

02267: Software Development of Web Services

Purpose of the exercise

- The purpose of this exercise is to learn to precisely control the contents of SOAP messages through WSDL and XML Schema.
- Please read all the exercise text before you start implementing.

White Pages Service

- Write a White Pages Web service that adds a person to a phone book (called addPerson) and then finds persons in that phone book (called findPerson) given a partly filled person object.
- For example, if in the query only the last name of the person is known, then the Web service should return a list of person objects having the same last name and which have all of the information filled (e.g. first name, phone number, address, e.t.c.).
- The exercise uses the same development process as with last weeks exercise: first, a WSDL is created from which the Web service implementation stubs are generated. The difference is, that now the Web service uses user defined datatypes, defined as XML schema in the WSDL document, and that the WSDL document completely defines the body of the SOAP message using document style.

4.1 White Pages Service: AddPerson Operation

- The addPerson operation takes as argument a person and returns a status element containing the string "done" when adding the person was successful
- If there is exists already a person with the same first- and last name in the phone book, a fault message should be returned

An example of a successful message exchange is: Request

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:person xmlns:ns2="urn:ws.imm.dtu.whitepages"
      phone="342" lastName="Rasmussen"
      firstName="Anita">
      <address>
        <street>Kastrup vej</street>
        <postcode>1234</postcode>
        <city>København</city>
      </address>
    </ns2:person>
  </S:Body>
</S:Envelope>

```

If everything went okay, the response should be

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:status xmlns:ns2="urn:ws.imm.dtu.whitepages">done</ns2:status>
  </S:Body>
</S:Envelope>

```

In case there exists a person with the same first- and last name, a SOAP fault should be returned. The SOAP fault should contain an error message and the person object in the phone book in the **detail** element. Note that it is okay if the detail element also contains the stack trace.

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <S:Fault xmlns:ns4="http://www.w3.org/2003/05/soap-envelope">
      <faultcode>S:Server</faultcode>
      <faultstring>error</faultstring>
      <detail>
        <ns2:wpfault xmlns:ns2="urn:ws.imm.dtu.whitepages">
          <errorMessage>Person already exists</errorMessage>
          <ns2:person firstName="Anita" lastName="Rasmussen" phone="342">
            <address>
              <street>Kastrup vej</street>
              <postcode>1234</postcode>
              <city>København</city>
            </address>
          </ns2:person>
        </ns2:wpfault>
      </detail>
    </S:Fault>
  </S:Body>
</S:Envelope>

```

Tasks:

- Define an XML Schema (i.e. `<xsd:schema>...</xsd:schema>`) representing a person as XML, where a person has a lastname, phone and address. Note that the XML Schema can go into its own file (e.g. `wp.xsd`), but can also be included into the type section of the WSDL document. The IDE has some bugs dealing with importing XML schema (see the course home page for more details), therefore I recommend for this exercise to inline all schemata in the WSDL document. After you have a working version of your service, you can play around with putting the schema definitions in a file of their own, which is recommended for domain datatypes (WSDL specific datatypes, like definition of messages using Document-style bindings, still go into the type section of the WSDL document).
 - Use the WSDL editor for this and put the schema as an inlined schema in the type section of the WSDL document
 - An address should have a street, postcode and city.
 - Represent firstname, lastname, phone by XML attributes whereas address, street, postcode and city should be represented by elements.
- Define the XML Schema for WPFault which contains an errorMessage and a Person as the result (the person that is already in the phone book)
- Write the remaining XML schemata and the WSDL file, so that it describes the message exchange above
 - What binding type should be used: document or rpc?
- Implement the Web services and deploy it
 - Note that the logic does not need to be correctly implemented, but it should be possible to test both outcomes of the add person operation
- Implement a client using JUnit that tests the Web service
- Check, using the TCP monitor, that the SOAP messages have the required form

4.2 White Pages Service: FindPerson Operation

The findPerson operation returns for a partially filled person object a list of person objects matching the argument to the findPerson operation. For example, the request below finds two entries in the phone book.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:personquery xmlns:ns2="urn:ws.imm.dtu.whitepages" firstName="Anita"/>
  </S:Body>
</S:Envelope>
```

Response:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:personArray xmlns:ns2="urn:ws.imm.dtu.whitepages">
      <ns2:person phone="23545" lastName="Nielson" firstName="Anita">
        <address>
          <street>Strandvejen</street>
          <postcode>1234</postcode>
          <city>Hvidover</city>
        </address>
      </ns2:person>
      <ns2:person phone="342" lastName="Rasmussen" firstName="Anita">
        <address>
          <street>Kastrup vej</street>
          <postcode>1234</postcode>
          <city>København</city>
        </address>
      </ns2:person>
    </ns2:personArray>
  </S:Body>
</S:Envelope>

```

The following request, however, finds no matches.

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:personquery xmlns:ns2="urn:ws.imm.dtu.whitepages" lastName="Hubert"/>
  </S:Body>
</S:Envelope>

```

Response

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:personArray xmlns:ns2="urn:ws.imm.dtu.whitepages"/>
  </S:Body>
</S:Envelope>

```

Another possible request is to ask for all students living in one city

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:personquery xmlns:ns2="urn:ws.imm.dtu.whitepages">
      <address>
        <city>København</city>
      </address>
    </ns2:personquery>
  </S:Body>
</S:Envelope>

```

Tasks

- Add the necessary XML schema declarations to the type section of the WSDL and add the operation `findPerson` as part of the `whitePagesPortType` to the WSDL file *from the previous exercise*, so that it describes Web services that follow the above message exchange
- Implement a Web service based on that WSDL file
 - Again the correctness of the implementation logic is not important. However, it should be possible to see the various outcomes (one person found, several persons found, and no person found)
- Implement a client using JUnit that test the Web services
- Check, using the TCP monitor, that the SOAP messages have the required form