



# **SMSCREATOR**

## **Schnittstellen- beschreibung**

# Inhaltsverzeichnis

<b>1 Allgemein</b> .....	<b>5</b>
<b>1.1 Einleitung – Was kann man mit SMSCreator eigentlich machen?</b> .....	<b>5</b>
<b>2 SMS Typen</b> .....	<b>6</b>
<b>3 SOAP</b> .....	<b>7</b>
<b>3.1 SOAP Funktionen</b> .....	<b>7</b>
3.1.1 SendSimpleSMS.....	7
3.1.2 SendSMS und SendSMSText.....	10
3.1.3 SendUDH.....	13
3.1.4 SMSBin-Typen.....	15
3.1.5 SendBinData.....	16
3.1.6 SendBinURL.....	18
3.1.7 Senden von Logos und Klingeltönen.....	20
3.1.8 VB Sample mit SMSComAPI.....	21
<b>3.2 Versand von mehreren SMS pro Auftrag</b> .....	<b>24</b>
3.2.1 Alle Empfänger selber Text.....	24
3.2.2 Mehrere Empfänger unterschiedlicher Text.....	24
<b>3.3 Versand mit Java</b> .....	<b>24</b>
3.3.1 Struktur des Archivs.....	24
3.3.2 Wie integriere ich die SMSCreator API in mein Projekt.....	25
3.3.3 Unterstützte Java Versionen.....	25
<b>4 SMTP (e-Mail 2 SMS)</b> .....	<b>25</b>
<b>4.1 Allgemein</b> .....	<b>25</b>
<b>4.2 Aufbau einer Email</b> .....	<b>25</b>
4.2.1 XML-Beschreibung einer Email.....	26
<b>4.3 Beispiel</b> .....	<b>26</b>
<b>5 RückSMS / Delivery Reports</b> .....	<b>28</b>
<b>5.1 Check-SMS</b> .....	<b>28</b>
<b>5.2 HLR Abfrage</b> .....	<b>28</b>
<b>5.3 Zustellung per HTTP-Request</b> .....	<b>28</b>
<b>5.4 Zustellung per Email</b> .....	<b>28</b>
5.4.1 Alternativer Empfänger der RückSMS.....	28
<b>5.5 Per Abfrage</b> .....	<b>29</b>
5.5.1 Abfrage auf einer Webseite.....	29
5.5.2 Received SMS.....	29
5.5.3 GetNewRecivedSMS.....	30
<b>6 CMDSendSMS</b> .....	<b>33</b>
<b>6.1 Parameter</b> .....	<b>33</b>
<b>6.2 Verwendung</b> .....	<b>33</b>
<b>7 Statistik</b> .....	<b>34</b>
<b>7.1 Account-Abfrage</b> .....	<b>34</b>
7.1.1 QueryBalance.....	34
<b>8 Send MMS</b> .....	<b>36</b>
8.1.1 Aufbau.....	36
<b>9 Error Codes</b> .....	<b>39</b>
<b>10 Glossar</b> .....	<b>40</b>

*11 Sonstiges* ..... **42**

## Versions-Historie

<b>Datum</b>	<b>Autor</b>	<b>Bemerkung</b>
06.06.2001	Jürgen Weisental	Erstellung der Schnittstellenbeschreibung
07.04.2004	Jürgen Weisental	Send MMS
21.09.2004	Jürgen Weisental	SMS Typ 19 – Lange SMS mit Absendererkennung über Direktanbindung
22.09.2004	Martin Ammer	Rücksms / delivery Reports
07.03.2005	Boris Stark	Redaktionelle Überarbeitung der Schnittstellen-Beschreibung
07.04.2005	Boris Stark	XML Beschreibung einer E-Mail
30.05.2005	Boris Stark	cmdSendSMS – Shell-basierendes Tool zum Versenden von SMS
13.12.2006	Jürgen Weisental	Korrektur eMail 2 SMS
11.06.2007	Christoph Sperl	Java Schnittstelle hinzugefügt
03.04.2008	Christoph Sperl	SOAP Schnittstellen hinzugefügt
21.04.2008	Christoph Sperl	HLR, DLR Schnittstellen hinzugefügt

# 1 Allgemein

## 1.1 Einleitung – Was kann man mit SMSCreator eigentlich machen?

Die SMSCreator Produktlinie bietet die Möglichkeit wichtige Informationen, wie zum Beispiel Werbung, Termine, neue Produkte, Geburtstagsgrüße und vieles mehr per SMS auf die Mobiltelefone von Mitarbeitern, Gästen, Geschäftspartnern und Kunden zu senden. Hierbei werden alle deutschen Mobilfunkunternehmen unterstützt. Des Weiteren besteht die Möglichkeit, SMS auch weitest gehend weltweit (130 Länder) zu versenden.

Um Ihnen die Anbindung an unsere SMSCreator Produktreihe zu erleichtern, finden Sie im folgenden eine ausführliche Schnittstellenbeschreibung. Es wurde versucht die einzelnen Möglichkeiten so einfach und präzise wie möglich zu erläutern.

## 1.2 Möglichkeiten eine SMS über eine Schnittstelle zu versenden

SOAP	Protokoll, mit dem man Daten zwischen Systemen austauschen kann und Remote Procedure Calls durchgeführt werden können. SOAP verwendet zur Darstellung der Daten XML, zur Übertragung der Nachrichten Internet-Protokolle wie z.B. TCP und HTTP
HTTP GET	Die Einfachste und wohl am meist genutzten Methode, SMS per Schnittstelle zu Übergeben.
HTTP POST	HTTP Schnittstelle
SMTP	Übertragung als E-Mail (E-Mail to SMS) <b>Achtung: diese Schnittstelle ist nicht für den Versand von Massen-SMS geeignet. Es kann immer nur eine SMS an mehrere Empfänger verschickt werden</b>
UCP	Diese Schnittstelle ermöglicht den Empfang und den Versand von Daten im UCP-Format. Grundlage dieser Schnittstelle ist das von der Firma GMC entwickelte Universal Computer Protocol (UCP), welches ursprünglich für die Kommunikation zwischen einem Short Message Service Center (SMSC) eines Mobilfunk-Netzes und einem UCP-Client entworfen wurde. <b>Achtung: zurzeit noch nicht verfügbar</b>
SMPP	Ein auf TCP/IP basierendes Industriestandard-Protokoll zur Anbindung von SMSC zu SMSC (Short Message Service Centern) bzw. zu SMS-Service-Plattformen wie z.B. SMSCreator. <b>Achtung: in der Testphase</b>
FTP	Schnittstelle um SMS über FTP-Zugang (File Transfer Protocol) zu übertragen. <b>Achtung: zurzeit noch nicht verfügbar</b>

## 2 SMS Typen

Nach folgend erhalten Sie eine Beschreibung der möglichen SMS-Typen:

<b>Typ</b>	<b>Beschreibung</b>
1	SMS mit Absenderkennung über ausländische Routen
2	Flash SMS
3	kleines Logo (72 x 14 Pixel – Nokia)
4	Picture SMS (maximal 120 Zeichen – Nokia)
5	Klingelton im RTTL Format (Nokia)
6	SMS ohne Absenderkennung (Standard SMS / Rück-SMS)
7	UDH
8	Bookmark (Test)
9	VCARD 1
10	VCARD 2 (Test)
12	EMS Sound (Test)
13	EMS Animation (Test)
14	EMS Bitmap (Test)
15	großes Logo (72 x 28 Pixel – Nokia)
16	SMS ohne Absenderkennung mit Auslieferungsstatus (Delivery Notification – Versand-Benachrichtigung)
17	lange SMS (wird mit 7 Bit codiert; pro 152 Zeichen wird eine SMS benötigt)
18	SMS mit Absenderkennung über Direktanbindung (High Quality Routen). Nach Absprache kann hier eine Delivery Report (siehe: 5.1) angefordert werden.
19	lange SMS mit Absendererkennung über Direktanbindung (wird mit 7 Bit codiert: pro 152 Zeichen wird eine SMS benötigt)
30	HLR Abfrage auf Mobilfunknummer
50	Festnetz SMS mit variabler Absenderkennung

## 3 SOAP

Die Daten werden über eine HTTP SOAP Schnittstelle gesendet.

Die WSDL Datei (XML Formatbeschreibung zum Senden von SMS an das SMSCreator Gateway) finden Sie unter:

<http://soap.smscreator.de/send.asmx?WSDL>

Um das Versenden von SMS zu unterbinden, verwenden Sie bitte als User und Passwort „Test“.

**Achtung** Es gibt von jeder Funktion zwei Versionen. Die Erste unterstützt SOAPRPC (z.B. PoketSoap). Version 2 ist für andere Anwendungen zu verwenden.

### 3.1 SOAP Funktionen

#### 3.1.1 SendSimpleSMS

Bei mehreren Empfängern oder unterschiedlichen Texten siehe: Punkt 3.2.2

Die einfachste Möglichkeit eine SMS zu übertragen.

Bitte öffnen Sie <http://soap.smscreator.de/send.asmx> und füllen Sie die Felder zum Testen aus.

Folgende Parameter können verwendet werden:

User	String	Benutzername
Password	String	Passwort
Recipient	String	Empfänger
SMSText	String	Text

Es können alle möglichen Zeichen verwendet werden, da die SOAP-Schnittstelle als Kodierung der Zeichen UTF-8 verwendet.

Nach Versand der SMS erhalten Sie diese Rückgabewerte:

OK = Alles erfolgreich!

ERROR + Beschreibung des Fehlers

### 3.1.1.1 SOAP

Hier ein SOAP Beispiel. Es enthält Aufruf und Antwort. Die gezeigten **Platzhalter** müssen mit tatsächlichen Werten ersetzt werden:

```
POST /send.asmx HTTP/1.1
Host: soap.smscreator.de
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://cetix.de/SendSMS/SendSimpleSMS"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="http://cetix.de/SendSMS"
xmlns:types="http://cetix.de/SendSMS/encodedTypes"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body
soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <tns:SendSimpleSMS>
      <User xsi:type="xsd:string">string</User>
      <Password xsi:type="xsd:string">string</Password>
      <Recipient xsi:type="xsd:string">string</Recipient>
      <SMSText xsi:type="xsd:string">string</SMSText>
    </tns:SendSimpleSMS>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="http://cetix.de/SendSMS"
xmlns:types="http://cetix.de/SendSMS/encodedTypes"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body
soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <tns:SendSimpleSMSResponse>
      <SendSimpleSMSResult
xsi:type="xsd:string">string</SendSimpleSMSResult>
    </tns:SendSimpleSMSResponse>
  </soap:Body>
</soap:Envelope>
```

### 3.1.1.2 HTTP GET

Hier ein HTTP GET Beispiel. Es enthält Aufruf und Antwort. Die gezeigten Platzhalter müssen mit tatsächlichen Werten ersetzt werden.

```
GET
/send.asmx/SendSimpleSMS?User=string&Password=string&Recipient=string&SMSText=string HTTP/1.1
Host: soap.smscreator.de
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>
```

### 3.1.1.3 HTTP POST

Hier ein HTTP POST Beispiel. Es enthält Aufruf und Antwort. Die gezeigten Platzhalter müssen mit tatsächlichen Werten ersetzt werden.

```
POST /send.asmx/SendSimpleSMS HTTP/1.1
Host: soap.smscreator.de
Content-Type: application/x-www-form-urlencoded
Content-Length: length

User=string&Password=string&Recipient=string&SMSText=string
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>
```

### 3.1.2 SendSMS und SendSMSText

Bei mehreren Empfängern oder unterschiedlichen Texten siehe: Punkt 3.2.2

Sendet SMS mit dem gleichen Text an mehrere Empfänger.

Bitte <http://soap.smscreator.de/send.asmx> öffnen und die Felder zum Testen füllen.

User	String	Benutzername
Password	String	Passwort
Caption	String	Beschreibung des SMS Auftrages (dient der Identifizierung im User Menü)
Sender	String	Absender (maximal 11 Zeichen oder eine Nummer)
Recipient	String Array	Empfänger der SMS und parametrisierte Felder <a href="#">Beispiel: +491711234567;Hans;Peter;</a>
SMSText	String	Text der SMS mit 2 parametrisierbaren Feldern #FELD1# und #FELD2# <a href="#">Beispiel: Hallo Herr #FELD1# alles Gute zum #FELD2# Geburtstag.</a>
SmsTyp	INT	Siehe Tabelle
SendDate	DateTime	Sendezeit des Auftrages

Nach Versand der SMS erhalten Sie diese Rückgabewerte:

OK = Alles erfolgreich!

ERROR + Beschreibung des Fehlers

### 3.1.2.1 SOAP

Hier ein SOAP Beispiel. Es enthält Aufruf und Antwort. Die gezeigten **Platzhalter** müssen mit tatsächlichen Werten ersetzt werden

```
POST /send.asmx HTTP/1.1
Host: soap.smscreator.de
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://cetix.de/SendSMS/SendText"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="http://cetix.de/SendSMS"
xmlns:types="http://cetix.de/SendSMS/encodedTypes"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body
soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <tns:SendText>
      <User xsi:type="xsd:string">string</User>
      <Password xsi:type="xsd:string">string</Password>
      <Caption xsi:type="xsd:string">string</Caption>
      <Sender xsi:type="xsd:string">string</Sender>
      <Recipient xsi:type="xsd:string">string</Recipient>
      <SMSText xsi:type="xsd:string">string</SMSText>
      <SmsTyp xsi:type="xsd:string">string</SmsTyp>
      <SendDate xsi:type="xsd:string">string</SendDate>
    </tns:SendText>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="http://cetix.de/SendSMS"
xmlns:types="http://cetix.de/SendSMS/encodedTypes"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body
soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <tns:SendTextResponse>
      <SendTextResult xsi:type="xsd:string">string</SendTextResult>
    </tns:SendTextResponse>
  </soap:Body>
</soap:Envelope>
```

### 3.1.2.2 HTTP GET

Hier ein http GET Beispiel. Es enthält Aufruf und Antwort. Die gezeigten Platzhalter müssen mit tatsächlichen Werten ersetzt werden.

```
GET
/send.asmx/SendText?User=string&Password=string&Caption=string&Sender=string&Recipient=string&SMSText=string&SmsTyp=string&SendDate=string HTTP/1.1
Host: soap.smscreator.de
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>
```

### 3.1.2.3 HTTP POST

Hier ein HTTP POST Beispiel. Es enthält Aufruf und Antwort. Die gezeigten Platzhalter müssen mit tatsächlichen Werten ersetzt werden.

```
POST /send.asmx/SendText HTTP/1.1
Host: soap.smscreator.de
Content-Type: application/x-www-form-urlencoded
Content-Length: length

User=string&Password=string&Caption=string&Sender=string&Recipient=string&SMSText=string&SmsTyp=string&SendDate=string
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>
```



### 3.1.3.1 SOAP

Hier ein SOAP Beispiel. Es enthält Aufruf und Antwort. Die gezeigten **Platzhalter** müssen mit tatsächlichen Werten ersetzt werden.

```
POST /send.asmx HTTP/1.1
Host: soap.smscreator.de
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://cetix.de/SendSMS/SendUDH"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendUDH xmlns="http://cetix.de/SendSMS">
      <User>string</User>
      <Password>string</Password>
      <Recipient>string</Recipient>
      <Sender>string</Sender>
      <Data>string</Data>
      <sendDate>string</sendDate>
    </SendUDH>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendUDHResponse xmlns="http://cetix.de/SendSMS">
      <SendUDHResult>string</SendUDHResult>
    </SendUDHResponse>
  </soap:Body>
</soap:Envelope>
```

### 3.1.3.2 HTTP GET

Hier ein HTTP GET Beispiel. Es enthält Aufruf und Antwort. Die gezeigten **Platzhalter** müssen mit tatsächlichen Werten ersetzt werden.

```
GET
/send.asmx/SendUDH?User=string&Password=string&Recipient=string&Sender=string&Data=string&sendDate=string HTTP/1.1
Host: soap.smscreator.de
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>
```

### 3.1.3.3 HTTP POST

Hier ein HTTP POST Beispiel. Es enthält Aufruf und Antwort. Die gezeigten [Platzhalter](#) müssen mit tatsächlichen Werten ersetzt werden.

```
POST /send.asmx/SendUDH HTTP/1.1
Host: soap.smscreator.de
Content-Type: application/x-www-form-urlencoded
Content-Length: length

User=string&Password=string&Recipient=string&Sender=string&Data=string&send
Date=string
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>
```

### 3.1.4 SMSBin-Typen

Nach folgend erhalten Sie eine Auflistung der möglichen SMSBin-Typen (eine Beschreibung der Abkürzungen finden Sie im Glossar):

<b>Typ</b>	<b>Beschreibung</b>
10	BMP
11	JPG
12	GIF
13	NOL
14	PNG
31	RTTL

### 3.1.5 SendBinData

#### Senden von Binären Daten

**User** Benutzerkennung  
**Password** Passwort  
**Recipient** Empfänger  
**strMsg** Nachricht  
**MsgType** siehe Api Doku 4.1  
**binType** siehe Api Doku 3.1.4  
**Data** ByteArray der Datei  
**sendDate** Sendezeit der Nachricht

#### Ergebnis:

OK MsgID oder Error Nachricht

#### 3.1.5.1 SOAP

Hier ein SOAP Beispiel. Es enthält Aufruf und Antwort. Die gezeigten **Platzhalter** müssen mit tatsächlichen Werten ersetzt werden.

```
POST /send.asmx HTTP/1.1
Host: soap.smscreator.de
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://cetix.de/SendSMS/SendBinData"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendBinData xmlns="http://cetix.de/SendSMS">
      <User>string</User>
      <Password>string</Password>
      <Recipient>string</Recipient>
      <strMsg>string</strMsg>
      <MsgType>int</MsgType>
      <binTyp>int</binTyp>
      <Data>base64Binary</Data>
      <sendDate>dateTime</sendDate>
    </SendBinData>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendBinDataResponse xmlns="http://cetix.de/SendSMS">
      <SendBinDataResult>string</SendBinDataResult>
    </SendBinDataResponse>
  </soap:Body>
</soap:Envelope>
```

### 3.1.5.2 HTTP GET

Hier ein HTTP GET Beispiel. Es enthält Aufruf und Antwort. Die gezeigten Platzhalter müssen mit tatsächlichen Werten ersetzt werden

```
GET
/send.asmx/SendBinData?User=string&Password=string&Recipient=string&strMsg=
string&MsgType=string&binTyp=string&Data=string&Data=string&sendDate=string
HTTP/1.1
Host: soap.smscreator.de
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>
```

### 3.1.5.3 HTTP POST

Hier ein HTTP POST Beispiel. Es enthält Aufruf und Antwort. Die gezeigten Platzhalter müssen mit tatsächlichen Werten ersetzt werden.

```
POST /send.asmx/SendBinData HTTP/1.1
Host: soap.smscreator.de
Content-Type: application/x-www-form-urlencoded
Content-Length: length

User=string&Password=string&Recipient=string&strMsg=string&MsgType=string&b
inTyp=string&Data=string&Data=string&sendDate=string
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>
```

### 3.1.6 SendBinURL

<b>User</b>	Benutzerkennung
<b>Password</b>	Passwort
<b>Recipient</b>	Empfänger
<b>MsgType</b>	siehe Api Doku 4.1
<b>binType</b>	siehe Api Doku 3.1.4
<b>URL</b>	URL der Binaren Datei muss erreichbar sein.
<b>sendDate</b>	Sendezeit der Nachricht

**Ergebnis:**

OK MsgID oder Error Nachricht

#### 3.1.6.1 SOAP

Hier ist ein SOAP Beispiel, dieses enthält Aufruf und Antwort. Die gezeigten [Platzhalter](#) müssen mit tatsächlichen Werten ersetzt werden

```
POST /send.asmx HTTP/1.1
Host: soap.smscreator.de
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://cetix.de/SendSMS/SendBinURL"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendBinURL xmlns="http://cetix.de/SendSMS">
      <User>string</User>
      <Password>string</Password>
      <Recipient>string</Recipient>
      <MsgType>int</MsgType>
      <binTyp>int</binTyp>
      <URL>string</URL>
      <sendDate>dateTime</sendDate>
    </SendBinURL>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendBinURLResponse xmlns="http://cetix.de/SendSMS">
      <SendBinURLResult>string</SendBinURLResult>
    </SendBinURLResponse>
  </soap:Body>
</soap:Envelope>
```

### 3.1.6.2 HTTP GET

Hier ist ein HTTP GET Beispiel, dieses enthält Aufruf und Antwort. Die gezeigten Platzhalter müssen mit tatsächlichen Werten ersetzt werden

```
GET
/send.asmx/SendBinURL?User=string&Password=string&Recipient=string&MsgType=
string&binTyp=string&URL=string&sendDate=string HTTP/1.1
Host: soap.smscreator.de
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>
```

### 3.1.6.3 HTTP POST

Hier ist ein HTTP POST Beispiel, dieses enthält Aufruf und Antwort. Die gezeigten Platzhalter müssen mit tatsächlichen Werten ersetzt werden

```
POST /send.asmx/SendBinURL HTTP/1.1
Host: soap.smscreator.de
Content-Type: application/x-www-form-urlencoded
Content-Length: length

User=string&Password=string&Recipient=string&MsgType=string&binTyp=string&U
RL=string&sendDate=string
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>
```

## 3.1.7 Senden von Logos und Klingeltönen

### 3.1.7.1 Senden über die SOAP-Schnittstelle

Unter <http://soap.smscreator.de/send.asmx> kann der Webservice angesprochen werden. Um eine Logo oder Klingelton zu senden wird die Funktion SendHex verwendet.

### 3.1.7.2 SendHex

### 3.1.7.3 Parameter

User	Benutzername
Password	Passwort
Recipient	Empfänger des Logos/Klingelton
HexData*	Daten des zu senden Objektes in Hex codiert
MsgType	Endung der Datei Mögliche Optionen
	Logos: GIF, BMP, NOL, JPG (72x 14 Pix)
	Klingeltöne: MID und RTTL

- Die Datei muss von Binär nach Hex codiert werden dabei ist darauf zu achten das jedes Byte in 2 Hex Zeichen umgewandelt wird.

Beispiel: Binär 0 wird zu Hex 00

### 3.1.8 VB Sample mit SMSComAPI

Bei mehreren Empfängern oder unterschiedlichen Texten siehe: Punkt 3.2

Vor dem Versenden über COM-Schnittstelle erst:

([www.smscreator.de/download/comapi/SMSCreatorCOM.zip](http://www.smscreator.de/download/comapi/SMSCreatorCOM.zip)) installieren!

Hier ein Codebeispiel:

```
Dim oSMS
```

```
Set oSMS = CreateObject("SmscreatorCOM.QuickSMS")
```

```
Einstellen eines anderen Proxies (IE Proxy wird default verwendet)
```

```
oSMS.Proxy = "http://proxyserver:2001"
```

```
Wird Benutzername und Passwort benötigt einfach setzen
```

```
oSMS.ProxyUser = "puser"
```

```
oSMS.ProxyPasswort = "pw"
```

```
Senden der Nachricht Parameter Benutzer,Passwort,Empfänger,Text
```

```
MsgBox oSMS.SendSMS("test", "test","0171123456", "Hallo Welt")
```

### 3.1.8.1 Perl Sample

Bei mehreren Empfängern oder unterschiedlichen Texten siehe: Punkt 3.2

Benötigt [SOAP::Lite](#)  
und [Crypt::SSLeay](#), wenn HTTPS verwendet werden soll.

```
use SOAP::Lite;
use strict;
my $s = SOAP::Lite

    -> uri('http://cetix.de/SendSMS')
#   HTTPS Verbindung
    -> proxy('http://soap.smscreator.de/send.asmx')

    -> on_action(sub{sprintf '%s/%s', @_ })
#   -> on_debug(sub{print@_});

#Senden über die Einfach Methode

my $User = SOAP::Data->name('User' => 'test')->type('string')->
>uri('http://cetix.de/SendSMS');
my $Pw = SOAP::Data->name('Password' => 'test')->type('string')->
>uri('http://cetix.de/SendSMS');
my $Empf = SOAP::Data->name('Recipient' => '49171123457')->type('string')->
>uri('http://cetix.de/SendSMS');
my $Text = SOAP::Data->name('SMSText' => 'Hallo Welt')->type('string')->
>uri('http://cetix.de/SendSMS');

my $result = $s->SendSimpleSMS2($User,$Pw, $Empf, $Text)->result;

print "\nResult: " . $result . "\n";
```

### 3.1.8.2 PHP Sample

Bei mehreren Empfängern oder unterschiedlichen Texten siehe: Punkt 3.2

```
<?php
function PostToHost($host, $path, $data_to_send) {
    $fp = fsockopen($host, 80);
    fputs($fp, "POST $path HTTP/1.1\r\n");
    fputs($fp, "Host: $host\r\n");
    fputs($fp, "Content-type: application/x-www-form-urlencoded\r\n");
    fputs($fp, "Content-length: ". strlen($data_to_send) ."\r\n");
    fputs($fp, "Connection: close\r\n\r\n");
    fputs($fp, $data_to_send);
    while(!feof($fp)) {
        $res .= fgets($fp, 128);
    }
    fclose($fp);
    return $res;
}

function SendSMS($user, $pw, $jobid, $Msg, $Rcp, $MsgTyp, $Sender)
{
    //Vorbereiten der Daten
    $data = "User=$user&Password=$pw&Caption=" . rawurlencode($jobid) .
"&Sender=" . rawurlencode($Sender) . "&SMSText=" . rawurlencode($Msg) .
"&Recipient=" . rawurlencode($Rcp) . "&SmsTyp=$MsgTyp&SendDate=";

    //Senden an den Server
    $x = PostToHost(
        "soap.smscreator.de",
        "/send.asmx/SendText",
        $data
    );

    //Auswerten des Ergebnisses

    $iStart = strpos($x, "SendSMS\>") + 9;
    $iEnde = strpos($x, ">", $iStart);

    return substr($x, $iStart, $iEnde-$iStart);
}

?>

<html>
<head><title>PHP Form Post SMS Sample</title></head>
<body>

        SMS Result: <?php echo SendSMS("test", "test", "Caption", "Hallo
PHP", "01711234567", 6, ""); ?>

</body>
</html>
```

## 3.2 Versand von mehreren SMS pro Auftrag

Diese Funktionen gelten für alle oben genannten SMS Übertragungsarten.

### 3.2.1 Alle Empfänger selber Text

Mehrere Empfänger erhalten die gleiche SMS.

- im Empfängerfeld (Recipient) werden die einzelnen Empfänger durch ein \n (Hex 10) getrennt

Beispiel: Normale SMS

```
0171124567;FELD1;FELD2/n
```

SMS an mehrere Empfänger

```
Nummer  Feld1  Feld2
0171124567;Michael;Müller\n
0171124568;Klaus;Mustermann\n
0171124569;Helmut;Schmit
```

- SMS Text: Hier könnte Ihr Nachricht stehen ...

### 3.2.2 Mehrere Empfänger unterschiedlicher Text

Im Text können mehrere Felder vergeben werden (siehe Beispiel).

- Die einzelnen Empfänger werden durch ein \n (Hex10) getrennt.

Beispiel:

```
Nummer  Feld1          Feld2
0171124567;Michael Müller;45trx45\n
0171124568;Klaus;Mustermann\n
0171124569;Helmut;Schmit
```

- SMS Text

Beispiel:

Hallo #FELD1# am Do haben wir eine Party. Dein Code ist #FELD2#.

## 3.3 Versand mit Java

Um diese Schnittstelle nutzen zu können, müssen Sie sich das Schnittstellenarchiv:

<http://www.smscreator.de/docs/java.zip>

### 3.3.1 Struktur des Archivs

Bat	Ordner der ausführbare Skripte zur Entwicklung.
Bin	Ordner in den die .class Dateien ausgegeben werden.
Example	Beispiele der Benutzung.
Lib	Ordner mit den Bibliotheken
Src	Ordner für den Quellcode
.classpath	Eclipse Klassenpfad Informationen
.project	Eclipse Projekt in dem Entwickelt wurde.

### 3.3.2 Wie integriere ich die SMSCreator API in mein Projekt

Um die API in ein Projekt zu integrieren müssen dem neuen Projekt alle Libraries aus dem "lib"-Ordner hinzugefügt werden.

Um die Funktionen der API zu nutzen muss ein neuer Service erzeugt werden:  
`de.cetix.SendSMS.Send service = new de.cetix.SendSMS.SendLocator();`

Von diesem Service kann dann das Soap Objekt geholt werden:  
`de.cetix.SendSMS.SendSoap sendSoap = service.getSendSoap();`

Über das erhaltene Soap Objekt können nun alle verfügbaren Operationen ausgeführt werden.

z.B. eine einfache Text SMS versendet werden:

```
sendSoap.sendText("user","pw","caption","sender","recipient","text","typ","date");
```

Achtung: Das Zeichenformat für die Soap Schnittstelle ist nicht UTF8, sondern ISO-8859-1 (Latin1). Um hier Sonderzeichen-Fehlern vorzubeugen sollte schon beim Einlesen auf das richtige Charset geachtet werden.

In der Beispielanwendung etwa durch:

```
new InputStreamReader(System.in, Charset.forName("ISO-8859-1"));
```

### 3.3.3 Unterstützte Java Versionen

Die API wurde auf der Java 1.4.2 und Java 6 getestet und funktionierte hier einwandfrei.

Java Versionen vor 1.4 werden nicht unterstützt.

Sollte eine Inkompatibilität auftreten, bitten wir um eine kurze Information an [support@netxp.de](mailto:support@netxp.de), damit dies behoben werden kann. Danke.

## 4 SMTP (e-Mail 2 SMS)

### 4.1 Allgemein

Empfänger der SMS ist [Handynummer@sms.smscreator.de](mailto:Handynummer@sms.smscreator.de)

Es wird nur das Feld ‚AN‘ ausgewertet.

Sie können als SMTP Server [mail.smscreator.de](mailto:mail.smscreator.de) angeben (Port 25).

Kann der Betreff nicht geändert werden können Sie alternativ auch im ‚AN‘-Feld Ihre Zugangsdaten übertragen.

Aufbau: Benutzer.Passwort.HandyNr@sms.smscreator.de

### 4.2 Aufbau einer Email

Empfänger: [Handynummer@sms.smscreator.de](mailto:Handynummer@sms.smscreator.de)

Body: Text der Nachricht (die ersten 160 Zeichen werden verwendet)

Empfänger: Empfänger der SMS

Betreff:

Benutzer: Account Name

Passwort: Account Passwort

SendeZeit: Optional die Zeit wann die SMS übertragen werden soll.  
Format: dd.mm.yyyy HH:MM:SS

SenderId: Abesender ID (max 11 Zeichen)  
Wird diese Option gewählt kann keine RückSMS (oder Nummern  
Überprüfung) aktiviert werden

Beispiel für den Aufbau des Betreffs:  
Test1;Test2;13.12.2006 13:30;01701234567 oder  
Test1;Test2;13.12.2006 13:30;Feuerwehr

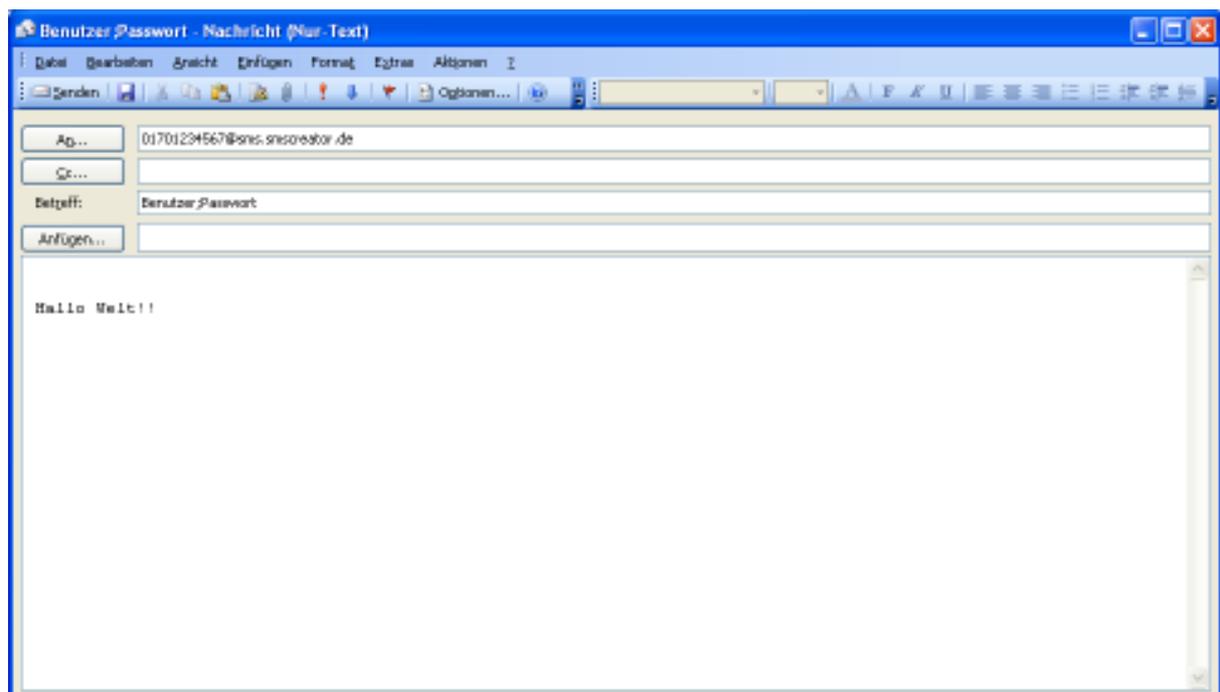
## 4.2.1 XML-Beschreibung einer Email

Mit Hilfe dieser XML-Struktur sind auch mehrere Empfänger möglich. Bei mehreren Empfängern diese XML-Struktur in den Body der Email kopieren.

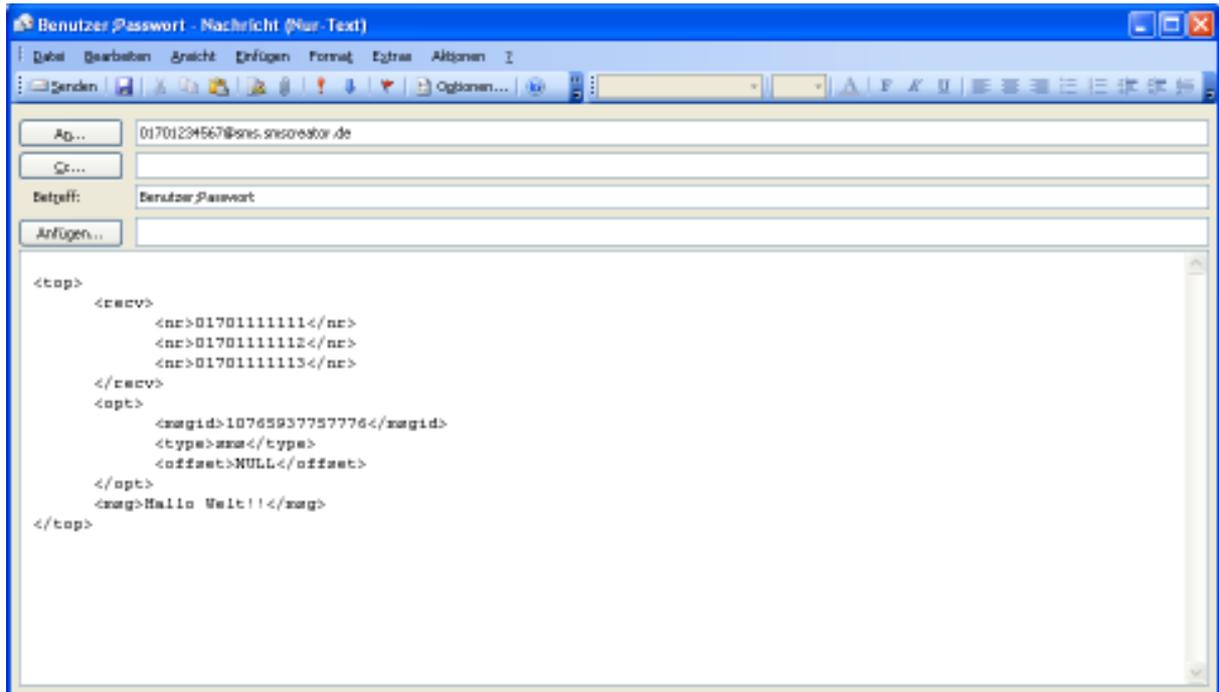
```
<top>
  <recv>
    <nr>Handynummer1</nr>      <!-- Rufnummer des
                                Empfänger -->
    <nr>Handynummer2</nr>      <!-- Rufnummer des
                                Empfänger -->
  </recv>
  <opt>
    <msgid>10765937757776</msgid> <!-- interne
                                    NachrichtenID; wird bei
                                    RückSMS benötigt -->
    <type>sms</type>              <!-- SMS Typ: sms =
                                    RückSMS, flash = FlashSMS
                                    -->
    <offset>NULL</offset>        <!-- Datum/Zeit für
                                    Versendung: null = sofort;
                                    ansonsten im Format TT-MM-
                                    JJJJ HH:MM -->
  </opt>
  <msg>test</msg>               <!-- Nachricht -->
</top>
```

## 4.3 Beispiel

1 Empfänger



## Mehrere (hier 3) Empfänger



## 5 RückSMS / Delivery Reports

RückSMS bzw. Delivery Reports können auf mehrere Arten zugestellt und abgefragt werden:

- Zustellung per http-Request
- Zustellung per Email
- Per Abfrage auf Webseite

### 5.1 Check-SMS

Mit dem SMS Typ 16 kann eine so genannte Check-SMS versendet werden. Mit dieser Check-SMS erhalten Sie einen Zustellreport über alle mit diesem Typ versendeten SMS. Abruf siehe: 5.5.1. Eine variable Absenderkennung ist bei diesem Typ nicht möglich.

Sollte ein Versand mit variabler Absenderkennung (Typ 18) erfolgen und Sie für diese SMS ebenfalls eine Statusreportabfrage benötigen, setzen Sie sich bitte mit uns in Verbindung. Hier besteht die Möglichkeit, so genannte Delivery Reports anzufordern.

### 5.2 HLR Abfrage

Das HLR (Home Location Register) ist eine Datenbank, das Auskunft darüber gibt, ob es eine Mobilfunknummer eines Benutzers gibt und in welchem Heimatnetz sie sich befindet (Provider). Bei dieser Abfrageart, wird dem Kunden keine SMS gesendet.

Die HLR Abfrage kann über alle Schnittstellen, die die Einstellung des SMS-Typs (HLR=SMS-Typ 30) zulassen, erfolgen.

### 5.3 Zustellung per HTTP-Request

Bei dieser Zustellung, benötigen Sie eine öffentlich zugängliche Internetseite, auf die wir diese Nachrichten zustellen können.

Hier ein Beispiellink:

<http://gateway.test.de/netxp.php?id=#ID#&text=#MSG#&from=#MOBIL#&smstyp=#STATUS#>

### 5.4 Zustellung per Email

Hier benötigen wir eine Email-Adresse von Ihnen, auf die wir die RückSMS zustellen können. Delivery Reports werden nicht per Email zugestellt.

Ausgabepunkte sind hier:

Sender :  
Datum :  
Nachricht:

#### 5.4.1 Alternativer Empfänger der RückSMS

Sie können in jedem Job einen alternativen Empfänger für die Rücksms angeben.

In der Caption [backmail:empfaenger@email.de](mailto:backmail:empfaenger@email.de) angeben.

Jede auf dieses Job Empfangene SMS wird nun an die angegebene Adresse gesendet.

## 5.5 Per Abfrage

### 5.5.1 Abfrage auf einer Webseite

Mit dem Link: <http://soap.smscreator.de/smscrecived/checkdata.aspx>

Können Sie eine Internetseite aufrufen und mit Hilfe Ihrer Seriennummer und des Passwortes, alle Rücksms bzw. delivery Reports abrufen.

Abrufkriterien sind hier:

Datum  
Format (csv oder HTML Tabelle)  
JobId  
Nur Statusberichte (Alle, Nur Erfolgreiche oder nur Fehler)

### 5.5.2 Received SMS

**Liefert alle Nachrichten ab einem Zeitpunkt**

**User** Benutzerkennung  
**Password** Passwort  
**strLastDate** von Datum

**Ergebnis:** CSV String

Aufbau:  
Sender;Datum;Nachricht;NewLineZeichen

#### 5.5.2.1 SOAP

Hier ist ein SOAP Beispiel, dieses enthält Aufruf und Antwort. Die gezeigten **Platzhalter** müssen mit tatsächlichen Werten ersetzt werden.

```
POST /send.asmx HTTP/1.1
Host: soap.smscreator.de
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://cetix.de/SendSMS/RecivedSMS"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <RecivedSMS xmlns="http://cetix.de/SendSMS">
      <User>string</User>
      <Password>string</Password>
      <strLastDate>string</strLastDate>
    </RecivedSMS>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <RecivedSMSResponse xmlns="http://cetix.de/SendSMS">
      <RecivedSMSResult>string</RecivedSMSResult>
    </RecivedSMSResponse>
  </soap:Body>
</soap:Envelope>
```

### 5.5.2.2 HTTP GET

Hier ist ein HTTP GET Beispiel, dieses enthält Aufruf und Antwort. Die gezeigten Platzhalter müssen mit tatsächlichen Werten ersetzt werden.

```
GET /send.asmx/RecivedSMS?User=string&Password=string&strLastDate=string
HTTP/1.1
Host: soap.smscreator.de
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>
```

### 5.5.2.3 HTTP POST

Hier ist ein HTTP POST Beispiel, dieses enthält Aufruf und Antwort. Die gezeigten Platzhalter müssen mit tatsächlichen Werten ersetzt werden.

```
POST /send.asmx/RecivedSMS HTTP/1.1
Host: soap.smscreator.de
Content-Type: application/x-www-form-urlencoded
Content-Length: length

User=string&Password=string&strLastDate=string
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>
```

## 5.5.3 GetNewRecivedSMS

**Liefert alle noch nicht abgeholten Nachrichten zurück**

**User** Benutzerkennung

**Password** Passwort

**Ergebnis:** CSV String

Aufbau:

Sender;Nachricht;Datum;IstStatusReport;NewLineZeichen

### 5.5.3.1 SOAP

Hier ist ein SOAP Beispiel, dieses enthält Aufruf und Antwort. Die gezeigten Platzhalter müssen mit tatsächlichen Werten ersetzt werden.

```
POST /send.asmx HTTP/1.1
Host: soap.smscreator.de
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://cetix.de/SendSMS/GetNewRecivedSMS"
```

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetNewRecivedSMS xmlns="http://cetix.de/SendSMS">
      <User>string</User>
      <Password>string</Password>
    </GetNewRecivedSMS>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetNewRecivedSMSResponse xmlns="http://cetix.de/SendSMS">
      <GetNewRecivedSMSResult>string</GetNewRecivedSMSResult>
    </GetNewRecivedSMSResponse>
  </soap:Body>
</soap:Envelope>

```

### 5.5.3.2 HTTP GET

Hier ist ein HTTP GET Beispiel, dieses enthält Aufruf und Antwort. Die gezeigten **Platzhalter** müssen mit tatsächlichen Werten ersetzt werden.

```

GET /send.asmx/GetNewRecivedSMS?User=string&Password=string HTTP/1.1
Host: soap.smscreator.de
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>

```

### 5.5.3.3 HTTP POST

Hier ist ein HTTP POST Beispiel, dieses enthält Aufruf und Antwort. Die gezeigten **Platzhalter** müssen mit tatsächlichen Werten ersetzt werden.

```

POST /send.asmx/GetNewRecivedSMS HTTP/1.1
Host: soap.smscreator.de
Content-Type: application/x-www-form-urlencoded
Content-Length: length

User=string&Password=string
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>

```



## 6 CMDSendSMS

Mit dem Shell-Tool cmdSendSMS versenden Sie SMS über die Windows Shell (das .NET-Framework muss auf Ihrem Rechner installiert sein) bzw. über die Linux-Shell (das .NET-Framework ‚Mono‘ muss auf Ihrem Rechner installiert sein).

### 6.1 Parameter

Folgende Parameter stehen Ihnen zur Verfügung:

User:            Seriennummer  
PW:              Passwort  
RCP:             Empfänger der SMS. Mehrere Empfänger müssen durch ein Komma getrennt, angegeben werden.  
Text:            Geben Sie hier Ihre Nachricht ein (alle Sonderzeichen möglich)

Optionale Parameter sind:

Date:            Sende Datum. Falls die SMS zu einem bestimmten Zeitpunkt versendet werden soll. Format: HH:MM JJJJ.MM.TT  
Caption:         Job Überschrift.  
SMSTyp:         Die einzelnen SMS-Typen entnehmen Sie bitte Punkt 6  
Sender:         Absenderkennung. Geben Sie hier einen bestimmten Absender ein.  
Proxy:           Für den Webzugriff  
Url:             Port  
Account:         Zeigt Ihnen Ihr restliches SMS Kontingent an

### 6.2 Verwendung

Rufen Sie das Tool folgendermaßen auf:

```
../cmdSendSMS /User:test /PW:123 /RCP:01771234567 /Text:"Hallo Welt!!!"
```

Selbstverständlich können Sie das Tool auch mit einer Batch-Datei aufrufen.

Die Hilfefunktion rufen Sie mit `cmdSendSMS /?` oder `cmdSendSMS -help` auf.

## 7 Statistik

### 7.1 Account-Abfrage

#### 7.1.1 QueryBalance

Liefert den aktuellen Account. Es werden nur Haupt-Accounts zurückgegeben.

##### 7.1.1.1 SOAP

Hier ist ein SOAP Beispiel, dieses enthält Aufruf und Antwort. Die gezeigten [Platzhalter](#) müssen mit tatsächlichen Werten ersetzt werden.

```
POST /send.asmx HTTP/1.1
Host: soap.smscreator.de
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://cetix.de/SendSMS/QueryBalance"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <QueryBalance xmlns="http://cetix.de/SendSMS">
      <User>string</User>
      <Password>string</Password>
    </QueryBalance>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <QueryBalanceResponse xmlns="http://cetix.de/SendSMS">
      <QueryBalanceResult>string</QueryBalanceResult>
    </QueryBalanceResponse>
  </soap:Body>
</soap:Envelope>
```

##### 7.1.1.2 HTTP GET

Hier ist ein HTTP GET Beispiel, dieses enthält Aufruf und Antwort. Die gezeigten [Platzhalter](#) müssen mit tatsächlichen Werten ersetzt werden.

```
GET /send.asmx/QueryBalance?User=string&Password=string HTTP/1.1
Host: soap.smscreator.de
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>
```

### 7.1.1.3 HTTP POST

Hier ist ein HTTP POST Beispiel, dieses enthält Aufruf und Antwort. Die gezeigten [Platzhalter](#) müssen mit tatsächlichen Werten ersetzt werden.

```
POST /send.asmx/QueryBalance HTTP/1.1
Host: soap.smscreator.de
Content-Type: application/x-www-form-urlencoded
Content-Length: length

User=string&Password=string
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>
```

## 8 Send MMS

### 8.1.1 Aufbau

Bitte <http://soap.smscreator.de/send.asmx> öffnen und die Felder zum Testen füllen.

User	String
Password	String
Data	String

Benutzername
Passwort
Aufbau als XML Struktur

```
<MMS>
<SUBJECT>Titel der MMS</SUBJECT>
<IMAGE1>Link auf das Bild</IMAGE1>
<IMAGEDATA1>Base64 Kodiertes
Bild</IMAGEDATA1>
<TEXT1>Text der MMS</TEXT1>
<IMAGE2> Anderes Bild </IMAGE2>
<SOUND1>Link auf den Sound </SOUND1>
alt. <SOUND1DATA1> Base64 Kodiertes Sound im
</SOUND1DATA1>
<TEXT2>Text für 2 Bild</TEXT2>
<SMIL> Um eine SMS Speziell zu formatieren
kann eine SMIL Datei angegeben werden.
Infos unter http://www.w3.org/TR/REC-smil
</SMIL>
</MMS>
```

Es können n Bilder und Text Elemente vorkommen.  
Werden Links angegeben, müssen diese zum Übertragungszeitpunkt erreichbar sein.

Bilder in den Formaten: JPG,GIF,PNG  
Größe 160 x 120 Pixel. Sollte ein Bild größer sein, wird automatisch eine Verkleinerung auf diese Werte durchgeführt.  
Sounds im Format **amr**

(Um wav/mp3-dateien in amr-Files zu konvertieren empfehlen wir von Nokia (<http://www.forum.nokia.com>) den Nokia Multimedia Converter 2.0 Beta 2. Downloadbar, nach kostenloser Registrierung unter:

<http://www.forum.nokia.com/main/1,6566,030,00.html?fsrPar am=2-3-/main/1,6566,030,00.html&fileID=2998> )

Beispiel:

```
<MMS>
<SUBJECT>Michael</SUBJECT>
<IMAGE1>http://www.smscreator.de/mms/image1.jpg</IMAGE1>
<TEXT1>Alle Gute zum Geburtstag</TEXT1>
</MMS>
```

SendDate	String
----------	--------

Sendezeit: Wenn keine Sendezeit angegeben wird, wird die MMS sofort versendet.
--

### 8.1.1.1 SOAP

Hier ein SOAP Beispiel. Es enthält Aufruf und Antwort. Die gezeigten **Platzhalter** müssen mit tatsächlichen Werten ersetzt werden.

```
POST /send.asmx HTTP/1.1
Host: soap.smscreator.de
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://cetix.de/SendSMS/SendMMS"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendMMS xmlns="http://cetix.de/SendSMS">
      <User>string</User>
      <Password>string</Password>
      <Recipient>string</Recipient>
      <Sender>string</Sender>
      <Data>string</Data>
      <SendDate>string</SendDate>
    </SendMMS>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendMMSResponse xmlns="http://cetix.de/SendSMS">
      <SendMMSResult>string</SendMMSResult>
    </SendMMSResponse>
  </soap:Body>
</soap:Envelope>
```

### 8.1.1.2 HTTP GET

Hier ein HTTP GET Beispiel. Es enthält Aufruf und Antwort. Die gezeigten **Platzhalter** müssen mit tatsächlichen Werten ersetzt werden.

```
GET
/send.asmx/SendMMS?User=string&Password=string&Recipient=string&Sender=string&Data=string&SendDate=string HTTP/1.1
Host: soap.smscreator.de
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>
```

### 8.1.1.3 HTTP POST

Hier ein HTTP POST Beispiel. Es enthält Aufruf und Antwort. Die gezeigten [Platzhalter](#) müssen mit tatsächlichen Werten ersetzt werden.

```
POST /send.asmx/SendMMS HTTP/1.1
Host: soap.smscreator.de
Content-Type: application/x-www-form-urlencoded
Content-Length: length

User=string&Password=string&Recipient=string&Sender=string&Data=string&Send
Date=string
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://cetix.de/SendSMS">string</string>
```

## 9 Error Codes

Folgende Errorcodes können Sie erhalten.

401: Invalid User

falscher oder nicht vorhandener User

401: No Recipient

kein Empfänger vorhanden

## 10 Glossar

AMR	Abkürzung für „Audio Modem Riser“
BMP	Bitmap Format - zur verlustfreien Speicherung von Rastergrafiken
EMS	Enhanced Message Service – Dienst für die Kommunikation zwischen Mobiltelefonen (es können damit auch Logos, Bilder und Klingeltöne versendet werden). EMS ist nicht auf 160 Zeichen beschränkt.
Flash SMS	“blinkende SMS” – das Handydisplay blinkt beim Erhalt dieses SMS-Typen. Die SMS kann nur gelesen werden. Das Speichern der SMS ist nicht möglich.
FTP	File Transfer Protocol – Netzwerkprotokoll zur Übertragung von Daten
GIF	Graphics Interchange Format – digitales Bildformat mit guter verlustfreier Komprimierung für Bilder mit geringer Farbtiefe
High Quality Routen	SMS werden direkt an die jeweiligen SMSCs gesendet
HTTP	Hypertext Transfer Protocol – Protokoll zur Übertragung von Daten
JPG	Joint Photographic Experts Group – verlustbehaftetes Kompressionsverfahren für digitale Bilder
MIDI	MIDI - Musical Instrument Digital Interface
NOL	Operator Logo Dateien (Hersteller Logos)
PNG	Portable Network Graphics – Dateiformat zur Speicherung von Bilddaten
RTTL	Ring Tone Transfer Language
SOAP	Protokoll, mit dem man Daten zwischen Systemen austauschen kann und Remote Procedure Calls durchgeführt werden können. SOAP verwendet zur Darstellung der Daten XML, zur Übertragung der Nachrichten Internet-Protokolle wie z.B. TCP und HTTP
SMPP	Short Message Peer to Peer Protocol
SMS ohne Absenderkennung	SMS werden per Modem versendet. Im Absender steht eine festgelegte Handynummer. Diese Nummer wird für alle weiteren SMS als Absendernummer verwendet.
SMSC	Short Message Service Center (der Mobilfunk-Anbieter)
SMTP	Simple Mail Transfer Protocol – Protokoll zur Versendung von E-Mails
TCP/IP	Transmission Control Protocol / Internet Protocol
UCP	Universal Computer Protocol
UDH	User Data Handy
VCARD 1 / 2	elektronische Visitenkarten für Mobilfunkgeräte (wird nur von bestimmten Herstellern unterstützt)

WSDL	Web Services Discription Language – plattform-, programmiersprachen- und protkollunabhängiger XML-Standard zur Beschreibung von Netzwerkdiensten zum Austausch von Nachrichten
XML	Extensible Markup Language - Standard zur Erstellung strukturierter, maschinen- und menschen lesbarer Dateien

## 11 Sonstiges

Sollten Sie dennoch Probleme, Fragen und/oder Anregungen haben, können Sie uns diese gerne per e-Mail an: [Support@SMSCreator.de](mailto:Support@SMSCreator.de) senden.

Wir bedanken uns für den Erwerb unseres Produktes.

Ihr SMSCreator-Team

