

# Некешируемые области.

Модуль позволяет добавить в любое место сайта (в т.ч. кешируемый компонент) некешируемую область. Данные можно выводить как с помощью назначения контента области, так и с помощью функций обратного вызова, передавая для каждой области свои параметры.

В административную панель добавляется пункт меню, позволяющий выделять некешируемые области для отладки. В окне отладки в разделе Эпилог > Ядро добавляется профайлинг каждого колбэка и функции подстановки контента в некешируемые области (время её работы - суммарное с учётом колбэков).

## Пример.

### Задача.

Если товар интернет-магазина находится в корзине пользователя, выводить ссылку "Уже в корзине" с указанием количества заказанного товара. В ином случае - вывести ссылки "Купить" и в "Корзину".

### Решение.

За основу взят стандартный шаблон books интернет-магазина.

1. Подключаем модуль некешируемых областей в файле `./bitrix/php_interface/init.php`:

```
CModule::IncludeModule('itconstruct.uncachedarea');
```

2. Формируем вёрстку блоков ссылок "Купить", "В корзину" и "Уже в корзине". Контент кладём в соответствующие переменные. Для удобства можно использовать функции захвата вывода:

```
<?itc\CUncachedArea::startCapture();?>
    <a href="<?echo $arResult["BUY_URL"]?>" rel="nofollow">Купить</a> <a href="<?echo
    $arResult["ADD_URL"]?>" rel="nofollow">В корзину</a>
<?$buy = itc\CUncachedArea::endCapture();?>
<?itc\CUncachedArea::startCapture();?>
    <span class="in-basket">Уже <a href="/personal/cart/">в корзине (#COUNT#)</a></span>
<?$inBasket = itc\CUncachedArea::endCapture();?>
```

3. Открываем шаблон вывода списка товаров и помечаем место, в которое необходимо вставить одну из ссылок следующим кодом:

```
<?itc\CUncachedArea::show('productBuyBlock', array('id' => $arResult['ID'], 'buy' =>
    $buy, 'inBasket' => $inBasket))?>
```

Первым аргументом передаётся ключ размеченной области. Во втором - массив данных, который поступает в функцию обратного вызова (см. ниже). Колбэк вызывается всегда один раз вне зависимости от количества размещённых блоков с одинаковым кодом.

4. В любом месте объявляем и регистрируем обработчик - либо функцию, либо метод класса, либо анонимную функцию (PHP >= 5.3):

```
itc\CUncachedArea::registerCallback('productBuyBlock', function ($key, $data) {
    CModule::IncludeModule('sale');

    $resBasket = CSaleBasket::GetList(
        array("NAME" => "ASC"),
        array("FUSER_ID" => CSaleBasket::GetBasketUserID(), "LID" => SITE_ID, "ORDER_ID"
=> "NULL")
    );

    $products = array();
```

```

while ($arItem = $resBasket->Fetch()) {
    $products[$arItem['PRODUCT_ID']] = $arItem['QUANTITY'];
}

$output = array();

foreach($data as $item) {
    $subkey = itc\CUncachedArea::getSubkey($item);

    if (isset($products[$item['id']])) {
        $output[$subkey] = str_replace('#COUNT#', round($products[$item['id']],
$item['inBasket']));
    } else {
        $output[$subkey] = $item['buy'];
    }
}

return $output;
});

```

## Рекомендуемый вариант подключения модуля

Рекомендуется вынести все callback-функции в отдельный файл, например, `uncachedarea.php`, и разместить в `init.php` подключение этого файла, совмещённое с подключением модуля:

```

if (CModule::IncludeModule('itconstruct.uncachedarea') &&
file_exists($_SERVER['DOCUMENT_ROOT'] . '/bitrix/php_interface/uncachedarea.php')) {
    require_once($_SERVER['DOCUMENT_ROOT'] . '/bitrix/php_interface/uncachedarea.php');
}

```

## Class CUncachedArea

Package: Default

API

### Properties

  `array $_callbacks = array()`

static

Зарегистрированные обработчики

**Default value**

`array()`

**Details**

**Type**

array

  `string $_captureKey`

static

Ключ области для захватываемого контента

**Details**

**Type**

string

  `int $_captureSort`

`static`

Индекс сортировки захватываемого контента для области.

#### Details

#### Type

int

  `array $_contents = array()`

`static`

Назначенный через CUncachedArea::setContent() контент для размеченных областей

#### Default value

`array()`

#### Details

#### Type

array

## Methods

  `_allowAddToDebugPanel() : bool`

`static`

Проверка прав на добавление информации в панель дебага.

#### Returns

Type	Description
bool	



  `_allowHighlightUncachedArea() : bool`

`static`

Проверка прав для обозначения областей.

#### Returns

Type	Description
bool	



  `_cmpSort(array $a, array $b) : void`

`static`

Сортировка по полю sort.

#### Parameters

Name	Type	Description
<code>\$a</code>	array	
<code>\$b</code>	array	



  `_escapeKey(string $key) : void`

static

Экранирование служебных символов в ключе (коде) области

#### Parameters

Name	Type	Description
<code>\$key</code>	string	



  `_parseAreaData(string $data) : void`

static

Десериализация данных.

#### Parameters

Name	Type	Description
<code>\$data</code>	string	

  `_wrapContentToAttribute(string $key, string $value) : string`

static



Оборачивает контент в блок div, содержащий служебные данные.

#### Parameters

Name	Type	Description
<code>\$key</code>	string	
<code>\$value</code>	string	

#### Returns

Type	Description
string	



  `endCapture() : null | string`

static

Завершение захвата контента.

#### Returns

Type	Description
null   string	Если в <code>startCapture()</code> не передавался ключ, функция вернёт контент.

  `getSubkey(string $data) : string | array`

static

Возвращает второстепенный ключ по данным некешируемой области.

Используется в пользовательских колбэках.

#### Parameters

Name	Type	Description
\$data	string	Данные.

#### Returns



Type	Description
string   array	Если данные пусты, возвращается ключ в виде пустой строки.

  `onPageStart()` : void

`static`

Обработчик для OnPageStart.

Добавляет кнопку в верхнюю панель. Добавляет стиль, если включена подсветка областей.

  `processAreas(string $pageContent)` : void



`static`

Вставка контента в размеченные области.

Обработчик для OnEndBufferContent.

#### Parameters

Name	Type	Description
\$pageContent	string	

  `registerCallback(int $key, \Closure | string $callback, string $sort = 100)` : int

`static`



Регистрация функции обратного вызова для генерации некешируемого контента

#### Parameters

Name	Type	Description
\$key	int	
\$callback	<a href="#">\Closure</a>   string	
\$sort	string	Индекс сортировки.

#### Returns

Type	Description
int	Индекс колбэка (для удаления).



  `setContent(string $key, string $content, string $sort = 100)` : void

`static`

Назначает контент для определённого ключа.

#### Parameters

Name	Type	Description
<b>\$key</b>	string	Ключ (код) области
<b>\$content</b>	string	
<b>\$sort</b>	string	Индекс сортировки.

  `show(string $key, int | string | array | null $data = null) : bool`

**static**



Помечает область для вставки некешируемых данных

#### Parameters

Name	Type	Description
<b>\$key</b>	string	Ключ (код) для области.
<b>\$data</b>	int   string   array   null	Данные. Опционально.

#### Returns

Type	Description
bool	

  `startCapture(string $key = null, int $sort = 100) : void`



**static**

Старт захвата (capturing) контента.

Вся выводимая информация будет удалена после окончания захвата. Если ключ не передан, endCapture() вернёт контент. Если передан - назначит его для \$key.

#### Parameters

Name	Type	Description
<b>\$key</b>	string	
<b>\$sort</b>	int	

  `unregisterCallback(int $key, int $index) : bool`

**static**

Удаление функции обратного вызова.

#### Parameters

Name	Type	Description
<b>\$key</b>	int	
<b>\$index</b>	int	index Индекс колбэка.

#### Returns

Type	Description

Type	Description
bool	

Documentation was generated by [phpDocumentor 2.0.0a10](#).