

JAX-WS.
Пример написания клиента.

Преподаватель:
Денис Сергеевич Фокин

JAX-WS

- Java API for XML Web Services – Java программный интерфейс для XML Web сервисов
- Технология для создания Web сервисов и клиентских приложений использующих для обмена данными XML
- Позволяет обмениваться данными, как на основе сообщений, так и на основе RPC (remote procedure call)

JAX-WS основан на SOAP

SOAP –XML протокол, предназначенный для обмена сообщениями через HTTP соединение

Спецификация SOAP

- Определяет структуру обёрток
- Кодировки
- Соглашения для вызова и отклика Web сервисов

SOAP сообщения достаточно сложны

Пример SOAP сообщения

```
<?xml version="1.0"?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
  soap:encodingStyle="http://www.w3.org/2001/12/soap-
  encoding">
  <soap:Body>
    <m:GetPrice xmlns:m="http://www.w3schools.com/prices">
      <m:Item>Apples</m:Item>
    </m:GetPrice>
  </soap:Body>
</soap:Envelope>
```

JAXP-WS не требует знания SOAP

Разработчик не должен генерировать или разбирать (parse) SOAP сообщения. Этим занимается JAXP-WS

- На стороне сервера разработчик определяет операции Web сервиса, как реализацию Java интерфейса
- На стороне клиента – создаётся посредник (proxy) для Web сервиса

Технологии W3C используемые в JAX-WS

- HTTP - HyperText Transfer Protocol
- SOAP - Simple Object Access Protocol
- Web ServiceDescription Language (WSDL)

SOAP ссылки

- Спецификация SOAP
<http://www.w3.org/TR/soap/>
- What is SOAP? tutorial
<http://searchwebservices.techtarget.com/se>
- SOAP учебник для начинающих
<http://www.w3.org/2002/07/soap-translation/russian/part0.html>

WSDL

WSDL (**англ.** *Web Services Description Language*) — язык описания веб-сервисов, основанный на языке **XML**.

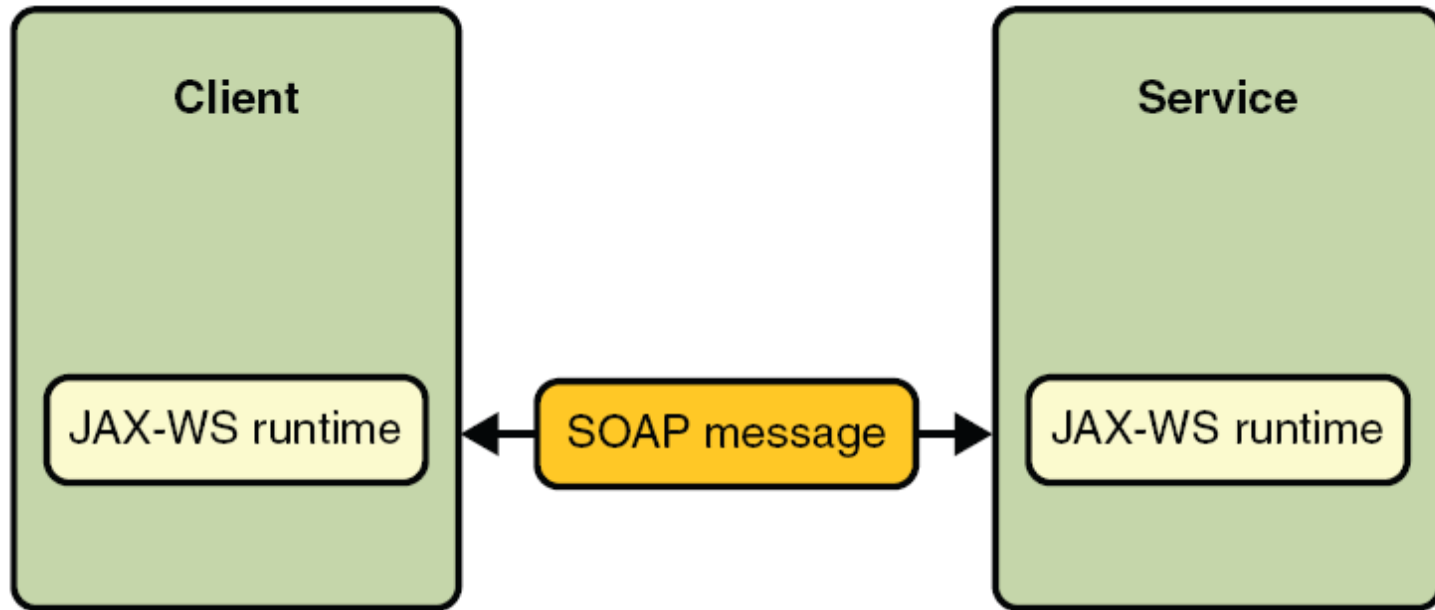
Составные части WSDL документа

- определение типов данных — определение вида отправляемых и получаемых сервисом XML сообщений
- абстрактные операции — список операций, которые могут быть выполнены с сообщениями
- связывание сервисов — способ, которым сообщение будет доставлено

Пример WSDL запроса

```
<mx:WebService id="BBAuthenticate" wsdl="http://10.1.1.1/auth.wsdl"
  useProxy="false" showBusyCursor="true"
  makeObjectsBindable="true"> <mx:operation name="Logon">
  <mx:request> <userName>{yourUserName.text}</userName>
    <password>{yourPassword.text}</password>
    <language>{yourLanguage.text}</language>
  </mx:request>
</mx:operation>
</mx:WebService>
<mx:DataGrid rowCount="2" allowMultipleSelection="true" x="10"
  y="380" id="authResponse" width="560"
  dataProvider="{BBAuthenticate.Logon.lastResult}" height="45">
```

Взаимодействие посредством JAX-WS



Реализация Web сервиса с помощью JAX-WS

Шаг первый

Java класс помеченный аннотацией `javax.jws.WebService`. Аннотация `@WebService` определяет класс как конечная точка сервиса

SEI (*service endpoint interface*)

SEI – Java интерфейс или класс, который объявляет методы, вызываемые на Web сервисе

Шаг второй

Используя утилиту `wsgen`, сгенерировать на базе класса реализацию, позволяющую клиенту соединяться с JAX-WS окружением*

**Sun Java System Application Server 9.1
ReferenceManual*

Требования к конечной точке JAX-WS (1)

- Реализация класса должна быть аннотирована `javax.jws.WebService` или `javax.jws.WebServiceProvider` аннотациями
- Класс реализации может неявно декларировать SEI
- Бизнес метод должен быть открытым (`public`) и не должен быть `final` или `static`
- Бизнес метод, вызываемый пользователем должен быть помечен аннотацией `javax.jws.WebMethod`.

Требования к конечной точке JAX-WS (2)

- Бизнес методы должны принимать параметры и возвращать значения совместимые с JAX-WS
- Класс реализации не может быть `final` или `abstract`
- Класс реализации обязан иметь публичный конструктор

Требования к конечной точке JAX-WS (3)

- Класс реализации не должен иметь финалайзеров
- В классе реализации могут быть использованы аннотации `@PostConstruct` и `@PreDestroy`

Пример реализации

```
package helloservice.endpoint;  
import javax.jws.WebService;  
@WebService  
public class Hello {  
    private String message = new String("Hello, ");  
    public void Hello() {}  
    @WebMethod  
    public String sayHello(String name) {  
        return message + name + ".";  
    }  
}
```

Что осталось за сценой?

- SOAP with Attachments API for Java (SAAJ)
- The Java™ Architecture for XML Binding (JAXB)

Вопросы и ответы