

# Rozproszone systemy internetowe

## Web serwisy

### JAX-WS (SOAP web services)

#### cz.4 SSL/TLS, @MTOM

Sources and help:

[JAX-WS Tutorial by mkyong](#)

[JAX-WS Release Documentation](#)

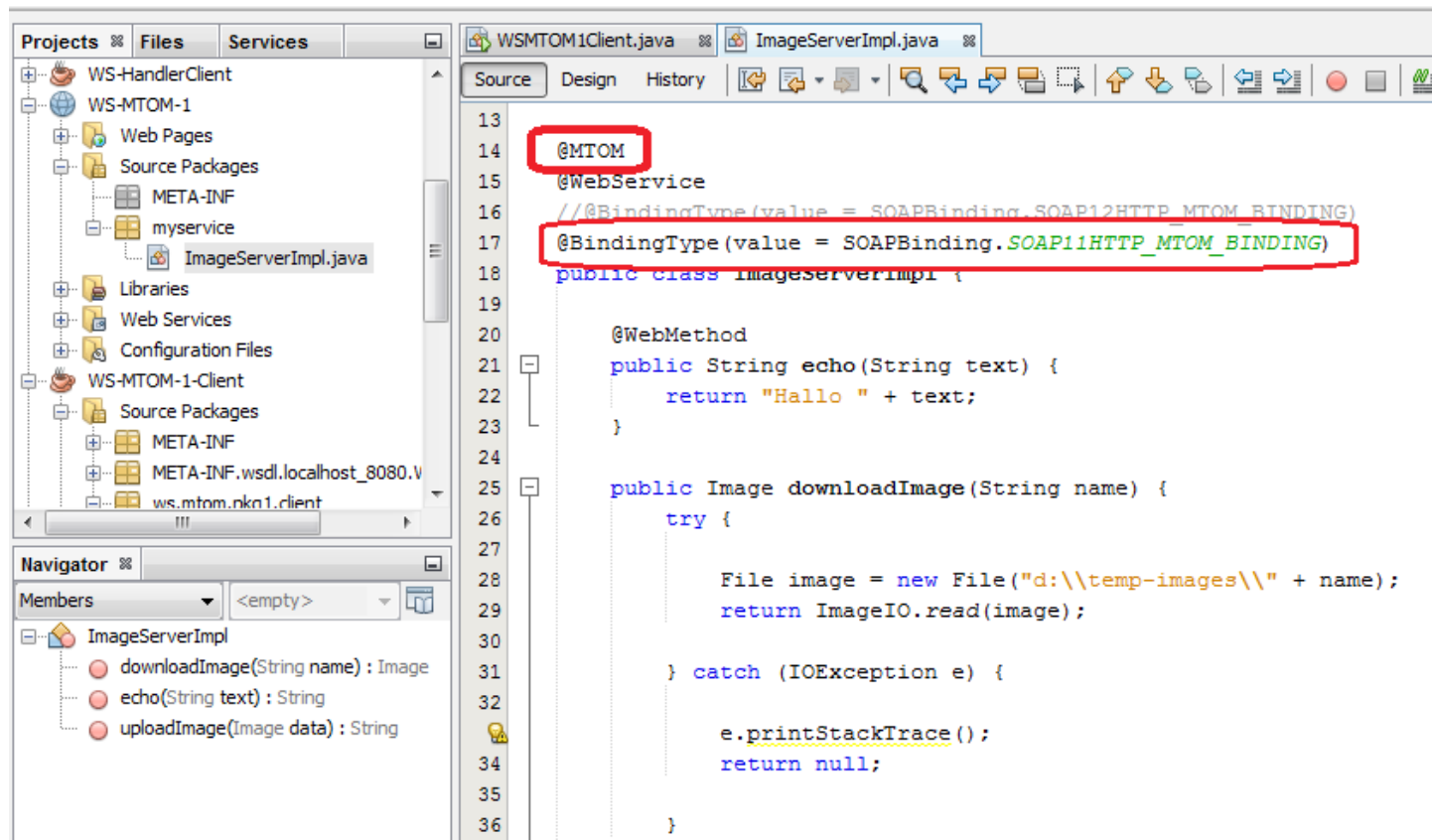
[Metro User Guide](#)

## Ćwiczenie 1. Przesyłanie binarnych załączników, użycie MTOM i XOP

Uruchomić przesyłanie załączników binarnych z wykorzystaniem MTOM

HELP: JAX-WS Release Documentation: [MTOM](#)

Enabling MTOM on server



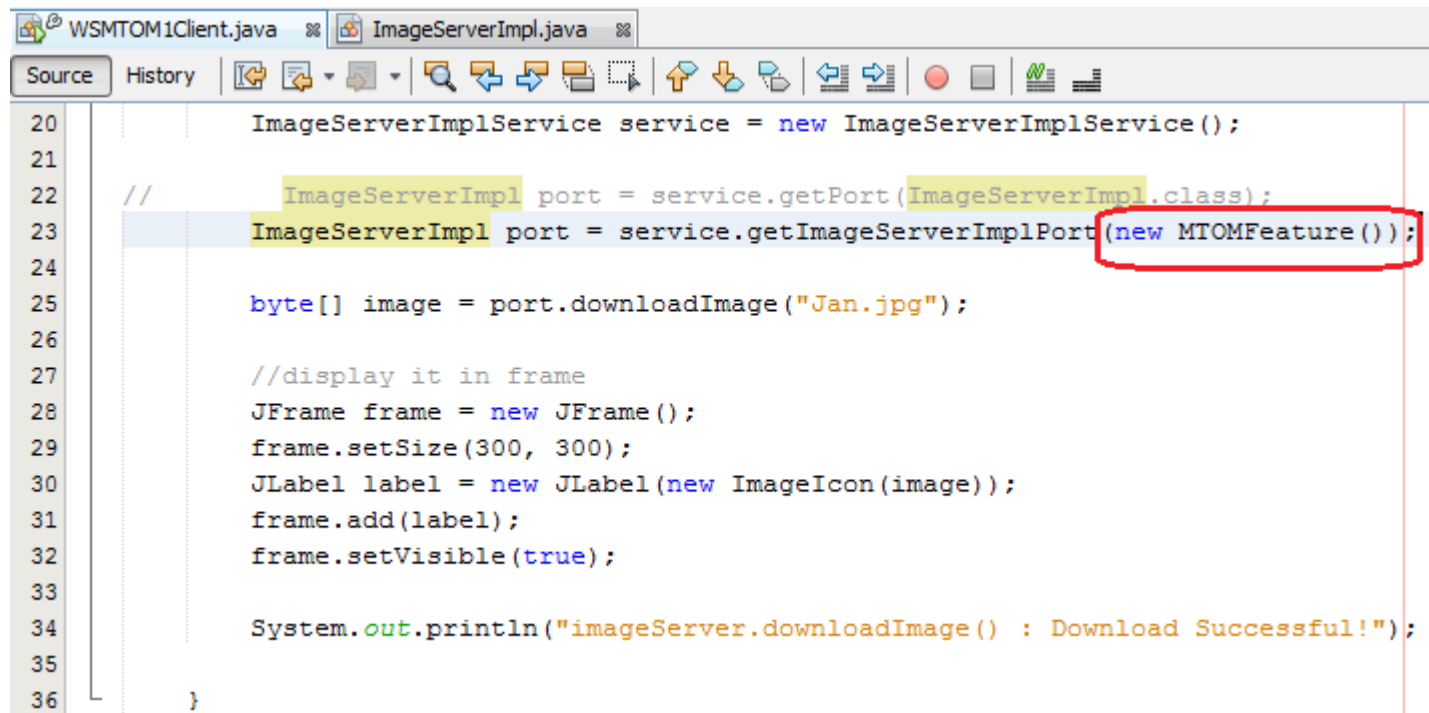
WSDL - co się zmieniło w WSDL?

```

- <!--
  Published by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.1-b419 (branches/2.3.1.x-7937; 2014-
-->
- <!--
  Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.1-b419 (branches/2.3.1.x-7937; 2014-
-->
- <definitions targetNamespace="http://myService/" name="ImageServerImplService">
- <wsp:Policy wsu:Id="ImageServerImplPortBinding_MTOM_Policy-ImageServerImplPortBinding_MTOM_Policy">
  <ns1:OptimizedMimeSerialization wsp:Optional="true"/>
</wsp:Policy>
+ <types></types>
+ <message name="uploadImage"></message>
+ <message name="uploadImageResponse"></message>
+ <message name="downloadImage"></message>
+ <message name="downloadImageResponse"></message>
+ <message name="echo"></message>
+ <message name="echoResponse"></message>
+ <portType name="ImageServerImpl"></portType>
- <binding name="ImageServerImplPortBinding" type="tns:ImageServerImpl">
  <wsp:PolicyReference URI="#ImageServerImplPortBinding_MTOM_Policy-ImageServerImplPortBinding_MTOM_Policy"/>
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
- <operation name="uploadImage">
  <soap:operation soapAction="">

```

Enabling MTOM on client



```

20     ImageServerImplService service = new ImageServerImplService();
21
22     // ImageServerImpl port = service.getPort(ImageServerImpl.class);
23     ImageServerImpl port = service.getImageServerImplPort(new MTOMFeature());
24
25     byte[] image = port.downloadImage("Jan.jpg");
26
27     //display it in frame
28     JFrame frame = new JFrame();
29     frame.setSize(300, 300);
30     JLabel label = new JLabel(new ImageIcon(image));
31     frame.add(label);
32     frame.setVisible(true);
33
34     System.out.println("imageServer.downloadImage() : Download Successful!");
35
36 }

```

Przeprowadzić monitorowanie przesyłania załączników

16-04-04 1...	localhost	localhost	ImageServerI...	downloadIm...	219	518	44057	POST	/WS-MTOM-...	multipart/rel...
16-04-04 1...	localhost	localhost	ImageServerI...	downloadIm...	204	518	44057	POST	/WS-MTOM-...	multipart/rel...

```
<return><xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include" href="cid:31f10792-9308-461a-9d61-90d17b74f61e@example.jaxws.sun.com"/>./return
```

```
~ôûûáíÇ{~Çû™s{ŤÇy_ôûâBL/t□L{ŝóóD×İççq?zDŽçst-□>y□ç±ÔŤýËüýŕXęmszkföëý~_Çsd~}□#ókeŘĀci□□óxt□ç+Řýñ□?B~ŧŽřZ□+9_>İŮûŮS?Ô÷İscÇésû</óó□O!z
```

```
İÄËTK~pDuÂ««□»□úy□□&□sn□™Ŕ%«ŝŝz~□,Ů'□÷ûuWŝRÇŷŝ°çBŝ□A□M□□ó{°,a@NuŇi□gÇ"+F©@aİ(uB©#JEk□û
```

Jak jest przesyłany załącznik, gdy nie został włączony MTOM u klienta?

HTTP Monitor [WS-MTOM-1/ImageServerImplService?WSDL] HTTP Proxy on port 4040

Traffic Log

Count.	Time	Request Host	Target Host	Interface	Operation	Time Taken	Req Sz	Resp Sz	Method	Path	Content-Type
0	2016-04-04 ...	localhost	localhost	ImageServe...	downloadI...	31	482	109502	POST	/WS-MTO...	multi...
1	2016-04-04 ...	localhost	localhost	ImageServe...	downloadI...	219	518	44057	POST	/WS-MTO...	multi...
2	2016-04-04 ...	localhost	localhost	ImageServe...	downloadI...	204	518	44057	POST	/WS-MTO...	multi...
3	2016-04-04 ...	localhost	localhost	ImageServe...	downloadI...	15	205	145134	POST	/WS-MTO...	text/x...

Raw XML

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:downloadImage xmlns:ns2="http://my.service/">
      <arg0>Jan.jpg</arg0>
    </ns2:downloadImage>
  </S:Body>
</S:Envelope>
```

Raw JSON

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:downloadImageResponse xmlns:ns2="http://my.service/">
      <return>iVBORw0KGgoAAAANSUhEUgAAASIAAAGCAAAArTW5AACAAEIEQVR42iy715Mr2ZbeV9f0MVUoe
    </ns2:downloadImageResponse>
  </S:Body>
</S:Envelope>
```

Head... Attachm... WS... JM...

Headers (5) Attachments (0) WSS (0) JMS (5)

**Request 1**  
 http://localhost:8080/WS-MTOM-1/ImageServerImplService

**Raw XML**

```
<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/'>
  <soapenv:Header/>
  <soapenv:Body>
    <mys:downloadImage>
      <!--Optional-->
      <arg0>Jan.png</arg0>
    </mys:downloadImage>
  </soapenv:Body>
</soapenv:Envelope>
```

**Raw XML**

```
<?xml version='1.0' encoding='UTF-8'>
<S:Envelope xmlns:S='http://schemas.xmlsoap.org/soap/envelope/'>
  <S:Body>
    <ns2:downloadImageResponse xmlns:ns2='http://myservice/'>
      <return>
        <xop:include href='cid:93340ce1-e84c-4e9e-9f22-dbd678e5d72@example.jaxws.sun.com' xmlns:xop='http://www.w3.org/2001/XMLSchema#xop'>
          </xop:include>
        </return>
      </ns2:downloadImageResponse>
    </S:Body>
  </S:Envelope>
```

**Request Properties**

Property	Value
Name	Request 1
Description	
Message Size	293
Encoding	UTF-8
Endpoint	http://localhost:80...
Timeout	
Bind Address	
Follow Redirects	true
Username	
Password	
Domain	
Authentication Type	Global HTTP Setti...
WSS-Password Type	
WSS TimeToLive	
SSL Keystore	
Skip SOAP Action	false
Enable MTOM	true
Force MTOM	true
Inline Response Att...	false
Expand MTOM Att...	false
Disable multipart	false

**Attachments**

Name	Content type	Size	Part	Type	ContentID
93340ce1-e84c-4e9e...	image/png	43229	93340ce1-e84c-4e9e...	XOP	<93340ce1-e84c-4e9e...

response time: 229ms (87286 bytes)

## Ćwiczenie 2. Użycie SSL na serwerze

Uruchom SSL na serwerze (GlassFish lub Tomcat). Odwołaj się do web serwisu za pomocą klienta przez SSL.

Pomoc 1: [SSL + Jax-WS i GlassFish](#) - do PART 2 włącznie. Na Glassfish 4.1 działa. Studenci zgłaszali, że na GlassFish 5.0 są problemy z SSL. Można próbować używając servera [Payara](#) (Payara Server is an open-source application server derived from GlassFish Server). GlassFish 5.1 i jdk1.8.0\_201 działa.

Pomoc 2: [link](#) SSL+ Tomcat

Do debugowania klienta javy można użyć komendy `System.setProperty("javax.net.debug", "all");`