

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 = "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 `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 dugmesi adi verilerek
4 string htmlForm = request.RenderPaymentForm("PayU ile Ödeme Yap");
5
6 // veya hem Submit dugmesi adi 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 parçasığı eklenmelidir:

```

1 <script>
2 // DİKKAT: 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 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	string	MERCHANT
OrderRef	string	ORDER_REF
OrderDate	DateTime	ORDER_DATE
PaymentMethod	string	PAY_METHOD
OrderHash	string	ORDER_HASH
PricesCurrency	string	PRICES_CURRENCY
TokenEnable	bool?	LU_ENABLE_TOKEN
TokenType	TokenType?	LU_TOKEN_TYPE
ReturnUrl	string	BACK_REF
ShippingCosts	decimal	ORDER_SHIPPING
ProductDetails[idx].Name	string	ORDER_PNAME[idx]
ProductDetails[idx].Code	string	ORDER_PCODE[idx]
ProductDetails[idx].UnitPrice	decimal	ORDER_PRICE[idx]
ProductDetails[idx].Quantity	int	ORDER_QTY[idx]
ProductDetails[idx].Information	string	ORDER_PINFO[idx]
ProductDetails[idx].VAT	decimal	ORDER_VAT[idx]
ProductDetails[idx].PriceType	PriceType	ORDER_PRICE_TYPE[idx]
Discount	decimal?	DISCOUNT
DestinationCity	string	DESTINATION_CITY
DestinationState	string	DESTINATION_STATE
DestinationCountry	string	DESTINATION_COUNTRY
InstallmentOptions	string	INSTALLMENT_OPTIONS
BillingDetails.LastName	string	BILL_LNAME
BillingDetails.FirstName	string	BILL_FNAME
BillingDetails.Email	string	BILL_EMAIL
BillingDetails.PhoneNumber	string	BILL_PHONE
BillingDetails.CountryCode	string	BILL_COUNTRYCODE
BillingDetails.Fax	string	BILL_FAX
BillingDetails.Address	string	BILL_ADDRESS
BillingDetails.Address2	string	BILL_ADDRESS2
BillingDetails.ZipCode	string	BILL_ZIPCODE
BillingDetails.City	string	BILL_CITY
BillingDetails.State	string	BILL_STATE
DeliveryDetails.LastName	string	DELIVERY_LNAME
DeliveryDetails.FirstName	string	DELIVERY_FNAME
DeliveryDetails.Email	string	DELIVERY_EMAIL
DeliveryDetails.PhoneNumber	string	DELIVERY_PHONE
DeliveryDetails.Company	string	DELIVERY_COMPANY
DeliveryDetails.Address	string	DELIVERY_ADDRESS
DeliveryDetails.Address2	string	DELIVERY_ADDRESS2
DeliveryDetails.ZipCode	string	DELIVERY_ZIPCODE
DeliveryDetails.City	string	DELIVERY_CITY
DeliveryDetails.State	string	DELIVERY_STATE

OrderDetails Alanı	Alan Tipi	PayU LiveUpdate Alanı
DeliveryDetails.CountryCode	string	DELIVERY_COUNTRYCODE
TestOrder	bool?	TESTORDER
Debug	bool?	DEBUG
Language	Language?	LANGUAGE
AutoMode	bool?	AUTOMODE
OrderTimeout	int?	ORDER_TIMEOUT
TimeoutURL	string	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 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 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	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]

IPNRequest Alanı	Alan Tipi	PayU IPN POST Alanı
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
ReferenceNumber	string	REFNO
SellerReferenceNumber	string	REFNOEXT
OrderNumber	string	ORDERNO
OrderStatus	IPNOrderStatus	ORDERSTATUS
PaymentMethod	string	PAYMETHOD
PaymentMethodCode	string	PAYMETHOD_CODE
TotalPaidAmount	decimal	IPN_PAID_AMOUNT
InstallmentProgramName	string	IPN_INSTALLMENTS_PROGRAM
InstallmentNumber	string	IPN_INSTALLMENTS_NUMBER
InstallmentProfit	string	IPN_INSTALLMENTS_PROFIT
FirstName	string	FIRSTNAME
LastName	string	LASTNAME
IdentityNumber	string	IDENTITY_NO
IdentityIssuer	string	IDENTITY_ISSUER
IdentityCNP	string	IDENTITY_CNP
Company	string	COMPANY
RegistrationNumber	string	REGISTRATIONNUMBER
FiscalCode	string	FISCALCODE
CompanyBankName	string	CBANKNAME
CompanyBankAccount	string	CBANKACCOUNT
Address1	string	ADDRESS1
Address2	string	ADDRESS2
City	string	CITY
State	string	STATE
ZipCode	string	ZIPCODE
Country	string	COUNTRY
PhoneNumber	string	PHONE
FaxNumber	string	FAX
CustomerEmail	string	CUSTOMEREMAIL
DeliveryFirstName	string	FIRSTNAME_D
DeliveryLastName	string	LASTNAME_D

IPNRequest Alanı	Alan Tipi	PayU IPN POST Alanı
DeliveryCompany	string	COMPANY_D
DeliveryAddress1	string	ADDRESS1_D
DeliveryAddress2	string	ADDRESS2_D
DeliveryCity	string	CITY_D
DeliveryState	string	STATE_D
DeliveryZipCode	string	ZIPCODE_D
DeliveryCountry	string	COUNTRY_D
DeliveryPhoneNumber	string	PHONE_D
IpAddress	string	IPADDRESS
Currency	string	CURRENCY
TotalGeneral	decimal	IPN_TOTALGENERAL
Shipping	decimal	IPN_SHIPPING
GlobalDiscount	decimal	IPN_GLOBALDISCOUNT
Commission	decimal	IPN_COMMISSION
Date	string	IPN_DATE
Hash	string	HASH
CreditCardToken	string	IPN_CC_TOKEN
CreditCardMask	string	IPN_CC_MASK
CreditCardExpiryDate	string	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 = "Yilmaz",
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 - Ilce/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 PayU.AutomaticLiveUpdate .OrderDetails tipindeki sipariş bilgisi sunucu tarafında PayU Automatic LiveUpdate adresine POST edilmelidir.

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

```

1 var response = PayU.AutomaticLiveUpdate.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 alanlari atanir
8         // ...
9
10        // 3D Secure zorunlu ise 3D Secure islemi sonrasi sonucun donmesini istedigimiz
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. Kullaniciyi verilen adrese yonlendirmeliyiz.
19         Response.Redirect(response.Url3DS);
20         Response.End();
21     }
22
23     // Buraya kadar geldiysek ALU'dan cevap gelmis demektir.
24     // response nesnesinin alanlarına bakarak sonucu anlayabiliriz.
25     var status = response.Status
26
27     if (status == Status.Success)
28     {
29         // Basarili bir islem durumu
30     }
31 }
32 catch (PayuException ex)
33 {
34     // Hata islemleri yapilir
35 }
36 }
37 }

```

```

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         // için bu alanlari kullanarak yeni bir AluResponse yaratmaliz.
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	string	MERCHANT
OrderRef	string	ORDER_REF
OrderDate	DateTime	ORDER_DATE
PaymentMethod	string	PAY_METHOD
OrderHash	string	ORDER_HASH
PricesCurrency	string	PRICES_CURRENCY
TokenEnable	bool?	LU_ENABLE_TOKEN
TokenType	TokenType?	LU_TOKEN_TYPE
ProductDetails[idx].Name	string	ORDER_PNAME[idx]
ProductDetails[idx].Code	string	ORDER_PCODE[idx]
ProductDetails[idx].UnitPrice	decimal	ORDER_PRICE[idx]
ProductDetails[idx].Quantity	int	ORDER_QTY[idx]
ProductDetails[idx].Information	string	ORDER_PINFO[idx]
ProductDetails[idx].Version	string	ORDER_VER[idx]
BillingDetails.LastName	string	BILL_LNAME
BillingDetails.FirstName	string	BILL_FNAME
BillingDetails.Email	string	BILL_EMAIL
BillingDetails.PhoneNumber	string	BILL_PHONE
BillingDetails.CountryCode	string	BILL_COUNTRYCODE
BillingDetails.Fax	string	BILL_FAX
BillingDetails.Address	string	BILL_ADDRESS
BillingDetails.Address2	string	BILL_ADDRESS2
BillingDetails.ZipCode	string	BILL_ZIPCODE
BillingDetails.City	string	BILL_CITY
BillingDetails.State	string	BILL_STATE
CardDetails.CardNumber	string	CC_NUMBER
CardDetails.ExpiryMonth	string	EXP_MONTH
CardDetails.ExpiryYear	string	EXP_YEAR
CardDetails.CardType	string	CC_TYPE
CardDetails.CVV	string	CC_CVV
CardDetails.CardOwnerName	string	CC_OWNER
DeliveryDetails.LastName	string	DELIVERY_LNAME
DeliveryDetails.FirstName	string	DELIVERY_FNAME
DeliveryDetails.Email	string	DELIVERY_EMAIL
DeliveryDetails.PhoneNumber	string	DELIVERY_PHONE
DeliveryDetails.Company	string	DELIVERY_COMPANY
DeliveryDetails.Address	string	DELIVERY_ADDRESS
DeliveryDetails.Address2	string	DELIVERY_ADDRESS2
DeliveryDetails.ZipCode	string	DELIVERY_ZIPCODE
DeliveryDetails.City	string	DELIVERY_CITY
DeliveryDetails.State	string	DELIVERY_STATE
DeliveryDetails.CountryCode	string	DELIVERY_COUNTRYCODE
ReturnUrl	string	BACK_REF

OrderDetails Alanı	Alan Tipi	PayU ALU İstek Alanı
ClientIpAddress	string	CLIENT_IP
ClientTime	DateTime?	CLIENT_TIME
SelectedInstallmentNumber	int?	SELECTED_INSTALLMENTS_NUMBER
CardProgramName	string	CARD_PROGRAM_NAME
OrderTimeout	int?	ORDER_TIMEOUT

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