



Paynet System – Document Web Services (DWS)

Handbuch für Entwickler

Revisionsnachweis

Version	Datum	Änderungen
1.0	11.05.2011	Erste Version
1.1	18.05.2011	Orthographie Fehler korrigiert
1.2	21.11.2012	Hinweis auf Limite von 500 Shipments ergänzt
1.3	17.01.2013	Neue URL
1.4	26.04.2013	Überarbeitung Wording
1.5	24.07.2015	Dokument-Typen ausführlicher spezifiziert
1.6	10.08.2015	Korrektur ShipmentPriority

Inhaltsverzeichnis

1.	Einleitung	3
1.1	Authentisierung.....	3
1.2	Voraussetzungen C#	3
1.3	Voraussetzungen Java	3
1.3.1	Hilfreiche Tools	4
1.4	Begriffserklärung Shipment	4
1.5	Darstellungskonventionen	4
1.6	Beispieldaten	4
1.7	Disclaimer	4
2.	Sequenzdiagramm für Sender.....	5
3.	Sequenzdiagramm für Empfänger.....	6
4.	Generelle Hinweise.....	7
4.1	Aufbau der beschreibenden Tabellen.....	7
5.	takeShipment.....	8
5.1	Definition Request.....	8
5.2	Definition Response.....	9
6.	getShipmentList	10
6.1	Definition Request.....	10
6.2	Definition Response.....	12
7.	getShipmentContent	14
7.1	Definition Request.....	14
8.	confirmShipment.....	15
8.1	Definition Request.....	15
8.2	Definition Response.....	15
9.	ping	16
9.1	Definition Request.....	16
9.2	Definition Response.....	16
10.	Fehlerverhalten	17
10.1	Liste der Fehlercodes	17
11.	Web Service URL	18
11.1	Test-System.....	18
11.2	Produktions-System.....	18
	Web Server Zertifikate.....	19

1. Einleitung

Für den Dokumentenaustausch mit dem Paynet System bietet SIX Paynet AG unterschiedliche Schnittstellen an, um den verschiedenen Bedürfnissen von Sendern und Empfängern gerecht zu werden. Eine wichtige Schnittstelle sind die Document Web Services des Paynet Systems (nachfolgend DWS genannt).

Dieses Handbuch bringt Ihnen die Funktionsweise und die Anwendung des DWS näher und erklärt, wie Sie einen Web Service Client erstellen. Zum Handbuch haben Sie auch Demo-Projekte in C# und Java erhalten, aber selbstverständlich können Sie Ihren Client in der Programmiersprache Ihrer Wahl implementieren. Vorliegendes Handbuch setzt Kenntnisse in den Sprachen (SOAP, XML, Java, C#) voraus.

1.1 Authentisierung



Die Authentisierung erfolgt bei jedem Aufruf entweder mittels Client-Zertifikat, oder Benutzernamen und Passwort in der SOAP Meldung. Bitte kontaktieren Sie uns, damit wir Ihnen Ihre Zugangsdaten zustellen können.

1.2 Voraussetzungen C#

Voraussetzungen für eine Implementation mit dem .NET Framework ist die Installation des .NET SDK. Die Empfehlung ist jeweils das neuste SDK einzusetzen (zum Zeitpunkt dieses Dokuments liegt das .NET Framework in der Version 4.0 vor). Wird Visual Studio .NET (2010, 2012) eingesetzt, sind alle notwendigen Tools für eine Implementierung vorhanden.

1.3 Voraussetzungen Java

Wenn Sie einen Client in Java entwickeln möchten, benötigen Sie ein JDK ab der Version 1.4 oder höher. Unsere Beispiel basiert auf dem Java SDK 1.6.0_13. Zusätzlich wurde das Axis Framework in der Version 1.4 eingesetzt. Als IDE wurde Netbeans 7.0 verwendet.

Unter Java gibt es verschiedene Frameworks, welche es Ihnen ermöglichen, einen Web Service Client zu implementieren. Welchen Framework Sie wählen steht Ihnen frei, vorausgesetzt, er deckt sich mit den Standards.

1.3.1 Hilfreiche Tools

Folgende Tools sind hilfreich:



- soapUI (<http://www.soapui.org>). Ist ein Open Source Tool für das Testen von Service Orientierten Architekturen.
- Netbeans (www.netbeans.com). Ist eine Entwicklungsumgebung für Java, oder
- Eclipse (<http://www.eclipse.org>), ebenfalls eine Entwicklungsumgebung für Java.
- Commons CLI für das vereinfachte Parsen von Commandline Argumenten für Java Programme (<http://commons.apache.org>)
- Axis 1.4 (<http://axis.apache.org/axis/java/releases.html>), SOAP Implementierung für Java Programme.

1.4 Begriffserklärung Shipment

Unter einem Shipment verstehen wir eine Datei, welche zwischen dem Paynet System und Ihrem Client ausgetauscht wird.

1.5 Darstellungskonventionen

Folgende Symbole werden in diesem Benutzerhandbuch verwendet:

Symbol	Bedeutung
	<i>Interessante Zusatzinformationen</i> Markiert Hinweise zur Vertiefung eines Themas.
	<i>Wichtige Zusatzinformationen</i> Markiert Anweisungen, die zwingend beachtet werden müssen.

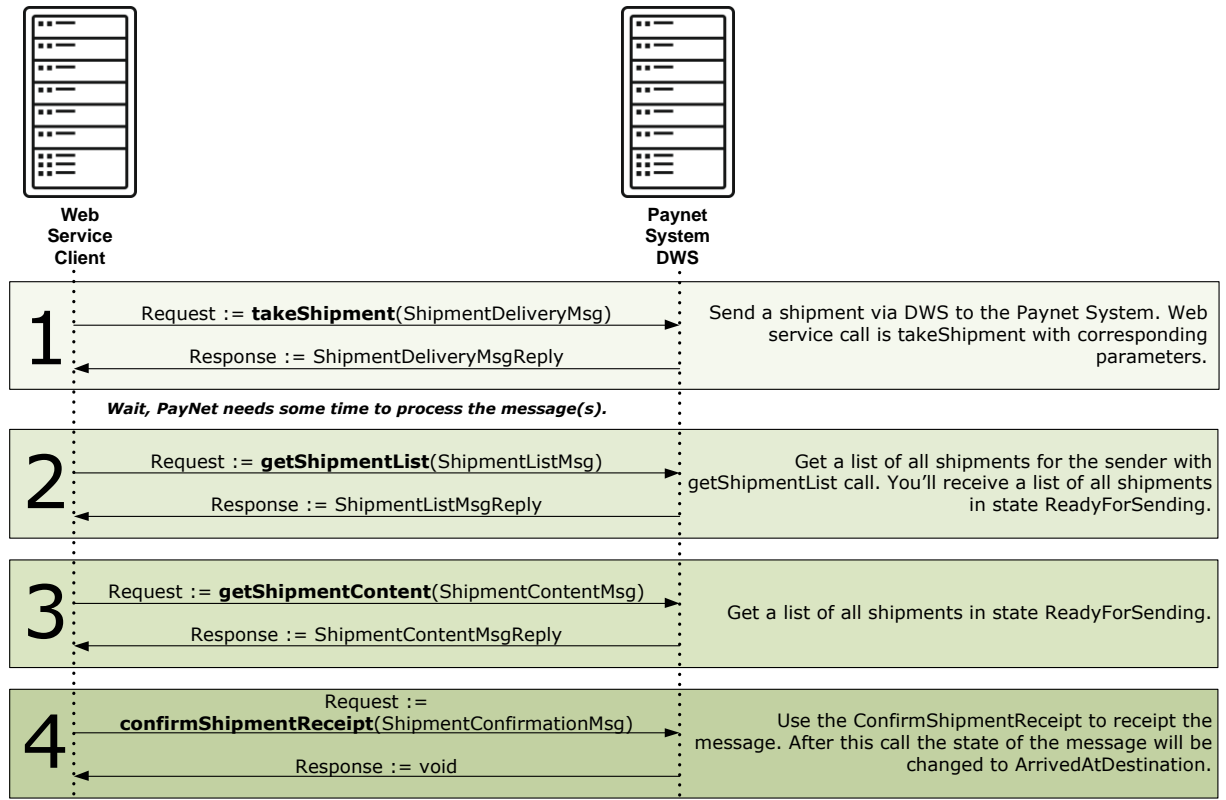
1.6 Beispieldaten

Alle in diesem Benutzerhandbuch verwendeten Beispieldaten (Firmenbezeichnungen, Dokumente, Beziehungen usw.) sind fiktiv.

1.7 Disclaimer

Dieses Benutzerhandbuch ist mit grösster Sorgfalt erstellt worden, doch können Fehler und Ungenauigkeiten nicht vollständig ausgeschlossen werden. SIX Paynet kann für Fehler in dieser Dokumentation und deren Folgen weder eine juristische Verantwortung noch irgendwelche Haftung übernehmen. Die in diesem Benutzerhandbuch enthaltenen Angaben können jederzeit ohne vorherige Benachrichtigung geändert werden.

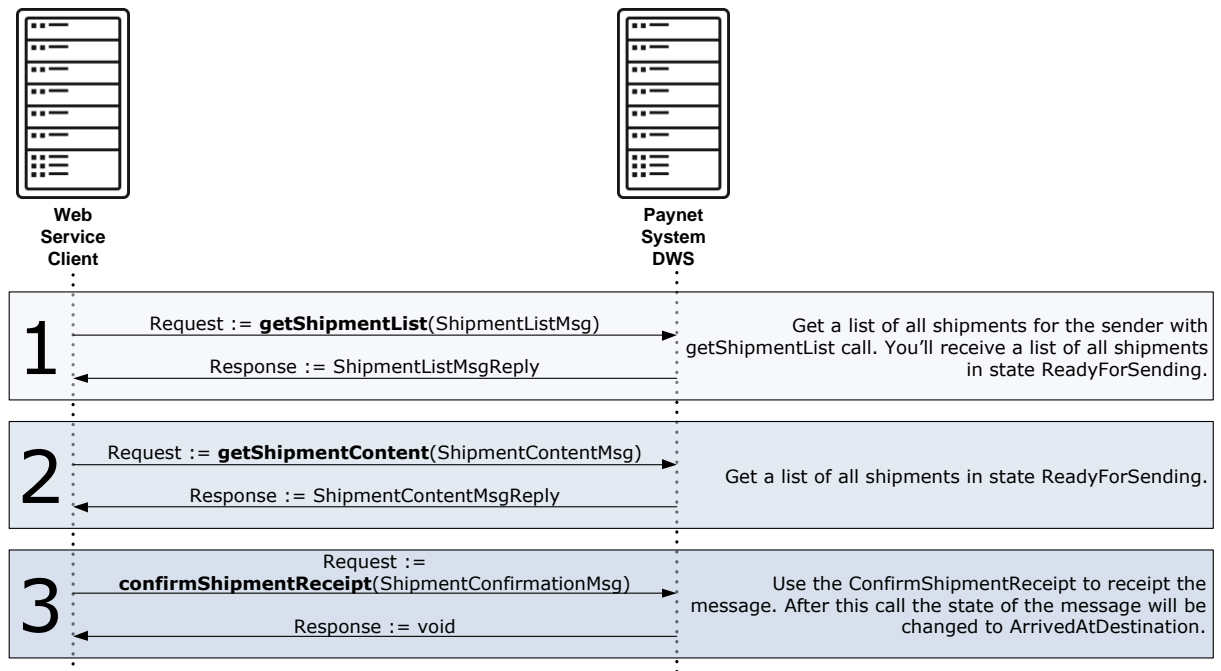
2. Sequenzdiagramm für Sender



Als Sender müssen Sie die folgenden vier Methoden implementieren:

- **takeShipment:** upload eines Shipments ans Paynet System. Wird diese Methode aufgerufen, wird das Shipment vom Paynet System entgegengenommen und für die weitere Verarbeitung gespeichert.
- **getShipmentList:** abrufen einer Liste von Shipments, welche für den Sender bereitstehen. Nachdem das Paynet System das zuvor eingelieferte Shipment verarbeitet hat, werden z.B. Quittungen für den Sender bereitgestellt. Alle bereitstehenden Meldungen werden dem Sender in dieser Liste präsentiert.
- **getShipmentContent:** der Sender durchläuft die Liste, die er zuvor mittels getShipmentList abgerufen hat, lädt nun jedes einzelne Shipment separat herunter, speichert und verarbeitet den Inhalt.
- **confirmShipmentReceipt:** nachdem das Shipment erfolgreich empfangen und verarbeitet wurde, muss der Meldungsempfänger den Erhalt der Meldung mittels dieser Methode bestätigen.

3. Sequenzdiagramm für Empfänger



Als Empfänger müssen Sie die folgenden drei Methoden implementieren:

- **getShipmentList:** abrufen einer Liste von Shipments, welche für den Empfänger bereitstehen. Alle bereitstehenden Meldungen werden in dieser Liste präsentiert.
- **getShipmentContent:** der Empfänger durchläuft die Liste, die er zuvor mittels getShipmentList abgerufen hat, lädt nun jedes einzelne Shipment separat herunter und speichert den Inhalt für die weitere Verarbeitung.
- **confirmShipmentReceipt:** nachdem das Shipment erfolgreich empfangen und verarbeitet wurde, muss der Meldungsempfänger den Erhalt der Meldung mittels dieser Methode bestätigen.

4. Generelle Hinweise

Jeder Aufruf (Request) eines Web Services verlangt als Eingabeparameter eine XML-Meldung und gibt in der Antwort (Response) eine XML-Meldung zurück.

4.1 Aufbau der beschreibenden Tabellen

Die Requests und Responses sind in Tabellen in den folgenden Kapiteln definiert. Die Tabellen enthalten folgende Rubriken:

- XML-Tag
- Format
- Typ
- Beschreibung

Unter XML-Tag werden die Aggregate, Elemente und Attribute aufgelistet. Aggregate werden immer mit Tag und End-Tag gekennzeichnet. Elemente werden einzeln gekennzeichnet. Attribute werden ohne Tags beschrieben. In der Spalte Typ wird die Unterscheidung nochmals verdeutlicht:

- A = Aggregat
- E = Element
- Attr. = Attribut

Ebenfalls im Typ wird die zulässige Menge des Tags erwähnt:

- 1 = nur ein Mal
- 0..1 = max. ein Mal (XML-Syntax: ?)
- 0..n = unbeschränkt (XML-Syntax: *)
- 1..n = beliebig viele, aber eines muss mind. kommen (XML-Syntax: +)

Weiterführende Informationen zu den Formaten finden Sie auch in den offiziellen Angaben vom www.w3.org.

5. takeShipment

Mit diesem Aufruf wird ein Shipment ans Paynet System gesendet.

5.1 Definition Request

XML-Tag	Format	Typ	Beschreibung
<ShipmentDeliveryMsg>	-	A [1]	-
..<Authorization>	-	A [0..1]	Wird nicht benötigt, wenn Sie über ein Client Zertifikat verfügen.
....<UserName>	String	E [1]	Ihr Benutzername wird von SIX Paynet vergeben.
....<Password>	String	E [1]	Ihr Passwort wird von SIX Paynet vergeben.
..<Mandant>	-	A [0..1]	Bitte lassen Sie dieses Aggregat weg!
....<ExternalID>	String	E [1]	Identifikation
.... <ExternalIDType>	String	E [1]	Einer der folgenden Werte: APPSYSNAME, BC, BCN, BIC, BLZ, BPID, DN, EAN, FTPLUSID, IBAN, MACID, MIME, PATHNAME, PID, SMSNUMBER, STATIONID, UNKNOWN, USERNAME, X400ADDRESS, XISENDERID, XIRECEIVERID
..<ProcessingDate>	DateTime	E [0..1]	Gewünschtes Verarbeitungsdatum, wenn nicht angegeben, wird das Shipment so rasch als möglich verarbeitet.
..<ShipmentPriority>	Integer	E [0..1]	Ein Wert zwischen 1..9 (Default ist 5).
..<Content>	Byte[]	E [1]	Der Inhalt der Datei, welche hochgeladen wird.
....<encoding>	String	Attr. [0..1]	Angabe des Encodings (z.B. UTF-8, ISO-8859-1). Kann auch ein leerer String sein.
..<Monitoring>	-	A [0..1]	Bitte lassen Sie dieses Aggregat weg!
....<MessageID>	String	E [1]	Referenz auf eine Meldung
....<MonitorCaseReference>	String	E [1]	Referenz auf ein Monitoringobjekt

Beispiel minimaler SOAP Request (ohne SOAP-Header und SOAP-Body):

```
<ShipmentDeliveryMsg>
  <Authorization>
    <UserName>PAYNET-USERNAME</UserName>
    <Password>PAYNET-PASSWORD</Password>
  </Authorization>
  <Content encoding="">SU5IQUxULURFUilEQVRFSQ==</Content>
</ShipmentDeliveryMsg>
```

5.2 Definition Response

XML-Tag	Format	Typ	Beschreibung
<ShipmentDeliveryMsgReply>-		A [1]	-
..<ShipmentID>	String	E [1]	Vom Paynet System erzeugte Identifikation des Shipments.

Beispiel SOAP-Response (ohne SOAP-Header und SOAP-Body):

```
<ShipmentDeliveryMsgReply>
  <ShipmentID>SC-99999999999999999999</ShipmentID>
</ShipmentDeliveryMsgReply>
```

6. getShipmentList

Der Aufrufer fordert eine Liste von Shipments an. Dabei können verschiedene Filterkriterien verwendet werden wie Datumsbereich, Status und Verarbeitungspriorität.

6.1 Definition Request

XML-Tag	Format	Typ	Beschreibung
<ShipmentListMsg>	-	A [1]	-
..<fromEntry>	Integer Unsigned	Attr. [0..1]	Positionsnummer, ab der Shipments geholt werden sollen (Default ist 1).
..<maxEntries>	Integer Unsigned	Attr. [0..1]	Maximale Anzahl Shipments, die in der Liste aufgeführt werden (Default ist 100).
..<Authorization>	-	A [0..1]	Wird nicht benötigt, wenn Sie über ein Client Zertifikat verfügen.
....<UserName>	String	E [1]	Ihr Benutzername wird von SIX Paynet vergeben.
....<Password>	String	E [1]	Ihr Passwort wird von SIX Paynet vergeben.
..<FromDate>	DateTime	E [0..1]	Start des Datumbereichs des Verarbeitungsdatums. Format: yyyy-mm-ddThh:ssZ
..<ToDate>	DateTime	E [0..1]	Ende des Datumbereichs des Verarbeitungsdatums. Format: yyyy-mm-ddThh:ssZ
..<DocumentIdentifier>	-	A [0..1]	Bitte lassen Sie dieses Aggregat weg!
....<Name>	String	E [0..1]	Name des Bezeichners.
....<Type>	String	E [0..1]	Typ des Bezeichners. Eine Liste Möglicher Werte ist: <ul style="list-style-type: none"> • ArchiveMetaData • Bansta • BBXCustomerRegistrationMessage • BBXCustomerRegistrationMessageResponse • Contrl • DocumentSet • DocumentSetReply • EDI Confirmation • Invoice • TransactionJournalReceiver • TransactionJournalSender
....<Format>	String	E [0..1]	Format des Bezeichners. Eine Liste möglicher Werte: <ul style="list-style-type: none"> • XML • MIME • EDIFACT.
....<Version>	String	E [0..1]	Version des Bezeichners, z.B. InternalD96A.
....<Encoding>	String	E [0..1]	Codierung, z.B. UTF-8, ISO-8859-1.

....<Extension>	String	E [0..1]	Datei-Erweiterung.
....<Category>	String	E [0..1]	Mögliche Werte für Kategorie sind Backpack, EdifactStructured, ISF, Unknown, Container, Structured, VerifyLog.
..<ShipmentStates>	-	A [0..1]	-
....<ShipmentState>	String	E [1..3]	Status in welchem sich das Shipment befindet. Mögliche Werte sind ReadyForSending, Submitted, ArrivedAtDestination.
..<FromShipmentPriority>	Integer	E [0..1]	Startbereich der Verarbeitungspriorität. Wert zwischen 1..9.
..<ToShipmentPriority>	Integer	E [0..1]	Endbereich der Verarbeitungspriorität. Wert zwischen 1..9. Sinnvollerweise grösser als der Wert von FromShipmentPriority.

Beispiel minimaler SOAP Request (ohne SOAP-Header und SOAP-Body):

```
<ShipmentListMsg fromEntry="1" maxEntries="100">
  <Authorization>
    <UserName>PAYNET-USERNAME</UserName>
    <Password>PAYNET-PASSWORD</Password>
  </Authorization>
</ShipmentListMsg>
```

6.2 Definition Response

XML-Tag	Format	Typ	Beschreibung
<ShipmentListMsgReply>	-	A [1]	-
..<entriesFound>	Integer Unsigned	Attr. [1]	Anzahl Einträge in der Liste. Es werden maximal 500 Einträge zurückgegeben. Existieren mehr als 500 Shipments im Paynet System, müssen die Shipments in Teilen zu maximal 500 Shipments abgearbeitet werden.
..<Shipments>	-	A [0..500]	-
....<ShipmentID>	String	E [1]	Vom Paynet System erzeugte Identifikation des Shipments.
....<DocumentIdentifier>	-	A [0..1]	-
.....<Name>	String	E [0..1]	Name des Bezeichners
.....<Type>	String	E [0..1]	Typ des Bezeichners. Mögliche Werte sind Bansta, Contrl, DocumentSet.
.....<Format>	String	E [0..1]	Format des Bezeichners. Mögliche Werte sind XML, MIME, EDIFACT.
.....<Version>	String	E [0..1]	Version des Bezeichners, z.B. InternalID96A
.....<Encoding>	String	E [0..1]	Codierung, z.B. UTF-8, ISO-8859-1.
.....<Extension>	String	E [0..1]	Datei-Erweiterung.
.....<Category>	String	E [0..1]	Mögliche Werte für Kategorie sind Backpack, EdifactStructured, ISF, Unknown, Container, Structured, VerifyLog.
....<SignatureIdentifier>	-	A [0..1]	-
.....<Name>	String	E [0..1]	Name des Bezeichners.
.....<Type>	String	E [0..1]	Typ des Bezeichners. Mögliche Werte sind Bansta, Contrl, DocumentSet.
.....<Extension>	String	E [0..1]	Datei-Erweiterung.
....<CreationDate>	DateTime	E [1]	Erstellungsdatum des Shipments.
....<ShipmentState>	String	E [1]	Status in welchem sich das Shipment befindet. Möglich ist einer der Werte ReadyForSending, Submitted, ArrivedAtDestination.
....<SenderID>	String	E [1]	Identifikation des Senders.
....<ReceiverID>	String	E [1]	Identifikation des Empfängers.

7. getShipmentContent

Der Aufrufer fordert ein Shipment an.

7.1 Definition Request

XML-Tag	Format	Typ	Beschreibung
<ShipmentContentMsg>	-	A [1]	-
..<Authorization>	-	A [0..1]	Wird nicht benötigt, wenn Sie über ein Client Zertifikat verfügen.
....<UserName>	String	E [1]	Ihr Benutzername wird von SIX Paynet vergeben.
....<Password>	String	E [1]	Ihr Passwort wird von SIX Paynet vergeben.
..<ShipmentID>	String	E [1]	Vom Paynet System erzeugte Identifikation des Shipments.

Beispiel minimaler SOAP Request (ohne SOAP-Header und SOAP-Body):

```
<ShipmentContentMsg>
  <Authorization>
    <UserName>PAYNET-USERNAME</UserName>
    <Password>PAYNET-PASSWORD</Password>
  </Authorization>
  <ShipmentID>SC-99999999999999999999</ShipmentID>
</ShipmentContentMsg>
```

Definition Response

XML-Tag	Format	Typ	Beschreibung
<ShipmentContentMsgReply>	-	A [1]	-
..<Content>	Byte[]	E [1]	Der Inhalt der Datei, welche hochgeladen wird.
....<encoding>	String	Attr. [0..1]	Angabe des Encodings (z.B. UTF-8, ISO-8859-1). Kann auch ein leerer String sein.
..<ShipmentPriority>	Integer	E [0..1]	Ein Wert zwischen 1..9 (Default ist 5).

Beispiel SOAP-Response (ohne SOAP-Header und SOAP-Body):

```
<ShipmentContentMsgReply>
  <Content>MIKazwYJKoZIhvcNAQcCoIKawDCCmrwCAQE...
  .
  .
  ....qNO6MQ=</Content>
  <ShipmentPriority>5</ShipmentPriority>
</ShipmentContentMsgReply>
```

8. confirmShipment

Der Aufrufer bestätigt den Empfang eines Shipments.

8.1 Definition Request

XML-Tag	Format	Typ	Beschreibung
<ShipmentConfirmationMsg>	-	A [1]	-
..<Authorization>	-	A [0..1]	Wird nicht benötigt, wenn Sie über ein Client Zertifikat verfügen.
....<UserName>	String	E [1]	Ihr Benutzername wird von SIX Paynet vergeben.
....<Password>	String	E [1]	Ihr Passwort wird von SIX Paynet vergeben.
..<ShipmentID>	String	E [1]	Vom Paynet System erzeugte Identifikation des Shipments.

Beispiel minimaler SOAP Request (ohne SOAP-Header und SOAP-Body):

```
< ShipmentConfirmationMsg>
  <Authorization>
    <UserName>PAYNET-USERNAME</UserName>
    <Password>PAYNET-PASSWORD</Password>
  </Authorization>
  <ShipmentID>SC-999999999999999999</ShipmentID>
</ ShipmentConfirmationMsg>
```

Der Status des Shipments wechselt nach diesem Aufruf von Submitted auf ArrivedAtDestination.

8.2 Definition Response

Der Web Service antwortet mit einer leeren Response. Der Returnwert der Funktion ist void.

9. ping

Um die Verfügbarkeit des Dienstes zu prüfen, kann der Aufrufer die ping-Methode aufrufen. Diese Methode benötigt keine Authentisierung. Der Aufrufer sendet dabei eine beliebige Zeichenfolgen an den DWS, welcher die Zeichenfolge wieder an den Aufrufer zurücksendet.

9.1 Definition Request

XML-Tag	Format	Typ	Beschreibung
<PingMsg>	-	A [1]	-
..<ClientData>	String	E [0..1]	Beliebige Zeichenfolge

Beispiel minimaler SOAP Request (ohne SOAP-Header und SOAP-Body):

```
< PingMsg>
  <ClientData>Hello PayNet!</ClientData>
</ PingMsg>
```

9.2 Definition Response

XML-Tag	Format	Typ	Beschreibung
<PingMsgReply>	-	A [1]	-
..<State>	String	E [1]	Status immer „OK“.
..<TimeStamp>	DateTime	E [1]	Aktuelle Zeit des Servers.
..<Version>	String	E [1]	Aktuelle Version des DWS, sofern freigegeben.
..<ClientData>	String	E [0..1]	Gesendete Zeichenfolge.

Beispiel SOAP-Response (ohne SOAP-Header und SOAP-Body):

```
< PingMsgReply>
  <State>OK</State>
  <TimeStamp>2011-05-05T11:09:11.629Z</TimeStamp>
  <Version>Not defined</Version>
  <ClientData>Hello PayNet!</ClientData>
</ PingMsgReply>
```

10. Fehlerverhalten

Wenn eine Methode nicht erfolgreich ausgeführt werden kann, erhält der Aufrufer eine Fehlermeldung vom Typ FaultMsg. Die Response hat in diesem Fall folgenden Aufbau:

XML-Tag	Format	Typ	Beschreibung
<FaultMsg>	-	A [1]	-
..<ErrorCode>	String	E [1]	4-stellige Zahl.
..<ErrorMessage>	String	E [0..1]	Fehlertext

Beispiel SOAP-Response (ohne SOAP-Header und SOAP-Body):

```
<Fault>
  <faultcode> env:Server</faultcode>
  <faultstring dxpp::FaultMsg</faultstring>
  <detail>
    <FaultMsg>
      <ErrorCode>3130</ErrorCode>
      <ErrorMessage>"urn:#getShipmentContent": Shipment [SC-
000111100000000000xxxx] für Request-Sender [PAYNET-USERNAME] wurde nicht gefunden!
(LogEventID:999999)</ErrorMessage>
    </FaultMsg>
  </detail>
</Fault>
```

10.1 Liste der Fehlercodes

Error	Beschreibung (Englisch)
0	Request successfully processed
1000	Unknown Error, maybe used faulty password
1500	Technical sender not identified
1501	Technical sender not authorized
3000	Unknown mandant
3001	Technical sender not authorized for mandant
3002	Error inserting Intermediate Document (only XI)
3100	ShipmentID not unique
3110	Technical sender not authorized to confirm shipment
3130	Unknown ShipmentID
3140	Shipment has not the required state for confirming (allowed: ReadyForSending, InTransmission, Submitted or ArrivedAtDestination)
3150	Empty shipment content
3200	Unknown SenderID in DocumentSetMsg
3201	Technical sender not authorized for SenderID in DocumentSetMsg

11. Web Service URL

Bitte beachten Sie, dass die Kommunikation zwischen dem Transfer Client und dem Paynet System über einen gesicherten Kanal erfolgt. Deshalb ist es wichtig, dass sie dem Herausgeber des Server-Zertifikates vertrauen und die entsprechende Root CA und allfällige intermediate CA's in ihrem Truststore erfassen. Die Web Services des Paynet Systems sind unter den folgenden URL erreichbar.

11.1 Test-System

Wenn die Client-Identifikation mittels Benutzername/Passwort erfolgt, ist folgende URL zu benutzen:

- <https://dws-test.paynet.ch/DWS/DWS>

Wenn die Client-Identifizierung mittels Client-Zertifikat erfolgt, ist die folgende URL zu benutzen:

- <https://dws-test-cert.paynet.ch/DWS/DWS>

11.2 Produktions-System

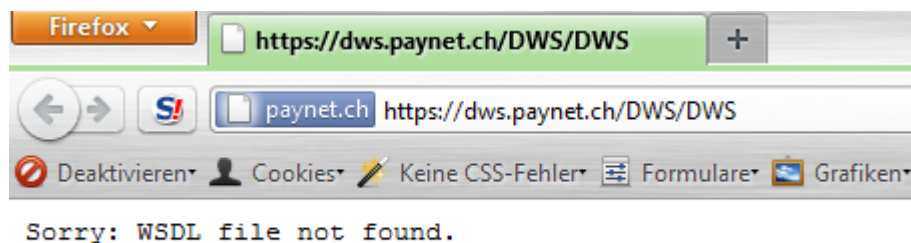
Wenn die Client-Identifikation mittels Benutzername/Passwort erfolgt, ist folgende URL zu benutzen:

- <https://dws.paynet.ch/DWS/DWS>

Wenn die Client-Identifizierung mittels Client-Zertifikat erfolgt, ist die folgende URL zu benutzen:

- <https://dws-cert.paynet.ch/DWS/DWS>

Hinweis: die Web Server sind so konfiguriert, dass keine WSDL Dokumente ausgeliefert werden, daher werden Sie folgende Antwort vom Server erhalten, wenn Sie die URL aufrufen:



Web Server Zertifikate

Die Kommunikation zwischen Client und Server wird durch den Einsatz von Web Server Zertifikate sichergestellt. D.h. der Datenstrom zwischen Client und Server ist verschlüsselt.

Die Server Zertifikate, welche zum Einsatz kommen werden von TC TrustCenter GmbH herausgegeben (<http://www.trustcenter.de>). Es handelt sich um mehrstufige Zertifikate. Der Client muss der ganzen Kette von Herausgebern vertrauen, damit die Kommunikation erfolgreich aufgebaut werden kann. Andernfalls werden Sie eine entsprechende Exception erhalten, mit dem Hinweis, dass die SSL Verbindung nicht aufgebaut werden kann.

Kette der Zertifikats-Herausgeber, denen der Client trauen muss (Stand Mai 2011):

TC TrustCenter Class 3 CA II	Root CA	Dieser Instanz muss der Client trauen...
TC TrustCenter Class 3-II L1 CA IV	Intermediate CA	...und dieser Instanz muss der Client trauen...
dws.paynet.ch	Server-Zertifikat (Produktion)	...dann ist dieses Server Zertifikat vertrauenswürdig...
dws-test.paynet.ch	Server Zertifikat (Test)	...,und dieses Server Zertifikat auch.

Weitere Informationen zu diesem Thema finden Sie auch in den Demo-Projekten.

Abkürzungen/Glossar

Abkürzung	Beschreibung
E-Rechnung	Elektronische Rechnung
EDI	Electronic Data Interchange
Paynet System	System von SIX Paynet für die Abwicklung elektronischer Meldungen (z.B. MWST-konforme elektronische Rechnungen)
Shipment	Ein Shipment ist ein Objekt, welches zwischen SIX Paynet AG und dem Paynet Teilnehmer übermittelt wird.
PKCS#7 envelopping	PKCS#7 envelopping ist ein international standardisiertes Signaturverfahren, welches die zu sichernden Daten ähnlich einem Couvert umschliesst. Dieses Verfahren eignet sich sehr gut, um beliebige Daten zu signieren.
Transport-Signatur	Die Transport-Signatur dient der Sicherstellung der Integrität während der Übermittlung. Sie darf nicht mit der MWST-relevanten Beleg-Signatur verwechselt werden.
SOAP	Simple Object Access Protocol ist ein Netzwerkprotokoll, mit dessen Hilfe Daten zwischen Systemen ausgetauscht und Remote Procedure Calls durchgeführt werden können.
XML	Extensible Markup Language ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdaten. XML wird u. a. für den plattform- und implementationsunabhängigen Austausch von Daten zwischen Computersystemen eingesetzt
Java	Java ist eine objektorientierte Programmiersprache und eingetragenes Warenzeichen des Unternehmens Sun Microsystems, welches Ende Januar 2010 von Oracle übernommen wurde. Die Programmiersprache ist ein Bestandteil der Java-Technologie.
C#	C# ist eine vom Softwarehersteller Microsoft im Rahmen seiner .NET-Strategie entwickelte Programmiersprache. C# ist bei ECMA und ISO als Standard registriert.
DWS	Document Web Service ein von SIX Paynet AG zur Verfügung gestellter Web Service, welcher die Kommunikation zwischen dem Paynet System und den Paynet Teilnehmern ermöglicht.

Kontaktinformationen

SIX Paynet AG
Hardturmstrasse 201
Postfach
8021 Zürich
Schweiz

www.six-paynet.com

Informationen und Beratung

058 399 9511
paynet-info@six-group.com

Support-Anfragen

058 399 9577
paynet-support@ six-group.com