

🏠 (/) > Курсы обучения (training/) > Создание компонентов (training/course1/) >

* Пишем интерфейс: виджеты ExtJS и процессоры

✎ Редактировать (<http://bezumkin.ru/topic.html?tid=2166>)

На прошлом уроке мы разобрались с контроллерами Custom Manager Page (CMP) нашего компонента, и выяснили, что основной смысл их существования — подготовить все нужные файлы для вывода страницы.

Пока у нас используются js файлы от modExtra, и сегодня нам нужно их изучить и переписать для Sendex.

Первым делом нужно понять, что ExtJS — это javascript фреймфорк, куда более мощный чем jQuery. Он не требует HTML верстки, он все генерирует на лету. То есть, вы пишете javascript код, а на странице получаете готовые таблички, кнопочки, пагинацию и т.д.

Причем, все эти элементы уже подчиняются какой-то логике: одна кнопка переключает странички, другая вызывает модальное окно и т.д.

Сначала довольно трудно въехать в ExtJS, особенно если не знаешь javascript, но со временем начинаешь понимать его преимущества, и реально экономить время.

Виджеты

Сразу говорю, я не гуру ExtJS, освоил его на минимальном уровне, необходимом для написания компонентов MODX. Поэтому, буду писать «своими словами», и возможно не всегда верно.

На мой взгляд, 2 основных момента для понимания ExtJS:

1. Все элементы ExtJS являются объектами javascript, со своими свойствами. Когда вы пишете какой-то свой виджет, нужно просто унаследовать и расширить уже готовый от авторов ExtJS или MODX. Например, пишете свою таблицу — расширяете **MODx.grid.Grid**, в таблицах будут всплывающие окошки — пишете виджет, расширяющий **MODx.Window**

Это очень похоже на Объектно-Ориентированное Программирование (ООП) в PHP и если принять эту логику — жизнь становится очень простой и лёгкой.

2. Практически все виджеты или уже работают, или умеют работать через Ajax. Все свои данные они отправляют на сервер без перезагрузки страницы, и ответ получают в формате JSON.

То есть, здесь полнейшее отделение логики от представления. Вам не нужно что-то там парсить или рендерить на PHP, чтобы отдать готовый HTML. Вставлять его тоже куда не нужно, сервер должен отдать массив в таком формате:

```
$array = array(  
    'success' => false,  
    'message' => 'Возника ошибка при сохранении формы',  
    'total' => 1  
    'data' => array(  
        'id' => 'alias',  
        'msg' => 'Это поле обязательно'  
    )  
);
```

А виджет ExtJS смотрит в этот массив, понимает, что сервер отдал ошибку и выводит сообщение о ней. В данном примере — это требование заполнить alias страницы при её сохранении.

Уверяю вас, такая схема делает работу **очень простой и удобной**. Вы отдельно пишете код на javascript, отдельно пишете процессоры на PHP, которые с ним будут работать, а потом открываете админку — и начинается магия!

Всё структурировано, никакой мешанины в коде, недоработки легко находятся и исправляются. Готовые виджеты, которые можно вызвать

на странице одной строчкой я называю **xtype** — потому что, именно это и указывается при их вызове.

Процессоры

Мы не будем рассматривать старые скучные процессоры, в которых были просто портянки кода, которые возвращали JSON. С версии MODX 2.2 у нас появились «классные процессоры», о которых вы можете почитать вот здесь (<http://bezumkin.ru/sections/php/59/>) — с ними мы и будем работать.

Итак, процессор это php файл, который содержит класс, наследующий **modProcessor** и возвращающий своё имя. В MODX есть уже куча готовых классов на разные случаи жизни, которые можно наследовать для своих целей:

- **modObjectGetProcessor** — Получает и возвращает указанный объект
- **modObjectGetListProcessor** — Выводит список объектов для таблицы
- **modObjectCreateProcessor** — Создаёт новый объект
- **modObjectUpdateProcessor** — Обновляет существующий объект
- **modObjectDuplicateProcessor** — Делает копию существующего объекта
- **modObjectRemoveProcessor** — Удаляет объект

То есть, здесь нам тоже, как правило, нужно просто унаследовать уже имеющийся класс, переопределить пару свойств и методов — и всё готово.

Сам файл-процессор состоит из класса, и обязан содержать в конце вернуть своё имя. Вот, простейший пример:

```
<?php
class SendexItemCreateProcessor extends modObjectCreateProcessor {
    // Тип объекта, с которым будем работать - это имя выводится в логе операций MODX
    // и подставляется в сообщения об ошибках
    public $objectType = 'sxNewsletter';
    // xPDO класс объекта
    public $classKey = 'sxNewsletter';
    // Используемый лексикон - нужен для вывода ошибок
    public $languageTopics = array('sendex');
    // Разрешение, которое проверяется при работе с процессором
    // Можно оставить пустым - тогда смогут работать все менеджеры
    public $permission = 'new_document';
}
// Обязательно возвращаем имя класса, чтобы MODX знал, что именно нужно инициализировать в процессоре
return 'SendexItemCreateProcessor';
```

Это процессор абсолютно рабочий, его можно вызвать через `$modx->runProcessor()`, передать нужные параметры и будет создан объект `sxNewsletter` (или получена ошибка).

Для того, чтобы узнать, какие методы можно переопределить — просто смотрите в исходник наследуемого `modObjectCreateProcessor` (<https://github.com/modxcms/revolution/blob/c2d6582fe7a570ab634de2b9bb42029ec4909905/core/model/modx/modprocessor.class.php#L599>). Напоминаю, что в PhpStorm нужно просто кликнуть на класс и нажать **Ctrl+B**.

С теорией покончено, давайте закрепим на практике.

Основные js файлы админки

От `modExtra` у нас осталась демонстрационная страница, которая создаётся следующими скриптами в `/assets/components/sendex/js/mgr/`:

- **sendex.js** — скрипт с объявлением самого объекта `Sendex`. В него будут вписываться все остальные объекты компонента. Этот файл мы не трогаем.
- **sections/home.js** — Скрипт единственной страницы компонента. Также не трогаем, но обращаем внимание на строку:
`,renderTo: 'sendex-panel-home-div'`

И вспоминаем, что именно такой `div` у нас указан в шаблоне

(<https://github.com/bezumkin/Sendex/blob/5416d620300261025420f9e73c41ee3a6fb9fd5a/core/components/sendex/elements/templates/home.tpl>), который загружается контроллером

(<https://github.com/bezumkin/Sendex/blob/5416d620300261025420f9e73c41ee3a6fb9fd5a/core/components/sendex/controllers/home.class.php#L45>).

А потом смотрим на строку, где объявляется **xtype** нашей страницы:

```
xtype: 'sendex-panel-home'
```

Именно этот `xtype` у нас и запускает контроллер

(<https://github.com/bezumkin/Sendex/blob/5416d620300261025420f9e73c41ee3a6fb9fd5a/core/components/sendex/controllers/home.class.php#L35>)

после полной загрузки страницы.

То есть, вот она связь HTML страницы, сгенерированной PHP, и запуск виджета ExtJS на ней.

- **widgects/home.panel.js** — виджет «Панель», который содержит в себе заголовок, описание и табы (вкладки). Его мы немножко будем менять, когда будем добавлять новые вкладки на страницу.
- **widgects/items.grid.js** — виджет с таблицей предметов modExtra. Основной рабочий файл, который выводит записи из БД и позволяет редактировать их во всплывающих окошках. В основном с ним мы и работаем.

Давайте еще раз: у нас есть раздел в админке, он состоит из одного пункта меню и одной страницы. На странице вызывается панель с вкладками, и на первой вкладке у нас таблица. Окошки, создающие новую запись и меняющие существующую, являются частью этой таблицы.

Сегодня работаем именно в этом объеме, другие вкладки добавим позже.

Готовим таблицу для работы с sxNewsletter

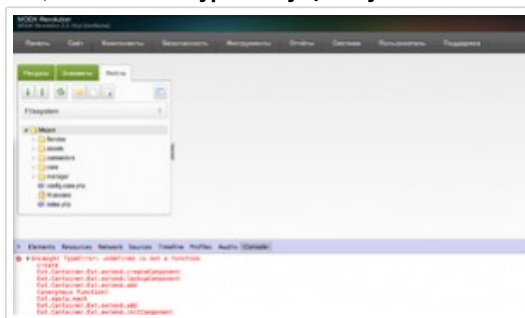
Первая вкладка посвящена работе с объектами **sxNewsletter**, то есть — с подписками.

Нужно указать это в **widgects/home.panel.js**:

```
{
    // Имя вкладки
    title: _('sendex_newsletters')
    // Массив с содержимым
    ,items: [
        // Блок HTML с описанием этой вкладки
        {
            html: _('sendex_newsletters_intro')
            ,border: false
            ,bodyCssClass: 'panel-desc'
            ,bodyStyle: 'margin-bottom: 10px'
        }
        // Вызов рабочей таблицы
        ,{
            xtype: 'sendex-grid-newsletters'
            ,preventRender: true
        }
    ]
}
```

Как видите, я просто переименовал кое-что. Самый главный момент здесь, это вызов **xtype** таблицы `xtype: 'sendex-grid-newsletters'`

Как только мы его указали — админка ломается, ибо **такого xtype не существует**.



(<http://st.bezumkin.ru/files/0/b/a/0baeaa76c71f0bbac0677ab549c53032.png>)

Как понять, из-за чего сломалось? Ну так нам ExtJS пишет в логе, что перед ошибкой было `create`

```
Ext.Container.Ext.extend.createComponent  
Ext.Container.Ext.extend.lookupComponent
```

То есть: поиск и создание виджета, после чего ошибка. Мы же помним, что мы его переименовали — так что всё логично.

Таким образом, нужно поменять имя и в файле `widgects/items.grid.js`. Хотя стоп, его сначала нужно переименовать в `widgects/newsletters.grid.js`.

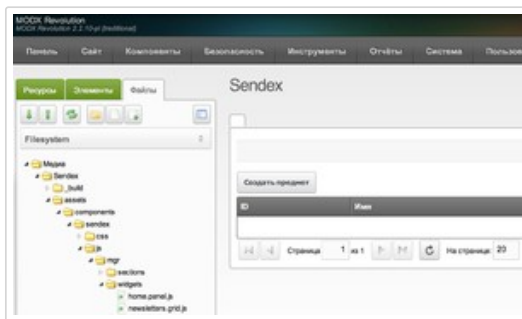
Сразу нужно указать новый файл для загрузки в контроллере **controllers/home.class.php** на строке 30:

```
$this->addJavascript($this->Sendex->config['jsUrl'] . 'mgr/widgets/newsletters.grid.js');
```

Ну а теперь просто меняем в `newsletters.grid.js` строку 108 с регистрацией виджета:

```
Ext.reg('sendex-grid-newsletters',Sendex.grid.Items);
```

Синхронизируем с MODXCloud и обновляем страницу — ошибки нет.



(<http://st.bezumkin.ru/files/d/f/f/df9bc32abb127247928863adbb65f0c.png>)

Кстати говоря, обычно с переименованием через SFTP возникают какие-то ошибки, и часто PhpStorm просто создаёт на сервере новый файл рядом со старым. Это не страшно, просто удаляйте его через админку.

У вас должно получиться всё как на скриншоте выше. Обратите внимание, что у меня не отображается заголовок вкладки. Так и должно быть — я же указал несуществующую запись из лексикона.

Вообще, поддержка лексиконов в компонентах сделана отлично, вам нужно просто указывать ключ из ваших файлов лексикона в виде: `_('ключ')`

Нижнее подчеркивание здесь — это функция MODX, которая выведет нужную фразу. Очень элегантное решение, не правда ли?

Поэтому редактируем

`/core/components/sendex/lexicon/ru/default.inc.php`

```
$_lang['sendex'] = 'Sendex';  
$_lang['sendex_menu_desc'] = 'Управление подписками';
```

```
$_lang['sendex_newsletters'] = 'Подписки';  
$_lang['sendex_newsletters_intro'] = 'На этой странице вы создаёте и редактируете ваши подписки.';
```

и `/core/components/sendex/lexicon/en/default.inc.php`.

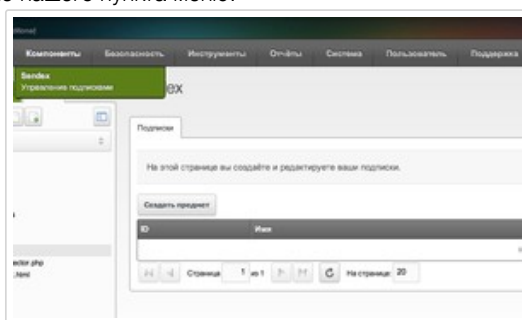
```
$_lang['sendex'] = 'Sendex';  
$_lang['sendex_menu_desc'] = 'Newsletters management';
```

```
$_lang['sendex_newsletters'] = 'Newsletters';  
$_lang['sendex_newsletters_intro'] = 'On this page you create and edit your subscription.';
```

Напоминаю, что лексиконы Sendex загружает контроллер `home.class.php`

(<https://github.com/bezumkin/Sendex/blob/5416d620300261025420f9e73c41ee3a6fb9fd5a/core/components/sendex/controllers/home.class.php#L22>).

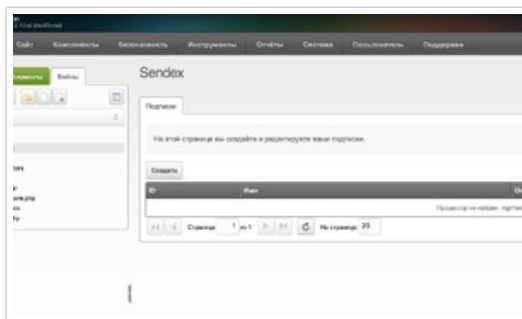
После синхронизации языковых файлов с сервером, нужно обязательно почистить кэш сайта, ибо MODX постоянно кэширует лексиконы. Как видите, заодно я еще поправил описание нашего пункта меню.



(<http://st.bezumkin.ru/files/2/4/d/24de785719334d72379f30b510afe01e.png>)

Английский у меня плохонький, поэтому пишу абы как, через Яндекс.Переводы (<http://translate.yandex.ru/>).

Ну а теперь в `newsletters.grid.js` осталось переименовать все встречающиеся **Items** в **Newsletters**, а **item** в **newsletter**. В результате этого пропадает кнопка создания (меняем ей запись на `sendex_btn_create` и добавляем это в лексиконы) и получается ошибка

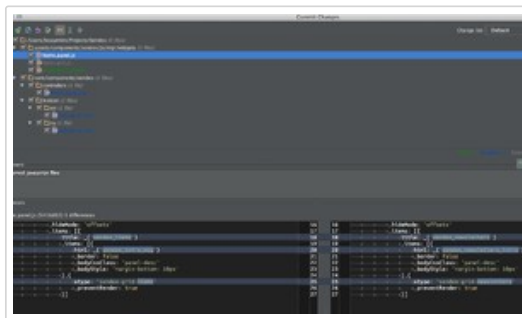


(<http://st.bezumkin.ru/files/f/f/5/ff59297a107de799057d5d3497c9d96f.png>)

Процессор не найден: mgr/newsletter/getlist

Конечно не найден, мы же переименовали всё только в javascript файлах, а php остались где были. Кстати, вот мой коммит со всеми изменениями (<https://github.com/bezumkin/Sendex/commit/fbb80934e3e220da52521ecf0f98b5388930acb8>) на текущий момент — сравнивайте.

Вот картинка с изменёнными файлами:



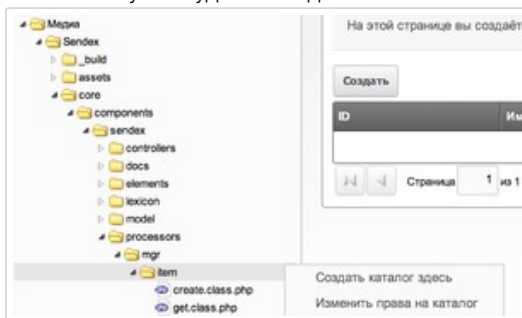
(<http://st.bezumkin.ru/files/6/8/e/68e1d17d30dadc47bfef706d59e170dc.png>)

Теперь нужно переделать процессоры.

Готовим процессоры для работы с sxNewsletter

Первым делом переименовываем `/core/components/sendex/processors/mgr/core/components/sendex/processors/mgr/newsletter/create.class.php` в `/core/components/sendex/processors/mgr/newsletter/` и заливаем новую директорию на сервер.

У меня получилось, что старая тоже осталась — поэтому я её удаляю в админке:



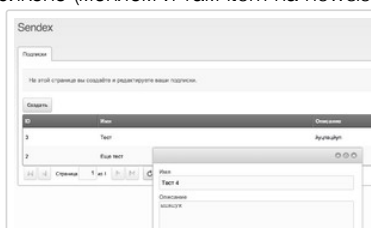
(<http://st.bezumkin.ru/files/e/c/8/ec818433909575f2b17bf86c5b6ea3f8.png>)

Затем меняем классы в файлах процессорах. Просто заменяем **SendexItem**, получившийся при изначальном переименовании modExtra, на **sxNewsletter** — наш объект с подписками.

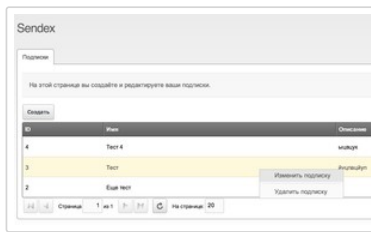
После переименования синхронизируем изменения файлов, обновляем страницу в админке и ошибка пропадает.

Более того, можно **создавать, изменять и удалять подписки**, так как основные поля объекта modExtraItem и sxNewsletter совпадают (name и description).

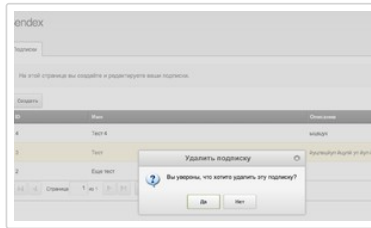
Затем приводим в порядок остальные записи в лексиконе (меняем и там item на newsletter) и получается все довольно приятно:



(<http://st.bezumkin.ru/files/9/f/f/9ffd0a52efc09ee47d173b5130b39e96.png>)



(<http://st.bezumkin.ru/files/f/0/0/f00ef55401d99f7a53736d0a9791ae6f.png>)

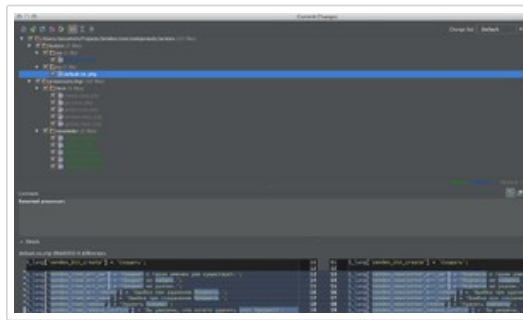


(<http://st.bezumkin.ru/files/e/a/4/ea460718e8f77fb57bc6ac780a89c863.png>)

Вот и всё, для начала. Мы уже можем управлять объектами — у нас есть базовый интерфейс на ExtJS, который работает с процессорами и поддерживает сразу 2 языка: русский и английский.

Вот мой коммит с изменениями (<https://github.com/bezumkin/Sendex/commit/3e3d6d5c420e9a6d452ed61b6bd573613e11d467>).

Вот какие файлы я редактировал:



(<http://st.bezumkin.ru/files/0/1/c/01c28188d02c5ef4f579c9fd457db279.png>)

Заключение

Дальнейшая работа сводится к трём пунктам:

- Придумали и нарисовали что-то на ExtJS
- Добавили работу с этим в процессоры
- Добавили нужные записи в словари

Конечно, нужно будет создавать новые javascript файлы и процессоры, подключать их в контроллерах и вызывать из виджетов, но в целом — вот она, основная база дальнейшей разработки.

Выходит несложно, если работать с заготовкой modExtra.

На следующем уроке мы плотнее поработаем с ExtJS и сделаем свой интерфейс для sxNewsletter.

← Следующая заметка (training/course1/2184/)

Пишем интерфейс: таблица подписок и окошко создания

Предыдущая заметка (training/course1/2150/) →

Собираем и устанавливаем первую версию пакета

📅 19 ноября 2013, 12:31 | 👤 Василий Наумкин (/user/topics/bezumkin) | 👁 19

Комментарии (10)

☑ Уведомлять о новых комментариях



Andrei Kilin (/user/comments/k07n)

20 ноября 2013, 10:11

(<http://bezumkin.ru/training/course1/2166/#comment-17601>)

Нужно указать это в sections/home.js:

в widgets/home.panel.js

 ответить



Василий ([user/comments/5636504/](/user/comments/5636504/)) 20 ноября 2013, 16:13 # (<http://bezumkin.ru/training/course1/2166/#comment-17625>)

У вас должно получиться всё как на скриншоте выше.

До этого момента проделал все трижды с откатами. Таблица как на скриншоте так и не появилась =\
В консоли — «Uncaught TypeError: Cannot call method 'request' of undefined».

Конечные файлы пока не синхронизирую с твоими коммитами, т.к., видимо, ошибка где-то в промежуточном шаге =\
 ответить



Василий Наумкин ([user/comments/bezumkin/](/user/comments/bezumkin/)) 20 ноября 2013, 16:14 *(комментарий был изменён)*

(<http://bezumkin.ru/training/course1/2166/#comment-17626>)

Что-то где-то не переименовал.

А с сервером то не забываешь синхронизировать изменения? У тебя есть на сервере файл widgets/newsletters.grid.js?

Ну и еще бывает, что браузер сильно кэширует — нужно обновлять через Ctrl+F5.

 ответить



Василий ([user/comments/5636504/](/user/comments/5636504/)) 20 ноября 2013, 16:18 # (<http://bezumkin.ru/training/course1/2166/#comment-17627>)

Вроде как да, после всех изменений синхронизация. Но так как при сохранении файла идет его автоматическая загрузка — таблица синхронизации пустая. Единственный момент — вручную удалял item.grid.js, как ты и предупреждал.

 ответить



Василий Наумкин ([user/comments/bezumkin/](/user/comments/bezumkin/)) 20 ноября 2013, 16:19 *(комментарий был изменён)*

(<http://bezumkin.ru/training/course1/2166/#comment-17628>)

Пришли данные для входа на сайт на bezumkin@ya.ru

 ответить

2



Василий Наумкин ([user/comments/bezumkin/](/user/comments/bezumkin/)) 20 ноября 2013, 16:31

(<http://bezumkin.ru/training/course1/2166/#comment-17629>)

Ты на прошлом уроке пропустил подготовку контроллеров.

Там нужно заменить

```
$this->modx->regClientCSS()
```

на

```
$this->addCSS()
```

и т.д.

 ответить



Василий ([user/comments/5636504/](/user/comments/5636504/)) 20 ноября 2013, 16:38

(<http://bezumkin.ru/training/course1/2166/#comment-17630>)

Если вы хотите, чтобы ваш компонент дружил с AjaxManager — их нужно заменить

Видимо, повелся на «Если», т.к. AjaxManager там не установлен.

Спасибо, будем продолжать!

 ответить



Василий Наумкин ([user/comments/bezumkin/](/user/comments/bezumkin/)) 20 ноября 2013, 16:39

(<http://bezumkin.ru/training/course1/2166/#comment-17631>)

Ну, просто дальше уже идут расхождения в коде и будет труднее.

miniShop2 ▲

Поэтому — делай как я =)

 ответить

Tickets ▲



Илья Ершов ([user/comments/WEB-master/](/user/comments/WEB-master/)) 2 часа назад *(комментарий был изменён)*

(<http://bezumkin.ru/training/course1/2166/#comment-17754>)

Кому надо — могу поделиться книжкой по ExtJS 4, издательство РАСКТруб, с архивом с примерами кода. Добавляйтесь в скайп: erшов.ilya (да простят меня буржуйские издатели)

Книжка в формате .erub (формат родной Apple, читалка для Win тоже есть), на английском языке. В оригинале бумажная стоит 100\$,

цифровое издание я покупал за 30\$.

С целью поддержания MODX сообщества книжку кину безвозмездно ;)

Василий, этот коммент можно через пару дней удалить :)

 [ответить](#)





Илья Ершов ([user/comments/WEB-master/](/user/comments/WEB-master/)) 1 минута назад # (<http://bezumkin.ru/training/course1/2166/#comment-17760>)

Рабочий день заканчивается, решил распечатать две последние недочитанные статьи, дочитать и осознать уже дома. Из принтера выскочил заголовок, 9 пустых листов, и все комментарии к статье...

Мда... я понимаю защита авторских прав, но всё равно, неожиданно, неудобно и слегка не приятно...

Нет никакой защиты, скорее это скрипт смены размера шрифта мешается.


 [Предпросмотр](#)

 [Написать](#)



Для вставки изображений нужно использовать "Файлохранилище (<http://st.bezumkin.ru>)".

[Оставить новый комментарий](#)

Читатель

 **Василий Наумкин** (</user/topics/bezumkin/>)



 [Настройки](http://bezumkin.ru/profile/settings/) (<http://bezumkin.ru/profile/settings/>)  [Выход](http://bezumkin.ru/training/course1/2166/?hauth_action=logout) (http://bezumkin.ru/training/course1/2166/?hauth_action=logout)

 [Написать](http://bezumkin.ru/topic.html) (<http://bezumkin.ru/topic.html>)

Поиск

Поиск



Новости магазина

Новый способ оплаты заказов в miniShop2 -

mSpQIwi

([http://store.simplifiedream.ru/mspqiwi?](http://store.simplifiedream.ru/mspqiwi?msfrom=bezumkin)

[msfrom=bezumkin](http://store.simplifiedream.ru/mspqiwi?msfrom=bezumkin))

Прямой эфир


 Илья Ершов (/user/comments/WEB-master/) меньше минуты назад

* Пишем интерфейс: виджеты ExtJS и процессоры (training/course1/2166/#comment-17760)  10

 Владимир Кисилица (/user/comments/Владимир Кисилица/) 8 минут назад

Tickets и miniShop2 (sections/help/1031/#comment-17759)  4

 Александр Москвин (/user/comments/Александр Мос/) 34 минуты назад

Страницы сайта очень долго открываются (15-30 сек) (sections/help/2129/#comment-17758)  19


 Илья Ершов (/user/comments/WEB-master/) 2 часа назад

* Собираем и устанавливаем первую версию пакета (training/course1/2150/#comment-17756)  19


 Сергей Шлоков (/user/comments/sergant/) 3 часа назад

pdoTools + Bootstrap3 + MinifyX = Theme.Bootstrap (sections/components/2180/#comment-17744)  25

 Пётр Молчанов (/user/comments/Ofeda8da047df78a35bbec27556e536a/) 3 часа назад


[HybridAuth] Сильно улучшенная версия 0.7.0-pl1 (sections/components/1717/#comment-17743)  60

 Василий Наумкин (/user/comments/bezumkin/) 4 часа назад

[MinifyX] Новая бета версия со встроенным Muneer (sections/components/2123/#comment-17742)  33

 Василий Наумкин (/user/comments/bezumkin/) 5 часов назад

Быстрый старт (modx/tickets/quickstart.html#comment-17734)  18

 Серго Арбузов (/user/comments/164787961/) 5 часов назад

Переписал MinifyX (sections/components/178/#comment-17732)  50

 Василий Наумкин (/user/comments/bezumkin/) 5 часов назад

Настройка DKIM и SPF на Ubuntu 12.04 + Sendmail (sections/hosting/1698/#comment-17729)  6

A- A A+

Твиттер @bezumkin (<https://twitter.com/bezumkin>)



Антон @korobkov 21 ноября
Все помнят ролик Volvo с Ван Дамом?
youtube.com/watch?v=M7Flvf...

Если бы его снимали в России, он выглядел так pic.twitter.com/5lxdeFsKZu

Ретвитнул(а) Василий Наумкин

Показать фотографию



Василий Наумкин @bezumkin 23ч
Большой урок по ExtJS "Пишем интерфейс: таблица подписок и окошко создания" - bezumkin.ru/training/cours...

Развернуть



Adam Michela @soopa 21 ноября
Steps in mastery of a programming skill:

- 1 Can I do this?
- 2 Can I do this more easily?
- 3 Can I do this more elegantly?
- 4 Can I not do this?

Ретвитнул(а) Василий Наумкин

Развернуть



SOLIDNY SNAEK @gulyaevr 21 ноября
В Советской России разработчики игр пишут разгромные рецензии на игровых журналистов.

Ретвитнул(а) Василий Наумкин

Развернуть



ВОЛЬГА @volgafilm 21 ноября

Обновления по RSS (<http://bezumkin.ru/rss>)

total time: 0.2561 s

query time: 0.0198 s

queries: 33

memory: 4.75 Mb

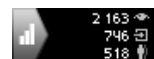
©2012—2013 Василий Наумкин (<http://bezumkin.ru/contacts.html>)



(<http://bezumkin.ru/modx/tickets/>)



(<http://www.modx.com>)



(<http://metrika.yandex.ru/stat/?id=12150862&from=informer>)