

Управление ярлыками элемента со страницы редактирования элемента

На странице редактирования информационного элемента (или товара) в центре администрирования создадим вкладку «Группы», на которой будет выводиться иерархический список групп информационной системы. Возле каждой группы пусть отображается чекбокс, который означает должен ли существовать ярлык элемента в этой группе.



Для размещения вкладки на странице редактирования информационных элементов будем использовать событие контроллера форм:

`Admin_Form_Action_Controller_Type_Edit.onAfterRedeclaredPrepareForm` (6.5.2+),

...а для сохранения ярлыков событие:

`Admin_Form_Action_Controller_Type_Edit.onAfterExecute`.

Для размещения вкладки на странице редактирования товаров используем событие:

`Admin_Form_Action_Controller_Type_Edit.onAfterRedeclaredPrepareForm`

...для сохранения аналогично:

`Admin_Form_Action_Controller_Type_Edit.onAfterExecute`.

Установка решения

Чтобы все работало понадобится следующий код в файле `bootstrap.php`:

```
// KAD: Управления ярлыками элементов информационной системы в отдельной вкладке
// Событие onAfterRedeclaredPrepareForm работает начиная с версии 6.5.2
Core_Event::attach('Admin_Form_Action_Controller_Type_Edit.onAfterRedeclaredPrepareForm',
array('Kad_Admin_Form_Observer_ShortcutsTab_Informationssystem',
'onAfterRedeclaredPrepareForm'));
Core_Event::attach('Admin_Form_Action_Controller_Type_Edit.onAfterExecute', array('Kad_Admin_Form_Observer_ShortcutsTab_Informationssystem',
```

```
'onAfterExecute'));
```

```
// KAD: Управления ярлыками товара в отдельной вкладке
```

```
Core_Event::attach('Admin_Form_Action_Controller_Type_Edit.onBeforeExecute',  
array('Kad_Admin_Form_Observer_ShortcutsTab_Shop', 'onBeforeExecute'));  
Core_Event::attach('Admin_Form_Action_Controller_Type_Edit.onAfterExecute',  
array('Kad_Admin_Form_Observer_ShortcutsTab_Shop', 'onAfterExecute'));
```

...а также наблюдатели ([скачать файлом](#)):

Kad_Admin_Form_Observer_ShortcutsTab

```
<?php
```

```
/*
```

```
* Управления ярлыками в отдельной вкладке
```

```
*
```

```
* Базовый класс для информационной системы и магазина
```

```
*
```

```
* @author Maxim Zasorin, KAD Systems (©)
```

```
* @date 13-01-2016
```

```
*
```

```
* Копирование и использование файлов модуля
```

```
* в коммерческих целях ЗАПРЕЩЕНО
```

```
*
```

```
*/
```

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

```
abstract class Kad_Admin_Form_Observer_ShortcutsTab
```

```
{
```

```
    static protected $_shortcutsGroups = array();
```

```
    // Добавляет вкладку с ярлыками
```

```
    static protected function shortcutsTab($controller, $object)
```

```
    {
```

```
        $oItemShortcutsTab = Admin_Form_Entity::factory('Tab')
```

```
            ->caption(
```

```
                Core_I18n::instance()->getLng() == 'ru' ? 'Группы' : 'Groups'
```

```
            )
```

```
            ->name('shortcuts_groups');
```

```
        $controller->addTabBefore(
```

```
            $oItemShortcutsTab,
```

```
            $controller->getTab('additional')
```

```
        );
```

```
        static::addGroupsToTab($oItemShortcutsTab, $object);
```

```
    }
```

```

// Добавляет все группы на вкладку с ярлыками
static protected function addGroupsToTab($oShopItemShortcutsTab,
$object)
{
    $aoModuleGroups = static::getModuleGroups($object);

    foreach ($aoModuleGroups as $oModuleGroup)
    {
        static::addGroupToTab($oShopItemShortcutsTab, $object,
$oModuleGroup);
    }
}

// Добавляет одну группы на вкладку с ярлыками
static protected function addGroupToTab($oShopItemShortcutsTab, $object,
$oModuleGroup, $level = )
{
    $oShopItemShortcutsTabGroupRow = Admin_Form_Entity::factory('Div')
        ->class('row')
        ->style('padding-left: ' . (25 * $level) . 'px');
    $oShopItemShortcutsTab->add($oShopItemShortcutsTabGroupRow);

    $oModuleGroupCheckbox = Admin_Form_Entity::factory('Checkbox');
    $oModuleGroupCheckbox
        ->value(
            in_array($oModuleGroup->id,
static::getShortcutsGroups($object)) ? 1 :
            )
        ->caption($oModuleGroup->name)
        ->name('shortcut_group_' . $oModuleGroup->id)
        ->divAttr(array('class' => 'form-group col-lg-6 col-md-6 col-
sm-6'))
        ;

    if (static::getObjectGroupId($object, true) == $oModuleGroup->id)
    {
        $oModuleGroupCheckbox->disabled = 'disabled';
    }

    $oShopItemShortcutsTabGroupRow->add($oModuleGroupCheckbox);

    $aoChildGroups = static::getChildGroups($oModuleGroup)->findAll();

    foreach ($aoChildGroups as $oChildGroup)
    {
        static::addGroupToTab($oShopItemShortcutsTab, $object,
$oChildGroup, $level + 1);
    }
}

// Возвращает группы, в которых присутствую ярлыки объекта

```

```

static protected function getShortcutsGroups($object)
{
    if (count(self::$_shortcutsGroups) > )
    {
        return self::$_shortcutsGroups;
    }

    $aoShortcuts = static::getShortcuts($object)->findAll();
    // $aShortcutsGroups = array();
    $aShortcutsGroups = array(static::getObjectGroupId($object, true));

    foreach ($aoShortcuts as $oShortcut)
    {
        $aShortcutsGroups[] = static::getObjectGroupId($oShortcut);
    }

    return (self::$_shortcutsGroups = $aShortcutsGroups);
}

// Сохраняем ярлыки групп
static protected function saveShortcuts($controller, $object)
{
    $aoModuleGroups = static::getModuleGroups($object, true);

    foreach ($aoModuleGroups as $oModuleGroup)
    {
        $oGroupShortcuts = static::getShortcuts($object, $oModuleGroup);
        $aoGroupShortcuts = $oGroupShortcuts->findAll();

        if (Core_Array::getPost('shortcut_group_' . $oModuleGroup->id))
        {
            if (count($aoGroupShortcuts) == )
            {
                $object->shortcut($oModuleGroup->id);
            }
        }
        else
        {
            foreach ($aoGroupShortcuts as $oGroupShortcut)
            {
                $oGroupShortcut->delete();
            }
        }
    }
}

// Возвращает подгруппы
abstract static protected function getChildGroups($oGroup);

// Возвращает все группы текущего модуля
abstract static protected function getModuleGroups($object, $all =

```

```

false);

    // Возвращает группу объекта
    abstract static protected function getObjectGroupId($object, $original =
false);

    // Возвращает ярлыки объекта
    abstract static protected function getShortcuts($object, $oGroup =
null);
}

```

Kad_Admin_Form_Observer_ShortcutsTab_Informationssystem

```

<?php

/*
 * Управления ярлыками элементов информационной системы в отдельной вкладке
 *
 * @author Maxim Zasorin, KAD Systems (©)
 * @date 13-01-2016
 *
 *
 * B bootstrap.php:
 *
 // KAD: Управления ярлыками элементов информационной системы в отдельной вкладке
 // Событие onAfterRedeclaredPrepareForm работает начиная с версии 6.5.2
Core_Event::attach('Admin_Form_Action_Controller_Type_Edit.onAfterRedeclared
PrepareForm',
array('Kad_Admin_Form_Observer_ShortcutsTab_Informationssystem',
'onAfterRedeclaredPrepareForm'));
Core_Event::attach('Admin_Form_Action_Controller_Type_Edit.onAfterExecute',
array('Kad_Admin_Form_Observer_ShortcutsTab_Informationssystem',
'onAfterExecute'));

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

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

```

class Kad_Admin_Form_Observer_ShortcutsTab_Informationssystem extends
Kad_Admin_Form_Observer_ShortcutsTab
{
    // Событие
    static public function onAfterRedeclaredPrepareForm($controller, $args)
    {
        list($object, $Admin_Form_Controller) = $args;

        // Данное событие будет вызываться для всех форм, определяем с каким

```

контроллером работаем

```
switch (get_class($controller))
{
    case 'Informationssystem_Item_Controller_Edit':
        if (get_class($object) == 'Informationssystem_Item_Model')
        {
            self::shortcutsTab($controller, $object);
        }
        break;
}

// Обрабатываем операцию сохранения формы
static public function onAfterExecute($controller, $args)
{
    list($operation, $Admin_Form_Controller) = $args;

    if ($operation == 'save' || $operation == 'apply')
    {
        $object = $controller->getObject();

        switch (get_class($controller))
        {
            case 'Informationssystem_Item_Controller_Edit':
                if (get_class($object) ==
'Informationssystem_Item_Model')
                {
                    self::saveShortcuts($controller, $object);
                }
                break;
        }
    }
}

static protected function getChildGroups($oGroup)
{
    return $oGroup->Informationssystem_Groups;
}

static protected function getModuleGroups($object, $all = false)
{
    $oModuleGroups =
$object->Informationssystem->Informationssystem_Groups;
    $oModuleGroups->queryBuilder()
        ->where('active', '=', 1)
        ->where('deleted', '=', );

    if (!$all)
    {
        $oModuleGroups->queryBuilder()->where('parent_id', '=', );
    }
}
```

```

        return $oModuleGroups->findAll();
    }

    static protected function getObjectGroupId($object, $original = false)
    {
        if ($object->shortcut_id != && $original)
        {
            $object = $object->Informationssystem_Item;
        }

        return $object->informationssystem_group_id;
    }

    static protected function getShortcuts($object, $oGroup = null)
    {
        if ($object->shortcut_id == )
        {
            $oShortcuts = $object->Informationssystem_Items;
        }
        else
        {
            $oShortcuts =
            $object->Informationssystem_Item->Informationssystem_Items;
        }

        if ($oGroup)
        {
            $oShortcuts->queryBuilder()
                ->where('informationssystem_group_id', '=', $oGroup->id);
        }

        return $oShortcuts;
    }
}

```

Kad_Admin_Form_Observer_ShortcutsTab_Shop

```
<?php
```

```

/*
 * Управления ярлыками товара в отдельной вкладке
 *
 * @author Maxim Zasorin, KAD Systems (©)
 * @date 13-01-2016
 *
 *
 * B bootstrap.php:
 *
 // KAD: Управления ярлыками товара в отдельной вкладке

```

```
Core_Event::attach('Admin_Form_Action_Controller_Type_Edit.onBeforeExecute',
array('Kad_Admin_Form_Observer_ShortcutsTab_Shop', 'onBeforeExecute'));
Core_Event::attach('Admin_Form_Action_Controller_Type_Edit.onAfterExecute',
array('Kad_Admin_Form_Observer_ShortcutsTab_Shop', 'onAfterExecute'));
```

** Копирование и использование файлов модуля*

** в коммерческих целях ЗАПРЕЩЕНО*

**/*

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

```
class Kad_Admin_Form_Observer_ShortcutsTab_Shop extends
Kad_Admin_Form_Observer_ShortcutsTab
```

```
{
```

```
    // Событие
```

```
    static public function onBeforeExecute($controller, $args)
```

```
    {
```

```
        list($operation, $Admin_Form_Controller) = $args;
```

```
        if (is_null($operation))
```

```
        {
```

```
            switch (get_class($controller))
```

```
            {
```

```
                case 'Shop_Item_Controller_Edit':
```

```
                    $object = $controller->getObject();
```

```
                    if (get_class($object) == 'Shop_Item_Model')
```

```
                    {
```

```
                        self::shortcutsTab($controller, $object);
```

```
                    }
```

```
                break;
```

```
            }
```

```
        }
```

```
    }
```

```
    // Обрабатываем операцию сохранения формы
```

```
    static public function onAfterExecute($controller, $args)
```

```
    {
```

```
        list($operation, $Admin_Form_Controller) = $args;
```

```
        if ($operation == 'save' || $operation == 'apply')
```

```
        {
```

```
            $object = $controller->getObject();
```

```
            switch (get_class($controller))
```

```
            {
```

```
                case 'Shop_Item_Controller_Edit':
```

```
                    if (get_class($object) == 'Shop_Item_Model')
```

```
                    {
```

```
                        self::saveShortcuts($controller, $object);
```

```
                    }
```

```

        break;
    }
}

static protected function getChildGroups($oGroup)
{
    return $oGroup->Shop_Groups;
}

static protected function getModuleGroups($object, $all = false)
{
    $oModuleGroups = $object->Shop->Shop_Groups;
    $oModuleGroups->queryBuilder()
        ->where('active', '=', 1)
        ->where('deleted', '=', );

    if (!$all)
    {
        $oModuleGroups->queryBuilder()->where('parent_id', '=', );
    }

    return $oModuleGroups->findAll();
}

static protected function getObjectGroupId($object, $original = false)
{
    if ($object->shortcut_id != && $original)
    {
        $object = $object->Shop_Item;
    }

    return $object->shop_group_id;
}

static protected function getShortcuts($object, $oGroup = null)
{
    if ($object->shortcut_id == )
    {
        $oShortcuts = $object->Shop_Items;
    }
    else
    {
        $oShortcuts = $object->Shop_Item->Shop_Items;
    }

    if ($oGroup)
    {
        $oShortcuts->queryBuilder()
            ->where('shop_group_id', '=', $oGroup->id);
    }
}

```

```
    return $oShortcuts;  
  }  
}
```