



Zachodniopomorski
Uniwersytet
Technologiczny
w Szczecinie



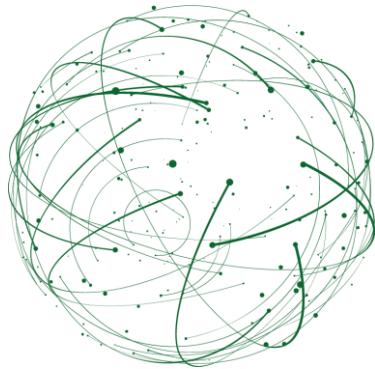
Wydział
Informatyki



KATEDRA INŻYNIERII OPROGRAMOWANIA

<http://wi.zut.edu.pl/>

PROJEKTOWANIE OPROGRAMOWANIA / SOFTWARE DESIGN



Aplikacje Webowe i Rozproszone

#03 : Java Architecture for XML Binding (JAXB)
Java API for XML Web Services (JAX-WS)

Prowadzący:

Krzysztof Kraska

email: kkraska@zut.edu.pl

Szczecin, 28 marca 2020 r.

JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

JAXB: Overview

- Allows reading and writing of XML documents
- Is an object-based model of XML document structure similar to DOM
- Provides an easy way to bind XML Schemas to Java representations
- Is an annotation-based configuration of Java-to-XML mapping
- Supports XML-Schema-to-Java-class generation and Java-class-to-XML-Schema generation
- Is used by JAX-WS and JAX-RS
 - The return values and method parameters that are JAXB-annotated class types are automatically converted for you.

ORACLE

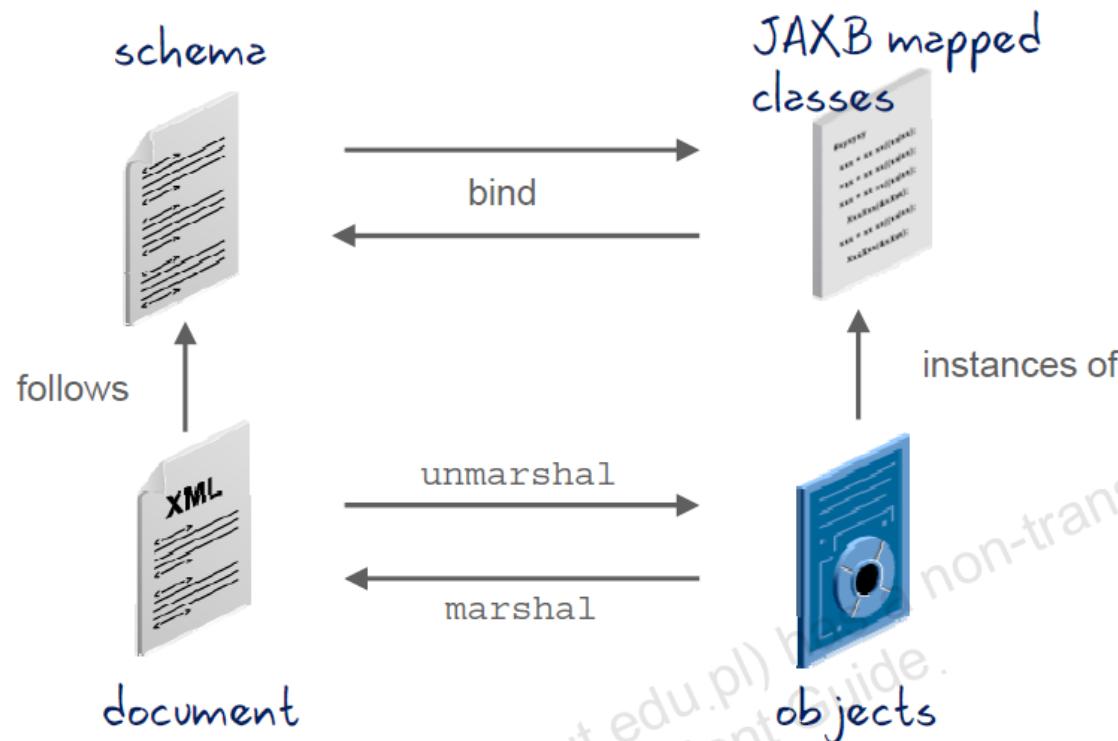
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 28 marca 2020 r.

JAVA ARCHITECTURE FOR XML BINDING

- APLIKACJE WEBOWE I ROZPROSZONE •

JAXB: Overview



JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

The general steps in the JAXB data binding process are:

- **Generate classes:** An XML Schema is used as input to the JAXB binding compiler to generate JAXB classes based on that schema.
- **Unmarshal:** The XML documents that are written according to the constraints in the source schema are unmarshalled by the JAXB binding framework.
- **Generate content tree:** The unmarshalling process generates a content tree of data objects instantiated from the generated JAXB classes. This content tree represents the structure and content of the source XML documents.
- **Validate (optional):** The unmarshalling process involves validation of the source XML documents before generating the content tree. Note that if you modify the content tree in the Process Content step, you can also use the JAXB Validate operation to validate the changes before marshalling the content back to an XML document.
- **Process content:** The client application can modify the XML data represented by the Java content tree by using the interfaces generated by the binding compiler.
- **Marshal:** The processed content tree is marshalled out to one or more XML output documents. The content may be validated before marshalling.

JAVA ARCHITECTURE FOR XML BINDING

- APLIKACJE WEBOWE I ROZPROSZONE •

XML and Java Class Comparison

Given the following XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<person>
    <name>Matt</name>
</person>
```

An equivalent Java class is:

```
@XmlRootElement
public class Person {
    private String name;
    public String getName() {/*...*/}
    public void setName(String name) {/*...*/}
}
```

JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

`@XmlRootElement`

The `@XmlRootElement` annotation is used to indicate that a class is used as a global (root) XML element.

```
@XmlRootElement (name="human")  
public class Person { /* ... */
```

Corresponds to an XML Schema of:

```
<xs:element name="human" type="person"/>  
<xs:complexType name="person">  
  <!-- ... -->  
</xs:complexType>
```

JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

Reading XML with JAXB

Reading XML is accomplished by using a JAXBContext, one or more JAXB-annotated classes, and an Unmarshaller.

```
try {
    JAXBContext jc =
        JAXBContext.newInstance(Person.class);
    Unmarshaller u = jc.createUnmarshaller();
    InputStream in =
        new FileInputStream("src/simple-read.xml");
    Person p = (Person)u.unmarshal(in);
    System.out.println("Name: " + p.getName());
} catch (JAXBException | IOException ex) {
}
```

JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

Unmarshalling the Document

Unmarshalling an XML document means creating a tree of content objects that represents the content and organization of the document. Unmarshalling provides a client application the ability to convert XML data into JAXB-derived Java objects.

To unmarshal an XML document, perform the following steps:

1. Create a `JAXBContext` object. This object provides the entry point to the JAXB API. When you create the object, you need to specify a context path. This is a list of one or more package names that contain interfaces generated by the binding compiler.
2. Create an `Unmarshaller` object. This object controls the process of unmarshalling. In particular, it contains methods that perform the actual unmarshalling operation.
3. Call the `unmarshal` method. This method does the actual unmarshalling of the XML document. In the example in the slide, the `Unmarshaller` instance (`u`) unmarshalls the XML data in `thesimple-read.xml` file.
4. Use the `get` methods in the schema-derived classes to access the XML data.

JAVA ARCHITECTURE FOR XML BINDING

- APLIKACJE WEBOWE I ROZPROSZONE •

Writing XML with JAXB

Writing XML is accomplished by using a JAXBContext, one or more JAXB-annotated classes, and a Marshaller.

```
try {
    Person p = new Person();
    p.setName("tom");
    JAXBContext jc =
        JAXBContext.newInstance(Person.class);
    Marshaller m = jc.createMarshaller();
    OutputStream out =
        new FileOutputStream("src/simple-write.xml");
    m.marshal(p, out);
} catch (JAXBException | IOException ex) {
}
```



JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

Marshalling

Marshalling is the opposite of unmarshalling. It creates an XML document from a content tree. Marshalling provides a client application the ability to convert a JAXB-derived Java object tree into XML data. To marshal a content tree, perform the following:

- Create a JAXBContext object and specify the appropriate context path—that is, the package that contains the classes and interfaces for the bound schema.
- Create a Marshaller object. This object controls the process of marshalling. In particular, it contains methods that perform the actual marshalling operation.
- Call the marshal method. This method does the actual marshalling of the content tree. When you call the method, you specify an object that contains the root of the content tree, and the output target.

JAVA ARCHITECTURE FOR XML BINDING

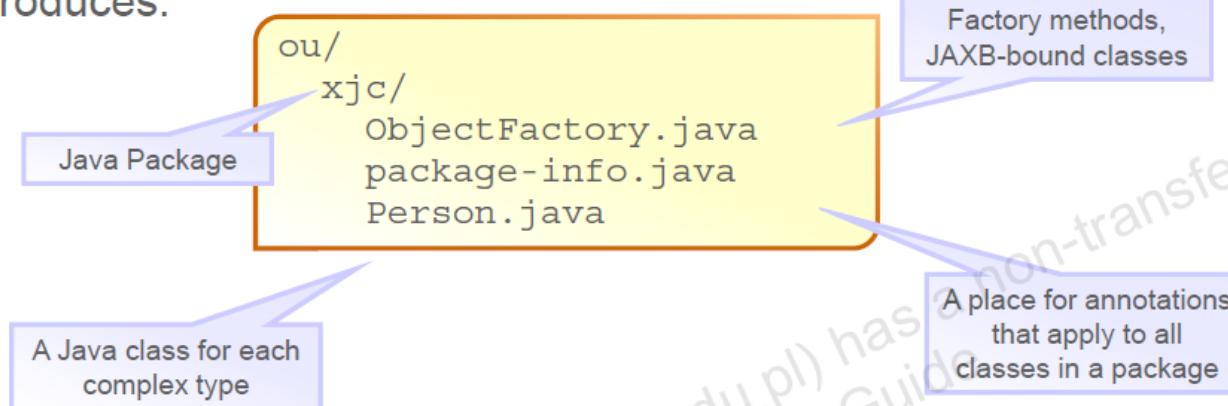
• APLIKACJE WEBOWE I ROZPROSZONE •

xjc

xjc is the JAXB binding compiler. It takes an XML Schema as input, and produces a Java package containing Java classes.

```
xjc -p ou.xjc person.xsd
```

Produces:



ORACLE

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 28 marca 2020 r.

JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

person.xsd XML Schema

```
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" targetNamespace="urn:example">
<xs:element name="person">
<xs:complexType>
<xs:sequence>
<xs:element name="name" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

JAVA ARCHITECTURE FOR XML BINDING

- APLIKACJE WEBOWE I ROZPROSZONE •

schemagen

schemagen is the JAXB Java-to-XML-Schema generator. It takes one or more Java files as input and produces XML Schemas.

```
schemagen ou\simple\Person.java
```

Produces schema1.xsd and ou\simple\Person.class

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 28 marca 2020 r.

JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

JAXBContext

The `JAXBContext` class is the entry point into the JAXB API. It is used to obtain:

- An `Unmarshaller` that can read XML
- A `Marshaller` that can write XML

`JAXBContext` can be passed a var-args class listing.

```
JAXBContext jc = JAXBContext.newInstance(Person.class);
```

`JAXBContext` can be passed a string of packages names.

```
JAXBContext jc = JAXBContext.newInstance("ou.schema");
```



JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

When supplying a string of package names, the following are applicable:

- Multiple package names can be supplied by separating the package names with colons
`JAXBContext jc = JAXBContext.newInstance("ou.schema:ou.other");`
- Every package specified must contain either of the following:
 - An `ObjectFactory.java` that is annotated with `@XmlRegistry` and contains a factory method for each Java class that is bound to an XML type
 - A `jaxb.index` file. The `jaxb.index` file contains a list of JAXB-bound class names (short name without package), one class name per line.

Did you know that you can switch JAXB implementations? EclipseLink Moxy is one available JAXB implementation. Moxy has additional features such as support for external binding files and JSON reading and writing. When you create a new `JAXBContext` instance, several checks are made to see if you have configured an alternative JAXB implementation. For more information, see the section titled “Discovery of JAXB Implementation” at <http://docs.oracle.com/javase/7/docs/api/javax/xml/bind/JAXBContext.html> and <http://www.eclipse.org/eclipselink/moxy.php>.

JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

JAXB Annotations

`@XmlRootElement` Indicates that a class is used as a global (root) XML element

`@XmlType`

- Specifies the name of a `complexType`
- Specifies the order of child elements

`@XmlAccessorType` Controls which members are bound to XML

`@XmlElement` Controls binding of class members to XML

`@XmlAttribute` Maps a class member to an XML attribute

`@XmlValue` Maps a class member to simple content within a complex type or a simple type

`@XmlEnum` Enables Java enums to be mapped to XML enumerated values

`@XmlElement` Is used to map a Java member to an XML Schema choice structure



JAVA ARCHITECTURE FOR XML BINDING

- APLIKACJE WEBOWE I ROZPROSZONE •

XmlRootElement

The `@XmlRootElement` annotation is used to indicate that a class is used as a global (root) XML element.

```
@XmlRootElement (name="human")
public class Person { /* ... */}
```

Corresponds to an XML Schema of:

```
<xs:element name="human" type="person"/>
<xs:complexType name="person">
  <!-- ... -->
</xs:complexType>
```



JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

XmlType

The `@XmlType` annotation is used to:

- Specify the name of the `complexType`
- Specify the order of child elements

```
@XmlRootElement (name="human")
@XmlType (name="individual",
          propOrder={ "name", "address" })
public class Person { /* ... */ }
```



JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

The `@XmlType` annotation can be used along with the `@XmlRootElement` or without it. When used without `@XmlRootElement`, the `@XmlType` annotation corresponds to a named global `complexType` but not a root element.

The code example in the slide corresponds to an XML Schema of

```
<xs:element name="human" type="individual"/>
<xs:complexType name="individual">
    <xs:sequence>
        <xs:element name="name" type="xs:string" minOccurs="0"/>
        <xs:element name="address" type="xs:string" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
```

You can map a Java class to a nested or anonymous `complexType` by using a zero-length string for the `name` attribute value.

```
@XmlType(name="", propOrder={"name", "address"})
```

JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

XmlAccessorType

The `@XmlAccessorType` annotation on a class controls which members are bound to XML. The default value is:

```
@XmlAccessorType(XmlAccessType.PUBLIC_MEMBER)
```

Possible values are:

- PUBLIC_MEMBER – All public fields and public getter/setter method pairs are bound to XML elements.
- FIELD – All fields, unless static or transient, are bound to XML elements.
- PROPERTY – All getter/setter method pairs are bound to XML elements.
- NONE – No members are bound to XML elements.

JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

The XML accessor type defines default behavior. You may override the specified behavior on class members by applying JAXB annotations on fields or getter/setter method pairs. The `@XmlTransient` annotation on a field, or getter/setter, indicates that the member should not be mapped to XML. Using other annotations such as `@XmlValue`, `@XmlAttribute`, or `@XmlElement` instructs JAXB to map the member to XML even if the XML accessor type excludes the member from mapping.

JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

XmlElement

The `@XmlElement` annotation is used to control binding of class members to XML.

```
@XmlRootElement (name="human")
@XmlAccessorType(XmlAccessType.NONE)
public class Person {
    @XmlElement(name="first-name", required=true)
    private String name;
```

- Map the field to XML even though the `XmlAccessorType` is `NONE`.
- The XML element name will be `first-name` instead of `name`.
- The `minOccurs` value is left at the default value of 1 instead of adding `minOccurs="0"`.



JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

XmlAttribute

The `@XmlAttribute` annotation maps a class member to an XML attribute.

```
@XmlRootElement()  
public class Person {  
    @XmlAttribute  
    public String name;  
    public String address;  
}
```

Corresponds to the following XML structure:

```
<person name="matt">  
    <address>221B Baker Street</address>  
</person>
```



JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

XmlValue

The `@XmlValue` annotation maps a class member to simple content within a complex type or a simple type where possible.

```
@XmlRootElement()  
public class Person {  
    @XmlAttribute  
    public String name;  
    @XmlValue  
    public String address;  
}
```

Corresponds to the following XML structure:

```
<person name="matt">221B Baker Street</person>
```



JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

Enumerations

Java enums can be mapped to XML enumerated values by using the `@XmlEnum` annotation.

```
@XmlType  
@XmlEnum  
public enum ProjectState {  
    @XmlEnumValue("0")  
    LATE,  
    @XmlEnumValue("1")  
    REALLY_LATE;  
}
```

Java identifiers cannot start with a number but XML enumerated values can.

JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

XmlElement

An `@XmlElement` annotation is used to map a Java member to an XML Schema choice structure.

```
@XmlElement(value = {  
    @XmlElement(name = "pie",  
    type = Pie.class,  
    required=true),  
    @XmlElement(name = "ice-cream",  
    type = IceCream.class,  
    required=true)  
})  
public Object obj;
```



JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

Errors and Validation

JAXB does not perform strict validation checking by default.

- When reading XML, unexpected elements and attributes that are not mapped to Java elements are ignored.
- When reading XML, malformed XML will cause a `javax.xml.bind.UnmarshalException` when calling `unmarshal`.
- If you want to keep track of validation failures, you can attach a `ValidationEventHandler` to the `Unmarshaller`.

```
Unmarshaller u = jc.createUnmarshaller();
ValidationEventCollector vec =
    new ValidationEventCollector();
u.setEventHandler(vec);
```



JAVA ARCHITECTURE FOR XML BINDING

• APLIKACJE WEBOWE I ROZPROSZONE •

NetBeans JAXB Support

- NetBeans supports the Ant `xjc` task through an XML Binding file type that is located in the XML new file category.
- However, there is no support for `schemagen`.
- You may run it on the command line, modify your `build.xml`, or generate the schema programmatically.

```
JAXBContext context =
JAXBContext.newInstance("mypackage");
context.generateSchema(new SchemaOutputResolver() {
    public Result createOutput(String namespaceUri,
                               String fileName) throws
IOException {
    return new StreamResult(new
File("src/myschema.xsd"));
}
});
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Objectives

After completing this lesson, you should be able to:

- Describe the basic structure of a Simple Object Access Protocol (SOAP) message and how it is encapsulated by transport mechanisms
- Explain how WSDL defines a web service, including its message representation and transport mechanism
- Create a JAX-WS web service
- Create a JAX-WS client by using Java SE
- Create a JAX-WS client by using Java EE



JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Need for Web Services

A computer program calling a subroutine located on a different machine on a network is not a new development.

- Remote Procedure Call (RPC) from Sun Microsystems was an early example of cross-system execution.
- Other examples include .Net Remoting, CORBA/IOP, and RMI.
- In all cases, the need is the same:
 - Run an operation on a remote machine (or even on the same machine but in a different address space).
- Many of the remoting technologies suffer from problems:
 - Platform-specific (CPU architecture or programming language)
 - Complexity



JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Characteristics of Web Services

- Platform neutral (both CPU architecture and programming language independent)
 - A platform-neutral data-interchange format is needed (text instead of binary).
- Client/server architecture
 - A server has a set of available operations.
 - A client can request the execution of an operation on a server.
- HTTP
 - “Web” services are most likely to use HTTP as a transport protocol.
 - In theory, both SOAP and RESTful services are not tied to HTTP.
 - In practice, you almost always use HTTP.
- The use of a service must be described.

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

XML and JSON in Web Services

To facilitate a platform-neutral exchange of data, a general purpose data-interchange format is needed.

- **Extensible Markup Language (XML):** Used by both SOAP and REST web services
 - A large number of processing libraries exist to support XML in almost every language.
 - Java developers can use SAX, DOM, StAX, and JAXB.
 - SOAP web service rely on XML but libraries hide a large portion of the XML work.
- **JavaScript Object Notation (JSON):** Used by REST web services
 - A subset of JavaScript
 - Less verbose than XML
 - Support evolving in other languages



JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Reasons for Using SOAP

The SOAP web services specification defines an interoperable, platform-independent means for component interaction. The SOAP web service requirements include:

- Decoupling message representation from transport mechanisms
- Supporting extensible frameworks

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Extensible Message Representation
Simple Object Access Protocol



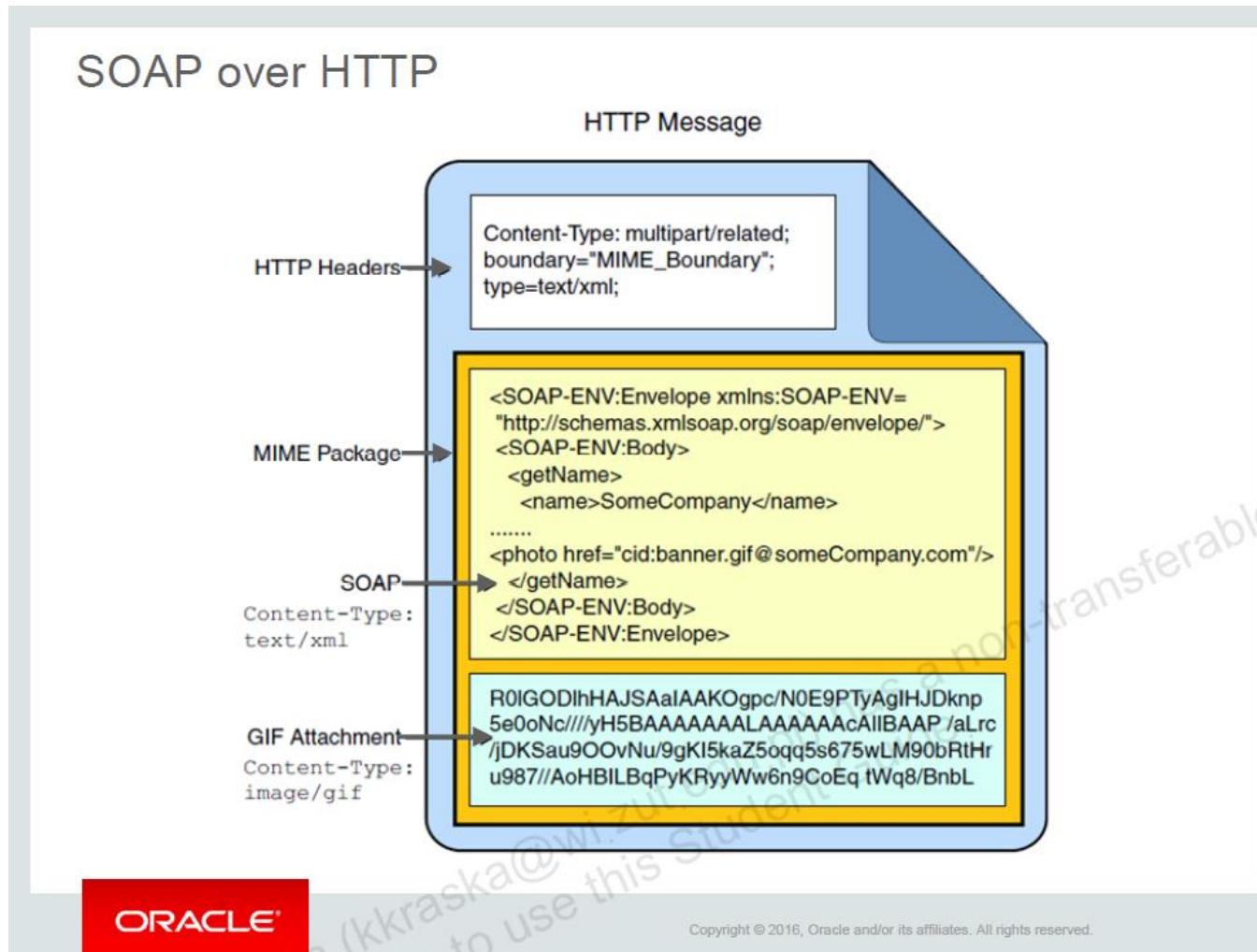
ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 28 marca 2020 r.

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •



HTTP is not the only transport supported by SOAP, but it is the most common.

JAVA API FOR XML (SOAP) WEB SERVICES

- APLIKACJE WEBOWE I ROZPROSZONE •

Raw SOAP/HTTP Request

```
Accept: text/html, image/gif, image/jpeg, */*; q=.2
Connection: Keep-Alive
Content-Length: 206
Content-Type: text/xml; charset=utf-8
Host: localhost:7001
SOAPAction: ""
User-Agent: Oracle JAX-RPC 1.1
<env:Envelope
    xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
<env:Header/>
<env:Body>
    <getHello xmlns="http://ou/">
        <arg0 xmlns="">matt</arg0>
    </getHello>
</env:Body>
</env:Envelope>
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 28 marca 2020 r.

JAVA API FOR XML (SOAP) WEB SERVICES

- APLIKACJE WEBOWE I ROZPROSZONE •

Raw SOAP/HTTP Response

```
HTTP/1.1 200 OK
Content-type: text/xml; charset=utf-8

<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope
    xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
    <ns2:getHelloResponse xmlns:ns2="http://ou/">
        <return>Hello matt!</return>
    </ns2:getHelloResponse>
</S:Body>
</S:Envelope>
```



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 28 marca 2020 r.

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

WSDL

Web Services Description Language (WSDL) is an XML-based interface definition language that is used for describing a web service.

- Provides an interface that exposes web services
- Maps web services to underlying programs and software systems
- Describes how to connect to and use a web service

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Structure of a WSDL File

<definitions>: Root WSDL Element

<types>: What data types will be transmitted?

<message>: What exact information is expected?

<portType>: What operations (functions) will be supported?

<binding>: How will the messages be transmitted on the wire?
What SOAP-specific details are there?

<service>: Define the collection of ports that make up the service and where
is the service located?

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Note that there are differences in the structure of a WSDL file between version 1.1 and 2.0 of the WSDL specification:

WSDL 1.1	WSDL 2.0
Service	Service
Port	Endpoint
Binding	Binding
Porttype	Interface
Operation	Operation
Message	n/a
Types	Types

Messages were removed in WSDL 2.0. In 1.1, the message defined the typing information of the binding. In WSDL 2.0, the schema handles all typing.

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

WSDL: Sample Structure

```
<?xml version='1.0' encoding='UTF-8'?>
<definitions xmlns:wsu="http://docs.oasis-
    open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
    1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
    xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://ou/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    targetNamespace="http://ou/" name="HelloService">
    <types><!-- ... --></types>
    <message><!-- ... --></message>
    <portType><!-- ... --></portType>
    <binding><!-- ... --></binding>
    <service><!-- ... --></service>
</definitions>
```

The WSDL file for a SOAP service can be downloaded by making an HTTP GET request to the URL of the web service with a wsdl query parameter, for example:

<http://localhost:7001/HellosWS>HelloService?wsdl>

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

JAX-WS

Java API for XML Web Services (JAX-WS) is a technology for building web services and clients that communicate by using XML.

- Web service operations are transmitted as SOAP messages (XML files) over HTTP.
- The JAX-WS API hides SOAP complexity from the application developer.
- On the server side, web service operations are defined by using annotations and POJOs.
- Client programs are easy to code.
- With JAX-WS, clients and web services are platform-independent.
 - A JAX-WS client can access a web service that is not running on the Java Platform and vice versa.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

The JAX-WS API is a Java API for creating SOAP-based web services.

Szczecin, 28 marca 2020 r.

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

JAX-WS Development Approaches

- WSDL First
 - Generate web service artifacts by using the information contained in a WSDL file.
- Code First
 - Create a service endpoint interface (SEI) or value classes as Java source files, and then use them as inputs to generate the associated WSDL descriptor and other portable artifacts.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 28 marca 2020 r.

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Benefits of a Code First Approach

A code first approach to providing a web service interface to an existing enterprise application offers several advantages:

- You can use the quickest development path.
- It is a natural approach, especially when business logic has already been implemented.
- You can map existing domain models directly to WSDL with little effort.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 28 marca 2020 r.

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Web Service Creation Process

To create a JAX-WS web service, perform the following steps:

1. Create a web or an enterprise application with NetBeans.
2. Create a class to represent your service.
3. Annotate the class with `@WebService`.
4. Build and deploy your project.
 - A WSDL file is automatically generated for you.
5. Create a web service client by referring to the WSDL URL.



JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Simple Greeter Web Service

```
@WebService  
public class Greeter {  
    public final String worldGreeting = "Hello World!";  
  
    public Greeter() { }  
  
    @WebMethod  
    public String greetWorld(){  
        return worldGreeting;  
    }  
}
```

Must be annotated with
@WebService

Must have a no-arg
constructor

Use this annotation to
define a web service
method.

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

The JAX-WS requirements on the Java class are as follows:

- Must be annotated with `javax.jws.WebService`
- Must not be declared `final`
- Must not be `abstract`
- Must have a `default public no argument constructor`
- Must not have a `finalize()` method

Note: The annotations, `@WebService` and `@WebMethod`, are actually part of another specification, JSR 181, Web Services Metadata.

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Testing a Web Service

- Using the WebLogic Server, the URL to test a web service has the following format:
`http://localhost:7001/appName/webService?Tester`
 - Operations with complex types are not supported.
- You can use any HTTP tool that can submit a POST request with a custom body:
 - Firefox RESTClient
 - cURL
- You can use JDeveloper's HTTP Analyzer to test a web service.
- You can use SoapUI for a dedicated SOAP web service testing application.



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Web Service URLs

The default URL for a web service is based on the class name of the service.

- POJO endpoints: The default URL is
<http://host:port/{app}/{classname}Service>.
 - Use of `@WebService(serviceName = "mypath")` results in:
<http://host:port/{app}/mypath>.
- EJB endpoints: The default URL is
<http://host:port/{classname}/{classname}Service>.
 - Use of `@WebService(serviceName = "mypath")` results in:
<http://host:port/{classname}/mypath>.

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

JAX-WS Requirements for Web Service Methods

- Must be `public`
 - By default, every public method in the class will be a part of the web service.
- Must not be `static` or `final`
- Must have JAXB-compatible parameters and return types
 - Parameters and return types must not implement the `java.rmi.Remote` interface.

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 28 marca 2020 r.

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Generating the WSDL

To publish an SEI, there must be a WSDL per service and JAXB classes for each SOAP message.

- JAX-WS can deliver WSDL and message classes dynamically. The WSDL is available at:
<http://host:port/path/to/service?WSDL>.

Obtaining the WSDL from a running service is preferred. However, if you need to generate a WSDL without running the SEI:

- Use the annotation processor in `jaxws-tools.jar`.

```
javac -processor  
com.sun.tools.ws.processor.modeler.annotation.WebServiceApp
```



JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Creating a JAX-WS Client

- The JAX-WS Reference Implementation (RI) is included in Java SE.
- The basic SOAP web services can be accessed by a Java SE client.
- Use NetBeans to generate client artifacts.
 - All you need is the web service's WSDL.
 - Right-click the package and select **New > Web Service from WSDL**.
 - Optionally use `wsimport` (similar to `xjc`) to generate client artifacts from the command line.

ORACLE®

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Szczecin, 28 marca 2020 r.

JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Classes Produced from WSDL

Running `wsimport` or using the wizard in NetBeans on the Greeter WSDL (one port with one operation) produces six .java files.

- `GreetWorld.java`: The JAXB-annotated class that is used to marshall and unmarshal the SOAP request body
- `GreetWorldResponse.java`: The JAXB-annotated class that is used to marshall and unmarshal the SOAP response body
- `Greeter.java`: The JAX-WS port class
- `GreeterService.java`: The JAX-WS service class
- `ObjectFactory.java`: The JAXB factory
- `package-info.java`: The JAXB package-level annotations



JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Local WSDL Files

When configuring a web service client reference in NetBeans, two local copies of the WSDL file are made.

- MyProject/xml-resources /
 - You must switch to the Files tab to inspect this directory.
 - This copy is used when running `wsimport` every time you build your project.
- MyProject/src/META-INF/wsdl /
 - This is packaged with the application so that a copy of the WSDL need not be downloaded when initializing the client.
 - Clients can improve performance by using this copy, but by default, it is not used.



JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Calling a Web Service

After the JAX-WS artifacts are generated, calling a web service is fairly easy.

```
public class Main {  
    public static void main(String[] args) {  
        GreeterService service = new GreeterService();  
        Greeter port = service.getGreeterPort();  
        System.out.println("The Message is: " +  
                           port.greetWorld());  
    }  
}
```

The Service subclass
that loads the WSDL.
This is a port factory.

The port with
methods that match
the operations

Call the web service.



JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Java EE Web Service Clients

When creating an EE client, you probably will perform the same steps every time. If your client is a managed component that supports injection of resources, you can obtain the service or port via injection.

```
@WebServiceRef(wsdlLocation = "WEB-INF/wsdl/localhost_7001/SOAPWS/GreeterService.wsdl")  
private GreeterService service;
```

Optional
performance-
enhancing attribute

```
@WebServiceRef(wsdlLocation = "WEB-INF/wsdl/localhost_7001/SOAPWS/GreeterService.wsdl")  
private Greeter port;
```



JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Custom WSDL Method Name

Web services require unique names for each method. Use the `@WebService` annotation to specify a different service name:

```
@WebMethod(operationName="GreetFirst")
public String greetName(String name) {
    return englishGreeting + name;
}

@WebMethod(operationName="GreetFull")
public String greetName(String first, String last) {
    return englishGreeting + first + " " + last;
}
```



JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Custom WSDL Operation Parameter Names

The default names of method parameters are arg0, arg1, and so on:

```
<xs:complexType name="GreetFirst">
<xs:sequence>
<xs:element name="arg0" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
```

To modify the element names used for method parameters, use the `@WebParam` annotation.

```
@WebMethod(operationName="GreetFirst")
public String greetName(
    @WebParam(name="firstName") String name) {
    return englishGreeting + name;
}
```



JAVA API FOR XML (SOAP) WEB SERVICES

• APLIKACJE WEBOWE I ROZPROSZONE •

Viewing SOAP Messages

Sometimes, it helps to dump all the SOAP messages exchanged by the server to the standard output.

- JAX-WS includes a generic framework, JAX-WS Handlers, which can be used to dump the messages.
- Use a system property to dump all JAX-WS messages.
-Dcom.sun.xml.ws.transport.http.HttpAdapter.dump=true

KONIEC

• APLIKACJE WEBOWE I ROZPROSZONE •



**DZIĘKUJĘ
ZA UWAGĘ!!!**

Szczecin, 28 marca 2020 r.