die von `getWeather` benötigten Parameter eingeben kann. Bei Klick auf einen Button bekommt man das Ergebnis unterhalb des Formulars präsentiert. Der Übersicht halber lagern Sie dabei den eigentlichen Webserviceaufruf innerhalb Ihrer Servletklasse in eine Methode `callTheGuy` aus. Beachten Sie zusätzlich: die Rückgabe des Webservices ist in einem XML-Format. Die dargestellte Information soll keine Tags enthalten sondern lediglich die enthaltene Information übersichtlich darstellen. Nutzen Sie dazu Ihre bereits erworbenen Kenntnisse zum Parsen eines XML-Files! Implementieren Sie eine Methode `parseResult` die die Rückgabe von `callTheGuy` als Parameter entgegen nimmt und die Information in einem String zurückgibt der auch gleich die nötigen HTML-Tags für eine übersichtliche Darstellung enthält so dass der Stringinhalt nur noch in die `response` des Servlets eingepasst werden muss.

Hinweis:

Informationen zur Funktionsweise der Axis-Tools *Java2WSDL* und *WSDL2Java* finden Sie unter <http://ws.apache.org/axis/java/user-guide.html> sowie unter <http://ws.apache.org/axis/java/reference.html>.

Anleitungen zur Verwendung der Tools unter Ant finden Sie unter <http://ws.apache.org/axis/java/ant/axis-wsdl2java.html> und <http://ws.apache.org/axis/java/ant/axis-java2wsdl.html>

Die Eclipse Web Tools Platform bietet Tools für Web services development.

Kontextrelevante Online-Informationen finden sie z.B. unter:

W3C Web Services	http://www.w3.org/2002/ws/
Axis	http://ws.apache.org/axis/
Axis Dokumentation	http://ws.apache.org/axis/java/
SOAP	http://www.w3.org/TR/SOAP/
WSDL	http://www.w3.org/TR/wsdl
Java Web Services Tutorial	http://java.sun.com/webservices/docs/1.3/tutorial/doc/index.html
Axis User's Guide	http://ws.apache.org/axis/java/user-guide.html
Web Services mit Java	ISBN 3-8272-6447-2

Viel Erfolg!