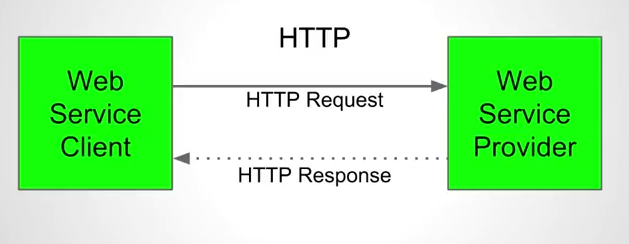
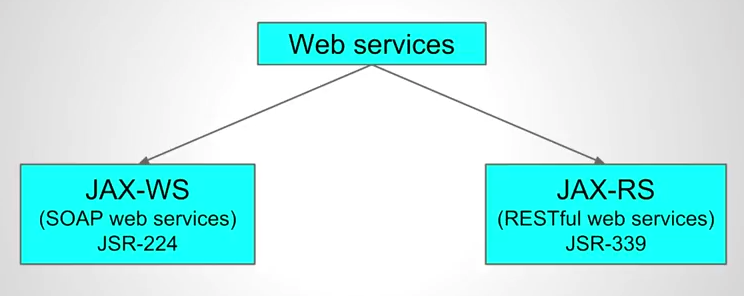
Webservice’y w świecie Javy

Webservice są usługami służącymi do wymiany komunikatów poprzez sieć między różnymi systemami.



JAXB służy do konwersji obiektów z i do XML’a.

W Javie mamy dwa rodzaje Web service’ow

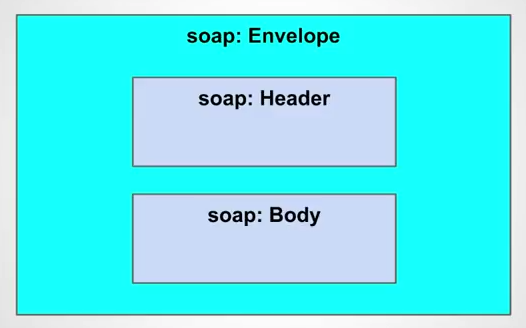


JAX-WS

Specyfikuje jak z Javovych service’ów i klas wygenerować WSDL i jak na podstawie WSDL’a wygenerować klasy JAVE’owe.

SOAP – Simple Object Access Protocol – opisuje wymianę komunikatów między naszymi web service’ami za pomocą tylko i wyłącznie XML’a.

Przykładowy format wiadomości SOAP:



WSDL (Web Service Description Language) – służy do opisu WebService’ów SOAP’owych. Może być przekazany komuś i służyć jako instrukcja do przygotowania wiadomości. Jest to XML’owy format, opisujący usługi sieciowe jako zbiór endpoint’ów operujących na wiadomościach, zawierających informacje zorientowane dokumentowo lub proceduralnie. WSDL jest bardzo rozszerzalny i pozwala na opis usług i ich wiadomości niezależni od formatu wiadomości, czy protokołu komunikacyjnego.

WSDL składa się z takich elementów:

* **Types** – kontener na definicje typów danych, wykorzystujący jakiś system typów np. XSD,
* **Message** – abstrakcyjna definicja przesyłanych danych, zgodnych ze zdefiniownaym typem,
* **Operation** – abstrakcyjny opis akcji obsługiwanych przez usługę,
* **Port Type** – abstrakcyjny zestaw operacji, obsługiwanych przez jeden lub więcej endpoint’ów,
* **Binding** – konkretny protokół i format danych wyspecyfikowany dla konkretnego typu portu,
* **Port** – pojedynczy endpoint zdefiniowany jako kombinacja bindinu i adresu sieciowego
* **Service** – kolekcja powiązanych endpoint’ów

<?xml version="1.0"?>

<definitions name="StockQuote"

targetNamespace="http://example.com/stockquote.wsdl"

          xmlns:tns="http://example.com/stockquote.wsdl"

          xmlns:xsd1="http://example.com/stockquote.xsd"

          xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"

          xmlns="http://schemas.xmlsoap.org/wsdl/">

    <types>

       <schema targetNamespace="http://example.com/stockquote.xsd"

              xmlns="http://www.w3.org/2000/10/XMLSchema">

           <element name="TradePriceRequest">

              <complexType>

                  <all>

                      <element name="tickerSymbol" type="string"/>

                  </all>

              </complexType>

           </element>

           <element name="TradePrice">

              <complexType>

                  <all>

                      <element name="price" type="float"/>

                  </all>

              </complexType>

           </element>

       </schema>

    </types>

    <message name="GetLastTradePriceInput">

        <part name="body" element="xsd1:TradePriceRequest"/>

    </message>

    <message name="GetLastTradePriceOutput">

        <part name="body" element="xsd1:TradePrice"/>

    </message>

    <portType name="StockQuotePortType">

        <operation name="GetLastTradePrice">

           <input message="tns:GetLastTradePriceInput"/>

           <output message="tns:GetLastTradePriceOutput"/>

        </operation>

    </portType>

    <binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">

        <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

        <operation name="GetLastTradePrice">

           <soap:operation soapAction="http://example.com/GetLastTradePrice"/>

           <input>

               <soap:body use="literal"/>

           </input>

           <output>

               <soap:body use="literal"/>

           </output>

        </operation>

    </binding>

    <service name="StockQuoteService">

        <documentation>My first service</documentation>

        <port name="StockQuotePort" binding="tns:StockQuoteBinding">

           <soap:address location="http://example.com/stockquote"/>

        </port>

    </service>

</definitions>

**Przykładowe annotacje JAX-WS:**

* @javax.jws.WebService
* @javax.jws.Webparam
* @javax.jws.WebResult
* @javax.jws.soap.SOAPBinding
* @javax.jws.OneWay

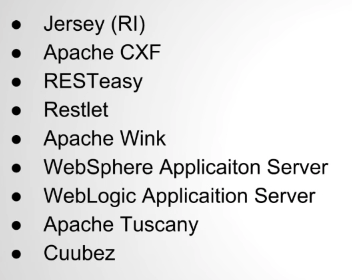
**WSDL Message Patterns**

* Request-Response
* One Way – bez odpowiedzi
* Notification – serwer wysyła notyfikacje bez odpowiedzi (nie jest w JAX-WS)
* Solicit-Response – serwer wysyła notyfikację i oczekuje odpowiedzi (nie jest w JAX-WS)

**JAX-RS (Representational State Transfer)**

Jest to taki pattern architektoniczny, który definiuje to jak powinny wyglądać nasze webservice’y.

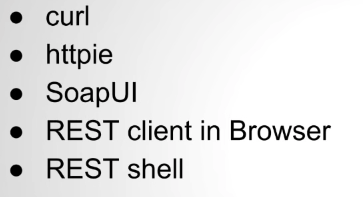
Implementacje JAX-RS:



REST (Representational State Transfer)

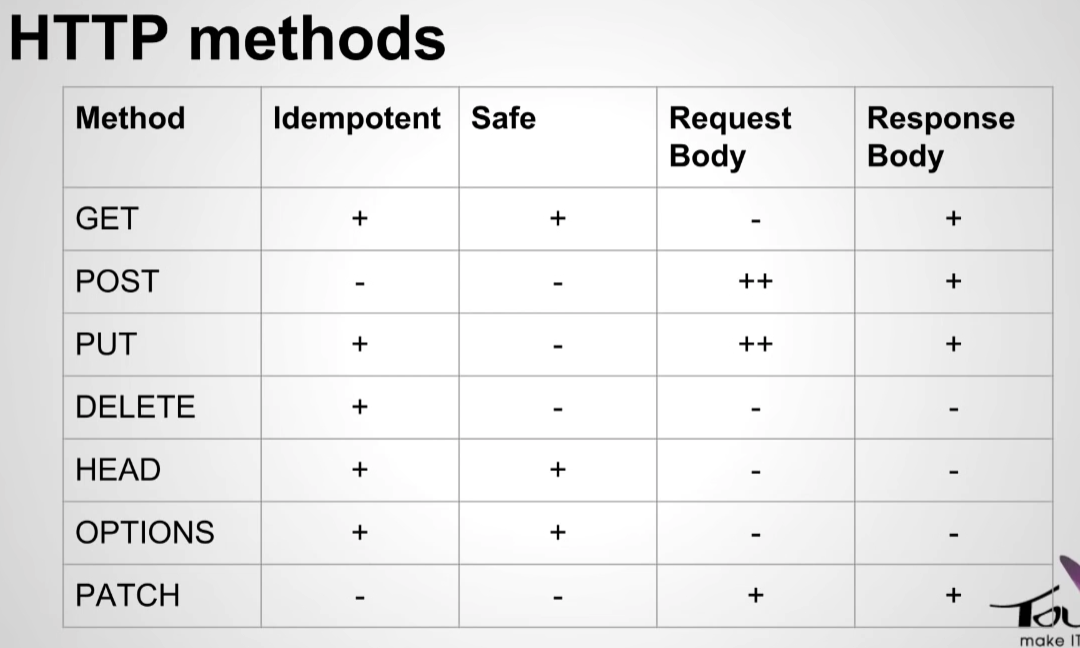
* Architektura klient-serwer,
* Bezstanowość,
* Cache – bardziej po stronie klienta,
* Jednolity kontrakt -
* Layeres System,
* Code-On-Demand,

Klienci REST’owi



Przykładowe annotacje JAX-RS

* @javax.ws.rs.GET
* @javax.ws.rs.POST
* @javax.ws.rs.DELETE
* @javax.ws.rs.PUT
* @javax.ws.rs.HEAD
* @javax.ws.rs.OPTION
* @javax.ws.rs.HttpMethod
* @javax.ws.rs.Path
* @javax.ws.rs.Consumes
* @javax.ws.rs.Produces



Parametry

* @javax.ws.rs.QueryParam
* @javax.ws.rs.PathParam
* @javax.ws.rs.HeaderParam
* @javax.ws.rs.CokkieParam
* @javax.ws.rs.MatrixParam
* @javax.ws.rs.FormParam
* @javax.ws.rs.BeanParam – do strzelania całym Bean’em

**WADL (Web Application**