

Streaming velikih datoteka preko SOAP web servisa

Mario Dijanošić

IN2 d.o.o.

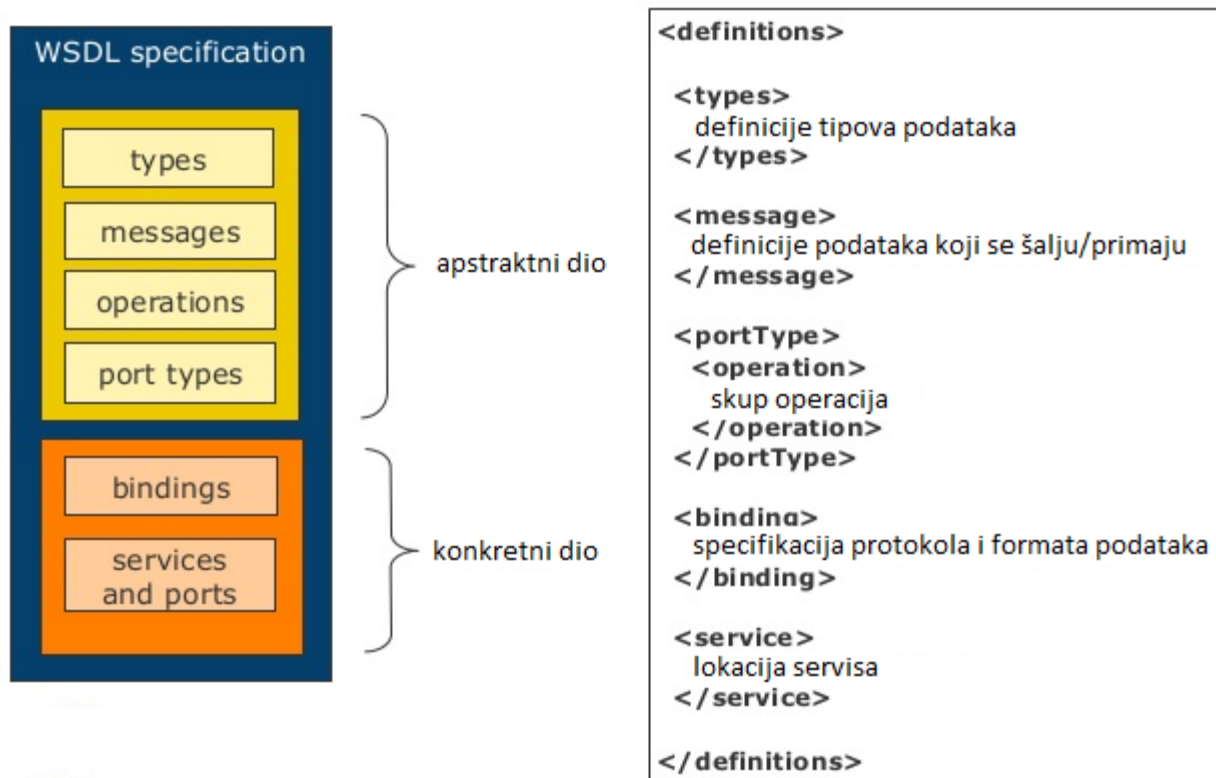
Web servisi



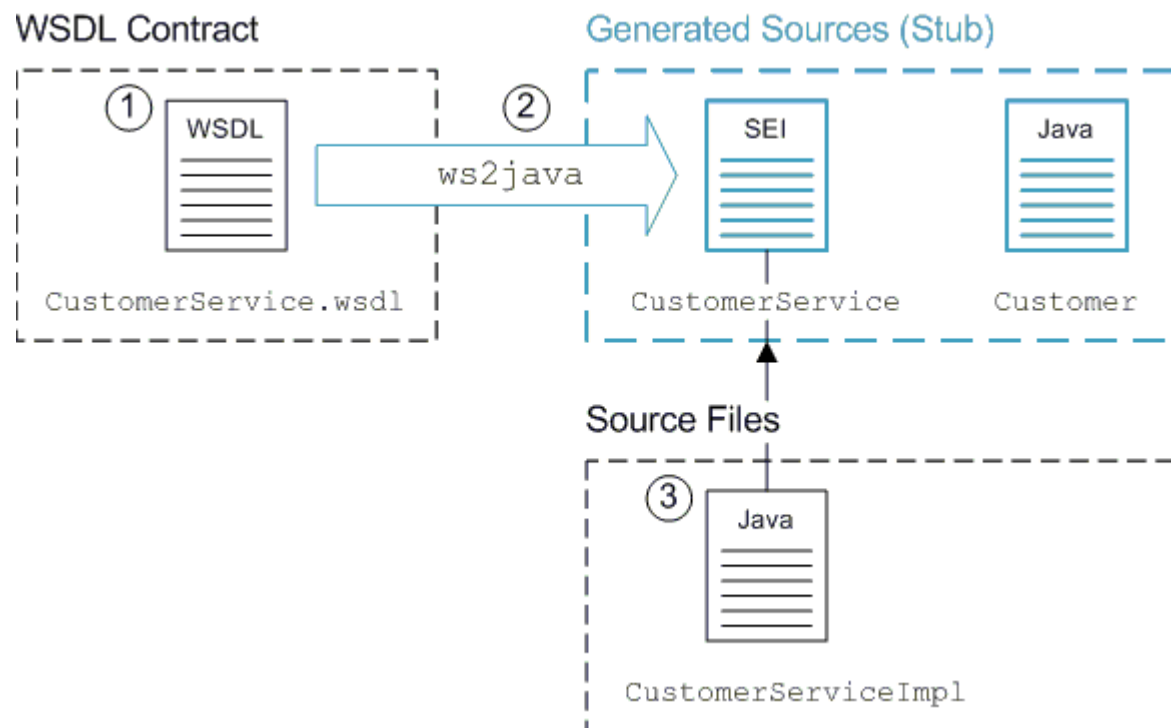
Vrste web servisa

- REST i SOAP web servisi
- REST
 - *Representational State Transfer*
 - Načelo arhitekture
- **SOAP**
 - *Simple Object Access Protocol*
 - Enterprise friendly
 - WS-Security, WS-AtomicTransaction, WS-ReliableMessaging
 - WSDL apstrakcija prema kojoj se radi
 - XML based

WSDL specifikacija

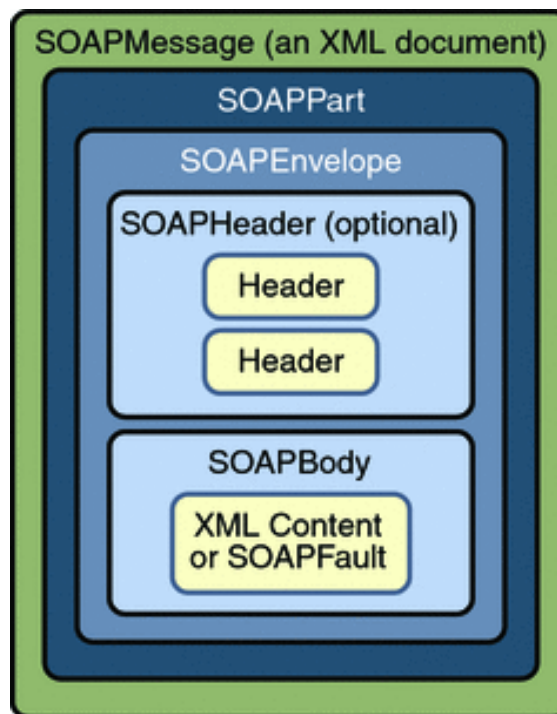


WSDL u Javi



SOAP Envelope

- SOAP poruka

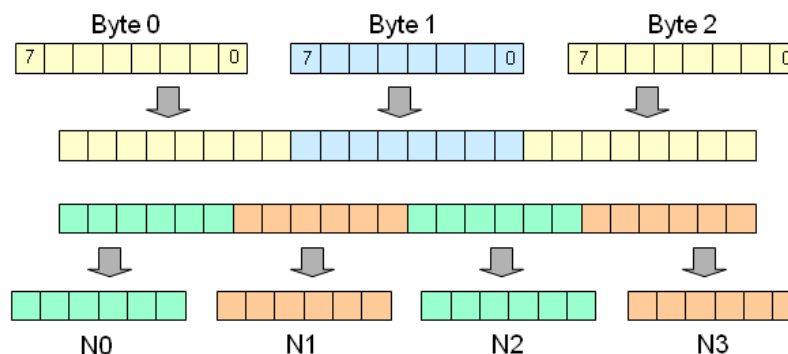


Slanje datoteka preko SOAP web servisa

- SOAP koristi XML
- Binarna datoteka – staviti u XML ili povezati iz XML-a
- Načini:
 - Base64 encoding
 - SOAP Attachments
 - MTOM

Base64

- Pretvorba binarnih podataka u ASCII text



- Overhead od teksta 20-30%
- Pogodno za male datoteke

Base64 u SOAP-u

- WSDL:

```
<element name="file" type="xsd:base64Binary" />
```

- Java kod:

```
private byte[] file;
```

- SOAP Envelope:

```
<file>UESDBBQAAGAIALBbiTxF3hbzNwAAAJ4AAAALAAAAZ2ZpdGVzdC50eHQzNIQAMyMoMEY  
AA0M9IDIyMLAEYQQPxDKCsAwMje1MjC3NzYEMCwNeLqhphuZQ00wQppmSYRwAUESBAHQAFAA  
CAAgAsFuJPEXeFvM3AAAANGAAAsAAAAAAAAAAAAAGAAAAAAAAAGdmaXRlc3QudHh0UESFBgAA  
AAABAAEAOQAAAGAAAAAAAAAA==</file>
```

SOAP Attachments

- Slanje binarnih podataka
- Nema enkodiranja
- 2 tipa:
 - MIME
 - SwaRef

SOAP attachments u praksi

- WSDL – posebno oblikovan

```
<binding name="xRayStorageBinding" type="tns:xRayStorage">
  <soap:binding style="document" />
  <operation name="store">
    <soap:operation soapAction="" style="document"/>
    <input name="storRequest">
      <mime:multipartRelated>
        <mime:part name="bodyPart">
          <soap:body use="literal"/>
        </mime:part>
        <mime:part name="imageData">
          <mime:content part="xRay" type="image/jpeg"/>
        </mime:part>
      </mime:multipartRelated>
    </input>
    <output name="storResponse">
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
```

- Java kod

```
private javax.activation.DataHandler;
```

SOAP attachments u praksi (2)

- MIME poruke

```
type=text/xml;
start="<rootpathID>"
Content-Length: ...
--_MIME_boundary_
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <rootpathID >
<soapenv:Envelope>
  <soapenv:Body>
    <sendImage>
      < filename>mooiemarjo.jpg</filename>
      <image href="cid:imgID"/>
    </sendImage>
  </soapenv:Body>
</soapenv:Envelope>
--_MIME_boundary_
Content-Type: image/jpeg
Content-Transfer-Encoding: binary
Content-ID: <imgID>
...JPEG image bytes...
--_MIME_boundary_--
```

SOAP Attachments

- Slično kao attachmenti u emailu
- SOAP poruka sadrži referencu na binarnu datoteku
- Binarna datoteka nije dio SOAP poruke, već je linkana
- Problem za neke stvari (npr. WS-Security)

MTOM

- *Message Transmission Optimization Mechanism*
- Koristi XOP - *XML-binary Optimized Packaging*
- Binarni data izvan SOAP omotnice
 - MIME attachment
 - uključen preko `<xop:Include>`
 - dio XML Infoseta

Princip rada MTOM-a

- XOP dokument + binarni privitak = XOP paket
- Primijena MTOM specifikacija -> XOP paket je MIME poruka u MTOM formatu
- MTOM šalje binarni sadržaj kao MIME privitak
 - Smanjuje količinu bajtova koji se prenose
 - Binarni sadržaj je semantički dio XML dokumenta
- MTOM konvertira poruku u MIME privitak prije slanja web servisu ili klijentu

MTOM – mehanizam optimizacije

- Mehanizam optimizacije – MIME poruka umjesto base64 stringa
- MTOM poruke - *multipart/related* MIME sekvence
- Root dio poruke - SOAP Envelope

MTOM – java i WSDL

- WSDL je isti kao kod base64

```
<element name="file" type="xsd:base64Binary" />
```

- U Javi se koristi DataHandler

```
private javax.activation.DataHandler;
```

MTOM SOAP Envelope

```

--uuid:8877a7f8-ac89-4482-a6f7-2b61fbf3702e
Content-Type: application/xop+xml; charset=UTF-8; type="text/xml"
Content-Transfer-Encoding: binary
Content-ID: <root.message@cxf.apache.org>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:uploadFile xmlns:ns2="http://mtom.ctk.in2.hr">
      <arg0>
        <datoteka>
          <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include"
            href="cid:eb5c4e5d-a7ae-4d0c-94cc-b1417cf-3@cxf.apache.org"/>
        </datoteka>
        <nazivDatoteke>testfile.txt</nazivDatoteke>
        <velicina>16</velicina>
      </arg0>
    </ns2:uploadFile>
  </soap:Body>
</soap:Envelope>
--uuid:8877a7f8-ac89-4482-a6f7-2b61fbf3702e
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <eb5c4e5d-a7ae-4d0c-94cc-b180362417cf-3@cxf.apache.org>

ovo je test file
--uuid:8877a7f8-ac89-4482-a6f7-2b61fbf3702e--

```

Pros & Cons MTOM-a

- Pros
 - Efikasnije slanje od base64 enkodirane datoteke
 - Dio XML-a -> omogućava npr. WS-Security
- Cons
 - Nije pogodan za puno malih datoteka
 - Dodatan overhead kod detekcije MTOM poruka za dispečiranje
 - Sve strane moraju podržavati

Streamanje datoteka

- Koristiti MTOM za privitak
- Koristiti odgovarajuće klase u Java kodu
 - Umjesto `byte[]` koristiti `javax.activation.DataHandler` klasu
- Podaci se čitaju u stream modu

Osobine streaminga

- Kod velikih datoteka poboljšava performanse web servisa
- Ne treba učitavati cijeli request u memoriju
- Kada ne streamati:
 - Poruka ne može biti rascijepana u sekvence
 - Poruke koje nisu u cijelosti dostupne prilikom početka slanja
 - Poruke koje moraju biti pravovremeno dostupne

Streamanje - datoteka

- Primjer streaminga za datoteku

```
import java.io.InputStream;
import java.io.FileInputStream;

import javax.activation.DataSource;
import javax.activation.DataHandler;

import org.apache.cxf.attachment.AttachmentDataSource;

...
InputStream inputStream = new FileInputStream(file);
AttachmentDataSource attachDataSource =
    new AttachmentDataSource("application/octet-stream", inputStream);
generiranaKlasaIzWsdla.setDataHandler(attachDataSource);
...

// ili
....
DataHandler dataHandler = generiranaKlasaIzWsdla.getDataHandler();
InputStream inputStream = dataHandler.getInputStream();
...
```

Streamanje - baza

- Primjer streaminga za bazu

```
import java.io.InputStream;
import java.sql.PreparedStatement;
import javax.activation.DataHandler;

DataHandler dataHandler = generiranaKlasaIzWsdla.getDataHandler();
InputStream inputStream = dataHandler.getInputStream();

PreparedStatement pstmt = conn.prepareStatement(
    "INSERT INTO TEST_TABLE (DATOTEKA) VALUES (?)");
pstmt.setBlob(1, inputStream);
pstmt.execute();
```

Zaključak

- Tri načina za slanje datoteka:
 - base64
 - SOAP with Attachments
 - MTOM
- MTOM - standard
 - najbolji izbor za veće datoteke
 - treba ga eksplicitno uključiti
- Za streaming koristiti *DataSource* i *DataHandler* klase iz `javax.activation` paketa

Hvala na pažnji!

Pitanja ?