

# **PayUSharp**

## **PayU Türkiye .NET Kütüphanesi**

### **Kullanım Kılavuzu**

**Kütüphane Versiyonu 2.0.2.0**

20/03/2015

Giriş . . . . .	2
LiveUpdate Servisi . . . . .	2
Başlangıç . . . . .	2
LiveUpdate Siparişinin Oluşturulması . . . . .	2
LiveUpdate Siparişinin Gönderilmesi . . . . .	3
LiveUpdate Sipariş Sonucunun İşlenmesi . . . . .	4
LiveUpdate Sipariş Alanları . . . . .	4
IPN Servisi . . . . .	7
Başlangıç . . . . .	7
IPN Alanlarının İşlenmesi . . . . .	7
IPN Cevabının Oluşturulması . . . . .	8
IPN Alanları . . . . .	8
Automatic LiveUpdate Servisi . . . . .	11
Başlangıç . . . . .	11
Automatic LiveUpdate Siparişinin Oluşturulması . . . . .	11
Automatic LiveUpdate Siparişinin Gönderilmesi . . . . .	12
Automatic LiveUpdate Sipariş Sonucunun İşlenmesi . . . . .	13
Automatic LiveUpdate Alanları . . . . .	14
Token Servisi . . . . .	17
Başlangıç . . . . .	17
Yeni Satış Oluşturulması . . . . .	17
Token Bilgilerinin Alınması . . . . .	18
Token İptali . . . . .	19
Token Alanları . . . . .	19

## Giriş

**PayUSharp** kütüphanesi .NET platformu üzerinde işlem yapan işyerlerinin PayU Türkiye servisleri ile entegrasyonunu kolaylaştırmak amacı ile geliştirilmiştir. Kütüphane PayU Türkiye'nin sunduğu [LiveUpdate](#), [IPN](#), [Automatic LiveUpdate](#) ve [Token](#) servisleri için kolay kullanılabilir nesne yönelimli sınıflar sağlamaktadır.

**PayUSharp** bir adet çekirdek, 4 adet servis kütüphanesi olmak üzere 5 adet DLL'den oluşmaktadır:

1. PayUSharp.Core.dll
2. PayUSharp.LiveUpdate.dll
3. PayUSharpAutomaticLiveUpdate.dll
4. PayUSharp.IPN.dll
5. PayUSharp.Token.dll

İşyerleri kullandıkları servislere ait tüm DLL'leri ve PayUSharp.Core DLL'ini projelerine Reference olarak eklemelidirler.

## LiveUpdate Servisi

LiveUpdate, son kullanıcıların PayU Ortak Ödeme Sayfası'na yönlendirilmesi ile ödeme yaptığı entegrasyon metodudur. Bu entegrasyon Merchant'ın sipariş ile ilgili bilgileri bir HTML Form'u olarak oluşturup bu form'un PayU LiveUpdate adresine POST edilmesini sağlaması ile gerçekleşir.

PayUSharp kütüphanesi ile bahsedilen HTML Form'unun oluşturulması ve, istenirse, bu form'un otomatik olarak POST edilmesi işlemleri kolaylaştırılmıştır.

### Başlangıç

Herhangi bir LiveUpdate işlemi gerçekleştirmeden önce yeni bir `PayU.LiveUpdate.LiveUpdateService` nesnesi yaratılmalıdır. Bu nesneye geçirilmesi zorunlu olan tek parametre `signatureKey` alanıdır ve PayU'dan alınan imza anahtarı değeri geçirilmelidir.

Örnek kullanım şu şekildedir:

```
1 var service = new PayU.LiveUpdate.LiveUpdateService('signatureKey');
```

Ayrıca servisin testler sırasında kullanılabilecek `endpointUrl` parametresi de bulunmaktadır. Bu parametrenin varsayılan değerlerini PayU tarafından tavsiye edilmedikçe değiştirmemeniz önerilir.

### LiveUpdate Siparişinin Oluşturulması

LiveUpdate sipariş bilgileri PayUSharp kütüphanesinde `PayU.LiveUpdate.OrderDetails` sınıfı ile temsil edilmektedir. Yeni bir `PayU.LiveUpdate.OrderDetails` nesnesi yaratılarak ve bu nesnenin [alanları](#) sipariş bilgileri ile doldurularak bir PayUSharp LiveUpdate siparişi oluşturulur:

```
1 var order = new PayU.LiveUpdate.OrderDetails();
2
3 order.Merchant = "PAYUDEMO";
4 order.OrderRef = "6112457";
```

```

5
6 var product1 = new PayU.LiveUpdate.ProductDetails
7 {
8     Code = "Product1 code",
9     Name = "Product1 name",
10    Quantity = 2,
11    VAT = 8.00M,
12    UnitPrice = 20.00M,
13    Information = "Product1 info",
14    PriceType = PriceType.GROSS
15 }
16
17 var product2 = new PayU.LiveUpdate.ProductDetails
18 {
19     Code = "Product2 code",
20     Name = "Product2 name",
21     Quantity = 1,
22     VAT = 8.00M,
23     UnitPrice = 40.00M,
24     Information = "Product2 info",
25     PriceType = PriceType.GROSS
26 }
27
28 order.ProductDetails.Add(product1);
29 order.ProductDetails.Add(product2);
30
31 order.ShippingCosts = 10.00M;
32 order.DestinationCity = "Ankara";
33 order.DestinationState = "Ankara";
34 order.DestinationCountry = "TR";
35
36 order.BillingDetails = new PayU.LiveUpdate.BillingDetails {
37     FirstName = "Ahmet",
38     LastName = "Yilmaz",
39     Email = "ahmet.yilmaz@payu.com.tr",
40     City = "Kagithane", // Ilce/Semt
41     State = "İstanbul", // Sehir
42     CountryCode = "TR"
43 };

```

### LiveUpdate Siparişinin Gönderilmesi

LiveUpdate sürecinin tamamlanabilmesi için oluşturulmuş olan `PayU.LiveUpdate.OrderDetails` tipindeki sipariş bilgisi HTML formu olarak Web sayfasına eklenmeli ve bu form PayU LiveUpdate adresine POST edilmelidir.

PayUSharp kütüphanesi HTML form işlemleri için [Başlangıç](#) adımıyla yarattığımız `LiveUpdateService` nesnesini kullanmaktadır. Sipariş bilgisi bu nesnenin `RenderPaymentForm` metoduna verilerek bahsedilen HTML formu string olarak oluşturulabilir:

```

1 var service = new LiveUpdateService('signatureKey');
2
3 //... Order yaratılır

```

```

4
5 // sadece Submit dugmesi adi verilerek
6 string htmlForm = service.RenderPaymentForm(order, "PayU ile Ödeme Yap");
7
8 // veya hem Submit dugmesi adi hem de Form Id'si belirterek
9 string htmlFormWithId = service.RenderPaymentForm(order, "PayU ile Ödeme Yap", "PayULiveUpdateForm");

```

Oluşturulan bu HTML formu kullanılan web framework'üne uygun bir şekilde müşteriye gösterilecek sipariş sayfasına eklenmelidir. Bu işlem framework'ler arasında farklı şekillerde olmaktadır. ASP.NET Web Forms ile sayfada bir Literal alan belirlenip, form datası bu alana yazdırılabilir:

```

1 ltrLiveUpdateForm.Text = service.RenderPaymentForm(order, "PayU ile Ödeme Yap", "PayULiveUpdateForm");

```

Son olarak, sipariş bilgilerini içeren bu HTML form'unun POST metodu ile submit edilmesi gerekmektedir. Eğer dükkânınızın akışı bu işlemi kullanıcının yapması üzerine kurulu ise yukarıda yapılanlar sizin için yeterli olacaktır. Fakat akışınız bu noktada kullanıcıyı otomatik olarak Ortak Ödeme Sayfasına yönlendirmek üzerine kurulu ise, o zaman sayfanızda aşağıdaki gibi bir JavaScript kod parçasığı eklenmelidir:

```

1 <script>
2 // DIKKAT: Oluşturulmuş olan PayU form'unun Form ID'si kullanılmalı...
3 document.getElementById('PayULiveUpdateForm').submit();
4 </script>

```

### LiveUpdate Sipariş Sonucunun İşlenmesi

Eğer LiveUpdate ile ilk gönderilen sipariş bilgilerinde returnUrl alanı doldurulmuşsa, kullanıcı ödeme işlemi bitiminde PayU tarafından verilen URL'e yönlendirilecektir. Bu adresi uygulamanızdaki bir sayfa olarak tanımlamanız durumunda kullanıcılarınız ödeme işlemi sonrasında sizin sayfanıza otomatik olarak geleceklerdir. Siz de bu sayfada ödemenin referans kodu ile işlem yaparak müşteriye ödemenin alındığını söyleyebilir, ödeme detaylarını gösterebilirsiniz.

Bu sayfaya PayU dışında başka kaynaklardan istek gelmemesi için PayU tarafından gönderilen parametrelerin arasında bir de doğrulama kodu bulunmaktadır. Güvenlik nedeniyle bu kod doğrulanmadan ödeme bilgilerine güvenmemeniz çok önemlidir. Doğrulama kodu yeni yaratılan bir PayU.LiveUpdate.LiveUpdateService nesnesinin VerifyControlSignature metodu çağırılarak kontrol edilebilir:

```

1 var service = new LiveUpdateService('signatureKey');
2
3 // Verify the signature in the "ctrl" query string parameter
4 var verified = service.VerifyControlSignature(Request);

```

### LiveUpdate Sipariş Alanları

OrderDetails Alanı	Alan Tipi	PayU LiveUpdate Alanı
Merchant	string	MERCHANT
OrderRef	string	ORDER_REF
OrderDate	DateTime	ORDER_DATE
PaymentMethod	string	PAY_METHOD
OrderHash	string	ORDER_HASH

OrderDetails Alanı	Alan Tipi	PayU LiveUpdate Alanı
PricesCurrency	<b>string</b>	PRICES_CURRENCY
TokenEnable	<b>bool?</b>	LU_ENABLE_TOKEN
TokenType	TokenType?	LU_TOKEN_TYPE
ReturnUrl	<b>string</b>	BACK_REF
ShippingCosts	<b>decimal</b>	ORDER_SHIPPING
ProductDetails[idx].Name	<b>string</b>	ORDER_PNAME[idx]
ProductDetails[idx].Code	<b>string</b>	ORDER_PCODE[idx]
ProductDetails[idx].UnitPrice	<b>decimal</b>	ORDER_PRICE[idx]
ProductDetails[idx].Quantity	<b>int</b>	ORDER_QTY[idx]
ProductDetails[idx].Information	<b>string</b>	ORDER_PINFO[idx]
ProductDetails[idx].VAT	<b>decimal</b>	ORDER_VAT[idx]
ProductDetails[idx].PriceType	PriceType	ORDER_PRICE_TYPE[idx]
Discount	<b>decimal?</b>	DISCOUNT
DestinationCity	<b>string</b>	DESTINATION_CITY
DestinationState	<b>string</b>	DESTINATION_STATE
DestinationCountry	<b>string</b>	DESTINATION_COUNTRY
InstallmentOptions	<b>string</b>	INSTALLMENT_OPTIONS
BillingDetails.LastName	<b>string</b>	BILL_LNAME
BillingDetails.FirstName	<b>string</b>	BILL_FNAME
BillingDetails.Email	<b>string</b>	BILL_EMAIL
BillingDetails.PhoneNumber	<b>string</b>	BILL_PHONE
BillingDetails.CountryCode	<b>string</b>	BILL_COUNTRYCODE
BillingDetails.Fax	<b>string</b>	BILL_FAX
BillingDetails.Address	<b>string</b>	BILL_ADDRESS
BillingDetails.Address2	<b>string</b>	BILL_ADDRESS2
BillingDetails.ZipCode	<b>string</b>	BILL_ZIPCODE
BillingDetails.City	<b>string</b>	BILL_CITY
BillingDetails.State	<b>string</b>	BILL_STATE
BillingDetails.FiscalCode	<b>string</b>	BILL_FISCALCODE
BillingDetails.RegNumber	<b>string</b>	BILL_REGNUMBER
BillingDetails.Bank	<b>string</b>	BILL_BANK
BillingDetails.BankAccount	<b>string</b>	BILL_BANKACCOUNT
DeliveryDetails.LastName	<b>string</b>	DELIVERY_LNAME
DeliveryDetails.FirstName	<b>string</b>	DELIVERY_FNAME
DeliveryDetails.Email	<b>string</b>	DELIVERY_EMAIL
DeliveryDetails.PhoneNumber	<b>string</b>	DELIVERY_PHONE
DeliveryDetails.Company	<b>string</b>	DELIVERY_COMPANY
DeliveryDetails.Address	<b>string</b>	DELIVERY_ADDRESS
DeliveryDetails.Address2	<b>string</b>	DELIVERY_ADDRESS2
DeliveryDetails.ZipCode	<b>string</b>	DELIVERY_ZIPCODE
DeliveryDetails.City	<b>string</b>	DELIVERY_CITY
DeliveryDetails.State	<b>string</b>	DELIVERY_STATE
DeliveryDetails.CountryCode	<b>string</b>	DELIVERY_COUNTRYCODE

OrderDetails Alanı	Alan Tipi	PayU LiveUpdate Alanı
TestOrder	<b>bool?</b>	TESTORDER
Debug	<b>bool?</b>	DEBUG
Language	Language?	LANGUAGE
AutoMode	<b>bool?</b>	AUTOMODE
OrderTimeout	<b>int?</b>	ORDER_TIMEOUT
TimeoutURL	<b>string</b>	TIMEOUT_URL

## IPN Servisi

PayU Teknik Entegrasyon Rehberi'nde IPN sisteminin çalışması:

Bir sipariş için provizyon ve onay verildiğinde, PayU sunucusu siparişle ilgili tüm bilgileri içeren bir veri yapısını sisteminizde daha önceden ayarlanmış olan bir URL'ye gönderir. Veriler HTTP POST aracılığıyla gönderilir. Veriler ayrıca bilgilerin doğrulanması için bir imza içerecektir. İmza, istenen veri seti üzerinde bir ortak PayU/Satıcı anahtarı ile bir HMAC\_MD5 işlevi uygulayarak elde edilir (HMAC, RFC 2104'te tanımlanmaktadır).

olarak anlatılmıştır.

PayUSharp kütüphanesi ile gönderilen IPN alanlarının işlenmesi ve yanıt olarak gönderilmesi gereken bilginin doğru şekilde oluşturulması işlemleri kolaylaştırılmıştır.

### Başlangıç

Herhangi bir LiveUpdate işlemi gerçekleştirmeden önce yeni bir PayU.IPN.IPNService nesnesi yaratılmalıdır. Bu nesneye geçirilmesi zorunlu olan tek parametre `signatureKey` alanıdır ve PayU'dan alınan imza anahtarı değeri geçirilmelidir.

Örnek kullanım şu şekildedir:

```
1 var service = new PayU.IPN.IPNService('signatureKey');
```

### IPN Alanlarının İşlenmesi

Sipariş bilgilerinde verdiğiniz IPN adresinizin `http://example.com/ipn/default.aspx` olduğunu farzedelim. Sipariş onayı verildiğinde bu adrese IPN bilgileri HTTP POST olarak gönderilecektir. Sayfa kodunuzda, gönderilen bu bilgileri yorumlamak için aşağıdaki örnekte olduğu gibi **Başlangıç** adımıyla yarattığımız IPNService nesnesinin ParseRequest metodu kullanılarak yeni bir IPNRequest nesnesi yaratılmalıdır. Bu metod ile IPNRequest nesnesi yaratılırken gelen POST alanları doğru bir şekilde işlenerek nesnenin ilgili alanlarına kolay erişim için eklenecektir.

```
1 public partial class Default: System.Web.UI.Page {
2     public void Page_Load() {
3         // Some code here
4         var service = new PayU.IPN.IPNService('signatureKey');
5
6         PayU.IPN.IPNRequest ipn = service.ParseRequest(Request);
7
8         Console.WriteLine("Siparis Durumu: {0}", ipn.OrderStatus);
9         // Some code here
10    }
11 }
```

## IPN Cevabının Oluşturulması

Eğer IPN isteği başarılı bir şekilde işlendiyse HTTP 200 kodu ile PayU dökümanında belirtilen XML formatında bir cevap dönülmelidir:

PayU, aşağıdaki formatta (sayfanın herhangi bir yerinde) bir yanıt bekler: DATE|HASH

Bu cevap alanındaki Hash'i hesaplamak ve doğru XML'i oluşturmak için de [Başlangıç](#) adımıyla yarattığımız IPNService nesnesinin GenerateResponseForRequest metodu kullanılmalıdır. Bunun için örnek kod şu şekildedir:

```
1 public partial class Default : System.Web.UI.Page
2 {
3     public void Page_Load() {
4         var service = new PayU.IPN.IPNService('signatureKey');
5
6         var ipn = service.ParseRequest(Request);
7
8         Console.WriteLine("Siparis Durumu: {0}", ipn.OrderStatus);
9
10        Response.ContentType = "text/xml";
11        Response.Write(service.GenerateResponseForRequest(ipn));
12        Response.End();
13    }
14 }
```

## IPN Alanları

IPNRequest Alanı	Alan Tipi	PayU IPN POST Alanı
Products[idx].Id	string	IPN_PID[idx]
Products[idx].Name	string	IPN_PNAME[idx]
Products[idx].Code	string	IPN_PCODE[idx]
Products[idx].Information	string	IPN_INFO[idx]
Products[idx].Quantity	int	IPN_QTY[idx]
Products[idx].Price	decimal	IPN_PRICE[idx]
Products[idx].Vat	decimal	IPN_VAT[idx]
Products[idx].Version	string	IPN_VER[idx]
Products[idx].Discount	decimal	IPN_DISCOUNT[idx]
Products[idx].PromotionName	string	IPN_PROMONAME[idx]
Products[idx].DeliveredCodes	string	IPN_DELIVEREDCODES[idx]
Products[idx].Total	string	IPN_TOTAL[idx]
SaleDateAsString	string	SALEDATE
Sale	DateTime	SALEDATE
PaymentDateAsString	string	PAYMENTDATE
Payment	DateTime	PAYMENTDATE
CompleteDateAsString	string	COMPLETE_DATE
Complete	DateTime	COMPLETE_DATE



IPNRequest Alanı	Alan Tipi	PayU IPN POST Alanı
ReferenceNumber	<b>string</b>	REFNO
SellerReferenceNumber	<b>string</b>	REFNOEXT
OrderNumber	<b>string</b>	ORDERNO
OrderStatus	IPNOrderStatus	ORDERSTATUS
PaymentMethod	<b>string</b>	PAYMETHOD
PaymentMethodCode	<b>string</b>	PAYMETHOD_CODE
TotalPaidAmount	<b>decimal</b>	IPN_PAID_AMOUNT
InstallmentProgramName	<b>string</b>	IPN_INSTALLMENTS_PROGRAM
InstallmentNumber	<b>string</b>	IPN_INSTALLMENTS_NUMBER
InstallmentProfit	<b>string</b>	IPN_INSTALLMENTS_PROFIT
FirstName	<b>string</b>	FIRSTNAME
LastName	<b>string</b>	LASTNAME
IdentityNumber	<b>string</b>	IDENTITY_NO
IdentityIssuer	<b>string</b>	IDENTITY_ISSUER
IdentityCNP	<b>string</b>	IDENTITY_CNP
Company	<b>string</b>	COMPANY
RegistrationNumber	<b>string</b>	REGISTRATIONNUMBER
FiscalCode	<b>string</b>	FISCALCODE
CompanyBankName	<b>string</b>	CBANKNAME
CompanyBankAccount	<b>string</b>	CBANKACCOUNT
Address1	<b>string</b>	ADDRESS1
Address2	<b>string</b>	ADDRESS2
City	<b>string</b>	CITY
State	<b>string</b>	STATE
ZipCode	<b>string</b>	ZIPCODE
Country	<b>string</b>	COUNTRY
PhoneNumber	<b>string</b>	PHONE
FaxNumber	<b>string</b>	FAX
CustomerEmail	<b>string</b>	CUSTOMEREMAIL
DeliveryFirstName	<b>string</b>	FIRSTNAME_D
DeliveryLastName	<b>string</b>	LASTNAME_D
DeliveryCompany	<b>string</b>	COMPANY_D
DeliveryAddress1	<b>string</b>	ADDRESS1_D
DeliveryAddress2	<b>string</b>	ADDRESS2_D
DeliveryCity	<b>string</b>	CITY_D
DeliveryState	<b>string</b>	STATE_D
DeliveryZipCode	<b>string</b>	ZIPCODE_D
DeliveryCountry	<b>string</b>	COUNTRY_D
DeliveryPhoneNumber	<b>string</b>	PHONE_D
IpAddress	<b>string</b>	IPADDRESS
Currency	<b>string</b>	CURRENCY
TotalGeneral	<b>decimal</b>	IPN_TOTALGENERAL
Shipping	<b>decimal</b>	IPN_SHIPPING

IPNRequest Alanı	Alan Tipi	PayU IPN POST Alanı
GlobalDiscount	<b>decimal</b>	IPN_GLOBALDISCOUNT
Commission	<b>decimal</b>	IPN_COMMISSION
Date	<b>string</b>	IPN_DATE
Hash	<b>string</b>	HASH
CreditCardToken	<b>string</b>	IPN_CC_TOKEN
CreditCardMask	<b>string</b>	IPN_CC_MASK
CreditCardExpiryDate	<b>string</b>	IPN_CC_EXP_DATE

## Automatic LiveUpdate Servisi

Automatic LiveUpdate ödeme bilgilerinin işyeri tarafından kullanıcından alındığı ve işlenmek için sipariş bilgileri ile birlikte PayU'ya gönderildiği entegrasyon metodudur. Bu entegrasyonda kullanıcılar işyerinin sipariş sayfalarından hiç ayrılmazlar (3D Secure durumu hariç) ve ödeme Merchant ile PayU arasında bitirilir.

PayUSharp kütüphanesi ile bahsedilen sipariş ve ödeme bilgilerinin oluşturulması, gereken formatta PayU'ya iletilmesi ve PayU'dan dönen cevabın işlenmesi işlemleri kolaylaştırılmıştır.

### Başlangıç

Herhangi bir Automatic LiveUpdate işlemi gerçekleştirmeden önce yeni bir `PayU.AutomaticLiveUpdate.ALUService` nesnesi yaratılmalıdır. Bu nesneye geçirilmesi zorunlu olan tek parametre `signatureKey` alanıdır ve PayU'dan alınan imza anahtarı değeri geçirilmelidir.

Örnek kullanım şu şekildedir:

```
1 var service = new PayU.AutomaticLiveUpdate.ALUService('signatureKey');
```

Ayrıca servisin testler sırasında kullanılabilecek `endpointUrl` ve `ignoreSSLCertificate` parametreleri de bulunmaktadır. Bu parametrelerin varsayılan değerlerini PayU tarafından tavsiye edilmedikçe değiştirmeniz önerilir.

### Automatic LiveUpdate Siparişinin Oluşturulması

Automatic LiveUpdate sipariş bilgileri PayUSharp kütüphanesinde `PayU.AutomaticLiveUpdate.OrderDetails` sınıfı ile temsil edilmektedir. Yeni bir `PayU.AutomaticLiveUpdate.OrderDetails` nesnesi yaratılarak ve bu nesnenin [alanları](#) sipariş bilgileri ile doldurularak bir PayUSharp Automatic LiveUpdate siparişi oluşturulur:

```
1 var order = new PayU.AutomaticLiveUpdate.OrderDetails();
2
3 order.Merchant = "PAYUDEMO";
4
5 order.OrderRef = "EXT_6112457";
6 order.OrderDate = DateTime.Now;
7
8 var product1 = new PayU.AutomaticLiveUpdate.ProductDetails
9 {
10     Code = "TCK1",
11     Name = "Ticket1",
12     Quantity = 1,
13     UnitPrice = 5.00M,
14     Information = "Barcelona flight"
15 };
16
17 var product2 = new PayU.AutomaticLiveUpdate.ProductDetails
18 {
19     Code = "TCK2",
20     Name = "Ticket2",
21     Quantity = 1,
22     UnitPrice = 10.00M,
23     Information = "London Flight"
```

```

24 };
25
26 order.ProductDetails.Add(product1);
27 order.ProductDetails.Add(product2);
28
29 order.PricesCurrency = "TRY";
30
31 order.CardDetails = new PayUAutomaticLiveUpdate.CardDetails
32 {
33     CardNumber = "4242424242424242",
34     ExpiryMonth = "12",
35     ExpiryYear = "2015",
36     CVV = "000",
37     CardOwnerName = "Ahmet Yılmaz"
38 };
39
40 order.BillingDetails = new PayUAutomaticLiveUpdate.BillingDetails
41 {
42     FirstName = "Ahmet",
43     LastName = "Yılmaz",
44     Email = "ahmet.yilmaz@payu.com.tr",
45     PhoneNumber = "2122223344",
46     CountryCode = "TR",
47     Address = "Billing address", //optional
48     Address2 = "Billing address ", //optional
49     ZipCode = "12345", //optional
50     City = "Kagithane", //optional - İlçe/Semt
51     State = "Istanbul", //optional - Şehir
52     Fax = "1234567890" //optional
53 };
54 order.DeliveryDetails = new PayUAutomaticLiveUpdate.DeliveryDetails
55 {
56     LastName = "John", //optional
57     FirstName = "Doe", //optional
58     Email = "shopper@payu.ro", //optional
59     PhoneNumber = "1234567890", //optional
60     Company = "Company Name", //optional
61     Address = "Delivery Address", //optional
62     Address2 = "Delivery Address", //optional
63     ZipCode = "12345", //optional
64     City = "City", //optional
65     State = "State / Dept.", //optional
66     CountryCode = "TR" //optional
67 };
68 order.ReturnUrl = "http://example.com/AutomaticLiveUpdate/ThreeDS.aspx";
69 order.ClientIpAddress = Request.UserHostAddress;

```

### Automatic LiveUpdate Siparişinin Gönderilmesi

Automatic LiveUpdate sürecinin tamamlanabilmesi için oluşturulmuş olan PayUAutomaticLiveUpdate.OrderDetails tipindeki sipariş bilgisi sunucu tarafında PayU Automatic LiveUpdate adresine POST edilmektedir.

PayUSharp kütüphanesi bu POST işlemi için [Başlangıç](#) adımıyla yarattığımız ALUService nesnesini kullanmaktadır. Sipariş bilgisi bu nesnenin ProcessPayment metoduna verilerek bu POST işleminin gerçekleşmesi sağlanır:

```
1 var response = service.ProcessPayment(order);
```

**DİKKAT:** Bu metod çağırıldığında sunucu kodu PayU Automatic LiveUpdate sayfası ile HTTP POST üzerinden iletişime geçer ve PayU tarafından işlem yapıp cevap dönene kadar bu metod dönmaz. Bu işlemin kullanıcıyı sayfada bekleteceği ve sunucu tarafında request block edeceği unutulmamalıdır.

### Automatic LiveUpdate Sipariş Sonucunun İşlenmesi

ProcessPayment metodunun çağırılması sonucunda dönen cevap PayU.AutomaticLiveUpdate.ALUResponse tipinden olacaktır. Bu cevap nesnesi gerçekleşen veya hata veren ödeme sonucunu veren [alanlara](#) sahiptir.

Normal akış dışında ProcessPayment metodu çağırıldığında iki olaya dikkat etmek gerekmektedir:

1. PayU ile iletişim sırasında bir hata olması durumunda PayU.PayUException tipinden bir exception alınacaktır. Bu nedenle bu metod çağırısı sırasında PayU.PayUException hataları yakalanmalı ve uygun şekilde işlenmelidir.
2. Eğer ödeme işlemi 3D-Secure ile yapılması gerekiyorsa o zaman PayU.AutomaticLiveUpdate.ALUResponse nesnesinin Is3DSResponse alanı **true** olacaktır. Bu durumda son kullanıcı ALUResponse nesnesinin Url3DS alanında döndürülen adrese yönlendirilmelidir. Bu adres PayU tarafından yönetilmektedir. Kullanıcı bu adreste 3D Secure işlemini tamamladıktan sonra PayU tarafından siparişte verilen PayU.AutomaticLiveUpdate.OrderDetails nesnesinin returnUrl alanındaki adrese geri yönlendirilecektir. Bu yönlendirme Automatic LiveUpdate sipariş sonucunu barındırmakta ve bu sonuç PayU.AutomaticLiveUpdate.ALUResponse.FromHttpRequest metodu kullanılarak alınmaktadır.

Tipik bir Automatic LiveUpdate request/response şu şekilde olmalıdır:

```
1 public partial class Default : System.Web.UI.Page
2 {
3     public void Page_Load() {
4         var service = new PayU.AutomaticLiveUpdate.ALUService('signatureKey');
5
6         var order = new PayU.AutomaticLiveUpdate.OrderDetails();
7
8         // ...
9         // order alanlari atanir
10        // ...
11
12        // 3D Secure zorunlu ise 3D Secure islemi sonrasini istedigimiz adres
13        order.ReturnUrl = "http://example.com/AutomaticLiveUpdate/ThreeDS.aspx";
14
15        try
16        {
17            var response = service.ProcessPayment(parameters);
```

```

18
19     if (response.Is3DSResponse) {
20         // 3D Secure zorunlu. Kullaniciyi verilen adrese yonlendirmeliyiz.
21         Response.Redirect(response.Url3DS);
22         Response.End();
23     }
24
25     // Buraya kadar geldiysek ALU'dan cevap gelmis demektir.
26     // response nesnesinin alanlarına bakarak sonucu anlayabiliriz.
27     var status = response.Status
28
29     if (status == Status.Success)
30     {
31         // Basarili bir islem durumu
32     }
33 }
34 catch (PayUException ex)
35 {
36     // Hata islemleri yapilir
37 }
38 }
39 }

```

```

1 public partial class ThreeDS : System.Web.UI.Page
2 {
3     public void Page_Load() {
4         // 3D Secure sonucu cevap gelmis. Cevap alanlari POST edilmiş olarak gelecegi
5         // icin bu alanlari kullanarak yeni bir ALUResponse yaratmalıyız.
6         var response = ALUResponse.FromHttpRequest (Request);
7
8         // Buraya kadar geldiysek ALU'dan cevap gelmiş demektir.
9         // response nesnesinin alanlarına bakarak sonucu anlayabiliriz.
10        var status = response.Status
11
12        if (status == Status.Success)
13        {
14            // Basarili bir islem durumu
15        }
16    }
17 }

```

### Automatic LiveUpdate Alanları

OrderDetails Alanı	Alan Tipi	PayU ALU İstek Alanı
Merchant	<b>string</b>	MERCHANT
OrderRef	<b>string</b>	ORDER_REF
OrderDate	DateTime	ORDER_DATE

OrderDetails Alanı	Alan Tipi	PayU ALU İstek Alanı
PaymentMethod	<b>string</b>	PAY_METHOD
OrderHash	<b>string</b>	ORDER_HASH
PricesCurrency	<b>string</b>	PRICES_CURRENCY
TokenEnable	<b>bool?</b>	LU_ENABLE_TOKEN
TokenType	TokenType?	LU_TOKEN_TYPE
ProductDetails[idx].Name	<b>string</b>	ORDER_PNAME[idx]
ProductDetails[idx].Code	<b>string</b>	ORDER_PCODE[idx]
ProductDetails[idx].UnitPrice	<b>decimal</b>	ORDER_PRICE[idx]
ProductDetails[idx].Quantity	<b>int</b>	ORDER_QTY[idx]
ProductDetails[idx].Information	<b>string</b>	ORDER_PINFO[idx]
ProductDetails[idx].Version	<b>string</b>	ORDER_VER[idx]
BillingDetails.LastName	<b>string</b>	BILL_LNAME
BillingDetails.FirstName	<b>string</b>	BILL_FNAME
BillingDetails.Email	<b>string</b>	BILL_EMAIL
BillingDetails.PhoneNumber	<b>string</b>	BILL_PHONE
BillingDetails.CountryCode	<b>string</b>	BILL_COUNTRYCODE
BillingDetails.Fax	<b>string</b>	BILL_FAX
BillingDetails.Address	<b>string</b>	BILL_ADDRESS
BillingDetails.Address2	<b>string</b>	BILL_ADDRESS2
BillingDetails.ZipCode	<b>string</b>	BILL_ZIPCODE
BillingDetails.City	<b>string</b>	BILL_CITY
BillingDetails.State	<b>string</b>	BILL_STATE
BillingDetails.FiscalCode	<b>string</b>	BILL_FISCALCODE
BillingDetails.RegNumber	<b>string</b>	BILL_REGNUMBER
BillingDetails.Bank	<b>string</b>	BILL_BANK
BillingDetails.BankAccount	<b>string</b>	BILL_BANKACCOUNT
CardDetails.CardNumber	<b>string</b>	CC_NUMBER
CardDetails.ExpiryMonth	<b>string</b>	EXP_MONTH
CardDetails.ExpiryYear	<b>string</b>	EXP_YEAR
CardDetails.CardType	<b>string</b>	CC_TYPE
CardDetails.CVV	<b>string</b>	CC_CVV
CardDetails.CardOwnerName	<b>string</b>	CC_OWNER
DeliveryDetails.LastName	<b>string</b>	DELIVERY_LNAME
DeliveryDetails.FirstName	<b>string</b>	DELIVERY_FNAME
DeliveryDetails.Email	<b>string</b>	DELIVERY_EMAIL
DeliveryDetails.PhoneNumber	<b>string</b>	DELIVERY_PHONE
DeliveryDetails.Company	<b>string</b>	DELIVERY_COMPANY
DeliveryDetails.Address	<b>string</b>	DELIVERY_ADDRESS
DeliveryDetails.Address2	<b>string</b>	DELIVERY_ADDRESS2
DeliveryDetails.ZipCode	<b>string</b>	DELIVERY_ZIPCODE
DeliveryDetails.City	<b>string</b>	DELIVERY_CITY
DeliveryDetails.State	<b>string</b>	DELIVERY_STATE
DeliveryDetails.CountryCode	<b>string</b>	DELIVERY_COUNTRYCODE

OrderDetails Alanı	Alan Tipi	PayU ALU İstek Alanı
ReturnUrl	<b>string</b>	BACK_REF
ClientIpAddress	<b>string</b>	CLIENT_IP
ClientTime	DateTime?	CLIENT_TIME
SelectedInstallmentNumber	<b>int?</b>	SELECTED_INSTALLMENTS_NUMBER
CardProgramName	<b>string</b>	CARD_PROGRAM_NAME
OrderTimeout	<b>int?</b>	ORDER_TIMEOUT

ALUResponse Alanı	Alan Tipi	PayU ALU Cevap Alanı
RefNo	<b>string</b>	REFNO
Alias	<b>string</b>	ALIAS
Status	Status	STATUS
ReturnCode	<b>string</b>	RETURN_CODE
ReturnMessage	<b>string</b>	RETURN_MESSAGE
Date	<b>string</b>	DATE
Currency	<b>string</b>	CURRENCY
InstallmentNumber	<b>string</b>	INSTALLMENTS_NO
CardProgramName	<b>string</b>	CARD_PROGRAM_NAME
Url3DS	<b>string</b>	URL_3DS
Hash	<b>string</b>	HASH
AuthCode	<b>string</b>	AUTH_CODE
RRN	<b>string</b>	RRN
BankResponse	<b>string</b>	RESPONSE
BankReturnCode	<b>string</b>	PROCRETURNCODE
TransactionId	<b>string</b>	TRANSID
BankErrorMessage	<b>string</b>	ERRORMESSAGE
BankClientId	<b>string</b>	CLIENTID
BankOrderId	<b>string</b>	OID
BankMDStatus	<b>string</b>	MDSTATUS
BankMDErrorMessage	<b>string</b>	MDERRORMSG
BankTransactionStatus	<b>string</b>	TXSTATUS
ECI	<b>string</b>	ECI
XID	<b>string</b>	XID
CAVV	<b>string</b>	CAVV
PAN	<b>string</b>	PAN
ExpiryYear	<b>string</b>	EXPYEAR
ExpiryMonth	<b>string</b>	EXPMONTH



## Token Servisi

Token servisi işyerlerinin bir sipariş sırasında müşterilerden aldıkları ödeme bilgilerini PayU aracılığı ile saklayabildikleri ve aynı müşterinin sonraki siparişlerinde kaydedilmiş ödeme bilgilerini kullanarak işlem yapabildikleri bir hizmettir. PayU kaydedilmesi istenen ödeme bilgilerini işyeri adına belirli bir süre PCI-DSS güvenlik kurallarına uygun bir şekilde saklar ve bu bilgilere karşılık bir token bilgisi üretir. Üretilen token kendi başına bir anlam taşımayan ama işyeri tarafından güvenli bir şekilde iletilindiğinde ödeme bilgisinin kullanılmasını sağlayan bir karakter dizisidir.

Token işlemleri **Tek Tıkla Ödeme** veya **Tekrarlayan Ödeme** gibi müşterilerden tekrar kart bilgisi girmesinin istenilmeyeceği akışlarda kullanılmak üzere yaratılmıştır.

### Başlangıç

PayUSharp kütüphanesi ile herhangi bir Token işlemi gerçekleştirmeden önce yeni bir `PayU.Token.TokenService` nesnesi yaratılmalıdır. Bu nesneye geçirilmesi zorunlu olan parametreler `merchant` ve `signatureKey` alanlarıdır ve bu alanlara PayU'dan alınan işyeri ismi ve imza anahtarı değerleri geçirilmelidir.

Örnek kullanım şu şekildedir:

```
1 var service = new PayU.Token.TokenService('merchant', 'signatureKey');
```

Ayrıca servisin testler sırasında kullanılacak `endpointUrl` ve `ignoreSSLCertificate` parametreleri de bulunmaktadır. Bu parametrelerin varsayılan değerlerini PayU tarafından tavsiye edilmedikçe değiştirmeniz önerilir.

Her token işleminde ödeme bilgisi yerine kullanmak için son kullanıcının PayU tarafında saklanan kart bilgisini temsil eden bir token alanı gereklidir. Bu token bilgisi [LiveUpdate](#) veya [Automatic LiveUpdate](#) servislerinde sipariş bilgilerinde `TokenEnable` ve `TokenType` alanlarının doldurularak gönderilmesi ve akabinde gelecek olan [IPN](#) bilgisinde `CreditCardToken` alanının okunması ile oluşturulur:

```
1 // LiveUpdate isteminde
2 var order = new PayU.LiveUpdate.OrderDetails();
3 order.TokenEnable = true;
4 order.TokenType = PayU.Core.Base.TokenType.PAY_BY_CLICK; // veya PAY_ON_TIME
5
6
7 // Daha sonra gelen IPN isteginde
8 var request = service.ParseRequest(Request);
9 var token = request.CreditCardToken;
```

### Yeni Satış Oluşturulması

Yeni bir satış yapmak için, [Başlangıç](#) adımıyla yaratılmış olan `TokenService` nesnesinin `NewSale` metodu bu müşteri için alınmış olan token, satış için işyeri tarafında tekil belirteç olarak kullanılacak olan `orderRef`

, satış bedelini temsil eden amount ve opsiyonel olarak satış para birimini temsilen currency bilgileri ile çağırılmalıdır. currency alanı verilmezse para biriminin TRY (Türk Lirası) olduğu varsayılacaktır. Bu metod TokenResponse tipinden bir sonuç döndürecektir. Bu sonuç bilgisinin [alanları](#) okunarak isteğin başarılı olarak gerçekleşip gerçekleşmediği ve varsa hata mesajları öğrenilebilir.

```
1 var service = new PayU.Token.TokenService('merchant', 'signatureKey');
2
3 var response = service.NewSale('token', 'Order-1234acf34', 12.50M);
4
5 if (response.Code == 0) {
6     // Sonuc basarili
7 } else {
8     // Sonuc basarisiz
9     var message = response.Message;
10 }
```

### Token Bilgilerinin Alınması

Elinizdeki bir token'a ait sipariş geçmişi, token'ın geçerlilik durumu ve son kullanma tarihi gibi bilgilere ulaşmak için [Başlangıç](#) adımıyla oluşturulmuş olan TokenService nesnesinin GetInfo metodu çağırılmalıdır. Bu metod sadece token bilgisini istemekte ve TokenResponse tipinden bir sonuç döndürmektedir. Sonuç bilgisinin [alanları](#) okunarak isteğin başarılı olarak gerçekleşip gerçekleşmediği, varsa hata mesajları ve token'a ait bilgiler öğrenilebilir.

```
1 var service = new PayU.Token.TokenService('merchant', 'signatureKey');
2
3 var response = service.GetInfo('token');
4
5 if (response.Code == 0) {
6     // Sonuc basarili
7     var status = response.TokenStatus;
8     foreach (var entry in response.History)
9     {
10         var index = entry.Key;
11         var date = entry.Value.Date;
12         var refNumber = entry.Value.ReferenceNumber;
13         var amount = entry.Value.Amount;
14         var currency = entry.Value.Currency;
15         // Yukaridaki bilgiler kullanılarak token gecmisi islenebilir
16     }
17
18 } else {
19     // Sonuc basarisiz
20     var message = response.Message;
21 }
```

## Token İptali

Elinizdeki bir token'ı müşteri talebi üzerine veya artık kullanılmadığı için iptal etmek isteyebilirsiniz. Bu durumda **Başlangıç** adımında yaratılmış olan `TokenService` nesnesinin `Cancel` metodu çağırılmalıdır. Bu metoda token bilgisi ve opsiyonel olarak iptal sebebini belirten `reason` bilgisi geçirilmelidir. Bu metod `TokenResponse` tipinden bir sonuç döndürmektedir ve sonuç bilgisinin **alanları** okunarak isteğin başarılı olarak gerçekleşip gerçekleşmediği ve varsa hata mesajları öğrenilebilir.

```
1 var service = new PayU.Token.TokenService('merchant', 'signatureKey');
2
3 var response = service.Cancel('token', 'musteri iptali');
4
5 if (response.Code == 0) {
6     // Iptal basarili
7 } else {
8     // Iptal basarisiz
9     var message = response.Message;
10 }
```

## Token Alanları

TokenResponse Alanı	Alan Tipi	PayU Token Cevap Alanı
Code	<b>int</b>	code
Message	<b>string</b>	message
Token	<b>string</b>	TOKEN
TokenStatus	<b>string</b>	TOKEN_STATUS
ExpirationDate	<b>string</b>	EXPIRATION_DATE

TokenHistory Alanı	Alan Tipi	PayU Token Cevap Alanı
Date	<b>string</b>	TOKEN_DATE
ReferenceNumber	<b>string</b>	REF_NO
Amount	<b>decimal</b>	AMOUNT
Currency	<b>string</b>	CURRENCY