



При создании собственного компонента часто необходимо ввести поле "опубликовано". Но я столкнулся с тем, что сами кнопки появляются, но не работают, то есть они отображают текущее состояние, но при нажатии не меняют его. Долго пытался найти решение, а в итоге "выдумал" его сам.

Joomla публикация в компоненте с использованием jgrid.published

Первым делом в нужной таблице надо создать поле **published**. Для экономии ресурсов достаточно tinyint. По умолчанию либо 1, либо 0 (опубликовано, неопубликовано, соответственно).

В установочном SQL-файле добавляем:

```
published TINYINT NOT NULL DEFAULT '1',
```

Добавим кнопки **публикации в компонент** в файл представления view.html.php:

```
protected function addToolBar()
{
    .....
    JToolBarHelper::publishList ();
```

```
JToolBarHelper::unpublishList ();  
.....  
}
```

Добавляем jgrid в шаблон компонента

В название колонки с публикацией вставляем:

```
sortDirection, $this->sortColumn); ?>
```

Если у вас по какой-то причине не используется сортировка, то можно вставить просто надпись "Публикация" или через JText и языковой файл.

В цикле вывода записей в таблицу вставляем:

```
echo JHTML::_('jgrid.published', $item->published, $i);
```

Об этом достаточно написано в интернете и проблем вызвать не должно.

Если после того, как вы проделали все выше написанное, **публикация в joomla компоненте** не работает, значит надо немного изменить контроллер и модель.

Изменяем контроллер для корректной работы публикации в компоненте

Проблема в том, что стандартная функция publish **публикации в joomla** работает если класс вашего контроллера наследуется от

JControllerAdmin

, а не от

JController

.

Если вы по какой-то причине хотите оставить JController, то редактируем функцию __construct()

:

```
function __construct()
{
parent::__construct();
$this->registerTask('unpublish', 'publish');
$this->registerTask('publish', 'publish');
}
```

И добавляем функцию publish:

```
public function publish()
{
JSession::checkToken() or die(JText::_('JINVALID_TOKEN'));

$cid = JRequest::getVar('cid', array(), '', 'array');
$data = array('publish' => 1, 'unpublish' => 0);
$task = $this->getTask();
$value = JFactoryHelper::getValue($data, $task, 0, 'int');

if (empty($cid))
{
JError::raiseWarning(500, JText::_($this->text_prefix . '_NO_ITEM_SELECTED'));
}
else
{
$model = $this->getModel('yourmodel');
JArrayHelper::toInteger($cid);
if (!$model->publish($cid, $value))
{
JError::raiseWarning(500, $model->getError());
}
else
{
if ($value == 1)
{
$text = JText::_('_ITEMS_PUBLISHED');
}
elseif ($value == 0)
{
$text = JText::_('_ITEMS_UNPUBLISHED');
}
$this->setMessage(JText::plural($text, count($cid)));
}
}
$this->setRedirect(JRoute::_("index.php?option=com_yourcomponent&controller=yourcontrol&task=yourtask"));
}
```

```
false
)
)
;
}
```

Не забудьте подставить в функцию ваши названия модели, контроллера и пр.

Добавление функции publish в модель компонента

Последнее, что нам надо сделать - это изменить модель для корректной работы встроенной **публикации Joomla!**:

```
public function publish(&$pks, $value = 1)
{
// Initialise variables.
$dispatcher = JDispatcher::getInstance();
$user = JFactory::getUser();
$table = $this->getTable();
$pks = (array) $pks;

// Include the content plugins for the change of state event.
JPluginHelper::importPlugin('content');

// Access checks.
foreach ($pks as $i => $pk)
{
$table->reset();
}

// Attempt to change the state of the records.
if (!$table->publish($pks, $value, $user->get('id')))
{
$this->setError($table->getError());
return false;
}

$context = $this->option . '.' . $this->name;

// Trigger the onContentChangeState event.
$result = $dispatcher->trigger($this->event_change_state, array($context, $pks, $value));

if (in_array(false, $result, true))
{
$this->setError($table->getError());
}
```

```
return false;  
}
```

```
// Clear the component's cache  
$this->cleanCache();
```

```
return true;  
}
```

Вот и все. Теперь **публикация в компоненте** работает.