

Временный купон

Решение реализует механизм скидки, которая применяется к заказу только в том случае, если со времени добавления последнего товара в корзину прошло меньше определенного времени. Например, скидка к заказу применяется в том случае, если со времени добавления товара в корзину прошло меньше 30 мин, при этом на странице корзины выводится таймер с отсчетом этого времени.

Установка решения:

1. Создать скидку и купон.
2. Добавить константы:
 - TEMPORARY_SALE_TIMEOUT — время, в течение которого работает скидка;
 - TEMPORARY_SALE_COUPON — текст используемого купона.
3. В настройки ТДС корзины добавить следующий код для события добавления товара в корзину:

```
if (Core_Array::getRequest('add'))
{
    $shop_item_id = intval(Core_Array::getRequest('add'));

    if ($shop_item_id)
    {
        // ...

        // Запоминаем последнее добавление товара в корзину
        $_SESSION['hostcmsCart']['last_addition'] = time();
    }
}
```

4. Создать наблюдатель:

```
<?php

defined('HOSTCMS') || exit('HostCMS: access denied.');
```

```
/*
 * @author Maxim Zasorin, KAD Systems (©) 2015
 * @date 04-04-2016
 *
 * В bootstrap.php:
 *
 * // Добавление купона, если со времени добавления товара в корзину прошло менее 30
 * мин.
 * Core_Event::attach('Shop_Purchase_Discount_Controller.onBeforeGetDiscounts', array('Shop_Purchase_Discount_Controller_Observer_Coupon',
 * 'onBeforeGetDiscounts'));
 *
 * Копирование и использование файлов модуля
 * в коммерческих целях ЗАПРЕЩЕНО
 *
```

```

*/

class Shop_Purchase_Discount_Controller_Observer_Coupon
{
    /**
     * Добавление купона, если со времени добавления товара в корзину прошло менее 30
     мин.
     *
     * @param Shop_Purchase_Discount_Controller $object
     * @param array $args
     */
    static public function
    onBeforeGetDiscounts(Shop_Purchase_Discount_Controller $object, $args)
    {
        if (defined('TEMPORARY_SALE_COUPON') &&
        defined('TEMPORARY_SALE_TIMEOUT'))
        {
            $couponText = TEMPORARY_SALE_COUPON;
            $timeout = TEMPORARY_SALE_TIMEOUT;
            // Не применен ни какой купон
            if (!$object->couponText || $object->couponText ==
            $couponText)
            {
                // В корзину добавлялся товар
                if (isset($_SESSION['hostcmsCart']['last_addition']))
                {
                    $lastAddition =
                    $_SESSION['hostcmsCart']['last_addition'];
                    // Время не вышло — добавляем купон
                    if (time() - $lastAddition <= $timeout * 60)
                    {
                        $object->couponText = $couponText;
                    }
                    else
                    {
                        $object->couponText = '';
                    }
                }
            }
        }
    }
}

```

5. Зарегистрировать наблюдатель в bootstrap.php:

```

Core_Event::attach('Shop_Purchase_Discount_Controller.onBeforeGetDiscounts', array('Shop_Purchase_Discount_Controller_Observer_Coupon', 'onBeforeGetDiscounts'));

```

6. Если требуется выводить таймер отсчета на странице корзины, то можно использовать следующий наблюдатель, который добавит узлы last_addition_timeout и

temporary_coupon, которые будут содержать оставшееся время с момента добавления последнего товара в корзину и код купона соответственно:

```
<?php

defined('HOSTCMS') || exit('HostCMS: access denied.');
```

```
/*
 * @author Maxim Zasorin, KAD Systems (©) 2015
 * @date 04-04-2016
 *
 * В bootstrap.php:
 *
 // Оставшееся время для скидки
Core_Event::attach('Shop_Cart_Controller_Show.onBeforeRedeclaredShow',
array('Shop_Cart_Controller_Show_Observer_Lastaddition',
'onBeforeRedeclaredShow'));

* Копирование и использование файлов модуля
* в коммерческих целях ЗАПРЕЩЕНО
*
*/

class Shop_Cart_Controller_Show_Observer_Lastaddition
{
    /**
     * Оставшееся время для скидки
     *
     * @param Shop_Cart_Controller_Show $object
     * @param array $args
     */
    static public function
onBeforeRedeclaredShow(Shop_Cart_Controller_Show $object, $args)
    {
        if (defined('TEMPORARY_SALE_COUPON') &&
defined('TEMPORARY_SALE_TIMEOUT'))
        {
            $couponText = TEMPORARY_SALE_COUPON;
            $temporaryTimeout = TEMPORARY_SALE_TIMEOUT;

            if (isset($_SESSION['hostcmsCart']['last_addition']))
            {
                $lastAddition =
$_SESSION['hostcmsCart']['last_addition'];

                // $object->addEntity(
                //     Core::factory('Core_Xml_Entity')
                //     ->name('last_addition')
                //
                ->value(Core_Date::timestamp2datetime($lastAddition))
                // );
            }
        }
    }
}
```

```

        $timeout = $lastAddition + $temporaryTimeout * 60 -
time();

        if ($timeout > ) {
            $object->addEntity(
                Core::factory('Core_Xml_Entity')
                    ->name('last_addition_timeout')
                    ->value($timeout)
            );

            $object->addEntity(
                Core::factory('Core_Xml_Entity')
                    ->name('temporary_coupon')
                    ->value($couponText)
            );
        }
    }
}

```

7. Зарегистрировать наблюдатель в файле bootstrap.php:

```

Core_Event::attach('Shop_Cart_Controller_Show.onBeforeRedeclaredShow',
array('Shop_Cart_Controller_Show_Observer_Lastaddition',
'onBeforeRedeclaredShow'));

```

8. На странице корзины вывод таймера можно организовать следующим образом с использованием библиотеки [FlipClock](#):

```

<xsl:if test="coupon_text = '' or coupon_text = temporary_coupon">
    <div style="text-align: center; margin-top: 30px;">
        <div id="last_addition_timeout" style="display: inline-block;
width: 340px;"></div>
    </div>

    <script>
        var clock = $('#last_addition_timeout').FlipClock(<xsl:value-of
select="last_addition_timeout" />, {
            countdown: true,
            defaultLanguage: 'russian'
        });
    </script>
</xsl:if>

```