



Einleitung

Die hier beschriebene Schnittstelle basiert auf dem SOAP Protokoll (Simple Object Access Protokoll) und wird als Web Service bereit gestellt. Hierbei werden XML-formatierte Anfragen an den Rechnung.de Server gesendet um Rechnungsbelege über das Trust-Center D-Trust (Bundesdruckerei) qualifiziert signiert zu versenden oder den Rechnungsbeleg per Post zu senden.

Um die Schnittstelle möglichst einfach zu halten, haben wir die Aufrufe stark reduziert. Mit diesen Aufrufen lassen sich alle Services sehr schnell und effizient aus allen möglichen Arten von Office- oder Webanwendungen bedienen. Die Aufrufe 4-12 sind Services, welche Sie nutzen können um die Anwendung komfortabel zu machen.

1. Register (Registrieren von Benutzern/Änderung Benutzerdaten)
2. Send (Senden von Rechnungsbelegen (digital/Postbeleg))
3. Status (Abfrage des Verarbeitungs-Status, Protokoll-Abfrage)
4. KundenName (Kundenname bereitstellen)
5. KundenDaten (Kundendaten bereitstellen)

Nachfolgend wird der Aufbau der einzelnen Abfragetypen und deren Rückantwort beschrieben.

Detaillierte XML Aufrufdokumentationen (Aufbau der WSDL-Strukturen) erhalten Sie unter folgenden Adressen:

Borland Delphi 7/8: <http://ccs.rechnung.de/xmlws3.exe?intf=IXMLWS>

Microsoft .NET, Microsoft Soap-tools, PHP4:
<http://ccs.rechnung.de/xmlws3.exe?intf=IXMLWS2>

Vereinfachtes Interface (z.B. Microsoft Foxpro):
<http://ccs.rechnung.de/xmlws3.exe?intf=IXMLWS3>

Anwendungsbeispiele für Visual-Basic, VB.net, C#.net und PHP4 finden Sie auf www.Rechnung.de / Download



Register

Int **Register**(string **AccountNo**,
string **Password** ,
TStringDynArray **RegData**)

Die Benutzerdaten werden in einem String-Array übergeben. Bei einer Neuanmeldung bleiben AccountNo und Password leer. Diese sind nach erfolgreicher Registrierung serverseitig mit der vergebenen Kunden-Nummer und Passwort gefüllt. Bei einer Datenänderung muss AccountNo und Password mit übergeben werden.

Die Registerdaten werden mit Token übergeben, Beispiel „ANREDE=Herr“:

TOKEN	Feldbeschreibung	Format	Hinweis
ACCOUNTNO=	Kunden-Nummer	999999	Leer übergeben bei Neuregistrierung Mit @<Nummer> übergeben bei Kartenregistrierung
ANREDE=	Anrede	X(15)	z.B. „Herr“ oder „Firma“
NAME1=	Kunden-Name	X(30)	Vorname, Nachname
NAME2=	Zusatz	X(30)	Zusatz Name
STRASSE=	Strasse	X(30)	
PLZORT=	Land/PLZ/Ort	X(40)	
TELEFON=	Telefon-1	X(20)	
ANSPRECH=	Ansprechpartner-1	X(30)	
MOBIL=	Telefon-2	X(20)	
ANSPRECH2=	Ansprechpartner-2	X(30)	
BANK-KK=	Bank oder KK-Inhaber	X(30)	Bankname, wenn Feld 33=„J“ KK-Inhaber, wenn Feld 33=K1..K4
BLZGUELTIG=	BLZ oder gültig bis	X(8)	BLZ, wenn Feld 33=„J“ Gültig bis, wenn Feld 33=K1..K4
KTOKARTNR=	Konto-Nr. oder Karten-Nr.	X(20)	Kto-Nr., wenn Feld 33=„J“ Karten-Nr., wenn Feld 33 = K1..K4
ZAHLART=	Lastschrift/Kreditkarte	X	J = Banklastschrift K1 = Eurocard/Mastercard K2 = Visa K3 = Diners Club K4 = American Express
KARTENPIN=	3-4 Stellige Karten-PIN	X(4)	Die Rückseitig letzten 3-4 Ziffern
FAX=	Fax-Nummer	X(20)	
EMAIL=	Email	X(50)	
MWSTID=	MwSt-IDNummer	X(15)	DE....
SCHALTER=	Schalter	X(20)	
AGENTURID=	Agentur-ID	9(10)	Rechnung.de AgenturNr.
CARDCODE=	Karten-Code	X(6)	KartenCode (Alternativ-Passwort)



Folgende Statusmeldungen sind möglich

0	Kunde gespeichert
1	<i>Fehler Kundendaten leer oder abgelehnt</i>
4	<i>Kunden-Aenderung...Kunde nicht gefunden</i>
5	<i>Sonst. Fehler</i>
6	<i>Datenfehler: Zahlungsart falsch</i>
7	<i>Datenfehler: Tarif falsch/unbekannt</i>
8	<i>Datenfehler: BLZ falsch/unbekannt</i>
9	<i>Datenfehler: Kto-Nummer falsch</i>
10	<i>Datenfehler: eMail fehlerhaft/leer</i>
11	<i>Datenfehler: Adresse leer/fehlerhaft</i>
12	<i>MAC-Adresse gesperrt</i>

Nur wenn Status = 0 ist, sind die Daten serverseitig gespeichert. Die Parameter

AccountNo = *Neue verwendete KundenNummer*
Password = *verwendetes Passwort/generiertes Passwort*

werden dann zurückgegeben. Das Passwort kann auch lokal vorgegeben werden, dieses muss dann beim Aufruf der Registrierung als Parameter mitgeliefert werden.



Int **Send**(
 string **Account**,
 string **Password**,
 int **JobArt**,
 string **Empfaenger**,
 base64Binary **EmpfaengerListe**,
 string **SendText**,
 base64Binary **SendFile1**,
 base64Binary **SendFile2**,
 base64Binary **SendFile3**,
 dateTime **SendeDatumZeit**,
 string **Schalter**,
 int **JobId**,
 int **AnzahlEmpfaenger**,
 int **AnzahlSeiten**,
 base64Binary **CheckDocFile**)

Diese Funktion veranlasst den Versand eines Dokumentes. Mit dem Testflag kann der Versand auch erst einmal getestet und verifiziert werden (von PDF zu TIFF etc.)

Account	Kundennummer des Versenders
Password	Passwort des Versenders
JobArt	3=Rechnungsbeleg (digitaler Schwarz/Weiss-Druck) 4=Rechnungsbeleg (digitaler Farbdruck) 8=Rechnungsversand als PDF mit qualifizierter Signatur der D-Trust (Bundesdruckerei)
Empfaenger	Empfaenger Mail-Adresse, bei mehreren mit Komma separiert
EmpfaengerListe	Für Jobart 3,4,8 derzeit nicht genutzt
SendText	Individueller Mailbody zur signierten Rechnung
SendFile1 ... SendFile3	Zu versendete Datei im Format TIF, BMP, HTML, DOC Jobart 8 erlaubt nur das PDF-Format. <u>Nur Sendfile1 wird signiert</u> , Sendfile2 + Sendfile3 werden mitgesendet (z.B. für Newsletter, Ergänzende Informationen).
SendeDatumZeit	Zeit-Format, wenn leer erfolgt Sofortversand
Schalter	Verschiedene Versandoptionen, siehe Tabelle
JobId	Laufende Nummer zur Identifikation bei der Abfrage Sendestatus (Status)
AnzahlEmpfaenger	Gibt die tatsächliche Anzahl Empfänger zurück, ungültige Empfänger wurden ausgefiltert
AnzahlSeiten	Gibt die Anzahl Seiten zurück, die durch die Konvertierung zum Sende-Format erstellt wurden
CheckDocFile	Schalter="TEST", gibt die Dokumente im GIF-Format zur Sichtkontrolle zurück.



Schalter

Über den Parameter Schalter können verschiedene Sonderoptionen veranlasst werden:

TEST	Kein Versand, nur Konvertierung zur Testsichtung
STATURL=<URL>	Optionale Meldung des Sendeergebnisses an eine URL*, maximal 128 Zeichen
QSIGNGET	Kein Versand des Beleges. Beleg und Signatur wird mit QSIGNGET() abgeholt

Es können mehrere Schalter übergeben werden, jeder Schalter wird mit „;“ getrennt.

Die folgenden Statis werden von der Funktion Send zurückgemeldet:

0	Auftrag angenommen, wird ausgeführt
-1	Passwort ungültig
-2	Kunde nicht bekannt
-3	Dokument nicht vorhanden
-4	Serverfehler
-5	EmpfängerNr generell gesperrt
-6	DokumentFormat nicht unterstützt
-7	Dokument nicht konvertierbar
-8	Kein Versand, da TEST Aufruf (Schalter=TEST)
-14	Dokument nicht konvertierbar
1	Nicht autorisiert
2	KundenNr ist leer
3	Kunde ist unbekannt (→ -2)
6	Guthaben noch nicht verfügbar (erste Lastschrift + 14 Tage)
7	Guthaben nicht ausreichend (nur Postrechnungversand, Jobart=3/4)
10	Mailvorlage konnte nicht angelegt werden
11	Zu viele Empfänger für zu Signierenden Beleg (>1)
12	Eingeliefertes Dokumenten-Format ist nicht PDF



Eigene Mailvorlage (HTML und Plain) speichern

Für den Versand der signierten Rechnungen wird als Mailbody der Standard von Rechnung.de verwendet. Wenn Sie die zugestellte Mail im eigenen Design, Bildern und Inhalten ausliefern möchten, müssen Sie einmalig mit einer Mail diesen Inhalt bei Rechnung.de speichern. Ab diesem Zeitpunkt werden Ihre Mails mit dem gesetzten Mailinhalt ausgeliefert. Dieser Vorgang kann beliebig oft wiederholt werden.

Die Mail an den Endkunden wird immer als Multipart-Mail ausgeliefert. Dies bedeutet, dass die Mail sowohl einen HTML-Teil, als auch einen Textteil hat. Damit wird sichergestellt, dass egal wie Ihr Kunde seine eMail-Software konfiguriert hat, die Mail richtig angezeigt wird. Viele eMail-Empfänger haben die HTML-Ansicht nicht aktiviert.

Beispielaufruf für das speichern der HTML-Mailvorlage:

```
Send(„150480“,„20X06“,8,„HTMLMAIL“,„“,„<!DOCTYPE HTML PUBLIC....“,„“,„“,„“,„“,„“,„“,„0,0,0,„“)
```

Beispielaufruf für das speichern der Plaintext-Mailvorlage:

```
Send(„150480“,„20X06“,8,„PLAINMAIL“,„“,„Sehr geehrte Damen und .....“,„“,„“,„“,„“,„“,„“,„0,0,0,„“)
```

Wichtig!

Wenn Bilder oder andere Objekte in der HTML-Vorlage eingebunden werden, können diese nur als Link eingesetzt werden. Bitte achten Sie unbedingt darauf, dass die eingebundenen Objekte über öffentliche Server erreichbar sind und angezeigt werden.



Mailvorlage mitliefern <HTMLBODY> <TEXTBODY>

Wenn mit dem signierten Beleg eine personalisierte Mail ausgeliefert werden soll, besteht die Möglichkeit, im **SendText** den Mailtext mitzuliefern. Da Rechnung.de generell die eMail als Multipart-Mail ausliefert, ist es hier sehr wichtig, den HTML-Teil und den Textteil durch spezielle Tags zu kennzeichnen:

```
<HTMLBODY>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
.....
.....
</HTMLBODY>
<TEXTBODY>
Sehr geehrte Frau Müller,
```

anbei erhalten Sie Ihre aktuelle Rechnung als PDF-Dokument.....

```
.....
.....
</TEXTBODY>
```

In den Tags <HTMLBODY> wird der HTML-Teil der Mail eingebettet, der Textteil wird durch <TEXTBODY> eingerahmt. Wird kein Textbody mit der Mail mitgeliefert, fügt unser System die allgemeine oder eigene Textvorlage automatisch ein. Das gleiche gilt bei fehlendem HTML-Teil.

Wichtig!

Wenn Bilder oder andere Objekte in der HTML-Vorlage eingebunden werden, können diese nur als Link eingesetzt werden. Bitte achten Sie unbedingt darauf, dass die eingebundenen Objekte über öffentliche Server erreichbar sind und angezeigt werden.

Eigener Mail-Betreff / Subject <SUBJECT>

Ein Eigener Mail-Betreff kann mit dem Tag <SUBJECT> in die Mail eingefügt werden. Beispiel:

```
<SUBJECT>Ihre Rechnung-Nr. 4711 als qualifiziert signierter Beleg</SUBJECT>
```

Dateiname Signatur-Beleg <QSIGNFILE>

Der Dateiname von Rechnung.de ist so aufgebaut, dass von unseren Mitarbeitern anhand der Zusammensetzung der exakte Belegvorgang verfolgt werden kann. Für den Kunden kann dies verwirrend sein, daher können Sie mit dem folgenden Tag den Namen des Beleges vorgeben, welcher dem Empfänger zugesendet wird.

Es ist jedoch notwendig bei Klärungen mit Rechnung.de mit dem Dateinamen, welcher auch im Belegstempel vorgegeben ist zu arbeiten.

Achten Sie bitte darauf, bei der Namensgebung nur gültige Zeichen vorzugeben. Verwenden Sie bitte möglichst keine Extension.

Gültige Zeichen sind: a-z, A-Z, 0-9, -_!() (space)

Alle anderen Zeichen werden als ungültig gewertet, das Rechnung.de System würde in diesem Fall den Standard-Dateinamen weiter verwenden.

Beispiel:

```
<QSIGNFILE>Rechnung-02-2006-456007</QSIGNFILE>
```



Int **QSIGNGET**(string **Account**,
string **Password**,
string **JobId**,
base64Binary **BelegPDF**,
base64Binary **SignaturP7S**)

Wird mit dem Send() Kommando ein Beleg zur Signatur eingeliefert und der Schalter hierbei auf „QSIGNGET“ gesetzt, versendet Rechnung.de diese Dateien NICHT per Mail. Diese werden 48 Stunden zur Abholung über den Aufruf QSIGNGET bereitgestellt.

Hierbei ist folgendes zu beachten:

- Die JobId, welche von Send() zurückgegeben wird, muss für den Aufruf QSIGNGET gespeichert werden
- Es darf frühestens nach 5 Minuten ein QSIGNGET Aufruf nach dem Send() Aufruf erfolgen. Um zu vermeiden, dass unser System durch laufende QSIGNGET Aufrufe blockiert wird, werden automatisch IP-Nummern, die mehr als 10 QSIGNGET Aufrufe innerhalb von 5 Minuten ohne Rückergebnis (Fehler 5) absenden, für 60 Minuten per IP-Sperre blockiert.

Fehlermeldungen QSIGNGET:

0	Beleg wurde signiert, PDF mit Stempel und Signatur-Datei stehen bereit
1	Account nicht gefunden
2	Passwort stimmt nicht zum Account
3	Konto Testzeit abgelaufen
4	Account gesperrt
5	Beleg steht zur Bearbeitung an, ist aber noch nicht erledigt



Int **KundenName**(string **Account**,
string **Password**,
string **KdName**)

Diese Funktion gibt den Namen zur Anzeige auf der Website eines bestimmten Kunden zurück.

Rueckgabe-Stati:

0	OK, KdName ist mit dem Kunden-Namen gefüllt
1	Account nicht gefunden
2	Password stimmt nicht zum Account
3	Konto Testzeit abgelaufen
4	Account gesperrt

Int **KundenDaten**(string **Account**,
string **Password**,
TStringDynArray **KdDaten**)

Diese Funktion gibt die Kundendaten zur Anzeige/Änderung auf der Website eines bestimmten Kunden zurück. Die Daten werden analog zur Funktion Register() zurückgegeben.

Rückgabe-Stati:

0	OK, KdDaten ist mit den Kunden-Daten als String-Array gefüllt, Inhalt siehe Funktion Register
1	Account nicht gefunden
2	Password stimmt nicht zum Account
3	Konto Testzeit abgelaufen
4	Account gesperrt