

# PayUSharp - PayU Türkiye .NET Kütüphanesi Kullanım Klavuzu

**Kütüphane Versiyonu:** 2.0.0.0

**Tarih:** 27/10/2014

## LiveUpdate (LU)

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.

## Ayarlar

Herhangi bir PayU işlemi gerçekleştirilmeden önce (tercihen 1 kere uygulama başlangıcında) PayUSharp kütüphanesinin ayarlarının doğru bir şekilde yapılması gerekmektedir. Bunun için PayU.Configuration sınıfı kullanılmaktadır.

LiveUpdate için zorunlu ayarlar SignatureKey ve Environment alanlarıdır. Örnek kullanım şu şekildedir:

```
1 PayU.Configuration.Instance
2   .SetSignatureKey('signaturekey')
3   .SetEnvironment("https://secure.payuodeme.com/order/");
```

## 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 = "Yılmaz",
39     Email = "ahmet.yilmaz@payu.com.tr",
40     City = "Kathane", // 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 `PayU.LiveUpdate.LiveUpdateRequest` sınıfını kullanmaktadır. Sipariş bilgisi ile yaratılan yeni bir `PayU.LiveUpdate.LiveUpdateRequest` nesnesinin `RenderPaymentForm` metodu çağırılarak bu HTML formu string olarak oluşturulabilir:

```

1 var request = new LiveUpdateRequest(order);
2

```

```

3 // sadece Submit düğmesi 1ad verilerek
4 string htmlForm = request.RenderPaymentForm("PayU ile Ödeme Yap");
5
6 // veya hem Submit düğmesi 1ad hem de Form Id'si belirterek
7 string htmlFormWithId = request.RenderPaymentForm("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 var request = new LiveUpdateRequest(order);
2
3 ltrLiveUpdateForm.Text = request.RenderPaymentForm("PayU ile Ödeme Yap", "
    PayULiveUpdateForm");

```

Son olarak, sipariş bilgilerini içeren bu HTML form'unun POST metodu ile submit edilmesi gerekmektedir. Eğer dükkanınızın 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 parçası eklenmelidir:

```

1 <script>
2 // DİKKAT: ŞşOluturulmu olan PayU form'unun Form ID'si 1ad kullanı1mal...
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 PayU.LiveUpdate.LiveUpdateRequest sınıfının VerifyControlSignature metodu çağırılarak kontrol edilebilir:

```

1 // Verify the signature in the "ctrl" query string parameter
2 var verified = LiveUpdateRequest.VerifyControlSignature(Request);

```

### LiveUpdate Sipariş Alanları

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

OrderDetails Alanı	Alan Tipi	PayU LiveUpdate Alanı
DeliveryDetails.CountryCode	<b>string</b>	DELIVERY_COUNTRYCODE
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

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.

## Ayarlar

Herhangi bir PayU işlemi gerçekleştirilmeden önce (tercihen 1 kere uygulama başlangıcında) PayUSharp kütüphanesinin ayarlarının doğru bir şekilde yapılması gerekmektedir. Bunun için PayU.Configuration sınıfı kullanılmaktadır.

IPN için zorunlu alan SignatureKey alanıdır. Örnek kullanım şu şekildedir:

```
1 PayU.Configuration.Instance.SetSignatureKey('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 IPNRequest.FromHttpRequest 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         IPNRequest ipn = IPNRequest.FromHttpRequest(Request);
5
6         Console.WriteLine("Siparis Durumu: {0}", ipn.OrderStatus);
7         // Some code here
8     }
9 }

```

### IPN Cevabının Oluşturulması

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

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 IPNRequest nesnesinin GenerateResponse 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 ipn = IPNRequest.FromHttpRequest(Request);
5
6         Console.WriteLine("Siparis Durumu: {0}", ipn.OrderStatus);
7
8         Response.ContentType = "text/xml";
9         Response.Write(ipn.GenerateResponse());
10        Response.End();
11    }
12 }

```

### IPN Alanları

IPNRequest Alanı	Alan Tipi	PayU IPN POST Alanı
Products[idx].Id	<b>string</b>	IPN_PID[idx]
Products[idx].Name	<b>string</b>	IPN_PNAME[idx]
Products[idx].Code	<b>string</b>	IPN_PCODE[idx]
Products[idx].Information	<b>string</b>	IPN_INFO[idx]
Products[idx].Quantity	<b>int</b>	IPN_QTY[idx]
Products[idx].Price	<b>decimal</b>	IPN_PRICE[idx]

IPNRequest Alanı	Alan Tipi	PayU IPN POST Alanı
Products[idx].Vat	<b>decimal</b>	IPN_VAT[idx]
Products[idx].Version	<b>string</b>	IPN_VER[idx]
Products[idx].Discount	<b>decimal</b>	IPN_DISCOUNT[idx]
Products[idx].PromotionName	<b>string</b>	IPN_PROMONAME[idx]
Products[idx].DeliveredCodes	<b>string</b>	IPN_DELIVEREDCODES[idx]
Products[idx].Total	<b>string</b>	IPN_TOTAL[idx]
SaleDateAsString	<b>string</b>	SALEDATE
Sale	DateTime	SALEDATE
PaymentDateAsString	<b>string</b>	PAYMENTDATE
Payment	DateTime	PAYMENTDATE
CompleteDateAsString	<b>string</b>	COMPLETE_DATE
Complete	DateTime	COMPLETE_DATE
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

IPNRequest Alanı	Alan Tipi	PayU IPN POST Alanı
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
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 (ALU)

Automatic LiveUpdate ödeme bilgilerinin Merchant tarafından kullanıcıdan alındığı ve işlenmek için sipariş bilgileri ile birlikte PayU'ya gönderildiği entegrasyon metodudur. Bu entegrasyonda kullanıcılar Merchant'ın 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.

## Ayarlar

Herhangi bir PayU işlemi gerçekleştirilmeden önce (tercihen 1 kere uygulama başlangıcında) PayUSharp kütüphanesinin ayarlarının doğru bir şekilde yapılması gerekmektedir. Bunun için PayU.Configuration sınıfı kullanılmaktadır.

Automatic LiveUpdate için zorunlu ayarlar SignatureKey ve Environment alanlarıdır. Örnek kullanım şu şekildedir:

```

1 PayU.Configuration.Instance
2     .SetSignatureKey('signaturekey')
3     .SetEnvironment("https://secure.payuodeme.com/order/");

```



## 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 PayU.AutomaticLiveUpdate.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 = "Kathane", //optional - İlce/Semt
51     State = "Istanbul", //optional - Sehir
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 edilmelidir.

PayUSharp kütüphanesi bu POST işlemi için PayUAutomaticLiveUpdate.AluRequest sınıfını kullanmaktadır. Sipariş bilgisi PayUAutomaticLiveUpdate.AluRequest sınıfının ProcessPayment metoduna verilerek bu POST işleminin gerçekleşmesi sağlanır:

```

1 var response = PayUAutomaticLiveUpdate.AluRequest.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 dönmez. Bu işlemin kullanıcıyı sayfada bekleteceği ve sunucu tarafında request block edeceği unutulmamalıdır.

### Automatic LiveUpdate Sipariş Sonucunun İşlenmesi

AluRequest.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 order = new PayU.AutomaticLiveUpdate.OrderDetails();
5
6         // ...
7         // order alanları atanır
8         // ...
9
10        // 3D Secure zorunlu ise 3D Secure işlemi sonrasında sonucun dönmesini gördüğümüz
11        // adres
12        order.ReturnUrl = "http://example.com/AutomaticLiveUpdate/ThreeDS.aspx";
13
14        try
15        {
16            var response = AluRequest.ProcessPayment(parameters);
17
18            if (response.Is3DSResponse) {
```

```

18         // 3D Secure zorunlu. Kullanıcı verilen adrese yönlendirmeliyiz.
19         Response.Redirect(response.Url3DS);
20         Response.End();
21     }
22
23     // Buraya kadar geldiysek ALU'dan cevap gelmiş demektir.
24     // response nesnesinin alanlarına bakarak sonucu anlayabiliriz.
25     var status = response.Status
26
27     if (status == Status.Success)
28     {
29         // Başarılı bir işlem durumu
30     }
31 }
32 catch (PayuException ex)
33 {
34     // Hata işlemleri yapılır
35 }
36 }
37 }

```

```

1 public partial class ThreeDS : System.Web.UI.Page
2 {
3     public void Page_Load() {
4         // 3D Secure sonucu cevap gelmiş. Cevap alanlar POST edilmiş olarak geleceği
5         // için bu alanlar kullanarak yeni bir AluResponse oluşturmalıyız.
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            // Başarılı bir işlem 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
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
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
ReturnUrl	<b>string</b>	BACK_REF

OrderDetails Alanı	Alan Tipi	PayU ALU İstek Alanı
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