



Patrick Hofmann

e-dec Web Services

Coding-Examples

Projektname: e-dec
Version: 0.2
Datum: 2017-12-16

Status	in Arbeit	in Prüfung	genehmigt zur Nutzung
	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Beteiligter Personenkreis	
Autoren:	Patrick Hofmann
Genehmigung:	PL
Benützer/Anwender:	Projektgruppe, Zollkunden
zur Information/Kenntnis:	Projektgruppe

Änderungskontrolle, Prüfung, Genehmigung			
Wann	Version	Wer	Beschreibung
21.08.2017	0.1	pho	Grundversion
24.08.2017	0.2	pho	Response-Beispiele hinzugefügt

Inhaltsverzeichnis

1	Einleitung.....	3
1.1	Referenzen.....	3
2	Java.....	4
2.1	Voraussetzungen.....	4
2.2	Installation	4
2.3	Bilden der Applikation	4
2.4	Starten der Applikation	4
3	C#	6
4	Antwort-Fallbeispiele	7
4.1	OK-Fall (#status# = HTTP/1.1 200 OK).....	7
4.2	Fehler-Fall (#status# = HTTP/1.1 500 Internal Server Error)	7
4.3	Timeout	9
5	Tools	10
5.1	Monitoring.....	10

1 Einleitung

In diesem Dokument wird die Verwendung der e-dec Webservice-Schnittstelle anhand von Beispiel-Code erläutert. Die Dokumentation beschränkt sich dabei auf das reine Senden und Empfangen eines Requests resp. einer Response. Das Erstellen der XML-Dokumentstrukturen sind nicht Gegenstand dieser Dokumentation.

1.1 Referenzen

Die folgenden Quellen werden im Dokument referenziert oder haben als Grundlage gedient:

Ref	Titel	Version

2 Java

2.1 Voraussetzungen

Damit der Beispielcode lauffähig ist, müssen folgende Voraussetzungen erfüllt sein:

- Java 1.7 oder neuer installiert
- Apache ANT v1.9.7 oder neuer installiert
- Es muss ein gültiges Zertifikat vorliegen, das für die Verwendung des Webservices autorisiert ist.

2.2 Installation

Der Beispielcode wird in einem gepackten Archiv ausgeliefert (edec_code_examples.tar.gz) und kann (unter Windows z.B. mit 7zip, unter Unix mit „tar -xzf“) an einen beliebigen Ort ausgepackt werden. Im Verzeichnis befindet sich danach unter anderem ein README.txt mit einer Kurzanleitung.

Als erstes muss das Zertifikat in den Keystore gespeichert werden. Dies kann mit dem Befehl

```
keytool -import -trustcacerts -alias <MeinAlias> -file <meinzert.crt> -keystore  
src/main/resources/keystore.jks
```

erledigt werden. Das Passwort ist „changeit“. Jetzt muss noch das Passwort für den Private-Key im config.properties gesetzt werden ("identity.password"). Diese Datei befindet sich ebenfalls unter „**src/main/resources/**“.

Die „config-properties“ sollte man sich noch genau durch sehen und gegebenenfalls die nötigen Anpassungen vornehmen.

2.3 Bilden der Applikation

Das Bilden der Applikation geschieht mittels Ausführen von

```
ant
```

im entpackten Verzeichnis. Dadurch wird ein JAR erstellt, das alleinstehend lauffähig ist.

2.4 Starten der Applikation

Das Starten der Applikation kann mit

```
ant run
```

ausgeführt werden. Alternativ geht das auch über

Best Practice WS-Stacks

```
java -jar <Erzeugtes-JAR-Archiv>
```

3 C#

TBD

4 Antwort-Fallbeispiele

4.1 OK-Fall (#status# = HTTP/1.1 200 OK)

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <goodsDeclarationsResponse
      schemaVersion="4.0"
      xsi:schemaLocation="http://www.e-dec.ch/xml/schema/edecResponse/v4
http://www.ezv.admin.ch/pdf\_linker.php?doc=edecResponse\_v4\_0"
      xmlns=http://www.e-dec.ch/xml/schema/edecResponse/v4
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <goodsDeclarationAcceptance>
        <traderDeclarationNumber>Test-1503563577382</traderDeclarationNumber>
        <traderReference>Test</traderReference>
        <customsDeclarationNumber>17CHEI000010511823</customsDeclarationNumber>
        <customsDeclarationVersion>1</customsDeclarationVersion>
        <accessCode>oCQUioX0ihNKLO+g</accessCode>
        <acceptanceDate>2017-08-24</acceptanceDate>
        <acceptanceTime>10:32:56</acceptanceTime>
      </goodsDeclarationAcceptance>
    </goodsDeclarationsResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Der OK-Fall wird immer zurückgegeben, wenn mit einem Request weder fachl. noch techn. Fehler aufgetreten sind. **Es ist zu beachten, dass z.Z. bei Receipt-/Bordereau-Requests als Antwort immer ein HTTP/1.1 500 Internal Server Error zurück geliefert wird.** Das ist auf eine ältere Implementation zurück zu führen, die momentan nicht geändert werden kann.

4.2 Fehler-Fall (#status# = HTTP/1.1 500 Internal Server Error)

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring>see in detail</faultstring>
      <detail>
        <goodsDeclarationsResponse schemaVersion="4.0"
          xsi:schemaLocation="http://www.e-dec.ch/xml/schema/edecResponse/v4
http://www.ezv.admin.ch/pdf\_linker.php?doc=edecResponse\_v4\_0"
          xmlns=http://www.e-dec.ch/xml/schema/edecResponse/v4
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <goodsDeclarationRejection>
            <rejectionDate>2017-08-24</rejectionDate>
            <rejectionTime>10:28:51</rejectionTime>
            <errors>

```

Im Falle von Receipt/Bordereau ist der SOAP-header zusätzlich noch mit Security-Infos versehen:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
    <wsse:Security soapenv:mustUnderstand="1"
      xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wsu:Timestamp wsu:Id="Timestamp-291"
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <wsu:Created>2017-08-24T09:04:13.710Z</wsu:Created>
        <wsu:Expires>2017-08-24T09:09:13.710Z</wsu:Expires>
        </wsu:Timestamp>
        <wsse:BinarySecurityToken EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" wsu:Id="CertId-65099AFD6B7DC325071503565453710436" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">MIIHRjCCBS6gAwIBAgIQZSkzggqoo2xhVeM8Ilsdn2jANBgkqhkiG9w0BAQsFADB9MQswCQYDVQQGEwJDSDEOM
```

[illegible]

Best Practice WS-Stacks

```
<wsa:RelatesTo soapenv:mustUnderstand="0">urn:uuid:22242daa-72ac-4db2-98f0-3413d8e15662</wsa:RelatesTo>
</soapenv:Header>
<soapenv:Body wsu:Id="Id-1649983746" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <soapenv:Fault>
    <faultcode>soapenv:Client</faultcode>
    <faultstring>see in detail</faultstring>
    <detail>
      <receiptRequestResponse schemaLocation="http://www.edec.ch/xml/schema/edecReceiptResponse/v3
http://www.ezv.admin.ch/pdf_linker.php?doc=edecReceiptResponse_v_3_0" schemaVersion="3.0">
        <requestorTraderIdentification-
Number>CHE205933294</requestorTraderIdentificationNumber>
        <receiptRequestRejection>
          <rejectionDate>2017-08-24</rejectionDate>
          <rejectionTime>11:04:13</rejectionTime>
        <errors>
```

4.3 Timeout

Timeouts treten u.A. dann auf, wenn ein falsches Zertifikat verwendet wird. In diesem Fall müssen die lokalen Logs zu Rate gezogen werden.

5 Tools

5.1 Monitoring

Name	Link
TCP Monitor	https://tcpmon.dev.java.net/
WSMonitor	https://wsmonitor.dev.java.net/