

```

<?php
/**
 * ТДС корзины с объединением всех шагов на одной странице v. 6.2
 * обязательно использовать с соответствующей страницей настроек ТДС
 * для вывода типов доставки и пересчета используется ajax
 *
 * @author Kuts Artem, KAD Systems (©) 2014
 * @date 01-09-2014
 */

// Игнорировать модуль "Пользователи"
define("SHORTCART_IGNOREUSERS", false);
// Режим отображения 2 - 2 шага оформления, 3 - 3 шага оформления
define("SHORTCART_CARTMODE", 2);
// Использовать captcha
define("SHORTCART_USECAPTCHA", false);
// Ошибка неправильного ввода
define("SHORTCART_ERROR_WC", "Введен неверный код подтверждения!");

$Shop = Core_Entity::factory('Shop',
Core_Array::get(Core_Page::instance()->libParams, 'shopId'));

// -----
// Обработка запросов от Яндекс.Денег
// -----
if (isset($_POST['action']) && isset($_POST['invoiceId']) &&
isset($_POST['orderNumber']))
{
    // Получаем ID заказа
    $order_id = intval(Core_Array::getPost('orderNumber'));

    $Shop_Order = Core_Entity::factory('Shop_Order')->find($order_id);

    if (!is_null($Shop_Order->id))
    {
        header("Content-type: application/xml");

        // Вызов обработчика платежной системы
        Shop_Payment_System_Handler::factory($Shop_Order->Shop_Payment_System)
            ->shopOrder($Shop_Order)
            ->paymentProcessing();
    }
}

if (isset($_REQUEST['type']))
{
    // Получаем ID заказа
    $order_id = intval(Core_Array::getPost('pay_for'));

    $Shop_Order = Core_Entity::factory('Shop_Order')->find($order_id);
}

```

```

        if (!is_null($oShop_Order->id))
        {
            header("Content-type: application/xml");

            // Вызов обработчика платежной системы
Shop_Payment_System_Handler::factory($oShop_Order->Shop_Payment_System)
                ->shopOrder($oShop_Order)
                ->paymentProcessing();
        }
    }

// -----
// Обработка уведомления об оплате от IntellectMoney
// -----
if (isset($_REQUEST['orderId']))
{
    // Получаем ID заказа
    $order_id = intval(Core_Array::getRequest('orderId'));

    $oShop_Order = Core_Entity::factory('Shop_Order')->find($order_id);

    if (!is_null($oShop_Order->id))
    {
        // Вызов обработчика платежной системы
Shop_Payment_System_Handler::factory($oShop_Order->Shop_Payment_System)
                ->shopOrder($oShop_Order)
                ->paymentProcessing();
    }
}

// -----
// Обработка уведомления об оплате от ROBOKASSA
// должно быть только в настройках типовой дин. страницы
// -----
if (isset($_REQUEST['SignatureValue']))
// для отличия от SuccessURL/FailURL
&& !isset($_REQUEST['Culture']))
{
    // Получаем ID заказа
    $order_id = intval(Core_Array::getRequest('InvId'));

    $oShop_Order = Core_Entity::factory('Shop_Order')->find($order_id);

    if (!is_null($oShop_Order->id))
    {
        // Вызов обработчика платежной системы
Shop_Payment_System_Handler::factory($oShop_Order->Shop_Payment_System)
                ->shopOrder($oShop_Order)
                ->paymentProcessing();
    }
}

```

```

        exit();
    }

    // Добавление товара в корзину
    if (Core_Array::getRequest('add'))
    {
        $shop_item_id = intval(Core_Array::getRequest('add'));

        if ($shop_item_id)
        {
            $oShop_Cart_Controller = Shop_Cart_Controller::instance();
            $oShop_Cart_Controller
                ->shop_item_id($shop_item_id)
                ->quantity(Core_Array::getRequest('count', 1))
                ->add();
        }

        // Ajax
        if (Core_Array::getRequest('_', FALSE))
        {
            ob_start();

            // Краткая корзина
            $Shop_Cart_Controller_Show = new Shop_Cart_Controller_Show(
                $oShop
            );
            $Shop_Cart_Controller_Show
                ->xsl(
                    Core_Entity::factory('Xsl')->getByName(
                        Core_Array::get(Core_Page::instance()->libParams,
'littleCartXsl')
                    )
                )
                ->couponText(Core_Array::get($_SESSION, 'coupon_text'))
                ->show();

            echo json_encode(ob_get_clean());

            exit();
        }
    }

    if (Core_Array::getGet('action') == 'repeat')
    {
        $guid = Core_Array::getGet('guid');
        if (strlen($guid))
        {
            $oShop_Order = $oShop->Shop_Orders->getByGuid($guid);

            if (!is_null($oShop_Order))
            {

```

```

        $aShop_Order_Items = $oShop_Order->Shop_Order_Items->findAll();

        $oShop_Cart_Controller = Shop_Cart_Controller::instance();

        foreach ($aShop_Order_Items as $oShop_Order_Item)
        {
            $oShop_Order_Item->shop_item_id && $oShop_Cart_Controller
                ->shop_item_id($oShop_Order_Item->shop_item_id)
                ->quantity($oShop_Order_Item->quantity)
                ->add();
        }
    }
}

if (!is_null(Core_Array::getGet('ajaxLoad')))
{
    $aObjects = array();

    if (Core_Array::getGet('shop_country_id'))
    {
        $oShop_Country_Location =
Core_Entity::factory('Shop_Country_Location');
        $oShop_Country_Location
            ->queryBuilder()
            ->where('shop_country_id', '=',
intval(Core_Array::getGet('shop_country_id')));
        $aObjects = $oShop_Country_Location->findAll();
    }
    elseif (Core_Array::getGet('shop_country_location_id'))
    {
        $oShop_Country_Location_City =
Core_Entity::factory('Shop_Country_Location_City');
        $oShop_Country_Location_City
            ->queryBuilder()
            ->where('shop_country_location_id', '=',
intval(Core_Array::getGet('shop_country_location_id')));
        $aObjects = $oShop_Country_Location_City->findAll();
    }
    elseif (Core_Array::getGet('shop_country_location_city_id'))
    {
        $oShop_Country_Location_City_Area =
Core_Entity::factory('Shop_Country_Location_City_Area');
        $oShop_Country_Location_City_Area
            ->queryBuilder()
            ->where('shop_country_location_city_id', '=',
intval(Core_Array::getGet('shop_country_location_city_id')));
        $aObjects = $oShop_Country_Location_City_Area->findAll();
    }

    $aArray = array('...');
}

```

```

foreach ($aObjects as $Object)
{
    $aArray['_' . $Object->id] = $Object->name;
    // Если не работает подгрузка городов - раскомментировать строчку выше
    // и закомментировать строчку ниже
    //$aArray[$Object->id] = $Object->name;
}

echo json_encode($aArray);
exit();
}

// Удаление товара из корзины
if (Core_Array::getGet('delete'))
{
    $shop_item_id = intval(Core_Array::getGet('delete'));

    if ($shop_item_id)
    {
        $oShop_Cart_Controller = Shop_Cart_Controller::instance();
        $oShop_Cart_Controller
            ->shop_item_id($shop_item_id)
            ->delete();
    }
}

// KAD: Удаление всех товаров из корзины
if (Core_Array::getGet('delete_all'))
{
    $oShop_Cart_Controller = Shop_Cart_Controller::instance();
    $oItemsInCart = $oShop_Cart_Controller->getAll($oShop);
    foreach( $oItemsInCart as $oItemInCart)
    {
        $oShop_Cart_Controller
            ->shop_item_id($oItemInCart->shop_item_id)
            ->delete();
    }
    if (isset($_SERVER['HTTP_REFERER']))
    {
        header('location: ' . $_SERVER['HTTP_REFERER']);
        exit();
    }
}

// KAD: ajax обработчик вывода типов доставки
if (Core_Array::getRequest('ajaxget') == "delivery")
{
    $_SESSION['hostcmsOrder']['shop_country_id'] =
intval(Core_Array::getRequest('shop_country_id', 0));
    $_SESSION['hostcmsOrder']['shop_country_location_id'] =
intval(Core_Array::getRequest('shop_country_location_id', 0));
}

```

```

        $_SESSION['hostcmsOrder']['shop_country_location_city_id'] =
intval(Core_Array::getRequest('shop_country_location_city_id', 0));
        $_SESSION['hostcmsOrder']['shop_country_location_city_area_id'] =
intval(Core_Array::getRequest('shop_country_location_city_area_id', 0));

        $Shop_Delivery_Controller_Show = new
Shop_Delivery_Controller_Show($oShop);

        $Shop_Delivery_Controller_Show
        ->shop_country_id($_SESSION['hostcmsOrder']['shop_country_id'])
->shop_country_location_id($_SESSION['hostcmsOrder']['shop_country_location_
id'])
->shop_country_location_city_id($_SESSION['hostcmsOrder']['shop_country_loca
tion_city_id'])
->shop_country_location_city_area_id($_SESSION['hostcmsOrder']['shop_country
_location_city_area_id'])
        ->couponText(Core_Array::get(Core_Array::get($_SESSION,
'hostcmsOrder', array()), 'coupon_text'))
        ->setUp()
        ->xsl(
            Core_Entity::factory('Xsl')->getByName(
                Core_Array::get(Core_Page::instance()->libParams,
'deliveryXsl')
            )
        )
        ->show();
        header("Content-Type: text/html; charset=utf-8");
        exit();
    }

// KAD: ajax обработчик вывода платежных систем
if (Core_Array::getRequest('ajaxget') == "payment")
{

    $Shop_Payment_System_Controller_Show = new
Shop_Payment_System_Controller_Show($oShop);

    // Если включена кэпча
    if (SHORTCART_USECAPTCHA)
    {
        $Shop_Payment_System_Controller_Show->addEntity(
            Core::factory('Core_Xml_Entity')
            ->name('siteuser_id')
            ->value((Core::moduleIsActive('siteuser') &&
!is_null(Core_Entity::factory('Siteuser')->getCurrent()))?Core_Entity::facto
ry('Siteuser')->getCurrent()->id:0)
        )->addEntity(
            Core::factory('Core_Xml_Entity')
            ->name('captcha_id')
            ->value(Core_Captcha::getCaptchaId())
        );
    }
}

```

```

    }
    $shop_delivery_condition_id =
strval(Core_Array::getRequest('shop_delivery_condition_id', 0));

    if (is_numeric($shop_delivery_condition_id))
    {
        $_SESSION['hostcmsOrder']['shop_delivery_condition_id'] =
intval($shop_delivery_condition_id);
    }
    else
    {
        $_SESSION['hostcmsOrder']['shop_delivery_condition_id'] = 0;

        // в shop_delivery_condition_id тогда "123#", ID элемента массива в сессии, в
котором
        // хранится стоимость доставки, налог, название специфичного условия доставки
        list($shopDeliveryInSession) = explode('#',
$shop_delivery_condition_id);

        if
(isset($_SESSION['hostcmsOrder']['deliveries'][$shopDeliveryInSession]))
        {
            $aTmp =
$_SESSION['hostcmsOrder']['deliveries'][$shopDeliveryInSession];

            $_SESSION['hostcmsOrder']['shop_delivery_id'] =
$aTmp['shop_delivery_id'];
            $_SESSION['hostcmsOrder']['shop_delivery_price'] =
$aTmp['price'];
            $_SESSION['hostcmsOrder']['shop_delivery_rate'] =
$aTmp['rate'];
            $_SESSION['hostcmsOrder']['shop_delivery_name'] =
$aTmp['name'];
        }
    }

    $oShop_Delivery_Condition =
Core_Entity::factory('Shop_Delivery_Condition',
$_SESSION['hostcmsOrder']['shop_delivery_condition_id']);

    $Shop_Payment_System_Controller_Show
->shop_delivery_id($oShop_Delivery_Condition->shop_delivery_id)
->xsl(
    Core_Entity::factory('Xsl')->getByName(
        Core_Array::get(Core_Page::instance()->libParams,
'paymentSystemXsl')
    )
)
->show();
header("Content-Type: text/html; charset=utf-8");
exit();

```

```

}

if (Core_Array::getPost('recount') || Core_Array::getPost('step') == 1)
{
    $oShop_Cart_Controller = Shop_Cart_Controller::instance();
    $aCart = $oShop_Cart_Controller->getAll($oShop);

    foreach ($aCart as $oShop_Cart)
    {
        $oShop_Cart_Controller
            ->shop_item_id($oShop_Cart->shop_item_id)
            ->quantity(Core_Array::getPost('quantity_' .
$oShop_Cart->shop_item_id))
            ->postpone(is_null(Core_Array::getPost('postpone_' .
$oShop_Cart->shop_item_id)) ? 0 : 1)
            ->shop_warehouse_id(Core_Array::getPost('warehouse_' .
$oShop_Cart->shop_item_id, 0))
            ->update();
    }

    // Запоминаем купон
    $_SESSION['hostcmsOrder']['coupon_text'] =
trim(strval(Core_Array::getPost('coupon_text')));
}

$Shop_Cart_Controller_Show = new Shop_Cart_Controller_Show($oShop);

/*
* KAD: ajax обработчик вывода корзины
* совместим с формой быстрого заказа
*/
if (Core_Array::getRequest('ajaxget') == "cart")
{
    $oShop_Cart_Controller = Shop_Cart_Controller::instance();
    $aCart = $oShop_Cart_Controller->getAll($oShop);

    foreach ($aCart as $oShop_Cart)
    {
        if (Core_Array::getRequest('quantity_' .
$oShop_Cart->shop_item_id))
        {
            $quantity = Core_Array::getRequest('quantity_' .
$oShop_Cart->shop_item_id);
            if ($quantity == -1)
            {
                $quantity = 0;
            }

            $oShop_Cart_Controller

```

```

        ->shop_item_id($oShop_Cart->shop_item_id)
        ->quantity($quantity)
        ->postpone(is_null(Core_Array::getRequest('postpone_' .
$oShop_Cart->shop_item_id)) ? 0 : 1)
        ->shop_warehouse_id(Core_Array::getRequest('warehouse_'
. $oShop_Cart->shop_item_id, 0))
        ->update();
    }
}

// Запоминаем купон
$_SESSION['hostcmsOrder']['coupon_text'] =
trim(strval(Core_Array::getRequest('coupon_text')));

$xmlName = Core_Array::get(Core_Page::instance()->libParams,
'cartXsl');
$Shop_Cart_Controller_Show
->couponText(Core_Array::get(Core_Array::get($_SESSION,
'hostcmsOrder', array()), 'coupon_text'))
->itemsProperties(FALSE)
//->itemsPropertiesList(FALSE)
->xsl(
    Core_Entity::factory('Xsl')->getByName($xmlName)
)
->show();

header("Content-Type: text/html; charset=utf-8");
exit();
}

Core_Page::instance()->object = $Shop_Cart_Controller_Show;

```