

Пример использования платежного модуля Onay, используя технологию asp.net

Настройки в кабинете для магазина demo: URL <http://secure.onpay.ru/user/login> логин demo
пароль demodemo

Подготовительные данные для работы примера: Представим, что у нас имеется интернет магазин, который продаёт товары, в котором для каждого пользователя существует USER_ID. Пользователь имеет баланс виртуальных денег на сайте, с помощью которых оплачивает товары из интернет-магазина. Задача - пополнить баланс пользователя, используя систему OnPay. То есть пользователь платит деньги провайдеру, а за это ему начисляются виртуальные деньги интернет-магазина Для начала создадим таблицу в базе данных: назовём её ONPAY_TRANSFER

```
CREATE TABLE ONPAY_TRANSFER (
```

```
ID INTEGER identity,-- Уникальный автоинкрементный идентификатор
USER_ID INTEGER,-- Пользователь интернет-магазина
REDIRECT_DTE DATETIME,-- Дата переадресации пользователя на платежную
систему OnPay
PAY_DTE DATETIME,-- Дата получения провайдером pay запроса
CHECK_DTE DATETIME,-- Дата получения провайдером check запроса
TRANSFER_SUM NUMERIC(18,2),-- Сумма к зачислению
CHECK_PAID INTEGER NOT NULL,-- Флаг того, что check запрос был произведен
PAY_PAID INTEGER NOT NULL,-- Флаг того, что pay запрос был произведен
ONPAY_ID nvarchar(100),-- Уникальный идентификатор платежа в системе OnPay
constraint PK_ONPAY_TRANSFER primary key (ID)
```

) Введём несколько дополнительных классов, для облегчения работы: Класс OnPayEngine using System;

```
using System.Collections.Generic;
```

```
using System.Globalization;
```

```
using System.Security.Cryptography;
```

```
using System.Text;
```

```
namespace Utils
```

```
{
```

```
    public class OnPayEngine
```

```
    {
```

```
        // Секретный пароль (Пароль для API IN)
```

```
        public static string strSecureApiIn = "a2su3ig8y";
```

```
// Ссылка, на которую пользователь будет перенаправлен для совершения платежа
```

```
public static string strRedirectLink =  
"http://secure.onpay.ru/pay/demo";
```

```
// Ссылка, по которой будет перенаправлен пользователь, после того как  
платеж успешно проведён
```

```
public static string strUrlSuccess =  
"http://www.domen.com/Success.aspx";
```

```
// Ссылка, по которой будет перенаправлен пользователь, если платеж  
прошёл неудачно
```

```
public static string strUrlFail = "http://www.domen.com/Fail.aspx";
```

```
// Идентификатор пользователя системы
```

```
public static int iUSER_ID = 123456;
```

```
// Идентификатор пользователя системы
```

```
public static string strUSER_EMAIL = "demouser@demodomen.ru";
```

```
/// <summary>
```

```
/// Функция кодирует исходную строку алгоритмом md5
```

```
/// </summary>
```

```
/// <param name="sCrcBase"></param>
```

```
/// <returns></returns>
```

```
public static string BuildCRCString(string sCrcBase)
```

```
{
```

```
    MD5CryptoServiceProvider md5 = new MD5CryptoServiceProvider();
```

```
    byte[] bSignature =  
md5.ComputeHash(Encoding.UTF8.GetBytes(sCrcBase));
```

```
    StringBuilder sbSignature = new StringBuilder();
```

```
    foreach (byte b in bSignature)
```

```
        sbSignature.AppendFormat("{0:x2}", b);
```

```
        return sbSignature.ToString().ToUpper();
    }

    /// <summary>
    /// Сгенерировать пользователю ссылки на платёжную систему
    /// </summary>
    /// <param name="fOutSum">сумма</param>
    /// <returns>сформированная ссылка на платёжную систему, по которой
    /// будет происходить запрос</returns>
    public static string GenerateUserPaymentLink(float fOutSum)
    {
        string sCrcBase;

        // OnPayTransfer - ORM объект представляющий запись в таблице
        ONPAY_TRANSFER

        OnPayTransfer new OnPayTransfer(); // Создать объект перевода
        средств

        onpayTransfer.USER_ID = iUSER_ID; // Перевод для пользователя с
        USER_ID = iUSER_ID

        onpayTransfer.CHECK_PAID = 0; // Установить флаг, что check запрос
        от системы onpay не был произведён

        onpayTransfer.REDIRECT_DTE = DateTime.Now; // Дата отсылки
        пользователя на форму оплаты

        onpayTransfer.PAY_PAID = 0; // Установить флаг, что pay запрос от
        системы onpay не был произведён

        onpayTransfer.TRANSFER_SUM = fOutSum; // Сумма перевода

        onpayTransfer.Create(); // Создать запись в таблице ONPAY_TRANSFER

        //pay_mode;price;currency;pay_for;convert; secret_key

        // Сформировать ссылку для кодирования её алгоритмом md5

        sCrcBase = string.Format("fix;{0};{1};{2};yes;{3}", fOutSum,
        "RUR", onpayTransfer.ID,
```

```
        strSecureApiIn);

        string strMD5String = BuildCRCString(sCrcBase); // Закодировать
строку

        onpayTransfer.Update();

        // Возвратить ссылку на платёжную систему
        // в переменную pay_for записывается уникальный идентификатор
        // перевода в системе провайдера(поле ID записи из таблицы
ONPAY_TRANSFER)

        //

        return strRedirectLink + "?" +

            "pay_mode=fix" +

            "&price=" + fOutSum +

            "&pay_for=" + onpayTransfer.ID +

            "&md5=" + strMD5String +

            "&convert=yes" +

            "&url_success=" + strUrlSuccess +

            "&url_fail=" + strUrlFail +

            "&note=" + "Пополнение_счёта" +

            "&user_email=" + strUSER_EMAIL;

    }

    /// <summary>

    /// Функция формирует md5 строку для ответа на check запрос платёжной
системы

    /// </summary>

    /// <param name="pay_for">параметр pay_for</param>

    /// <param name="strCurrency">параметр order_currency</param>

    /// <param name="order_amount">параметр order_amount </param>
```

```
/// <param name="code">параметр code</param>
```

```
/// <returns>закодированная строка методом md5</returns>
```

```
public static string GenerateCheckCRC(int pay_for, string strCurrency,  
string order_amount, int code)
```

```
{
```

```
    //type;pay_for;order_amount;order_currency;code;secret_key_api_in"
```

```
    // Сформировать строчку по заданному шаблону
```

```
    string sCrcBase = string.Format("check;{0};{1};{2};{3};{4}",  
pay_for, order_amount, strCurrency, code, strSecureApiIn);
```

```
    // Вычислить и вернуть закодированную строку
```

```
    return BuildCRCString(sCrcBase);
```

```
}
```

```
/// <summary>
```

```
/// Функция формирует md5 строку для ответа на pay запрос платёжной  
системы
```

```
/// </summary>
```

```
/// <param name="pay_for">параметр pay_for</param>
```

```
/// <param name="onpay_id">параметр onpay_id</param>
```

```
/// <param name="order_amount">параметр order_amount</param>
```

```
/// <param name="code">параметр code</param>
```

```
/// <returns>закодированная строка методом md5</returns>
```

```
public static string GeneratePayCRC(int pay_for, string onpay_id,  
string order_amount, int code)
```

```
{
```

```
    //type;pay_for;onpay_id;order_id;order_amount;order_currency;code;secret_key  
_api_in
```

```
    // Сформировать строчку по заданному шаблону
```

```
        string sCrcBase = string.Format("pay;{0};{1};{2};{3};RUR;{4};{5}",
pay_for, onpay_id, pay_for, order_amount, code, strSecureApiIn);

        // Вычислить и вернуть закодированную строку

        return BuildCRCString(sCrcBase);

    }

}
```

```
}
```

Класс SiteHelper using System;

using System.Collections.Generic;

using System.Text;

using System.Web;

using System.Web.UI;

namespace Helpers

```
{
```

```
    public class SiteHelper
```

```
    {
```

```
        public static string GetPrm(string sName)
```

```
        {
```

```
            string sValue;
```

```
            sValue = HttpContext.Current.Request.Form[sName] as string;
```

```
            if (string.IsNullOrEmpty(sValue))
```

```
                sValue = HttpContext.Current.Request.QueryString[sName] as
string;
```

```
            if (string.IsNullOrEmpty(sValue))
```

```
                sValue = String.Empty;
```

```
            return sValue;
```

```
}
```

```
}
```

```
}
```

Общая схема работы примера такова:

В интернет-магазине пользователь заходит на страницу пополнения счета, вводит сумму денег, на которую хочет пополнить свой баланс, нажимает на кнопку Пополнить. Далее происходит запись в таблицу ONPAY_TRANSFER: Заполняются следующие поля: 1) USER_ID - пользователь интернет-магазина 2) REDIRECT_DTE - текущее время(DateTime.Now) 3) CHECK_PAID = 0, флаг того, что check запрос не был произведен 3) PAY_PAID = 0, флаг того, что pay запрос не был произведен 3) TRANSFER_SUM , сумма перевода Далее происходит генерация ссылки на платёжную систему(Функция GenerateUserPaymentLink в классе OnPayEngine).

Код приведён ниже:
Разметка формы

```
<form id="form1" runat="server">
<table>

    <tr>
        <td>
            <asp:TextBox ID="txtAddSumm" runat="server"></asp:TextBox>
        </td>
        <td>
            <asp:Button ID="btnAddSumm" runat="server" Text="Button"
                onclick="btnAddSumm_Click" />
        </td>
    </tr>
</table>
</form>
```

Обработчик нажатия кнопки

```
protected void btnAddSumm_Click(object sender, EventArgs e)
```

```
{
```

```
    float fOutSumm;
```

```
    float.TryParse(txtAddSumm.Text, out fOutSumm);
```

```
    if (fOutSumm > 0)
```

```
    {
```

```
        if (SiteConfiguration.OnlinePaymentEnabled)
```

```
Response.Redirect(OnPayEngine.GenerateUserPaymentLink(fOutSumm));
```

```
}
```

```
}
```

Далее у пользователя должна появиться такая форма:

После начала оплаты на страницу провайдера от системы OnPay приходит запрос check, страница обрабатывает его, выдает серверу ответ. Если ответ от сервера положительный, система OnPay посылает странице провайдера запрос pay. Провайдер обрабатывает его и если результат обработки - удачный, то зачисляет пользователю сумму баллов, которые он указал при переходе на страницу оплаты.

Приведём полный код страницы api.aspx

Разметка страницы

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="api.aspx.cs"
Inherits="OnPay.api" %> <!DOCTYPE html PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN"
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»> <html
xmlns="http://www.w3.org/1999/xhtml" > <head runat="server"> <title></title> </head> <body>
<form id="form1" runat="server"> <div> </div> </form> </body> </html>
```

Далее приведём codebehind файл страницы:

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace OnPay {
public partial class api : System.Web.UI.Page {
protected void Page_Load(object sender, EventArgs e) {
string strType = SiteHelper.GetPrm("type");
// Проверить тип запроса
```

```
if (strType == "check") // Был произведён check запрос
{
// идентификатор платежа (поле ID таблицы ONPAY_TRANSFER)
int iOnPayTransferId;
// Считать необходимые параметры из строки запроса
string strOrder_amount = SiteHelper.GetPrm("order_amount");
string strPay_for = SiteHelper.GetPrm("pay_for");
string strOrder_currency =
SiteHelper.GetPrm("order_currency");
int.TryParse(strPay_for, out iOnPayTransferId);
OnPayTransfer new OnPayTransfer(); // Создать ORM объект перевода
onPayTransfer.Read(iOnPayTransferId); //
Считать свойства объекта перевода
if (onPayTransfer.ID != 0 &&
onPayTransfer.CHECK_PAID == 0) // Если такой объект есть в базе
check запрос для него не был произведён
{
onPayTransfer.CHECK_PAID = 1; // Установить флаг что check
запрос был произведён
onPayTransfer.CHECK_DTE = DateTime.Now; // Запомнить дату
check запроса
onPayTransfer.Update(); // Записать изменения в базу
// Сформировать строку с кодом 0 - удачно, с xml ответом
серверу
string strXMLResponse = GetCheckXML(0, onPayTransfer.ID,
strOrder_amount, "OK", strOrder_currency);
```



```

        WriteResponse(strXMLResponse);
    }
    else
    {
        // Сформировать строку с кодом 2 - неудачно, с xml
ответом серверу
        stringstrXMLResponse = GetCheckXML(2, onPayTransfer.ID,
strOrder_amount, "Ошибка в параметрах перевода", strOrder_currency);
        // Выставить кодировку для ответа
        WriteResponse(strXMLResponse);
    }
}
if (strType == "pay") // Был произведён pay запрос
{
    int pay_for;
    int order_amount;
    int balance_amount;
    double dexchange_rate;
    DateTime paymentDateTime;
    string stronpay_id = SiteHelper.GetPrm("onpay_id");
    string strpay_for = SiteHelper.GetPrm("pay_for");
    string strorder_amount = SiteHelper.GetPrm("order_amount");
    string strbalance_amount =
SiteHelper.GetPrm("balance_amount");
    string strexchange_rate = SiteHelper.GetPrm("exchange_rate");
    string strpaymentDateTime =
SiteHelper.GetPrm("paymentDateTime");
    int.TryParse(strpay_for, out pay_for);
    int.TryParse(strorder_amount, out order_amount);
    int.TryParse(strbalance_amount, out balance_amount);
    double.TryParse(strexchange_rate, out dexchange_rate);
    DateTime.TryParse(strpaymentDateTime, out paymentDateTime);
}

```

OnPayTransfer new OnPayTransfer(); Создать ORM объект перевода *onPayTransfer.Read(pay_for);*
 Считать свойства объекта перевода

```

    if (onPayTransfer.ID != 0 &&
        onPayTransfer.CHECK_PAID == 1 &&
        onPayTransfer.PAY_PAID == 0) // Если перевод найден в базе,
check запрос для него был произведён, а pay запрос нет
    {
        onPayTransfer.PAY_DTE = paymentDateTime.Date; // Записать
дату pay запроса
        onPayTransfer.PAY_PAID = 1; // Выставить флаг, что pay
запрос был произведен
        onPayTransfer.ONPAY_ID = stronpay_id; // Сохранить
внутренний идентификатор платежа в системе OnPay
        onPayTransfer.Update(); // Обновить запись о платеже -
платеж проведён
        // Сюда можно вставить код пополнения внутреннего счёта
пользователя интернет-магазина
    }

```

```

        stringstrXMLResponse = GetPayXML(0, onPayTransfer.ID,
strorder_amount, stronpay_id, "OK");
        WriteResponse(strXMLResponse);
    }
    else
    {
        string strXMLResponse = GetPayXML(2, onPayTransfer.ID,
strorder_amount, stronpay_id, "Ошибка в параметрах перевода");
        WriteResponse(strXMLResponse);
    }
}

```

```

}
///<summary>
///Функция возвращает xml ответ на pay запрос
///</summary>
///<param name="iCode">код операции</param>
///<param name="pay_for">параметр pay_for</param>
///<param name="order_amount">сумма зачислению</param>
///<param name="comment">комментарий</param>
///<param name="onpay_id">индекс идентификатор платежа в системе OnPay</param>
///<returns></returns>
private string GetPayXML(int iCode, int pay_for, string order_amount,
string onpay_id, string comment)
{
    StringBuilder stringBuilder = new StringBuilder();
    stringBuilder.AppendLine("<?xml version='1.0' encoding='UTF-
8'?'>");
    stringBuilder.AppendLine("<result>");
    stringBuilder.AppendLine("<code>" + iCode + "</code>");
    stringBuilder.AppendLine("<comment>" + comment + "</comment>");
    stringBuilder.AppendLine("<onpay_id>" + onpay_id + "</onpay_id>");
    stringBuilder.AppendLine("<pay_for>" + pay_for + "</pay_for>");
    stringBuilder.AppendLine("<order_id>" + pay_for + "</order_id>");
    stringBuilder.AppendLine("<md5>" +
OnPayEngine.GeneratePayCRC(pay_for, onpay_id, order_amount, iCode) +
"</md5>");
    stringBuilder.AppendLine("</result>");
    return stringBuilder.ToString();
}
///<summary>
/// Функция возвращает xml ответ на check запрос
///</summary>
///<param name="iCode">код операции</param>
///<param name="pay_for">параметр pay_for</param>
///<param name="order_amount">сумма зачислению</param>
///<param name="comment">комментарий</param>
///<param name="strCurrency">валюта перевода</param>
///<returns></returns>
private string GetCheckXML(int iCode, int pay_for, string

```

```
order_amount, string comment, string strCurrency)
{
    StringBuilder stringBuilder = new StringBuilder();
    stringBuilder.AppendLine("<?xml version=\"1.0\" encoding=\"UTF-8\"?>");
    stringBuilder.AppendLine("<result>");
    stringBuilder.AppendLine("<code>" + iCode + "</code>");
    stringBuilder.AppendLine("<pay_for>" + pay_for + "</pay_for>");
    stringBuilder.AppendLine("<comment>" + comment + "</comment>");
    stringBuilder.AppendLine("<md5>" +
OnPayEngine.GenerateCheckCRC(pay_for, strCurrency, order_amount, iCode) +
"</md5>");
    stringBuilder.AppendLine("</result>");
    return stringBuilder.ToString();
}
///  

///  

///  

///  

private void WriteResponse(string strXmlResponse)
{
    Response.Charset = "utf-8";
    Response.HeaderEncoding = Encoding.UTF8;
    Response.ContentEncoding = Encoding.UTF8;
    Response.Write(strXmlResponse);
    Response.End();
}
}
```

From:

<http://wiki.onpay.ru/> - Onpay.ru Wiki

Permanent link:

<http://wiki.onpay.ru/doku.php?id=asp.net&rev=1312282451>

Last update: **2011/08/02 11:54**

