

CS544

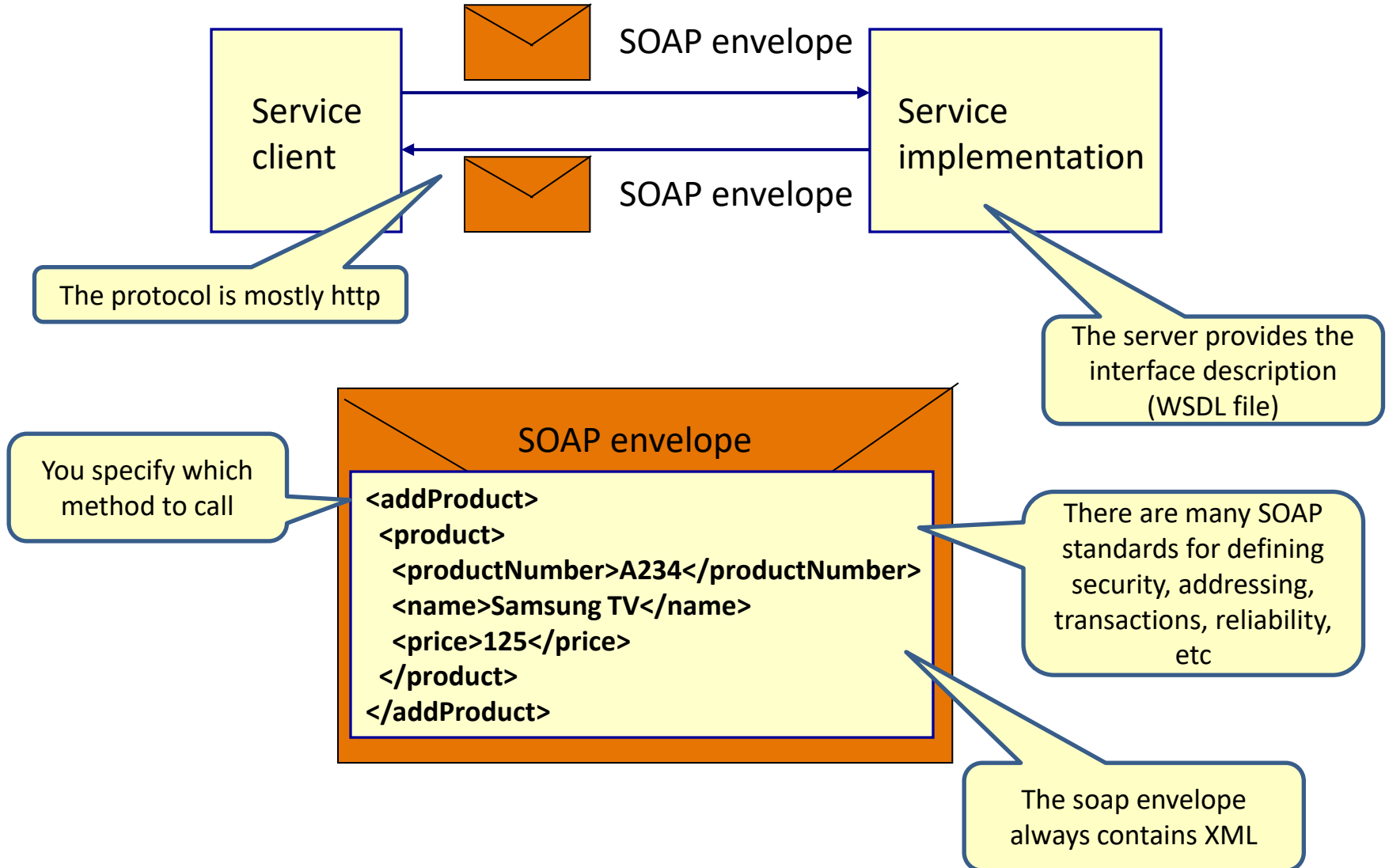
LESSON 10

SOAP WEBSERVICES

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
November 28 Lesson 1 Introduction Spring framework Dependency injection	November 29 Lesson 2 Spring Boot AOP	November 30 Lesson 3 JDBC JPA	December 1 Lesson 4 JPA mapping 1	December 2 Lesson 5 JPA mapping 2	December 3 Lesson 6 JPA queries	December 4
December 5 Lesson 7 Transactions	December 6 Lesson 8 MongoDB	December 7 Midterm Review	December 8 Midterm exam	December 9 Lesson 9 REST webservices	<div> December 10 Lesson 10 SOAP webservices </div>	December 11
December 12 Lesson 11 Messaging	December 13 Lesson 12 Scheduling Events Configuration	December 14 Lesson 13 Monitoring	December 15 Lesson 14 Testing your application	December 16 Final review	December 17 Final exam	December 18
December 19 Project	December 20 Project	December 21 Project	December 22 Presentations			

SOAP WEBSERVICES

Simple Object Access Protocol (SOAP)



Soap message

Envelope (required)

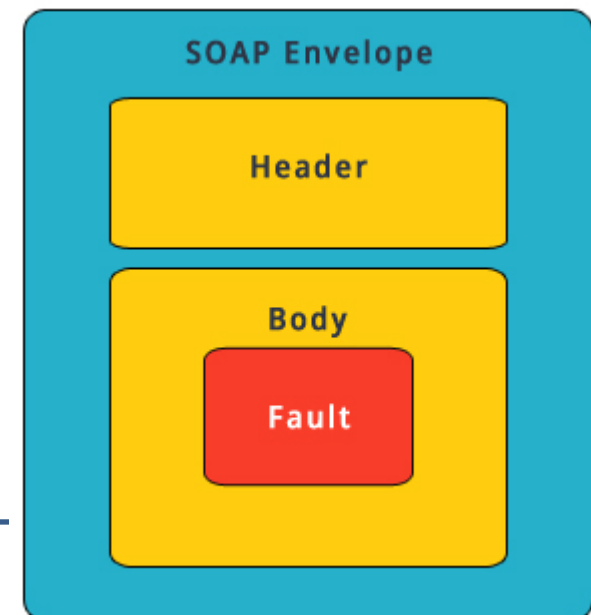
```
<?xml version = "1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV = "http://www.w3.org/2001/12/soap-envelope"
  SOAP-ENV:encodingStyle = "http://www.w3.org/2001/12/soap-encoding">

  <SOAP-ENV:Header>
    ...
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    ...
    <SOAP-ENV:Fault>
      ...
    </SOAP-ENV:Fault>
    ...
  </SOAP-ENV:Body>
</SOAP_ENV:Envelope>
```

Header (optional)

Body (required)

Fault (optional)



SAOP request and response

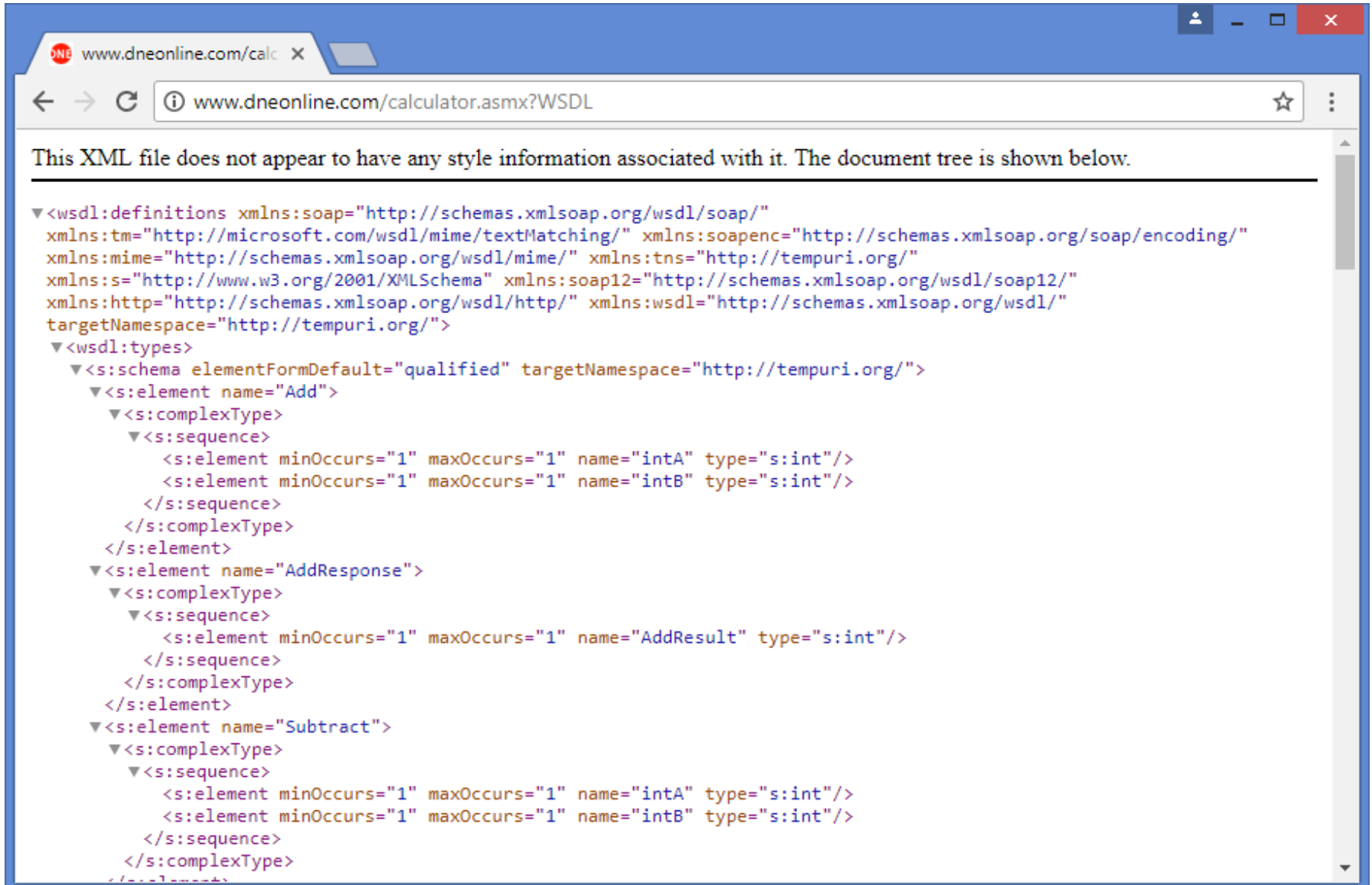
Request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:tem="http://tempuri.org/">
  <soap:Header/>
  <soap:Body>
    <tem:Add>
      <tem:intA>3</tem:intA>
      <tem:intB>6</tem:intB>
    </tem:Add>
  </soap:Body>
</soap:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <AddResponse xmlns="http://tempuri.org/">
      <AddResult>9</AddResult>
    </AddResponse>
  </soap:Body>
</soap:Envelope>
```

WSDL



www.dneonline.com/calculator.asmx?WSDL

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://tempuri.org/"
  xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:wsd1="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://tempuri.org/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
      <s:element name="Add">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="intA" type="s:int"/>
            <s:element minOccurs="1" maxOccurs="1" name="intB" type="s:int"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="AddResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="AddResult" type="s:int"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="Subtract">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="intA" type="s:int"/>
            <s:element minOccurs="1" maxOccurs="1" name="intB" type="s:int"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
  <wsdl:binding name="Calculator" type="tns:Calculator">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  </wsdl:binding>
  <wsdl:service name="Calculator" baseURI="http://tempuri.org/Calculator">
    <wsdl:port name="Calculator" binding="tns:Calculator" location="http://tempuri.org/Calculator" type="tns:Calculator">
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```


SOAP vs. REST

■ SOAP

- SOAP envelope
- Standards for
 - Security
 - Transactions
 - Reliability
 - Addressing
 - Etc
- More complex as REST
- XML messages
- Standard way to describe the interface
 - WSDL

■ REST

- No support for
 - Security
 - Transactions
 - Reliability
 - Addressing
 - Etc
- Simple
- Http messages containing text, XML or JSON
- No standard way to describe the interface

SPRING SOAP WEBSERVICES

Spring WS

- In a contract first approach we need to:
 1. Create a xsd (XML schema) for our web service
 2. Generate Java classes based on the xsd
 3. Create implementation based on the generated Java classes

Spring SOAP libraries

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-web-services</artifactId>  
</dependency>
```

Start with xsd: Greeting.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified"
  elementFormDefault="qualified"
  targetNamespace="http://springtraining/greeting"
  xmlns:tns="http://springtraining/greeting"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:complexType name="Person">
    <xs:sequence>
      <xs:element type="xs:string" name="FirstName" />
      <xs:element type="xs:string" name="LastName" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="GreetingRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Person" type="tns:Person" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="GreetingResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:string" name="Greeting" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Namespaces

Start with xsd: Greeting.xsd

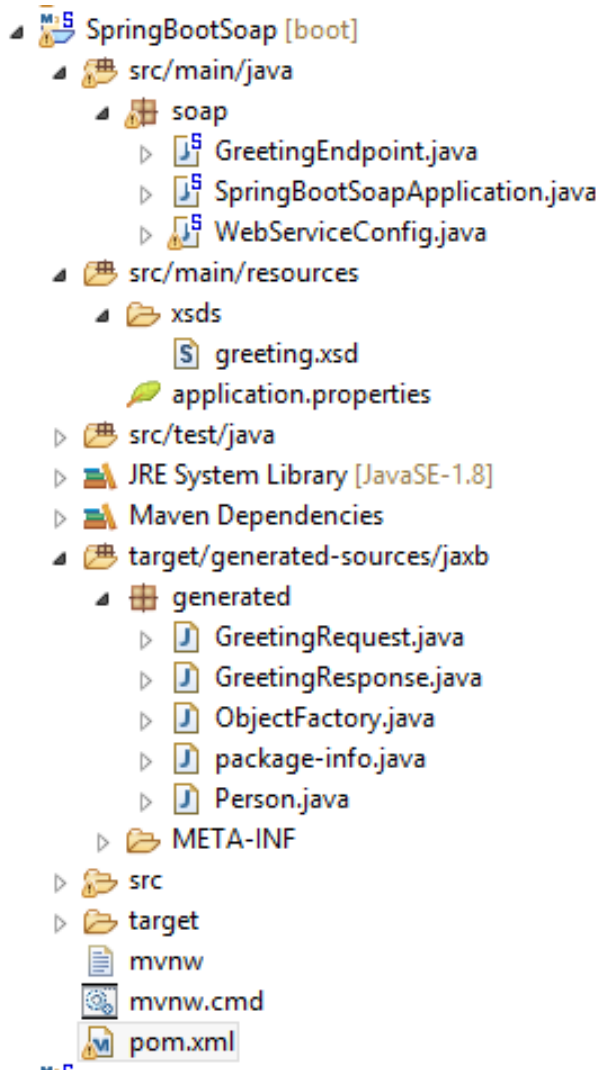
```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified"
  elementFormDefault="qualified"
  targetNamespace="http://springtraining/greeting"
  xmlns:tns="http://springtraining/greeting"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:complexType name="Person">
    <xs:sequence>
      <xs:element type="xs:string" name="FirstName" />
      <xs:element type="xs:string" name="LastName" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="GreetingRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Person" type="tns:Person" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="GreetingResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:string" name="Greeting" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

In POM file



```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>jaxb2-maven-plugin</artifactId>
  <version>2.3.1</version>
  <executions>
    <execution>
      <id>xjc-schema</id>
      <goals>
        <goal>xjc</goal>
      </goals>
    </execution>
  </executions>
  <configuration>
    <sources>
      <source>src/main/resources/xsds</source>
    </sources>
    <packageName>generated</packageName>
    <clearOutputDir>false</clearOutputDir>
  </configuration>
</plugin>
```

Maven plugin to
generate Java classes
out of xsd file

Generate Java classes from xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified"
  elementFormDefault="qualified"
  targetNamespace="http://springtraining/greeting"
  xmlns:tns="http://springtraining/greeting"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:complexType name="Person">
    <xs:sequence>
      <xs:element type="xs:string" name="FirstName" />
      <xs:element type="xs:string" name="LastName" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="GreetingRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Person" type="tns:Person" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="GreetingResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:string" name="Greeting" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

greeting.xsd

```
@XmlAccessorType(
@XmlType(
@XmlRootElement(
public class GreetingRequest {
    ...
}
```

```
@XmlAccessorType(
@XmlType(
public class Person {
    ...
}
```

```
@XmlAccessorType(
@XmlType(
@XmlRootElement(
public class GreetingResponse {
    ...
}
```


Generated classes

```
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name = "", propOrder = {
    "person"
})
@XmlRootElement(name = "GreetingRequest")
public class GreetingRequest {

    @XmlElement(name = "Person", required = true)
    protected Person person;
    ...
}
```

```
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name = "", propOrder = {
    "greeting"
})
@XmlRootElement(name = "GreetingResponse")
public class GreetingResponse {

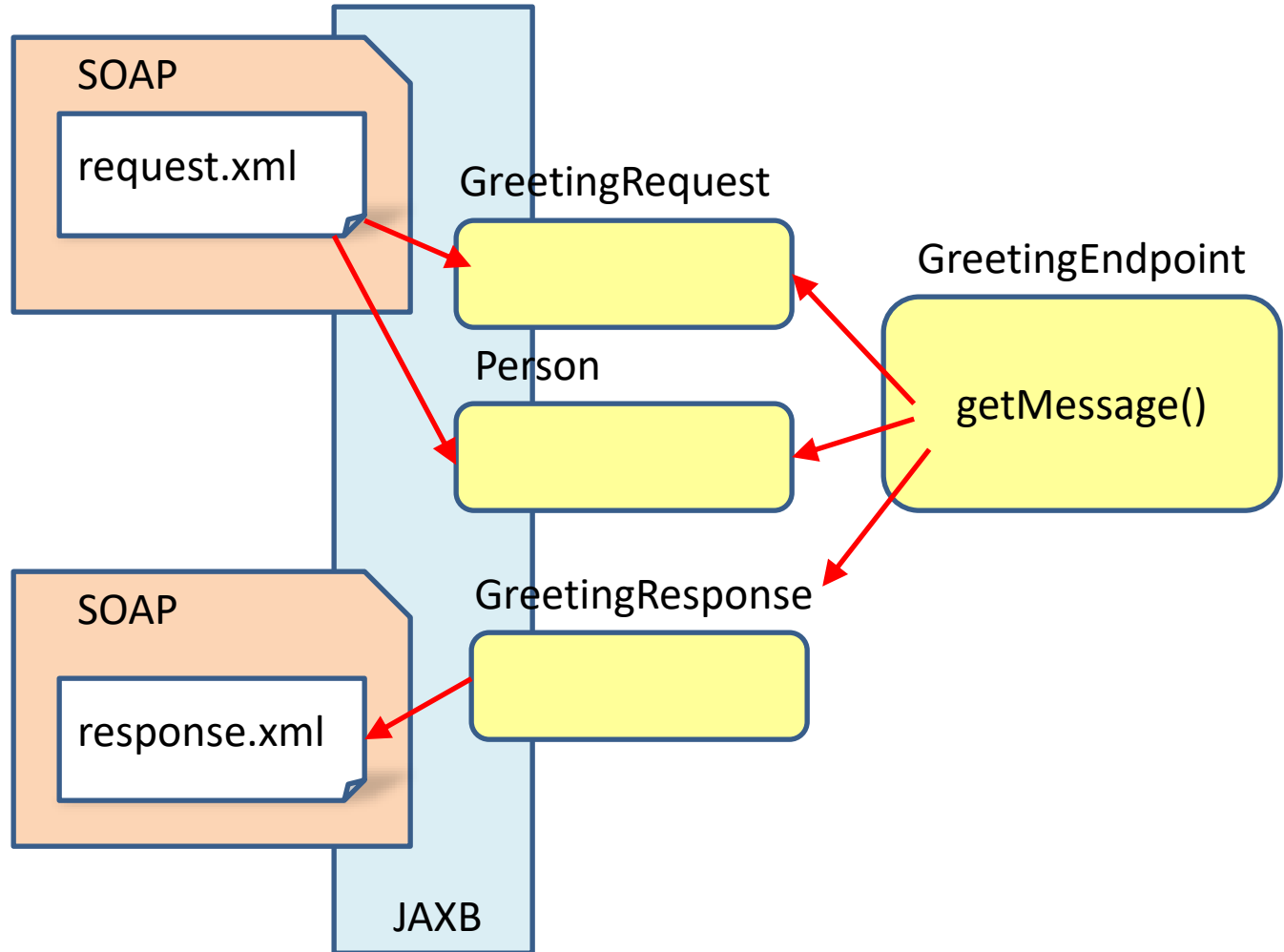
    @XmlElement(name = "Greeting", required = true)
    protected String greeting;
    ...
}
```

```
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name = "Person", propOrder = {
    "firstName",
    "lastName"
})
public class Person {

    @XmlElement(name = "FirstName", required = true)
    protected String firstName;
    @XmlElement(name = "LastName", required = true)
    protected String lastName;
    ...
}
```

JAXB annotation

Spring-WS service



GreetingEndpoint Implementation

@Endpoint

@Endpoint

Namespace

```
@Endpoint
public class GreetingEndpoint {
    private static final String NAMESPACE_URI = "http://springtraining/greeting";

    @PayloadRoot(namespace = NAMESPACE_URI, LocalPart = "GreetingRequest")
    @ResponsePayload
    public GreetingResponse getGreeting(@RequestPayload GreetingRequest request) {
        GreetingResponse response = new GreetingResponse();
        response.setGreeting("Hello " + request.getPerson().getFirstName() + " " +
            request.getPerson().getLastName());
        return response;
    }
}
```

Service configuration

```
@EnableWs
@Configuration
public class WebServiceConfig extends WsConfigurerAdapter {
    @Bean
    public ServletRegistrationBean messageDispatcherServlet(ApplicationContext
                                                            applicationContext) {
        MessageDispatcherServlet servlet = new MessageDispatcherServlet();
        servlet.setApplicationContext(applicationContext);
        servlet.setTransformWsdlLocations(true);
        return new ServletRegistrationBean(servlet, "/ws/*");
    }
    @Bean(name = "greeting")
    public DefaultWsd11Definition defaultWsd11Definition(XsdSchema
                                                         greetingSchema) {
        DefaultWsd11Definition wsdl11Definition = new DefaultWsd11Definition();
        wsdl11Definition.setPortTypeName("GreetingPort");
        wsdl11Definition.setLocationUri("/ws");
        wsdl11Definition.setTargetNamespace("http://springtraining/greeting");
        wsdl11Definition.setSchema(greetingSchema);
        return wsdl11Definition;
    }
    @Bean
    public XsdSchema greetingSchema() {
        return new SimpleXsdSchema(new ClassPathResource("xsds/greeting.xsd"));
    }
}
```

Namespace

Get the WSDL



The screenshot shows a web browser window with the address bar displaying `localhost:8080/ws/greeting.wSDL`. The page content indicates that the XML file does not have associated style information and displays the document tree. The XML is a WSDL file with the following structure:

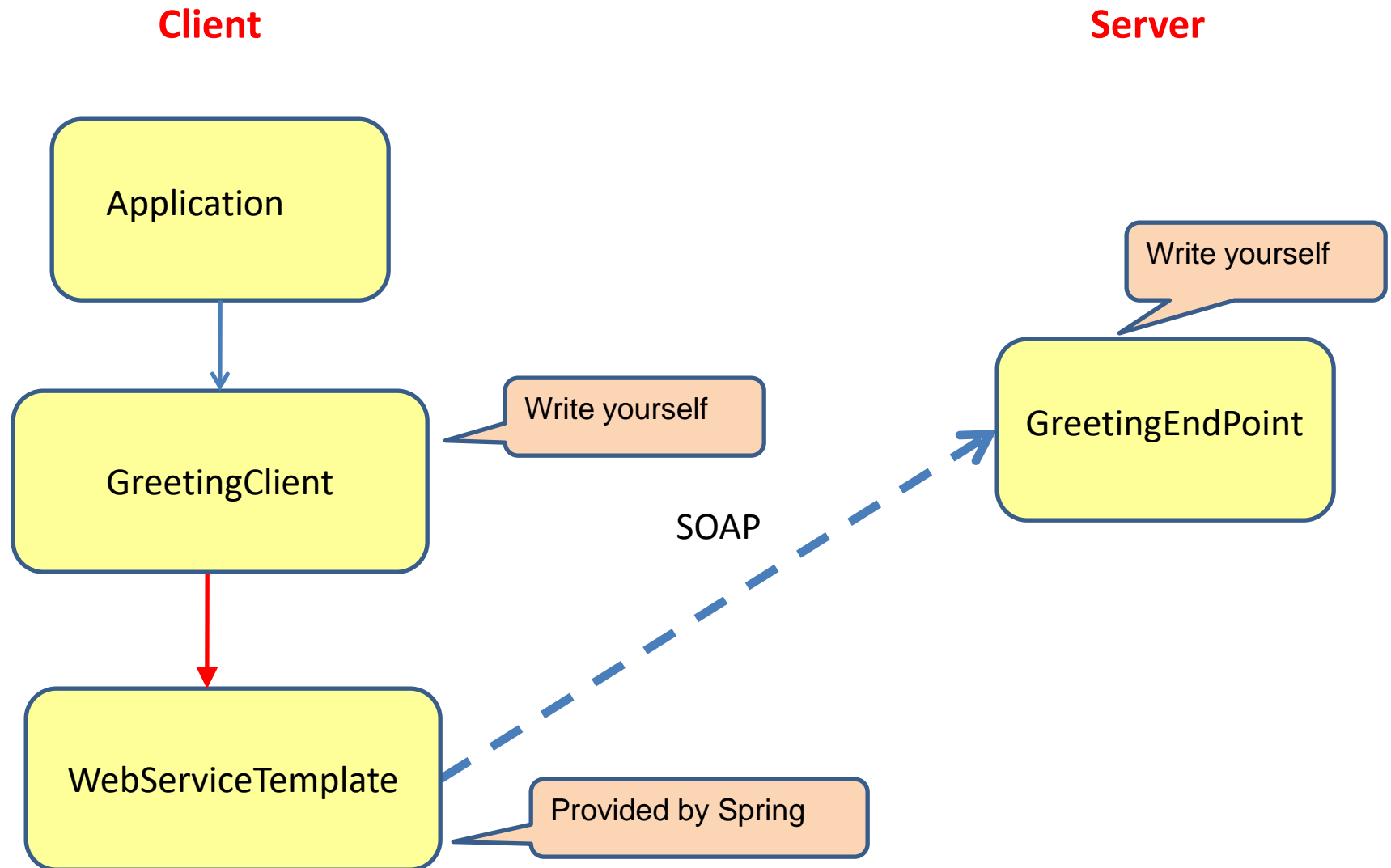
```
<?xml version='1.0' encoding='UTF-8'?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:sch="http://springtraining/greeting"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://springtraining/greeting"
  targetNamespace="http://springtraining/greeting">
  <wsdl:types>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" attributeFormDefault="unqualified"
      elementFormDefault="qualified" targetNamespace="http://springtraining/greeting">
      <xs:element name="GreetingRequest">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Person">
              <xs:complexType>
                <xs:all>
                  <xs:element name="FirstName" type="xs:string"/>
                  <xs:element name="LastName" type="xs:string"/>
                </xs:all>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="GreetingResponse">
```

Test the webservice

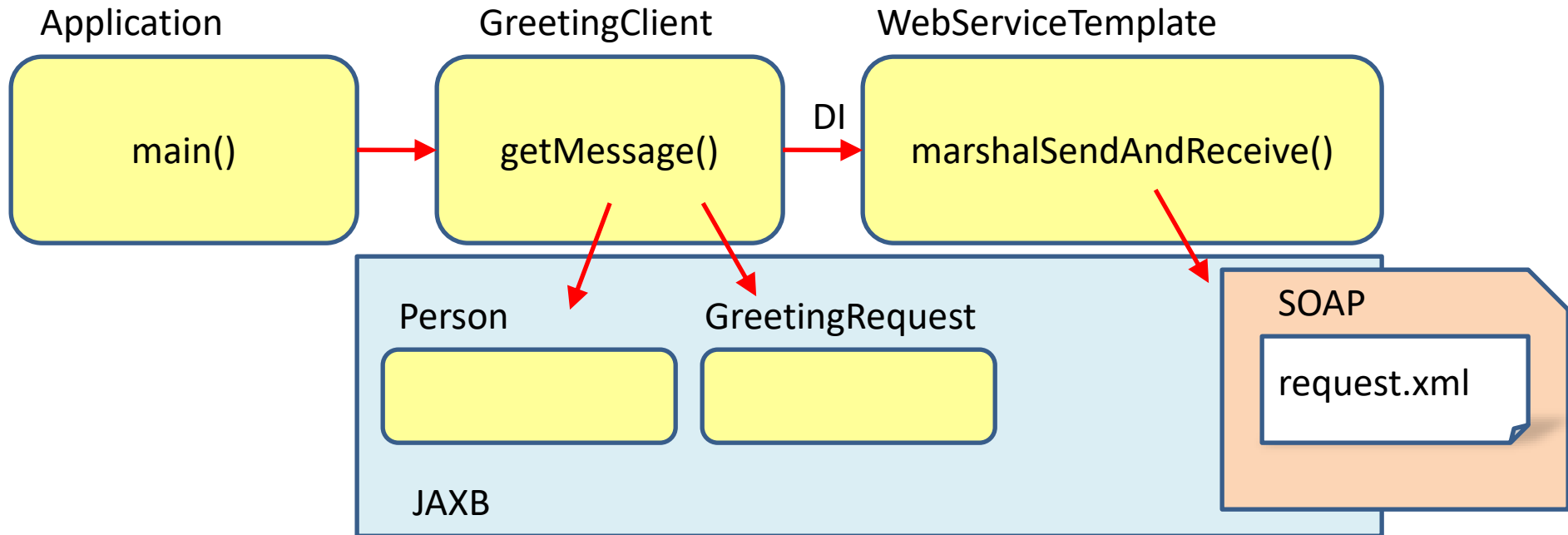
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:gre="http://springtraining/greeting">
  <soapenv:Header/>
  <soapenv:Body>
    <gre:GreetingRequest>
      <gre:Person>
        <gre:FirstName>Frank</gre:FirstName>
        <gre:LastName>Brown</gre:LastName>
      </gre:Person>
    </gre:GreetingRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:GreetingResponse xmlns:ns2="http://springtraining/greeting">
      <ns2:Greeting>Hello Frank Brown</ns2:Greeting>
    </ns2:GreetingResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

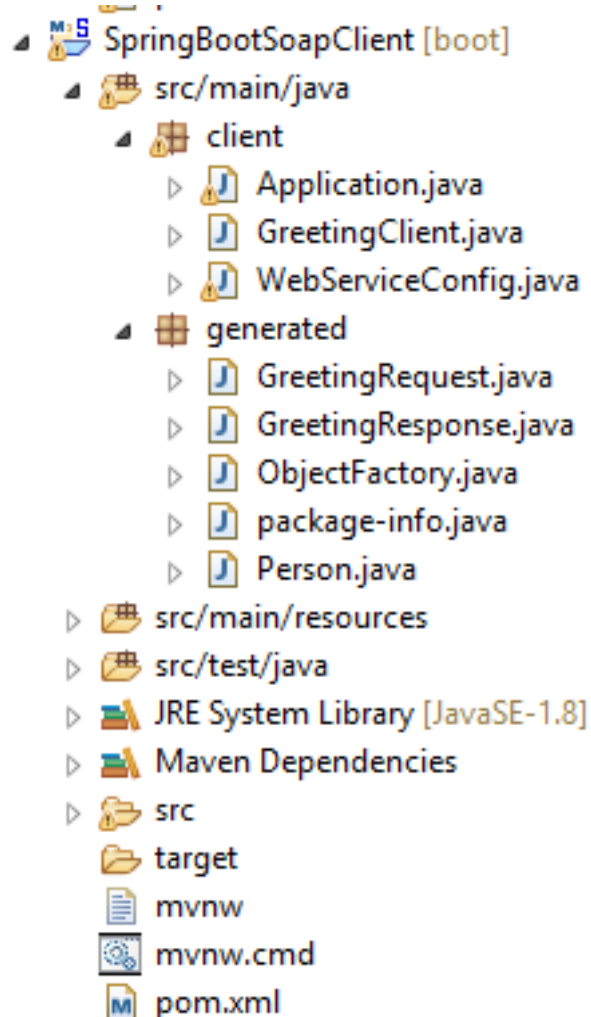
Spring WS



Spring-WS client



Spring-WS client



Spring WS Client

```
public class GreetingClient extends WebServiceGatewaySupport {

    public String getMessage(Person person) {
        GreetingRequest request = new GreetingRequest();
        request.setPerson(person);

        GreetingResponse response = (GreetingResponse)
            getWebServiceTemplate().marshalSendAndReceive(request);
        return response.getGreeting();
    }
}
```

```
@SpringBootApplication
public class Application {

    public static void main(String[] args) {
        AnnotationConfigApplicationContext context =
            new AnnotationConfigApplicationContext(Application.class);
        GreetingClient greetingClient = context.getBean(GreetingClient.class);
        Person person = new Person();
        person.setFirstName("Frank");
        person.setLastName("Brown");
        System.out.println(greetingClient.getMessage(person));
    }
}
```

Client configuration

```
@Configuration
public class WebServiceConfig {
    @Bean
    public Jaxb2Marshaller marshaller() {
        Jaxb2Marshaller marshaller = new Jaxb2Marshaller();
        // this package must match the package in the <generatePackage> specified in
        // pom.xml
        marshaller.setContextPath("generated");
    }
    return marshaller;
}

@Bean
public GreetingClient greetingClient(Jaxb2Marshaller marshaller) {
    GreetingClient client = new GreetingClient();
    client.setDefaultUri("http://localhost:8080/ws/greeting");
    client.setMarshaller(marshaller);
    client.setUnmarshaller(marshaller);
    return client;
}
}
```