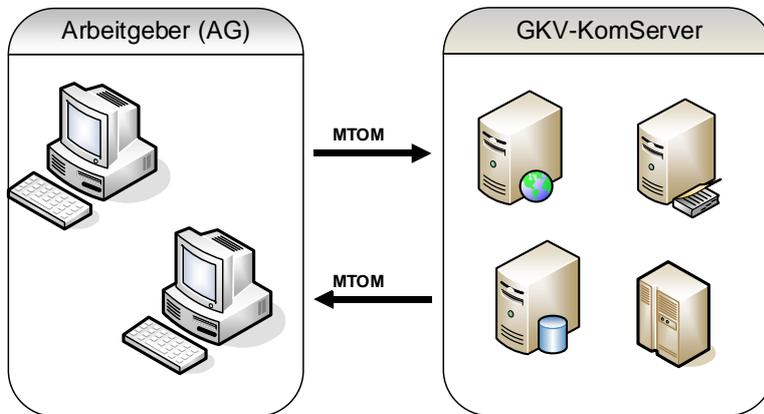

**WebService mit MTOM
an der AG-Schnittstelle des
GKV-Kommunikationsserver**

1 Einführung

Das vorliegende Dokument dient als Informationsgrundlage für die Kommunikation von Webservice via MTOM mit der Arbeitgeber-Schnittstelle (im folgenden AG-Schnittstelle) des GKV-Kommunikations-servers.



Der GKV-Kommunikationsserver folgt hierbei der W3C-Empfehlung für die Übertragung binärer Daten in Webservices via MTOM (*SOAP Message Transmission Optimization Mechanism*) und verwendet XOP (*XML-binary Optimized Packaging*) für die optimierte Übermittlung binärer Daten.

2 Technische Umsetzung

2.1 Allgemeines

MTOM (SOAP Message Transmission Optimization Mechanism) dient der Übertragung binärer Daten in Webservices. Der GKV-Kommunikationsserver stellt eine entsprechende Schnittstelle bereit. MTOM verwendet XML-binary Optimized Packaging (XOP) für die optimierte Übermittlung binärer Daten und ersetzt die sonst übliche Übertragung von Binärdaten mittels Base64-Kodierung in eXtra-XML-Dateien. Durch den Entfall des Base64 wird die zu übertragende Datenmenge um ca. 33 % verringert.

Der GKV-Kommunikationsserver ist über die neue Schnittstelle in der Lage, Meldungen bis maximal 16 MB (brutto) anzunehmen. Im Falle der zu übertragenden Rückmeldungen an den Arbeitgeber, erfolgt seitens des GKV-Kommunikationsserver weiterhin eine Begrenzung auf max. 2400 Packages oder 20MB (netto).

2.2 Webservice URL

Die produktive Webservice-Schnittstelle des GKV-Kommunikationsservers nimmt die MTOM-Dateien an den identischen, bereits bekannten geschäftsfallspezifischen URLs entgegen:

URL für Geschäftsfall „Meldung“:

<https://verarbeitung.gkv-kommunikationsserver.de/meldung/extra14.meldung>

URL für Geschäftsfälle „Statusanfrage“ und „RepeatResponse“:

<https://verarbeitung.gkv-kommunikationsserver.de/anfrage/extra14.anfrage>

URL für Geschäftsfall „Quittung“:

<https://verarbeitung.gkv-kommunikationsserver.de/quittung/extra14.quittung>

2.3 Webservice-WSDL und XSD-Schemadateien

2.3.1 WSDL Zugriff

Da die Webservice-URL nur über eine https-POST-Anforderung mit Client-Zertifikat-Authentifizierung erreichbar ist, wird auf die sonst übliche Bereitstellung der WSDL-Datei mit Hilfe einer http-GET-Anforderung und abschließender ?wsdl-Abfragezeichenfolge verzichtet. Die folgende Beispiel-URL ist somit **NICHT** gültig: <https://verarbeitung.gkv-kommunikationsserver.de/meldung/extra14.meldung?wsdl>

2.3.2 Hinweise zur Erzeugung eines Webservice-Clients

Die unter der obigen URL bereitgestellte WSDL-Datei bezieht sich auf den reinen eXtra 1.4 – Standard und referenziert daher intern die XSD-Schemadateien des *reinen* eXtra-Standards und *nicht* die Schemadateien der hiervon abgeleiteten KomServer-Profilierung!

Der reine eXtra-Standard bietet mehr Freiheitsgrade – bezogen auf die Erzeugung von XML-Dateien – als die eXtra-Profilierung für den GKV-Kommunikationsserver. Letztere bildet nur eine Untermenge des umfangreicheren eXtra-Standards ab!

Bei der Erstellung eines Webservice-Clients anhand der bereitgestellten WSDL-Datei müssen die hieraus generierten XML-Dateien somit nicht nur gegen die XSD-Schemadateien des reinen eXtra-Standard validieren, sondern auch gegen die für den GKV-Kommunikationsserver profilierten XSD-Schemadateien.

Die in der WSDL hinterlegten XSD-Schemadateien entsprechen dem weiter gefassten eXtra 1.4 - Standard und beziehen sich - geschäftsfallunabhängig - auf einen Request bzw. eine Response:

WSDL-Message	In WSDL-Datei referenzierte XSD-Schemadatei
executeRequest	xsd_extra/ eXtra-request-1.xsd
executeResponse	xsd_extra/ eXtra-response-1.xsd
ExtraFault	xsd_extra/ eXtra-error-1.xsd

Hinweis:

Die Responses des GKV-Kommunikationsserver sind grau hinterlegt, eine etwaige Fehlerantwort ist dunkelgrau hinterlegt.

Die von einem Webservice-Client erzeugten XML-Dateien müssen jedoch - je nach Geschäftsfall - zusätzlich gegen die folgenden XSD-Schemadateien der eXtra-Profilierung für den GKV-Kommunikationsserver validieren:

Geschäftsfall am GKV-Kommunikationsserver	Prüfung im GKV-Kommunikationsserver gegen folgende XSD-Schemadatei
Meldung-Request	xsd_mtom/ xsd_KomServer_1_request_senden_datenlieferungen_mtom.xsd
Meldung-Response	xsd_mtom/ xsd_KomServer_2_response_senden_datenlieferungen_mtom.xsd
Statusanfrage-Request	xsd_mtom/ xsd_KomServer_3_request_anfordern_rueckmeldungen_mtom.xsd
Statusanfrage-Response	xsd_mtom/ xsd_KomServer_4_response_abholen_rueckmeldungen_mtom.xsd
Quittung-Request	xsd_mtom/ xsd_KomServer_5_request_senden_empfangsquittungen_mtom.xsd
Quittung-Response	xsd_mtom/ xsd_KomServer_6_response_senden_empfangsquittungen_mtom.xsd
RepeatResponse-Request	xsd_mtom/ xsd_KomServer_7_request_repeatresponse_mtom.xsd
RepeatResponse-Response	xsd_mtom/ xsd_KomServer_8_response_repeatresponse_mtom.xsd
Fehler-Response	xsd_mtom/ xsd_KomServer_0_error_mtom.xsd

2.4 Einstellungen für den Http-Header des Request

Zum Versenden von Nachrichten mittels SOAP/MTOM an den GKV-Kommunikationsserver muss der http-Header „**Content-Type**“ den Wert

„**multipart/related; boundary=MIME-Multipart-Boundary**“

besitzen!

Hinweis:

Für die Kommunikation mit der Webanwendung findet üblicherweise der Wert „application/xml“ Verwendung.

2.5 Datenformat

Die Kommunikation mit dem GKV-Kommunikationsserver findet in einer profilierten (speziell für den GKV-Kommunikationsserver eingeschränkten) Variante des eXtra-Datenformats statt.

Das eXtra-Datenformat ist ein offener, frei verfügbarer Bundesstandard für den Datenaustausch.

Für die Verwendung von MTOM müssen die folgenden Punkte eingehalten werden:

1. Header: Der http-Header „Content-Type“ des http-Requests muss den Wert „multipart/related; boundary=MIME-Multipart-Boundary“ besitzen (für https-POST lautet der Wert „application/xml“) (→ siehe voriges Kapitel)
2. Datenformat: Die eXtra-XML-Dateien müssen in einen SOAP-Envelope eingebettet werden (siehe dieses Kapitel weiter unten)
3. Datenformat: Der üblicherweise BASE64-kodierte Inhalt muss als SOAP with Attachments unter Verwendung von MIME versendet werden (siehe dieses Kapitel weiter unten)

Im Folgenden werden die Punkte 2 und 3 näher erläutert:

Am Geschäftsfall Meldung-Request wird beispielhaft aufgezeigt, wie die eXtra-Nachricht vom Arbeitgeber an die Webservice-Schnittstelle des GKV-Kommunikationsserver gesendet werden muss, um dort korrekt verarbeitet zu werden.

Als Referenz (A) dient hierbei die etablierte Schnittstelle der Webanwendung.
Im Anschluß werden die Unterschiede (B) herausgearbeitet.

(A) Beispiel einer Nachricht ohne MTOM:

→ Kommunikation des AG mit der **Webanwendung** des GKV-Kommunikationsservers

In eXtra 1.4 ohne MTOM sieht der Aufbau bzw. die strukturelle Reihenfolge einer eXtra-XML Nachricht wie folgt aus:

1. Das eXtra-XML (ggf. mit eingebetteten, BASE64-kodierten Nutzdaten)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xreq:Transport
profile="http://www.extra-standard.de/profile/DEUEVGKV/1.4" version="1.4"
xmlns:xcpt="http://www.extra-standard.de/namespace/components/1"
xmlns:xplg="http://www.extra-standard.de/namespace/plugins/1"
xmlns:xreq="http://www.extra-standard.de/namespace/request/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.extra-standard.de/namespace/request/1
../xsd/xsd_KomServer_1_request_senden_datenlieferungen.xsd">
  <xreq:TransportHeader>
    <xcpt:Sender>
      [Hier kommt nun der mittlere Teil der Nachricht – gefolgt von diesem Ende:]
    <xreq:PackageBody>
      <xcpt:Data>
        <xcpt:Base64CharSequence>
          RG11cyBpc3QgZW1uZSBUZXR0emVpY2h1bmt1dHR1IGRpZSBmw7xyIGRpZSBCZW1z
          cG1lbGRhdGVpZW4gZGVzIEtvdW11bmlrYXRpb25zc2VydmVycyB1bnR3b3JmZW4g
          d3VyZGUGdW5kIG51ciB6dSBUZXR0end1Y2t1biBkaWVudC4NCKJpdHR1IGJlYWNo
          dGVuIFNpZSBkaWUgRG9rdW11bnRhdGlvbiBkZXMGs29tbXVuaWthdGlvbnNzZXJ2
          ZXJzLg==
        </xcpt:Base64CharSequence>
      </xcpt:Data>
    </xreq:PackageBody>
  </xreq:Package>
</xreq:TransportBody>
</xreq:Transport>
```

(B) Beispiel einer Nachricht mit MTOM:

→ Kommunikation des AG mit dem **WebService** des GKV-Kommunikationsservers

In eXtra 1.4 mit MTOM sieht der Aufbau bzw. die strukturelle Reihenfolge einer eXtra-XML Nachricht wie folgt aus:

1. MIME-Multipart-Boundary vom sogenannten "RootPart"
2. Der SOAP-Envelope um die eXtra-XML Nachricht
3. Das eXtra-XML (ggf. mit cid-Referenz auf die unkodierten, binären Nutzdaten)
4. Nutz- bzw. Binärdaten als SOAP-Attachment, eingeleitet mit jeweils einem MIME-Multipart-Boundary

Hinweis:

Referenziert werden etwaige, binäre Nutzdaten mit Hilfe eines Include-Tag:

```
<inc:Include href="cid:20161004155623000002999"/>
```

Die cid: referenziert somit die zugehörige Content-Id aus einem nach folgenden SOAP-Attachment.

Das vollständige MTOM-Dokument sieht somit wie folgt aus:

```
--MIME-Multipart-Boundary
Content-Type: application/xop+xml; charset=iso-8859-1
Content-Id: RootPart
Content-Transfer-Encoding: binary
```

```
<?xml version="1.0" encoding="iso-8859-1" ?>
```

```
<soapenv:Envelope
```

```
xmlns:xcpt="http://www.extra-standard.de/namespace/components/1"
```

```
xmlns:xplg="http://www.extra-standard.de/namespace/plugins/1"
```

```
xmlns:xreq="http://www.extra-standard.de/namespace/request/1"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
```

```
xmlns:inc="http://www.w3.org/2004/08/xop/include"
```

```
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
```

```
../xsd_mtom/xsd_KomServer_1_request_senden_datenerlieferung_mtom.xsd">
```

```
<soapenv:Header />
```

```
<soapenv:Body>
```

```
<xreq:Transport profile="http://www.extra-standard.de/profile/DEUEVGKV/1.4" version="1.4">
```

```
<xreq:TransportHeader>
```

```
<xcpt:Sender>
```

[Hier kommt nun der mittlere Teil der Nachricht – gefolgt von diesem Ende:]

```
<xreq:PackageBody>
```

```
<xcpt:Data>
```

```
<xcpt:Base64CharSequence>
```

```
<!-- Der Verweis auf den MIME-Content -->
```

```
<inc:Include href="cid:20161004155623000002999"/>
```

```
</xcpt:Base64CharSequence>
```

```
</xcpt:Data>
```

```
</xreq:PackageBody>
```

```
</xreq:Package>
```

```
</xreq:TransportBody>
```

```
</xreq:Transport>
```

```
</soapenv:Body>
```

```
</soapenv:Envelope>
```

```
--MIME-Multipart-Boundary
```

```
Content-Type: application/octet-stream
```

```
Content-Length: 20
```

```
Content-Id: 20161004155623000002999
```

```
Content-Transfer-Encoding: binary
```

```
Testdaten 001301 ÄÖÜ
```

```
--MIME-Multipart-Boundary--
```

Namespaces von
<xreq:Transport
umgezogen nach
<soapenv:Envelope

Verweis auf die binären Daten
(in eXtra ohne MTOM sind diese
BASE64 kodiert)

Die **grünen** Bereiche bezeichnen die MTOM-spezifischen Änderungen an der eXtra-XML-Nachricht (im Vergleich zu der vorigen Version ohne MTOM).

Hinweis:

Die bereitgestellte WSDL Datei enthält auch XML-Beispieldateien. Diese enthalten jedoch nicht die o.a. MIME-Multipart-Boundary um sie ohne weitere Anpassungen und mit allen gängigen XML-Tools gegen die zugehörigen Schemadateien zu validieren.