**Sanal Pos Üretici Kart İşlemleri**

**1 -** Esnek Vadeli İşlemler

Vade tarihinin ve vade tutarının üye işyeri tarafından belirlendiği işlem tipidir. Esnek Vadeli satış işlemi için gönderilecek parametreler ve anlamları aşağıdaki gibidir.

ShopCode : Mağaza Numarası (Zorunlu)

PurchAmount : İşlem Tutarı (Zorunlu)

Currency : Para Birimi Kodu (Zorunlu)

OrderId : Sipariş Numarası

InstallmentCount : Taksit Sayısı (Zorunlu)

TxnType : İşlem Tipi (Zorunlu)

UserCode : Kullanıcı Kodu (Zorunlu)

UserPass : Kullanıcı Şifre (Zorunlu)

SecureType : Güvenlik Tipi (Zorunlu)

Pan : Kart Numarası (Zorunlu)

Expiry : Son Kullanma Tarihi (Zorunlu)

Cvv2 : Cvv

BonusAmount : Bonus Tutarı

Lang : Dil

Description : Açıklama Satırı

Description2 : Açıklama Satırı 2

CustomerName : Müşteri Adı

AgricultureTxnFlag : Üretici Kart İşlem Tipi. (Zorunlu) Esnek Vadeli işlemler için F gönderilmelidir.

FlexInstallInfo : Taksit Bilgisi (Zorunlu). Her bir ayın taksit bilgisi 20 hane olacak şekilde gönderilmedir. İlk 8 hane tarih (yyyymmdd) geri kalan 12 hane ise 2 hanesi kuruş olmak üzere tutar bilgisidir. Taksit sayısı kadar tekrarlar.

Örn: 20160526000000000500 20160622000000000500 20160723000000000500

Esnek Vadeli İşlem için kullanılacak örnek aspx kod bloğu aşağıdaki gibidir.

<%@ Page Language="C#" %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >

<html>

<head id="Head1" runat="server">

<title>Non Secure Payment</title>

</head>

<body>

<%

if (Request.Form.Count > 0)

{

String format = "{0}={1}&";

StringBuilder sb = new StringBuilder();

sb.AppendFormat(format, "ShopCode", "3123");

sb.AppendFormat(format, "PurchAmount", "15");

sb.AppendFormat(format, "Currency", "949");

sb.AppendFormat(format, "OrderId", "");

sb.AppendFormat(format, "InstallmentCount", "3");

sb.AppendFormat(format, "TxnType", "Auth");

sb.AppendFormat(format, "UserCode", "InterTestApi");

sb.AppendFormat(format, "UserPass", "3");

sb.AppendFormat(format, "SecureType", "NonSecure");

sb.AppendFormat(format, "Pan", Request.Form["Pan"]);

sb.AppendFormat(format, "Expiry", Request.Form["Expiry"]);

sb.AppendFormat(format, "Cvv2", Request.Form["Cvv2"]);

sb.AppendFormat(format, "BonusAmount", "");

sb.AppendFormat(format, "Lang", "TR");

sb.AppendFormat(format, "Description", Request.Form["Description"]);

sb.AppendFormat(format, "Description2", Request.Form["Description2"]);

sb.AppendFormat(format, "CustomerName", Request.Form["CustomerName"]);

sb.AppendFormat(format, "AgricultureTxnFlag", "F");

sb.AppendFormat(format, "FlexInstallInfo", "201605260000000005002016062200000000050020160723000000000500");

try

{

System.Net.HttpWebRequest request = (System.Net.HttpWebRequest)System.Net.WebRequest.Create("http://sanaltest.denizbank.com/MPI/Default.aspx");

byte[] parameters = System.Text.Encoding.GetEncoding("ISO-8859-9").GetBytes(sb.ToString());

request.Method = "POST";

request.ContentType = "application/x-www-form-urlencoded";

request.ContentLength = parameters.Length;

System.IO.Stream requeststream = request.GetRequestStream();

requeststream.Write(parameters, 0, parameters.Length);

requeststream.Close();

Response.Write("Request:" + sb.ToString());

System.Net.HttpWebResponse resp = (System.Net.HttpWebResponse)request.GetResponse();

System.IO.StreamReader responsereader = new System.IO.StreamReader(resp.GetResponseStream(), System.Text.Encoding.GetEncoding("ISO-8859-9"));

String responseStr = responsereader.ReadToEnd();

Response.Write("Response:" + responseStr);

if (responseStr != null)

{

string[] paramArr = responseStr.Split(';', ';');

%><table>

<%

foreach (string p in paramArr)

{

string[] nameValue = p.Split('=');

%><tr>

<td>

<%

Response.Write(nameValue[0]);

%>

</td>

<td>

<% if (nameValue.Length > 1) Response.Write(nameValue[1]);%>

</td>

</tr>

<%}

%>

</table>

<%

}

}

catch (Exception ex)

{

Response.Write("Hata " + ex.Message + " " + ex.StackTrace + (ex.InnerException != null ? ex.InnerException.Message : ""));

}

}

%>

<center>

<form method="post">

<table>

<tr>

<td>Kredi Kart Numarası:

</td>

<td>

<input type="text" name="Pan" size="20" value="5584460004364732" />

</td>

</tr>

<tr>

<td>Güvenlik Kodu:

</td>

<td>

<input type="text" name="Cvv2" size="4" value="" />

</td>

</tr>

<tr>

<td>Son Kullanma Tarihi (MMYY):

</td>

<td>

<input type="text" name="Expiry" value="0516" />

</td>

</tr>

<tr>

<td>Description

</td>

<td>

<input type="text" name="Description" />

</td>

</tr>

<tr>

<td>Description2

</td>

<td>

<input type="text" name="Description2" />

</td>

</tr>

<tr>

<td>CustomerName

</td>

<td>

<input type="text" name="CustomerName" />

</td>

</tr>

<tr>

<td align="center" colspan="2">

<input type="submit" value="Ödemeyi Tamamla" />

</td>

</tr>

</table>

</form>

</center>

</body>

</html>

**2 - Şablonlu İşlemler :**

İşlemde kullanılacak vade ve ödeme sıklığı bilgilerinin,işyerine tanımlanan bir şablon üzerinden yönetildiği işlem tipidir.Girilen kart üretici kartsa ve işlem denenen işyerine ait tanımlı bir şablon varsa, şablon içeriğindeki vadelerden birisi seçilerek taksitli işlem gerçekleştirilebilir.

Bu işlem tipinde satış işlemi yapılmadan önce üye işyeri tanımlı şablon bilgileri InterVpos tarafından sağlanan bir servis aracılığı ile online olarak çeker. Servis sadece üretici kartlar için olumlu cevap döner.

**Servis URL :** <https://ssg.denizbank.com/SPOS/DenizbankIntegration?wsdl>

**Test ortam url :** <https://sanaltest.denizbank.com/VposAuthorizationService/AuthorizationService.asmx>

**Method Adı :** [GetAgricultureTemplate](https://spos.denizbank.com/AuthorizationService/AuthorizationService.asmx?op=GetAgricultureTemplate)

Şablon sorgulama servisi için ip adresinizin tanımlanması gerekmektedir.Sorgulama servisinde kullanacağınız ip adresini [sanalpos@denizbank.com](mailto:sanalpos@denizbank.com) adresine iletmeniz gerekmektedir.

**Request Parametreleri :**

UserCode : Kullanıcı Kodu (Zorunlu)

UserPass : Kullanıcı Şifre (Zorunlu)

ShopCode : Mağaza Numarası (Zorunlu)

Pan : Kart Numarası (Zorunlu)

**Response Parametreleri :**

ResponseCode : Cevap Kodu (00 olduğu durumlar başarılı, diğer durumlar başarısız kabul edilir.)

Response Desc : Cevap Açıklaması. Dönen cevap kodununn açıklamasını içerir.

GetAgricultureTemplateData : Liste şeklinde firmanın üretici kart şablon bilgilerini döner.

Bu bilgiler;

* MaturityPeriod (Vade Süresi),
* PaymentFrequency (Ödeme Sıklığı),
* ComisionRate (Komisyon Oranı),
* TemplateName (Şablon Adı)

Üye işyeri servisi çağırarak şablon bilgilerini ekranında listeler. İlgili şablon seçildiğinde, seçilen şablon bilgileri ile beraber satış işlemini başlatır.

Şablonlu Satış işlemi için gönderilecek parametreler ve anlamları aşağıdaki gibidir.

ShopCode : Mağaza Numarası (Zorunlu)

PurchAmount : İşlem Tutarı (Zorunlu)

Currency : Para Birimi Kodu (Zorunlu)

OrderId : Sipariş Numarası

InstallmentCount : Taksit Sayısı

TxnType : İşlem Tipi (Zorunlu)

UserCode : Kullanıcı Kodu (Zorunlu)

UserPass : Kullanıcı Şifre (Zorunlu)

SecureType : Güvenlik Tipi (Zorunlu)

Pan : Kart Numarası (Zorunlu)

Expiry : Son Kullanma Tarihi (Zorunlu)

Cvv2 : Cvv

BonusAmount : Bonus Tutarı

Lang : Dil

Description : Açıklama Satırı

Description2 : Açıklama Satırı 2

CustomerName : Müşteri Adı

AgricultureTxnFlag : Üretici Kart İşlem Tipi. (Zorunlu). Şablonlu satış işlemi için T gönderilmelidir. Net Taksit işlemi için  A  gönderilmelidir.

PaymentFrequency : Servisten dönen ve yapılmak istenen ilgili şablonun Ödeme Sıklığı bilgisi gönderilir. (Zorunlu)

MaturityPeriod : Servisten dönen ve yapılmak istenen ilgili şablonun Vade Süresi bilgisi gönderilir. (Zorunlu)

Şablonlu satış işlemi için kullanılacak örnek aspx kod bloğu aşağıdaki gibidir.

<%@ Page Language="C#" %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >

<html>

<head id="Head1" runat="server">

<title>Non Secure Payment</title>

</head>

<body>

<%

if (Request.Form.Count > 0)

{

String format = "{0}={1}&";

StringBuilder sb = new StringBuilder();

sb.AppendFormat(format, "ShopCode", "3123");

sb.AppendFormat(format, "PurchAmount", "15");

sb.AppendFormat(format, "Currency", "949");

sb.AppendFormat(format, "OrderId", "");

sb.AppendFormat(format, "InstallmentCount", "3");

sb.AppendFormat(format, "TxnType", "Auth");

sb.AppendFormat(format, "UserCode", "InterTestApi");

sb.AppendFormat(format, "UserPass", "3");

sb.AppendFormat(format, "SecureType", "NonSecure");

sb.AppendFormat(format, "Pan", Request.Form["Pan"]);

sb.AppendFormat(format, "Expiry", Request.Form["Expiry"]);

sb.AppendFormat(format, "Cvv2", Request.Form["Cvv2"]);

sb.AppendFormat(format, "BonusAmount", "");

sb.AppendFormat(format, "Lang", "TR");

sb.AppendFormat(format, "Description", Request.Form["Description"]);

sb.AppendFormat(format, "Description2", Request.Form["Description2"]);

sb.AppendFormat(format, "CustomerName", Request.Form["CustomerName"]);

sb.AppendFormat(format, "AgricultureTxnFlag", "T");

sb.AppendFormat(format, "PaymentFrequency", "3");

sb.AppendFormat(format, "MaturityPeriod", "1");

try

{

System.Net.HttpWebRequest request = (System.Net.HttpWebRequest)System.Net.WebRequest.Create("http://sanaltest.denizbank.com/MPI/Default.aspx");

byte[] parameters = System.Text.Encoding.GetEncoding("ISO-8859-9").GetBytes(sb.ToString());

request.Method = "POST";

request.ContentType = "application/x-www-form-urlencoded";

request.ContentLength = parameters.Length;

System.IO.Stream requeststream = request.GetRequestStream();

requeststream.Write(parameters, 0, parameters.Length);

requeststream.Close();

Response.Write("Request:" + sb.ToString());

System.Net.HttpWebResponse resp = (System.Net.HttpWebResponse)request.GetResponse();

System.IO.StreamReader responsereader = new System.IO.StreamReader(resp.GetResponseStream(), System.Text.Encoding.GetEncoding("ISO-8859-9"));

String responseStr = responsereader.ReadToEnd();

Response.Write("Response:" + responseStr);

if (responseStr != null)

{

string[] paramArr = responseStr.Split(';', ';');

%><table>

<%

foreach (string p in paramArr)

{

string[] nameValue = p.Split('=');

%><tr>

<td>

<%

Response.Write(nameValue[0]);

%>

</td>

<td>

<% if (nameValue.Length > 1) Response.Write(nameValue[1]);%>

</td>

</tr>

<%}

%>

</table>

<%

}

}

catch (Exception ex)

{

Response.Write("Hata " + ex.Message + " " + ex.StackTrace + (ex.InnerException != null ? ex.InnerException.Message : ""));

}

}

%>

<center>

<form method="post">

<table>

<tr>

<td>Kredi Kart Numarası:

</td>

<td>

<input type="text" name="Pan" size="20" value="5584460004364732" />

</td>

</tr>

<tr>

<td>Güvenlik Kodu:

</td>

<td>

<input type="text" name="Cvv2" size="4" value="" />

</td>

</tr>

<tr>

<td>Son Kullanma Tarihi (MMYY):

</td>

<td>

<input type="text" name="Expiry" value="0516" />

</td>

</tr>

<tr>

<td>Description

</td>

<td>

<input type="text" name="Description" />

</td>

</tr>

<tr>

<td>Description2

</td>

<td>

<input type="text" name="Description2" />

</td>

</tr>

<tr>

<td>CustomerName

</td>

<td>

<input type="text" name="CustomerName" />

</td>

</tr>

<tr>

<td align="center" colspan="2">

<input type="submit" value="Ödemeyi Tamamla" />

</td>

</tr>

</table>

</form>

</center>

</body>

</html>