



Universität Freiburg
Institut für Informatik
Prof. Dr. G. Lausen
Alexander Schätzle
Martin Przyjaciel-Zablocki

Georges-Köhler Allee, Geb. 51
D-79110 Freiburg
lausen@informatik.uni-freiburg.de
schaetzl@informatik.uni-freiburg.de
zablocki@informatik.uni-freiburg.de

Advanced Information Systems
Summerterm 2011
05.05.2011

1. Exercise Sheet: Web Services

Submission: 23.05.2011
Discussion: 26.05.2011

Submission Guidelines: If you want your solutions to be reviewed you have to send them by email until 23.05.2011. Solutions to exercise 1, 2 and 3 should be submitted as PDF and solutions to exercise 4 and 5 as separate runnable Java programs (including the corresponding source code). Source code without appropriate comments will not be reviewed!

Exercise 1 (Web Service architecture)

A general Web Service architecture consists of the following basic components:

- Communication (SOAP/XML or REST)
- Service Description (WSDL)
- Directory Service (UDDI)

WSDL (Web Service Description Language) is an XML-based language to describe Web Services such that clients know how to use the service. SOAP or REST is used to communicate with the Web Service and UDDI is a directory that helps to find available Web Services on demand. We will not use UDDI in this exercise as the services we use are already known.

Introduce yourself to Web Services, WSDL, REST and SOAP by using the given literature, tutorials from the W3C (<http://www.w3schools.com/>) or other materials you prefer.

Explain the basic components of a Web Service architecture in your own words and argue about possible advantages and drawbacks of this general architecture style.

Exercise 2 (REST)

Consider the WSDL declaration for a global weather Web Service:

`http://www.webservicex.net/globalweather.asmx?wsdl`

- a) Compose a REST (HTTP-GET) request that determines all German cities provided by the weather service. You can check your solution with any internet browser.
- b) Use the response to the previous request to determine the weather of a German city of your choice.
- c) What are the advantages and drawbacks of REST?

Exercise 3 (SOAP)

Consider the WSDL declaration for a FIFA WM 2010 Web Service:

<http://footballpool.dataaccess.eu/data/info.wso?wsdl>

- Which operations of the Web Service do not require any parameters?
- Describe the parameters of the operation *'PlayedAtWorldCup'*?
- Describe the return type of the operation *'Coaches'*.
- Complete the following SOAP request to determine all defenders of the German team. You can use *TCPmon* (<http://ws.apache.org/commons/tcpmon/>) to execute your request.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tns="http://footballpool.dataaccess.eu" >
  <SOAP-ENV:Body>
    ...
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- What are the advantages and drawbacks of SOAP?

Exercise 4 (Simple SOAP Web Service)

In this exercise you will implement your own simple SOAP Web Service and Client with Java.

We use the *Java API for XML - Web Services (JAX-WS)* which is included in *Java SDK 6* as it provides a fast way of getting started. We do not use more sophisticated Web Service frameworks like *Apache Axis* or *Java Glassfish (Metro)* as they would cause a greater installation effort.

A good introduction on how to implement Web Services and corresponding clients with JAX-WS can be found at (1) <http://www.theserverside.de/webservice-in-java/>.

- Analogous to the calculator example in (1) implement a simple SOAP Web Service that satisfies the following extract of a WSDL declaration.

```
...
<message name="randomNumber">
  <part name="lowerBound" type="xsd:int"/>
  <part name="upperBound" type="xsd:int"/>
</message>
<message name="randomNumberResponse">
  <part name="return" type="xsd:int"/>
</message>
<portType name="RandomNumberGenerator">
  <operation name="randomNumber" parameterOrder="lowerBound upperBound">
    <documentation>
      Returns a random number between given the lower and upper bound.
    </documentation>
    <input message="tns:randomNumber"/>
    <output message="tns:randomNumberResponse"/>
  </operation>
</portType>
...
```

- Implement a corresponding client that uses your Web Service to simulate the lottery drawing by requesting six random numbers between 1 and 49.

Exercise 5 (Advanced SOAP Web Service)

In this exercise you will combine two existing Web Services to build an advanced Web Service. For this purpose we use the WSDL declaration to automatically generate client stubs for accessing the corresponding Web Service.

- Use the *wsimport* tool of JAX-WS to automatically generate client stubs for the global weather Web Service and the FIFA WM 2010 Web Service of exercise 2 and 3.

- b) Implement a Web Service that provides the following two operations:
- *cardsOfTeam*: Return for all players of a given team the number of yellow and red cards.
 - *weatherOfWMCities*: Return the current weather for the cities of the FIFA WM 2010. (*Hint*: First request all WM cities using the WM service. Unfortunately, the weather service will not find all of these cities.)

Literature:

1. Ingo Melzer: *Service-orientierte Architekturen mit Web Services*, 4. Auflage, 2010. ISBN 978-3-8274-2549-2.
2. Patrick Finger, Klaus Zeppenfeld: *SOA und WebServices*, 2009. ISBN 978-3-540-76990-3.

Available as electronic version from the UB (<http://www.ub.uni-freiburg.de/index.php?id=opac>)