

www.1sms.com.tr

Toplu SMS Servisi

HTTP API Teknik Dokümantasyonu

Sürüm 1.1.2

Giriş

1. API ile çalışmaya başlamadan önce ilgili yazılım geliştirmeleri yaparken şu geçici hesabı kullanabilirsiniz : Kullanıcı adı : “demo” Şifre : “12345678”.
2. API ile çalışmaya başlamak için <http://panel.1sms.com.tr> adresinde yeni üyelik oluşturun, giriş yapın. Kullanıcı İşlemleri -> Api Ayarları -> Api Hesabı Oluştur yolunu izleyin. Kullanıcı adınız aynı olmakla birlikte şifre olarak **API SECRET olarak belirtilen 32 karakterlik ifadeyi kullanın. (Web şifreniz ile API şifreniz farklı şeylerdir)**
3. SMS yollayabilmek için hesabınızda kredi yüklendiğinden ve mesaj başlığınızın (originator) onaylanmış olduğuna emin olun.
4. API istekleri sunucularımızın 8080 numaralı portuna HTTP GET veya HTTP POST şeklinde gerçekleştirilir.
5. HTTP POST yönteminde içerik XML olarak formatlanmaktadır.
6. API isteklerinde istek sayısı olarak bir sınırlama yoktur.
7. Destek ve sorularınız için destek@1sms.com.tr e-posta adresinden iletişime geçebilirsiniz.

Kredi Sorgulama

Örnek :

İstek Adresi	http://panel.1sms.com.tr:8080/api/credit/v1?username=test&password=test
Cevap (Başarısız)	87
Cevap (Başarılı)	00 15600

Giden Parametreler

Parametreler :

Parametre	Tip	Açıklama
username	String	Web kullanıcı adınız
password	String	Api Hesabı bölümünde görüntülenen API SECRET ifadesi

Dönüş Değerleri

- Tüm dönüş kodları 2 basamaklı sayılardan oluşur.
- Başarısız dönüş olduğunda sadece kodun kendisini geri döner.
- Başarılı dönüşlerde koddan sonra bir boşluk karakteri bırakılır ve kredi miktarı döner.

Başarısız Dönüşler :

- Aşağıdaki hata kodları isteği yapan tarafın düzeltmesi gereken durumları belirtir.

Kod	İsim	Açıklama
95	USE_GET_METHOD	İsteği HTTP GET ile yollayınız
93	MISSING_GET_PARAMS	GET parametrelerinde eksik var
87	WRONG_USER_OR_PASSWORD	Kullanıcı adı veya şifre hatalı

Başarılı Dönüşler :

Kod	İsim	Açıklama
00	SUCCESS	İşlem başarı ile sonlandı.

SMS Yollama

- 1-N tipi mesajlaşma aynı mesaj metnini bir veya daha fazla numaraya yollamak amacıyla tercih edilir. N-N tipi mesajlaşmada ise her numaraya farklı mesaj metni gönderilebilir.
- Aşağıda gri arkaplına sahip olan satırlar opsiyonel özellikleri ifade eder. Bu özellikler kullanılmayacaksa XML yapısından kaldırılmalıdır, sadece gerekli olduğunda eklenmelidir.
- Boş SMS gönderilemez.

Örnek :

İstek Adresi	http://panel.1sms.com.tr:8080/api/smspost/v1
İstek Tipi	1-N Tipi Mesajlaşma
İstek İçeriği	<pre><sms> <username>test</username> <password>test</password> <header>BAŞLIK</header> <validity>2880</validity> <sendDateTime>2015.7.23.9.30.0</sendDateTime> <message> <gsm> <no>905559998877</no> <no>905559998866</no> </gsm> <msg><![CDATA[-MESAJ-METNİ-1-]]></msg> </message> </sms></pre>
İstek Tipi	N-N Tipi Mesajlaşma
İstek İçeriği	<pre><sms> <username>test</username> <password>test</password> <header>BAŞLIK</header> <validity>2880</validity> <sendDateTime>2015.7.23.9.30.0</sendDateTime> <messages> <mb><no>905559998877</no><msg><![CDATA[-MESAJ-METNİ-1-]]></msg></mb> <mb><no>905559998866</no><msg><![CDATA[-MESAJ-METNİ-2-]]></msg></mb> </messages> </sms></pre>
Cevap (Başarısız)	89
Cevap (Başarılı)	00 77300600

Giden Parametreler

- Parametreler boş bırakılmamalıdır, aksi halde 89 yani WRONG_XML_FORMAT hatası ile karşılaşılır.

Parametreler :

Parametre	Tip	Açıklama
username	String	Web kullanıcı adınız
password	String	Api Hesabı bölümünde görüntülenen API SECRET ifadesi
header	String	Onaylanmış mesaj başlıklarınızdan biri. Bu ifade büyük-küçük harf duyarlıdır. Max 11 karakterdir.
validity	Integer	Mesajın geçerlilik süresi. (dakika cinsinden) Min 1, Max 2880 olabilir. 2880 değeri 48*60 şeklinde hesaplanır, yani 48 saate tekabül eder. Hatalı değerler sistem tarafından 2880 olarak değiştirilir ve öyle kullanılır.
sendDateTime	String	Opsiyonel bir parametredir. Mesajın gönderileceği ileri bir tarihi işaret eder. Örneğin : “2015.7.23.9.30.0” ifadesi 2015 yılı, 7. ay, 23. gün, saat 09:30:00 manasına gelir. Tüm parametreler “.” (nokta) karakteri ile birbirinde ayrılmıştır. Zamanı geldiğinde sistem mesajı otomatik olarak yollar. Eğer geçmiş zamanı gösteren bir tarih bilgisi yollanırsa sistem SMS'i bekletmeden gönderir. Mesajın gönderim tarihi gelmeden önce yönetim panelinden mesaja ulaşip müdahale etme imkanınız bulunmaktadır.
no	String	Telefon numarası. Tüm numaralar Türkiye içi numaralardır. Numaralar 10, 11 veya 12 karakter olabilir. Sistem bu numaraları 12 karaktere tamamlar ve öyle kullanır. İletim raporlarında da 12 karakter olarak raporlar. Bu belirtilenler dışında, daha farklı uzunluklara sahip olan numaralara mesaj gönderilmez, iletim raporlarında numara aynen belirtilir, iletim durumu ise başarısız olarak yansıtılır.
msg	String	Gönderilecek olan mesaj metnidir. Metin içinde XML'e has özel karakterler bulunabilir. Böyle durumlarda hataların önüne geçmek amacıyla mesaj metninin soluna “<![CDATA[“ ve sağına “]]>” ifadelerinin eklenmesini tavsiye etmekteyiz. Karakter seti ve boy hesaplamaları ile ilgili detaylar için dokümanın sonlarında özel bir başlık yer almaktadır, detaylar için ilgili bölüme bakınız.

Dönüş Değerleri

- Tüm dönüş kodları 2 basamaklı sayılardan oluşur.
- Başarısız dönüş olduğunda sadece kodun kendisini geri döner.
- Başarılı dönüşlerde koddan sonra bir boşluk karakteri bırakılır ve daha sonra bu gönderim ile ilgili iletim raporu sorgusunda kullanılacak olan özgün (unique) bir sayısal ifade döner. (Kısaca buna dlrId diyoruz) Gelecekte iletim raporu sorgulanacaksa bu ifadeyi kaydetmeniz ve sorgu sırasında parametre olarak göndermeniz gerekmektedir.

Başarısız Dönüşler 1 :

- Normal şartlarda asla karşılaşmamanız gereken bir hata kodudur. Hatanın kaynağı sunucu veya istemci taraflı olabilir. Yoğun olarak bu hatayı alırsanız lütfen iletişime geçiniz.

Kod	İsim	Açıklama
99	UNKNOWN_ERROR	Henüz dokümente edilmemiş bir hatayı belirtir.

Başarısız Dönüşler 2 :

- Aşağıdaki hata kodları isteği yapan tarafın düzeltilmesi gereken durumları belirtir.

Kod	İsim	Açıklama
97	USE_POST_METHOD	İsteği HTTP POST ile yollayınız
91	MISSING_POST_DATA	POST verisi okunamadı veya yok
89	WRONG_XML_FORMAT	POST verisi XML olarak parse edilemedi
87	WRONG_USER_OR_PASSWORD	Kullanıcı adı veya şifre hatalı
85	WRONG_SMS_HEADER	Belirttiğiniz mesaj başlığı bulunamadı veya onaylanmamış
84	WRONG_SEND_DATE_TIME	İleri tarihli gönderim zamanı hatalı bir formata sahip veya 1 yıldan daha ileri bir zamanı gösteriyor
83	EMPTY_SMS	Mesaj metni ve numaralar incelendikten sonra sistem yollanacak bir SMS oluşturmaya yetecek en az 1 numara ve en az 1 karakterden oluşan mesaj metnine sahip olamadı. Gönderim yapılacak verilerin yeterli olmadığına karar verdi.
81	NOT_ENOUGH_CREDITS	Gönderilecek olan mesaj için yeterli krediye sahip değilsiniz.

Başarılı Dönüşler :

Kod	İsim	Açıklama
00	SUCCESS	İşlem başarı ile sonlandı.

İletim Raporu (DLR) Sorgulama

- SMS gönderimi yapıldıktan hemen sonra DLR isteği yaparsanız hata alabilirsiniz. Sistem mesajı göndermekle ilgileniyor olabilir ve DLR raporunu oluşturacak olan veriler henüz hazır olmayabilir.
- DLR isteğini, ilk olarak SMS gönderiminden sonraki 90. veya 180. saniyede yapmanızı tavsiye ederiz. (Sistem ilk olarak 60. saniyede DLR içeriğini güncelleyen bir görevi başlatır. 48 saat içinde başlarda daha sık olmak üzere sonlara doğru daha da seyrekleşen bir kural üstünden DLR içeriği güncellenmektedir)
- Eğer belirttiğiniz tüm numaralara SMS ulaşmış ise DLR raporu 48 saat dolmadan önce tamamlanabilir, yani son halini alabilir. Bunu başarılı dönüş koduna bakarak anlayabilirsiniz.
- Eğer gönderim listenizdeki numaralardan en az biri 48 saat boyunca kapalı kalmış ise DLR raporunun son halini alması toplamda 48 saat + 10 dakika gibi bir zaman alır. Bu 48 saat boyunca, DLR raporunun henüz son halini almadığı bilgisini başarılı dönüş koduna bakarak anlayabilirsiniz.

Örnek :

İstek Adresi	http://panel.1sms.com.tr:8080/api/dlr/v1?username=test&password=test&id=123456
Cevap (Başarısız)	93
Cevap (Başarılı)	25 905559998877 0 905559998866 5 905559998855 6 905559998844 9

Giden Parametreler

Parametreler :

Parametre	Tip	Açıklama
username	String	Web kullanıcı adınız
password	String	Api Hesabı bölümünde görüntülenen API SECRET ifadesi
id	Integer	SMS gönderimi sonrası sistem tarafından oluşturulan ve istemciye gönderilen dlrId isimli özgün (unique) sayısal ifade.

Dönüş Değerleri

- Tüm dönüş kodları 2 basamaklı sayılardan oluşur.
- Başarısız dönüş veya bilgilendirme dönüşü olduğunda sadece kodun kendisini geri döner.
- Başarılı dönüşlerde, dönüş kodu sonrasında birbirinden “|” karakteri ile ayrılmış bloklar döner. Bu bloklar GSM no ve SMS iletim durumu bilgisini içerir. (Detaylar sayfa sonunda)

Başarısız Dönüşler 1 :

- Normal şartlarda asla karşılaşmamanız gereken bir hata kodudur. Hatanın kaynağı sunucu veya istemci taraflı olabilir. Yoğun olarak bu hatayı alırsanız lütfen iletişime geçiniz.

Kod	İsim	Açıklama
99	UNKNOWN_ERROR	Henüz dokümente edilmemiş bir hatayı belirtir.

Başarısız Dönüşler 2 :

- Aşağıdaki hata kodları isteği yapan tarafın düzeltmesi gereken durumları belirtir.

Kod	İsim	Açıklama
95	USE_GET_METHOD	İsteği HTTP GET ile yollayınız
93	MISSING_GET_PARAMS	GET parametrelerinde eksik var
87	WRONG_USER_OR_PASSWORD	Kullanıcı adı veya şifre hatalı
79	DLR_ID_NOT_FOUND	Böyle bir dlrId değeri bulunamadı, yok

Bilgilendirme Dönüşleri :

Kod	İsim	Açıklama
29	MESSAGE_WAITING_TO_SEND	Mesaj henüz yollanmamış, yollanmak üzere
27	MESSAGE_SEND_ERROR	Mesaj yollanırken beklenmeyen bir hata oluşmuş

Başarılı Dönüşler :

Kod	İsim	Açıklama
25	DLR_OPERATION_STARTED	DLR raporu güncellenmeye başlamış
23	DLR_OPERATION_COMPLETED	DLR raporu artık güncellenmeyecek, tüm bilgiler toplanmış, rapor son halini almış

SMS İletim Durum Kodları :

Kod	İsim	Açıklama	Kod	İsim	Açıklama
0	WAITING	SMS gönderim için bekliyor	6	FAIL	SMS Ulaşmadı
5	PENDING	SMS gönderilmiş, bilgi bekleniyor	9	SUCCESS	SMS İletildi

SMS Karakter Seti ve Boy Hesaplamaları

- Bu bölümde bir SMS metninin kaç SMS olarak ücretlendirileceği, yani kaç boy olacağı ile ilgili hesaplamalara değinilecektir. Ayrıca gönderimine izin verilen karakterler hakkında da bilgi sahibi olacaksınız.
- Sistemimiz max 6 boy SMS kabul etmektedir. Mesaj içeriği 6 boydan fazla ise otomatik olarak sondaki karakterlerden gereken kadarı silinerek mesaj 6 boy'a indirgenir.
- Sistemimiz yollanacak olan her mesaj için boy hesabını otomatik olarak yapar, hesaplanan bu değerleri web arayüzünden görebilirsiniz.

Karakter Sınıfları :

- İlk olarak kullanılan karakterlere bakılarak mesajın Türkçe mi yoksa İngilizce mi olduğu belirlenir. En az 1 adet Türkçe karakter kullanılmış ise mesaj Türkçe olur, aksi halde İngilizce olacaktır.
- Eğer mesajı İngilizce mesaj olarak göndermek istiyorsanız aşağıdaki tabloda Türkçe olarak belirtilen karakterleri mesaj içeriğinden kaldırmanız veya İngilizce karakter sınıfından benzer karakterler ile yer değiştirdikten sonra mesajı yollamanız gerekmektedir.
- Mesaj içeriğindeki Türkçe karakterlerin bazıları 1, bazıları ise 2 karakter yer kaplar. İlgili bilgiler aşağıdaki tabloda belirtilmiştir.
- Bu aşamada mesajın İngilizce veya Türkçe mesaj olması ile birlikte mesajın kaç karakter olduğu hesaplanır. Dikkat ediniz, örneğin SMS içeriğindeki 'ğ' 2 karakter olarak hesaplanır.

Karakter Sınıfları	Liste
İngilizce (Tamamı 1-Karakter)	'i', 'I', 'a', 'A', 'b', 'B' ... 'x', 'X', 'w', 'W' ... '0', '1' ... '\$', '!', '+' ... vs
Türkçe 1-Karakter olanlar	'ü', 'Ü', 'ö', 'Ö', 'Ç'
Türkçe 2-Karakter olanlar	'ğ', 'Ğ', 'ş', 'Ş', 'ı', 'İ', 'ç'

Boyların Hesaplanması :

- Mesajın dili ve karakter sayısı dikkate alınarak boy aşağıdaki tablo yardımıyla belirlenebilir. Dikkat ederseniz İngilizce mesajlar Türkçe mesajlara göre birkaç karakter daha fazla bilgi taşıyabilmektedir.
- Tablodaki değerler aralık belirtir. Başlangıç ve bitiş değerleri de o aralığa dahildir.

	1 boy	2 boy	3 boy	4 boy	5 boy	6 boy
İngilizce	[1, 160]	[161, 306]	[307, 459]	[460, 612]	[613, 765]	[766, 917]
Türkçe	[1, 155]	[156, 292]	[293, 439]	[440, 587]	[588, 735]	[736, 882]

Mesaj Yollanmadan Önce Yaptığımız İçerik Dönüşümleri :

- İlk olarak şunlar yer değiştirir : | => I, ^ => ', € => E, { => (, [=> (, } =>),] =>), \ => /
- Sonrasında, mesaj içeriğinde aşağıdaki karakterler dışında bir karaktere rastlanırsa ilgili karakter '' (yani boşluk karakteri) olarak kabul edilir ve mesaj öyle yollanır :
ABCDEFGHIJJKLMNOPRSŞTUVYZÖÜQWXC
abcçdefgğhijklmnoprsştuvyzöüqwx@*!\$_#()+-;<=>./?'"%&0123456789

Örnek Kaynak Kodlar

- Fonksiyon : Kredi Sorgulama
- Programlama Dili : Java
- Lütfen kaynak kodu kullandığınız editörde yeniden formatlayınız. Kodlarda gereksiz gördüğünüz yerleri daha sonra silebilirsiniz.

```
public void kredi(){
URLConnection uc = null;
try{
URL url = new URL("http://panel.1sms.com.tr:8080/api/credit/v1?username=test&password=test");
uc = (URLConnection)url.openConnection();
uc.setRequestMethod("GET");
uc.setReadTimeout(5*1000);
uc.setConnectTimeout(5*1000);
uc.connect();

int resCode = uc.getResponseCode();
System.out.println("HTTP "+resCode);

int readed;
char[] buffer = new char[4*1024];
StringBuffer sb = new StringBuffer();
BufferedReader in = new BufferedReader(new InputStreamReader(uc.getInputStream(), "UTF-8"));
while( (readed = in.read(buffer)) > 0 ){
sb.append(buffer, 0, readed);
}
in.close();

String ret = sb.toString();
System.out.println("["+ret+"]");

}catch(Exception ex){
ex.printStackTrace();
}finally{
try{ uc.disconnect(); }catch(Exception ex){} uc = null;
}
}
```

- Fonksiyon : SMS Gönderme
- Programlama Dili : Java
- Lütfen kaynak kodu kullandığınız editörde yeniden formatlayınız. Kodlarda gereksiz gördüğünüz yerleri daha sonra silebilirsiniz.

```

public void send(){
    HttpURLConnection uc = null;
    try{
        URL url = new URL("http://panel.1sms.com.tr:8080/api/smspost/v1");
        uc = (HttpURLConnection)url.openConnection();
        uc.setRequestMethod("POST");
        uc.setDoInput(true);
        uc.setDoOutput(true);
        uc.setRequestProperty("Content-Type", "text/xml; charset=UTF-8");
        uc.setRequestProperty("Content-Encoding", "UTF-8");
        uc.setReadTimeout(5*1000);
        uc.setConnectTimeout(5*1000);
        uc.connect();

        OutputStreamWriter out = new OutputStreamWriter(uc.getOutputStream(), "UTF-8");
        out.write("
+ <sms>
+ <username>test</username>
+ <password>test</password>
+ <header>BAŞLIK</header>
+ <validity>2880</validity>
+ <message>
+ <gsm>
+ <no>905559998877</no>
+ </gsm>
+ <msg><![CDATA[-test-message-]]></msg>
+ </message>
+ </sms>");
        out.flush();
        out.close();

        int resCode = uc.getResponseCode();
        System.out.println("HTTP "+resCode);

        int readed;
        char[] buffer = new char[4*1024];
        StringBuffer sb = new StringBuffer();
        BufferedReader in = new BufferedReader(new InputStreamReader(uc.getInputStream(), "UTF-8"));
        while( (readed = in.read(buffer)) > 0 ){
            sb.append(buffer, 0, readed);
        }
        in.close();

        String ret = sb.toString();
        System.out.println("["+ret+"]");

    }catch(Exception ex){
        ex.printStackTrace();
    }finally{
        try{ uc.disconnect(); }catch(Exception ex){} uc = null;
    }
}

```

- Fonksiyon : Kredi Sorgulama
- Programlama Dili : PHP (curl eklentisi ile birlikte)
- Lütfen kaynak kodu kullandığınız editörde yeniden formatlayınız. Kodlarda gereksiz gördüğünüz yerleri daha sonra silebilirsiniz.

```
<?php
$postUrl =
"http://panel.1sms.com.tr:8080/api/credit/v1?username=test&password=test";

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $postUrl);
curl_setopt($ch, CURLOPT_TIMEOUT, 5);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

$response = curl_exec($ch);
curl_close($ch);

echo '['.$response.'],'.\n";
?>
```

- Fonksiyon : SMS Gönderme
- Programlama Dili : PHP (curl eklentisi ile birlikte)
- Lütfen kaynak kodu kullandığınız editörde yeniden formatlayınız. Kodlarda gereksiz gördüğünüz yerleri daha sonra silebilirsiniz.

```
<?php
$postUrl = "http://panel.1sms.com.tr:8080/api/smspost/v1";

$postData = "".
"<sms>".
"<username>test</username>".
"<password>test</password>".
"<header>BAŞLIK</header>".
"<validity>2880</validity>".
"<message>".
"<gsm>".
"<no>905559998877</no>".
"</gsm>".
"<msg><![CDATA[-test-message-]]></msg>".
"</message>".
"</sms>";

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $postUrl);
curl_setopt($ch, CURLOPT_POSTFIELDS, $postData);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_TIMEOUT, 5);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_HTTPHEADER, Array("Content-Type:
text/xml; charset=UTF-8"));

$response = curl_exec($ch);
curl_close($ch);

echo '['.$response.'],'.\n";

?>
```

- Fonksiyon : Kredi Sorgulama
- Programlama Dili : ASP.NET (C#)
- Lütfen kaynak kodu kullandığınız editörde yeniden formatlayınız. Kodlarda gereksiz gördüğünüz yerleri daha sonra silebilirsiniz.

```
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

string sUserID = "UserName";
string sPwd = "Password";
string sURL
="http://panel.1sms.com.tr:8080/api/credit/v1?username="+ sUserID
+ "&password=" + sPwd;
string sResponse = GetResponse (sURL);
Response.Write (sResponse);

public static string GetResponse (string sURL)
{
    HttpWebRequest request =
(HttpWebRequest)WebRequest.Create (sURL);
    request.MaximumAutomaticRedirections = 4;
    request.Credentials = CredentialCache.DefaultCredentials;
    try
    {
        HttpWebResponse response =
(HttpWebResponse)request.GetResponse ();
        Stream receiveStream = response.GetResponseStream ();
        StreamReader readStream = new StreamReader
(receiveStream, Encoding.UTF8);
        string sResponse = readStream.ReadToEnd();
        response.Close ();
        readStream.Close ();
        return sResponse;
    } catch {
        return "";
    }
}
```

- Fonksiyon : SMS Gönderme
- Programlama Dili : ASP.NET (C#)
- Lütfen kaynak kodu kullandığınız editörde yeniden formatlayınız. Kodlarda gereksiz gördüğünüz yerleri daha sonra silebilirsiniz.

```

using System.Net;
using System.IO;
using System.Web;
using System.Xml;

string sURL = "http://panel.1sms.com.tr:8080/api/smspost/v1";
string sResponse = GetResponse(sURL, createXml());
Response.Write(sResponse);

    public static string GetResponse(string sURL, string sXml)
    {
        try
        {
            HttpWebRequest request = WebRequest.Create(new
Uri(sURL)) as HttpWebRequest;
            request.Method = "POST";
            request.ContentType = "application/x-panel-form-
urlencoded";
            request.Timeout = 5000;
            byte[] data = UTF8Encoding.UTF8.GetBytes(sXml);
            request.ContentLength = data.Length;
            using (Stream postStream = request.GetRequestStream())
            {
                postStream.Write(data, 0, data.Length);
            }
            using (HttpWebResponse response = request.GetResponse()
as HttpWebResponse)
            {
                StreamReader reader = new
StreamReader(response.GetResponseStream());
                return reader.ReadToEnd();
            }
        }
        catch (Exception ex)
        {
            return ex.Message;
        }
    }

public static string createXml()
{
    StringBuilder sb = new StringBuilder();
    XmlWriterSettings settings = new XmlWriterSettings();
    settings.Encoding = Encoding.Unicode;

    settings.Indent = true;
    settings.IndentChars = ("    ");

```

```
using (XmlWriter writer = XmlWriter.Create(sb, settings))
{
    writer.WriteStartElement("sms");
    writer.WriteElementString("username", "test");
    writer.WriteElementString("password", "test");
    writer.WriteElementString("header", "baslik");
    writer.WriteElementString("validity", "2880");
    writer.WriteStartElement("message");
    writer.WriteStartElement("gsm");
    writer.WriteElementString("no", "905559998877");
    writer.WriteEndElement(); //gsm
    writer.WriteStartElement("msg");
    writer.WriteCData("-test-message-");
    writer.WriteEndElement(); //msg
    writer.WriteEndElement(); //message
    writer.WriteEndElement(); // sms
    writer.Flush();
}
return sb.ToString();
}
```