
SkeekS CMS Documentation

Выпуск 3.2.2

Semenov Aleksander

мая 21, 2018

1	Ссылки	3
2	Информация	5
2.1	Обзор	5
2.1.1	Требования	5
2.1.1.1	Программное обеспечение	5
2.1.1.2	Модули php	6
2.1.1.3	Настройки php	6
2.1.1.4	Прочее	6
2.1.2	Установка	6
2.1.2.1	1. Installation composer	6
2.1.2.2	2. Installation files	7
2.1.2.3	3. Configuring the database	7
2.1.2.4	4. Installation of migrations	7
2.1.2.5	5. Configuring the server	7
2.1.2.6	6. Authorization system	7
2.1.2.7	7. Check the working environment	8
2.1.3	Обновление	8
2.1.3.1	Стандартное обновление	8
2.1.4	Настройка сервера	8
2.1.4.1	Recommended Apache Configuration	8
2.1.4.2	Recommended Nginx Configuration	9
2.1.5	Reporting a security vulnerability	10
2.1.6	Work with documents	10
2.2	Структура	10
2.2.1	Директории	10
2.2.2	Глобальные константы	11
2.2.3	Предопределенные псевдонимы путей	11
2.2.4	Приложения	12
2.2.5	Конфигурирование	12
2.2.5.1	Простейшая конфигурация приложения	12
2.2.5.2	Автоматическая конфигурация приложения	12
2.2.5.3	Автоматическая конфигурация приложения + автообновление конфигураций	14
2.2.5.4	Варианты определения константы ENV	14
2.3	Быстрый старт	15

2.3.1	Работа с шаблонами	15
2.3.1.1	Настройка	15
2.3.1.2	Использование	15
2.3.1.3	Шаблоны для писем	16
2.3.2	Работа с URL	16
2.3.2.1	Ссылки на разделы	16
2.3.2.2	Ссылки в консольном приложении	17
2.3.3	Авторизация / Регистрация	17
2.3.3.1	Стандартная авторизация/регистрация	17
2.3.4	Компоненты CMS	19
2.3.5	Виджеты CMS	20
2.3.5.1	Редактируемые блоки	20
2.3.5.2	Меню	20
2.3.5.3	Хлебные крошки	22
2.3.5.4	Элементы контента	23
2.4	Javascript	24
2.4.1	Общее	24
2.4.2	Установка	24
2.4.3	Базовые классы	24
2.4.4	Пространство видимости	24
2.4.5	Наследование	25
2.4.6	Использование классов	25
2.4.7	Переопределение родительского конструктора	25
2.4.8	Базовый компонент	26
2.4.9	Подключение библиотеки к проекту на yii2	26
2.5	CMS Admin	27
2.5.1	Настройка и конфигурирование	27
2.5.1.1	Пример конфигурирования	27
2.5.1.2	Доступные настройки	27
2.5.2	Меню	27
2.5.2.1	Формат	27
2.5.3	Создание контроллера	28
2.5.3.1	В проекте	28
2.5.3.2	Создание контроллера	30
2.6	Расширения	30
2.6.1	Shop (skeeks/cms-shop)	30
2.6.1.1	Виджеты	30
2.6.1.2	Примеры	31
2.7	FAQ	38
2.7.1	Разные вопросы	38
2.7.1.1	Как правильно сделать <code>resize</code> изображений?	38
2.7.1.2	Как можно загрузить файл в хранилище?	38
2.7.1.3	Как правильно поймать событие отправки формы в конструкторе	39
2.7.1.4	Как отметить обязательные поля в формах *	39
2.7.1.5	Как изменить <code>timeout ajax</code> ?	40
2.7.1.6	Как вызвать событие в момент окончания работы <code>ajax</code> -скрипта с определенным <code>id</code> ?	40
2.7.1.7	Оптимальный <code>robots.txt</code>	40
2.7.1.8	Как включить <code>js</code> , <code>css</code> и <code>html</code> оптимизацию?	40
2.7.2	Перенос проекта на другой хостинг	41
2.7.2.1	Архивация	41
2.7.2.2	Восстановление	41

SkeekS CMS - современная система общего управления контентом на основе yii2 php framework.

- Сайт
- Автор
- Лог изменений

Предоставляет возможности для управления структурой сайта и элементами контента (новостями, публикациями, продуктами и т.д.).

Возможности управления и динамического создания дополнительных свойств разделов и элементов контента через систему администрирования.

Возможности управления пользователями, привилегиями, ролями и их назначениями.

Поддерживает точку входа для одного запроса (один `index.php`), для бэкэнд и фронтенд частей проекта. Удобно настраивать `nginx` и `apache`.

Almost every page of the site - content item or section. Each of these models has some povdeniem availability of additional properties. Therefore, any product publkatcii, news, etc. It has a set of common properties, which are described in the model, and a set of dynamically created properties, through the administration system.

This versatility allows you to easily create any site, without writing and design of additional models and migration. What idelalno for quickly writing conventional nevysokonagruzhennyh projects (this does not mean that you can not write highly loaded projects).

Just have <https://cms.skeeks.com/marketplace> marketpleys the CMS for this, which is constantly updated with useful solutions.

2.1 Обзор

2.1.1 Требования

2.1.1.1 Программное обеспечение

- `apache (mod_rewrite)` or `php-fpm`
- `php >= 7.0`
- `DB mysql ~ 5.5`

2.1.1.2 Модули php

- mbstring
- xml
- pdo
- pdo_mysql
- json
- phar
- simplexml
- timezonedb
- gd или imagik
- intl
- mcrypt
- fileinfo
- curl

2.1.1.3 Настройки php

- short_open_tag on

Внимание: Важно не забудьте проверить эту настройку!

2.1.1.4 Прочее

- SSH доступ

2.1.2 Установка

2.1.2.1 1. Installation composer

The recommended way to install SkeekS CMS is with [Composer](#). Composer is a dependency management tool for PHP that allows you to declare the dependencies your project needs and installs them into your project.

If you do not have Composer, follow the instructions in the [Installing Yii](#) section of the definitive guide to install it.

```
# Install Composer  
curl -sS https://getcomposer.org/installer | COMPOSER_HOME=.composer php
```

Примечание: Alternative commands, depending on the server configuration and your access rights:

```
#composer if not installed globally, you can use this command
COMPOSER_HOME=.composer php composer.phar
# or use if composer installed globally
composer
```

```
php yii
# or use (file yii must be executable)
yii
```

2.1.2.2 2. Installation files

Establish **example.com** site in example.com folder

Navigate to the folder where are your projects (such as `/var/www/sites/`).

```
# Download latest version of composer
curl -sS https://getcomposer.org/installer | COMPOSER_HOME=.composer php

# Installing the base project SkeekS CMS
COMPOSER_HOME=.composer php composer.phar create-project --prefer-dist --stability=dev skeeks/app-
↳basic demo.ru
# Going into the project folder
cd demo.ru
```

2.1.2.3 3. Configuring the database

Edit the file to access the database, it is located at `common/config/db.php`

2.1.2.4 4. Installation of migrations

```
#Installation of ready-dump
php yii dbDumper/mysql/restore
```

2.1.2.5 5. Configuring the server

By default, your site opens at `//example.com/frontend/web/`

On hostings are configured by default under the usual sites.

But it can be reconfigured (and even necessary) in detail about this here: [Server Configuration \(web-server\)](#)

2.1.2.6 6. Authorization system

Система управления по умолчанию доступна по следующему адресу (при желании его можно пере-конфигурировать)

```
//example.com/~sx/admin/auth/
```

root (логин)

skeeks (пароль)

2.1.2.7 7. Check the working environment

If the installation process has been completed, but there are still not clear to you the error, it is likely that something is wrong is configured on the server. To do so, download to `/frontend/web/` and run the file to test <https://github.com/skeeks-cms/cms/blob/master/requirements.php> environment. `//example.com/frontend/web/requirements.php` or `//example.com/requirements.php` (depends on item 4).

Внимание: It is important to remember to check the setting of php: `short_open_tag on`

2.1.3 Обновление

2.1.3.1 Стандартное обновление

```
# Composer update to the latest stable version
COMPOSER_HOME=.composer php composer.phar self-update && COMPOSER_HOME=.composer php composer.phar ↵
↵update -o
```

2.1.4 Настройка сервера

Примечание: Info: You may skip this subsection for now if you are just test driving Yii with no intention of deploying it to a production server.

The application installed according to the above instructions should work out of box with either an Apache HTTP server or an Nginx HTTP server, on Windows, Mac OS X, or Linux running PHP 5.5 or higher. Yii 2.0 is also compatible with facebook's HHVM. However, there are some edge cases where HHVM behaves different than native PHP, so you have to take some extra care when using HHVM.

On a production server, you may want to configure your Web server so that the application can be accessed via the URL `//www.example.com/index.php` instead of `//www.example.com/frontend/web/index.php`. Such configuration requires pointing the document root of your Web server to the `basic/web` folder. You may also want to hide `index.php` from the URL, as described in the Routing and URL Creation section. In this subsection, you'll learn how to configure your Apache or Nginx server to achieve these goals.

2.1.4.1 Recommended Apache Configuration

Use the following configuration in Apache's `httpd.conf` file or within a virtual host configuration. Note that you should replace `path/to/basic/web` with the actual path for `basic/web`.

```
# Set document root to be "frontend/web"
DocumentRoot "path/to/frontend/web"

<Directory "path/to/frontend/web">
  # use mod_rewrite for pretty URL support
  RewriteEngine on
  # If a directory or a file exists, use the request directly
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
```

(continues on next page)

(продолжение с предыдущей страницы)

```

# Otherwise forward the request to index.php
RewriteRule . index.php

# ...other settings...
</Directory>

```

2.1.4.2 Recommended Nginx Configuration

To use Nginx, you should install PHP as an FPM SAPI. You may use the following Nginx configuration, replacing path/to/frontend/web with the actual path for frontend/web and mysite.local with the actual hostname to serve.

```

server {
    charset utf-8;
    client_max_body_size 128M;

    listen 80; ## listen for ipv4
    #listen [::]:80 default_server ipv6only=on; ## listen for ipv6

    server_name mysite.local;
    root        /path/to/frontend/web;
    index       index.php;

    access_log  /path/to/frontend/log/access.log;
    error_log   /path/to/frontend/log/error.log;

    location / {
        # Redirect everything that isn't a real file to index.php
        try_files $uri $uri/ /index.php$is_args$args;
    }

    # uncomment to avoid processing of calls to non-existing static files by Yii
    #location ~ \.(js|css|png|jpg|gif|swf|ico|pdf|mov|fla|zip|rar)$ {
    #    try_files $uri =404;
    #}
    #error_page 404 /404.html;

    # deny accessing php files for the /assets directory
    location ~ ~/assets/.*\.php$ {
        deny all;
    }

    location ~ /\.php$ {
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_pass 127.0.0.1:9000;
        #fastcgi_pass unix:/var/run/php5-fpm.sock;
        try_files $uri =404;
    }

    location ~* /\. {
        deny all;
    }
}

```

When using this configuration, you should also set `cgi.fix_pathinfo=0` in the `php.ini` file in order to avoid

many unnecessary system `stat()` calls.

Also note that when running an HTTPS server, you need to add `fastcgi_param HTTPS on`; so that Yii can properly detect if a connection is secure.

2.1.5 Reporting a security vulnerability

Publicly disclosing a vulnerability can put the entire community at risk. If you've discovered a security concern, please email us at support@skeeks.com.

After a security vulnerability has been corrected, a security hotfix release will be deployed as soon as possible.

2.1.6 Work with documents

Для написания документации используется генератор документации [Sphinx](#). Стандартный синтаксис разметки `reStructuredText`

Полезные команды для генерации:

```
apt-get install python-pip
pip install Sphinx
pip install sphinx-intl
pip install sphinx_rtd_theme

make gettext
make html
sphinx-intl update -p _build/gettext -l ru
#make -e SPHINXOPTS="-D language='ru'" html

sphinx-build -D language='ru' ./ build/ru
sphinx-build ./ build/en
```

2.2 Структура

2.2.1 Директории

Стандартная структура проекта выглядит следующим образом (корневая директория проекта):

<code>common</code>	содержит общие файлы всех приложений
<code>config/</code>	содержит общие конфигурационные файлы приложений
<code>mail/</code>	содержит файлы представлений для электронной почты
<code>models/</code>	содержит классы моделей, используемые во всех приложениях
<code>runtime/</code>	временно генерируемые файлы используемые всеми приложениями
<code>widgets/</code>	классы виджетов приложений
<code>console</code>	консольное приложение, скрипты для крона и прочее
<code>config/</code>	содержит конфигурационные файлы
<code>controllers/</code>	содержит консольные контроллеры (<code>commands</code>)
<code>migrations/</code>	содержит миграции
<code>models/</code>	содержит классы моделей
<code>runtime/</code>	временно генерируемые файлы
<code>frontend</code>	приложение 1
<code>assets/</code>	описание и хранение Asset блоков Yii2
<code>config/</code>	содержит конфигурационные файлы

(continues on next page)

(продолжение с предыдущей страницы)

<code>controllers/</code>	contains Web controller classes
<code>models/</code>	содержит классы моделей
<code>runtime/</code>	временно генерируемые файлы
<code>templates/</code>	содержит набор шаблонов
<code>default/</code>	Файлы представлений шаблона по умолчанию
<code>web/</code>	публичная директория (файлы js, css, img...)
<code>assets/</code>	временные js, css, файлы
<code>widgets/</code>	классы виджетов приложения
<code>frontend2</code>	приложение 2
<code>//</code>	... полностью повторяет структуру предыдущего приложения...
<code>frontend3</code>	приложение 3
<code>//</code>	... полностью повторяет структуру предыдущего приложения...
<code>vendor/</code>	используемые дополнительные библиотеки в проекте
<code>tests</code>	contains various tests for the advanced application
<code>codeception/</code>	contains tests developed with Codeception PHP Testing Framework

Стандартный проект содержит следующие папки в корневой директории проекта:

- **common** - файлы общие для всех приложений.
- **console** - консольное приложение.
- **frontend** - frontend приложение.

2.2.2 Глобальные константы

- `ROOT_DIR` — путь до корневой директории проекта
- `ENV` — названия окружения (от окружения будет зависеть, то какие настройки будут подключены)

2.2.3 Предопределенные псевдонимы путей

Подробнее про псевдонимы <https://www.yiiframework.com/doc/guide/2.0/en/concept-aliases>

- `@yii` - фремворк директория.
- `@app` - базовый путь текущего запущенного приложения.
- `@common` - файлы общие для всех приложений.
- `@frontend` - frontend приложение.
- `@console` - console приложение.
- `@runtime` - runtime directory of currently running web application.
- `@vendor` - Composer vendor directory.
- `@bower` - vendor directory that contains the [bower packages](#).
- `@npm` - vendor directory that contains [npm packages](#).
- `@web` - base URL of currently running web application.
- `@webroot` - web root directory of currently running web application.
- `@root` - корневая директория проекта

Алиасы специфичные для ваших проектов, можно прописать в общем конфиге проекта, следующим образом:

```
'aliases' => [  
    'frontend2' => '@root/frontend2',  
    'frontend3' => '@root/frontend3',  
],
```

2.2.4 Приложения

По умолчанию в проекте есть два приложения: frontend и console. Frontend обычно представляет то, что представлено конечному пользователю, собственно сам сайт.

Консоль обычно используется для заданий stop и управления серверами низкого уровня. Также он используется во время развертывания приложений и обрабатывает миграции и т.д.

Существует также общий каталог, содержащий файлы, используемые более чем одним приложением. Например, модель пользователя.

Каждое приложение имеет собственное пространство имен и псевдоним, соответствующий его имени. То же самое относится к общему каталогу.

2.2.5 Конфигурирование

2.2.5.1 Простейшая конфигурация приложения

В файле `/frontend/web/index.php` определяется путь слияния конфигурационных файлов проекта.

В простейшем виде можно сконфигурировать приложение стандартным способом, вот так может выглядеть файл `/frontend/web/index.php`:

```
define("ENV", 'prod');  
define("ROOT_DIR", dirname(dirname(__DIR__)));  
  
require_once(ROOT_DIR . '/vendor/skeeks/cms/bootstrap.php');  
  
$config = \yii\helpers\ArrayHelper::merge([],  
    , require(__DIR__ . '/../../common/config/main.php')  
    , require(__DIR__ . '/../../frontend/config/main.php')  
);  
  
$application = new \yii\web\Application($config);  
$application->run();
```

В этом случае, как и в любом yii2 проекте, необходимо полностью сконфигурировать приложение самостоятельно.

2.2.5.2 Автоматическая конфигурация приложения

Слиянием файлов конфигураций занимается специальный composer-plugin `cms-composer`. Подробнее можно прочитать тут: <https://habr.com/post/329286/>

Идея в том, что любое расширение yii2 (модуль, компонент, пакет), может предоставить собственные настройки, которые автоматически подключатся к проекту.

Слиянием файлов конфигураций занимается `composer` по команде или после обновления зависимостей.

Пути слияния прописываются в *composer.json* проекта, по умолчанию следующим образом:

```
{
  "extra": {
    "config-plugin": {
      //Каждый из установленных расширений в проекте, уже предоставил конфиги для
      ↪соответствующих секций
      "web": [
        "common/config/main.php",
        "common/config/db.php",
        "frontend/config/main.php"
      ],
      "web-dev": [
        "$web",
        "?frontend/config/env/dev/main.php"
      ],
      "web-prod": [
        "$web",
        "?frontend/config/env/prod/main.php"
      ],
      "console": [
        "common/config/main.php",
        "common/config/db.php",
        "console/config/main.php"
      ],
      "console-dev": [
        "$console",
        "?console/config/env/dev/main.php"
      ],
      "console-prod": [
        "$console",
        "?console/config/env/prod/main.php"
      ]
    ]
  }
}
```

А файл */frontend/web/index.php*:

```
define("ENV", 'prod');
define("ROOT_DIR", dirname(dirname(__DIR__)));

require(ROOT_DIR . '/vendor/skeeks/cms/app-web.php');
```

В приведенной конфигурации проекта, если определить константу *ENV* как *prod*

То в *web* приложении результирующая конфигурация будет состоять из:

```
"web-prod": [
  "$web", //сюда попадут все конфиги расширений + "common/config/main.php" + "common/config/db.
  ↪php" + "frontend/config/main.php"
  "?frontend/config/env/prod/main.php"
],
```

Для того чтобы перекомпилировать конфигурацию приложения, необходимо выполнить команду:

```
composer du
```

Для того чтобы посмотреть пути наследования конфигураций:

```
composer du --verbose
```

Внимание: Не забывайте обновлять файл конфигураций во время разработки!

2.2.5.3 Автоматическая конфигурация приложения + автообновление конфигураций

```
define("ENV", 'dev');
define("ROOT_DIR", dirname(dirname(__DIR__)));

//Стандартная загрузка yii2 + всего необходимого для skeeks cms
require(ROOT_DIR . '/vendor/skeeks/cms/bootstrap.php');

//Если включен dev режим работы с сайтом, то слияния настроек будет происходить при выполнении
↳каждого сценария
if (ENV == 'dev') {
    \Yii::beginProfile('Rebuild config');
    error_reporting(E_ALL);
    ini_set('display_errors', 'On');
    \skeeks\cms\composer\config\Builder::rebuild();
    \Yii::endProfile('Rebuild config');
}

//Подключение стандартного слитого файла конфигураций для текущего окружения
$configFile = \skeeks\cms\composer\config\Builder::path('web-' . ENV);
if (!file_exists($configFile)) {
    $configFile = \skeeks\cms\composer\config\Builder::path('web');
}
$config = (array)require $configFile;

$application = new yii\web\Application($config);
$application->run();
```

2.2.5.4 Варианты определения константы ENV

.htaccess

Определение через .htaccess /frontend/web/index.php:

```
SetEnv ENV dev
```

/frontend/web/index.php:

```
$env = getenv('ENV');
if (!empty($env)) {
    defined('ENV') or define('ENV', $env);
}

define("ROOT_DIR", dirname(dirname(__DIR__)));
require(ROOT_DIR . '/vendor/skeeks/cms/app-web.php');
```

ip адрес

Определение окружения для определенного ip адреса /frontend/web/index.php:

```

$ip = isset($_SERVER['REMOTE_ADDR']) ? $_SERVER['REMOTE_ADDR'] : "";
if (in_array($ip, ['31.148.139...'])) {
    defined('ENV') or define('ENV', 'dev');
}

define("ROOT_DIR", dirname(dirname(__DIR__)));
require(ROOT_DIR . '/vendor/skeeks/cms/app-web.php');

```

Таким образом любой разработчик имеет возможность иметь собственную конфигурацию, а проект единую кодовую базу. Так же любое установленное расширение, которое предоставляет конфигурацию по текущим правилам, сразу приносит настройку в проект.

2.3 Быстрый старт

2.3.1 Работа с шаблонами

2.3.1.1 Настройка

Стандартным для yii2 способом, для определения пути к теме/шаблону сайта, можно путем конфигурирования компонента view. В файле конфига проекта frontend/config/main.php отредактировать секцию components

```

'view' =>
[
    'theme' =>
    [
        'pathMap' =>
        [
            '@app/views' =>
            [
                '@app/templates/default',
            ],
        ],
    ],
],

```

2.3.1.2 Использование

Пути к шаблонам обычно собираются отталкиваясь от алиаса @app/views - который выше сконфигурирован.

Пример подключения шаблона в шаблоне

```

<?=$this->render("@app/views/header", []); ?>

```

Пример глобального рендеринга шаблона

```
<?= \Yii::$app->views->render("@app/views/header", []); ?>
```

2.3.1.3 Шаблоны для писем

Шаблоны для отправки писем из расширений лежат непосредственно в расширении в папке mail-templates. При отправке письма идет проверка.

```
\Yii::$app->mailer->view->theme->pathMap = ArrayHelper::merge(\Yii::$app->mailer->view->theme->
↳ pathMap, [
    '@app/mail' =>
        [
            '@skeeks/cms/mail-templates'
        ]
    ]
]);
```

@app - это папка текущего приложения. Чтобы подложить для отправки свой шаблон, создайте папку mail в папке своего приложения. Положите туда свои шаблоны.

2.3.2 Работа с URL

Везде в своих проектах правильно формируйте url, на любое действие на любой раздел, на любой элемент и т.д. Это позволит избежать кучи проблем с ростом проекта. Особенно с добавлением мультиязычности на сайт.

И так, в yii2 на эту тему есть множество примеров, с ними можно ознакомиться, например тут: <https://github.com/yiisoft/yii2/blob/master/docs/guide/helper-url.md>

Здесь же, мы рассмотрим конкретные примеры всего что связано с базовым модулем cms

2.3.2.1 Ссылки на разделы

Ссылки на разделы сайта, по их id параметру

```
\yii\helpers\Url::to(['/cms/tree/view', 'id' => 10])
```

Ссылки на разделы сайта, по их объекту модели model

```
$model = \skeeks\cms\models\CmsTree::findOne(10);
\yii\helpers\Url::to(['/cms/tree/view', 'model' => $model])
```

Ссылки на разделы сайта, по их dir параметру

```
//Ссылка в раздел about
\yii\helpers\Url::to(['/cms/tree/view', 'dir' => 'about'])
```

Прочие примеры с параметрами

```
//Ссылка в раздел about с параметрами
\yii\helpers\Url::to(['/cms/tree/view', 'dir' => 'about', 'param1' => 'test1', '#' => 'test1'])

//Абсолютная ссылка на раздел about
```

(continues on next page)

(продолжение с предыдущей страницы)

```
\yii\helpers\Url::to(['/cms/tree/view', 'dir' => 'about'], true)

//Абсолютная https ссылка на раздел about
\yii\helpers\Url::to(['/cms/tree/view', 'dir' => 'about'], 'https')

//Ссылка на вложенный раздел
\yii\helpers\Url::to(['/cms/tree/view', 'dir' => 'about/level-2/level-3'])
```

Но cms поддерживает концепцию многосайтовости. Поэтому можно в параметрах указать код желаемого сайта:

```
\yii\helpers\Url::to(['/cms/tree/view', 'dir' => 'about/level-2/level-3', 'site_code' => 's2'])
```

2.3.2.2 Ссылки в консольном приложении

Об этом стоит сказать особенно. Частый случай, что в yii2 сыпятся ошибки при запуске каких либо консольных утилит. Для корректной работы ссылок, необходимо сконфигурировать компонент UrlManager в консольном приложении.

```
'urlManager' => [
    'baseUrl' => '',
    'hostInfo' => 'http://your-site.com'
]
```

А так же в bootstrap определить пару алиасов:

```
\Yii::setAlias('webroot', dirname(dirname(__DIR__)) . '/frontend/web');
\Yii::setAlias('web', '');
```

2.3.3 Авторизация / Регистрация

2.3.3.1 Стандартная авторизация/регистрация

В SkeekS CMS уже реализован процесс авторизации, регистрации и восстановления пароля (через email). Реализация находится в cms/auth контроллере.

Методы реализующие эти процессы:

- `login` — процесс авторизации
- `register` — процесс регистрации
- `register-by-email` — регистрация через email (только ajax)
- `forget` — запроса начала процедуры восстановления пароля
- `reset-password` — действие подтверждения смены пароля

Проверка текущего пользователя

Для проверки авторизации текущего пользователя на сайте, используется стандартная конструкция yii2.

```

if (\Yii::$app->user->isGuest)
{
    //Пользователь неавторизован
} else
{
    //Пользователь авторизован можно запросить его данные
    print_r(\Yii::$app->user->identity->toArray());
}

```

Ссылки на авторизацию

Как получить ссылку на действия связанные с авторизацией

```

echo \yii\helpers\Url::to(['cms/auth/login']);
echo \yii\helpers\Url::to(['cms/auth/register']);
echo \yii\helpers\Url::to(['cms/auth/forget']);

```

Еще один вариант через хелпер SkeekS CMS

```

echo \skeeks\cms\helpers\UrlHelper::construct('cms/auth/login')->setCurrentRef()

```

Форма авторизации

Эту форму можно вставить в любое место на сайте, работает через ajax.

```

$model = new \skeeks\cms\models\forms\LoginFormUsernameOrEmail();

<?php $form = skeeks\cms\base\widgets\ActiveFormAjaxSubmit::begin([
    'action' => \skeeks\cms\helpers\UrlHelper::construct('cms/auth/login')->setCurrentRef()->
    toString(),
    'validationUrl' => \skeeks\cms\helpers\UrlHelper::construct('cms/auth/login')->
    setSystemParam(\skeeks\cms\helpers\RequestResponse::VALIDATION_AJAX_FORM_SYSTEM_NAME)->toString()
]); ?>
    <?=$form->field($model, 'identifier') ?>
    <?=$form->field($model, 'password')->passwordInput() ?>
    <?=$form->field($model, 'rememberMe')->checkbox() ?>

    <div class="form-group">
        <?=\yii\helpers\Html::submitButton("<i class='glyphicon glyphicon-off'></i> Войти", [
            'class' => 'btn btn-primary', 'name' => 'login-button']) ?>
    </div>

<?php skeeks\cms\base\widgets\ActiveFormAjaxSubmit::end(); ?>

```

Форма регистрации

```

<?php $form = ActiveForm::begin([
    'action' => UrlHelper::construct('cms/auth/register-by-email')->toString(),
    'validationUrl' => UrlHelper::construct('cms/auth/register-by-email')->
    setSystemParam(\skeeks\cms\helpers\RequestResponse::VALIDATION_AJAX_FORM_SYSTEM_NAME)->
    toString(),
    'afterValidateCallback' => <<<JS

```

(continues on next page)

(продолжение с предыдущей страницы)

```

function(jForm, ajaxQuery)
{
    var handler = new sx.classes.AjaxHandlerStandartRespose(ajaxQuery, {
        'blockerSelector' : '#' + jForm.attr('id'),
        'enableBlocker' : true,
    });

    handler.bind('success', function()
    {
        _.delay(function()
        {
            $('#sx-login').click();
        }, 2000);
    });
}
JS

]); ?>
<?= $form->field($model, 'email') ?>

<div class="form-group">
    <?= Html::submitButton("<i class='\"glyphicon glyphicon-off\"></i> Зарегистрироваться", [
↪'class' => 'btn btn-primary', 'name' => 'login-button']) ?>
</div>

<?php ActiveForm::end(); ?>

```

Форма восстановления пароля

```

<?php $form = ActiveForm::begin([
    'action' => UrlHelper::construct('cms/auth/forget')->toString(),
    'validationUrl' => UrlHelper::construct('cms/auth/forget')->
↪setSystemParam(\skeeks\cms\helpers\RequestResponse::VALIDATION_AJAX_FORM_SYSTEM_NAME)->toString()
]); ?>
    <?= $form->field($model, 'identifier') ?>

    <div class="form-group">
        <?= Html::submitButton("Отправить", ['class' => 'btn btn-primary', 'name' => 'login-button
↪']) ?>
    </div>

<?php ActiveForm::end(); ?>

```

2.3.4 Компоненты CMS

```

$model = \Yii::$app->companySiteSettings;
$model->setCmsSite(\Yii::$app->project->company->site);
$model->refresh();

$model->load(\Yii::$app->request->post());
$model->setOverride(Component::OVERRIDE_SITE)->save();

```

2.3.5 Виджеты CMS

Виджеты cms наследуются от базового виджета: `skeeks\cms\base\Widget`

Преимущество их работы, состоит в том, что их можно редактировать в «**Панеле быстрого управления сайтом**».

2.3.5.1 Редактируемые блоки

`skeeks\cms\cmsWidgets\text\TextCmsWidget` — Предназначен для редактирования блоков текста или html кода

Пример использования

```
<? echo \skeeks\cms\cmsWidgets\text\TextCmsWidget::widget([
    'namespace' => 'unique-settings-code',
    'text' => 'Edited text'
]); ?>
```

```
<? echo \skeeks\cms\cmsWidgets\text\TextCmsWidget::widget([
    'namespace' => 'unique-settings-code',
    'text' => <<<HTML
<p class="cl-gray ">
    Edited text
</p>
HTML
]); ?>
```

```
<? \skeeks\cms\cmsWidgets\text\TextCmsWidget::begin([
    'namespace' => 'unique-settings-code',
]); ?>
<p class="cl-gray ">
    Edited text
</p>
<? \skeeks\cms\cmsWidgets\text\TextCmsWidget::end(); ?>
```

```
<? \skeeks\cms\cmsWidgets\text\TextCmsWidget::beginWidget('unique-settings-code'); ?>
<p class="cl-gray ">
    Edited text
</p>
<? \skeeks\cms\cmsWidgets\text\TextCmsWidget::end(); ?>
```

2.3.5.2 Меню

`skeeks\cms\cmsWidgets\treeMenu\TreeMenuCmsWidget` — Данный виджет, чаще всего предназначен для построения меню на сайте. При чем как главного меню, так и второстепенного. Добиться этого можно путем манипуляции с его параметрами и способом вызова. Так же, виджет может подойти для вывода подразделов определенного раздела сайта (например основные разделы услуг, на главную страницу сайта).

Пример использования

```
<?= \skeeks\cms\cmsWidgets\treeMenu\TreeMenuCmsWidget::widget([
    'namespace' => 'top-site-menu',
    'viewFile' => '@app/views/widgets/TreeMenuCmsWidget/top-site-menu',
    'label' => 'Title menu',
    'level' => '1',
    'enabledRunCache' => \skeeks\cms\components\Cms::BOOL_N,
]); ?>
```

Пример содержимого файла: @app/views/widgets/TreeMenuCmsWidget/top-site-menu

```
<?php
/* @var $this yii\web\View */
/* @var $widget \skeeks\cms\cmsWidgets\treeMenu\TreeMenuCmsWidget */
/* @var $trees \skeeks\cms\models\Tree[] */
?>
<ul class="menu">
    <? if ($trees = $widget->activeQuery->all()) : ?>
        <? foreach ($trees as $tree) : ?>
            <?= $this->render("menu-top-item", [
                "widget" => $widget,
                "model" => $tree,
            ]); ?>
        <? endforeach; ?>
    <? endif; ?>
</ul>
```

Пример содержимого файла: @app/views/widgets/TreeMenuCmsWidget/menu-top-item

```
<?php
/* @var $this yii\web\View */
/* @var $widget \skeeks\cms\cmsWidgets\treeMenu\TreeMenuCmsWidget */
/* @var $model \skeeks\cms\models\Tree */
$dir = $model->dir;
if ($model->redirect_tree_id) {
    $dir = $model->redirectTree->dir;
};
$activeClass = '';
if (strpos(Yii::$app->request->pathInfo, $dir) !== false) {
    $activeClass = ' active';
}
?>
<li>
    <a href="<?= $model->url; ?>" title="<?= $model->name; ?>" class="<?= $activeClass; ?>">
        <?= $model->name; ?>
    </a>
</li>
```

Пример с переопределением настроек

```
<? $widget = \skeeks\cms\cmsWidgets\treeMenu\TreeMenuCmsWidget::beginWidget('menu-top-1', [
    'viewFile' => '@app/views/widgets/TreeMenuCmsWidget/menu-top',
    'label' => 'Верхнее меню',
    'level' => '1',
```

(continues on next page)

(продолжение с предыдущей страницы)

```

'enabledRunCache' => \skeeks\cms\components\Cms::BOOL_N,
]); ?>
<?
//Переопределение шаблона, то есть не важно что укажут в настройках виджета, шаблон все равно
↳ будет использоваться этот!
$widget->viewFile = '@app/views/widgets/TreeMenuCmsWidget/menu-top-2';
//Изменение запроса
$widget->activeQuery->andWhere(['code' => 'dostavka']);
?>
<? \skeeks\cms\cmsWidgets\treeMenu\TreeMenuCmsWidget::end(); ?>

```

Второстепенное меню каталога

Например, при хождении по каталогу, необходимо чтобы подразделы каталога, отображались всегда слева. Для этого можно опереться на параметр текущего раздела сайта, и передать его в один из параметров виджета в качестве `treePid`. В этом случае, в выборке нужных разделов будет всегда участвовать условие родительского раздела. И при этом это условие будет перекрывать настройки указанные администратором через админку, а значит администратор не сможет сломать виджет, но при этом сможет поменять некоторые параметры.

```

<? $widget = \skeeks\cms\cmsWidgets\treeMenu\TreeMenuCmsWidget::beginWidget('menu-top-1', [
'viewFile' => '@app/views/widgets/TreeMenuCmsWidget/menu-top',
'label' => 'Вложенное меню',
'enabledRunCache' => \skeeks\cms\components\Cms::BOOL_N,
]); ?>
<?
//Если задан текущий раздел, и у него есть достаточная вложенность
if (\Yii::$app->cms->currentTree && \Yii::$app->cms->currentTree->parent && isset(\Yii::$app->
↳ cms->currentTree->parents[1]))
{
    $currentParentTree = \Yii::$app->cms->currentTree->parents[1];
    $widget->treePid = $currentParentTree->id; //Переопределение параметра родительского
↳ раздела

    $widget->initActiveQuery(); //Применение новых настроек виджета
}
?>
<? \skeeks\cms\cmsWidgets\treeMenu\TreeMenuCmsWidget::end(); ?>

```

2.3.5.3 Хлебные крошки

`\skeeks\cms\cmsWidgets\breadcrumbs\BreadcrumbsCmsWidget` — виджет для построения хлебных крошек на сайте.

Пример вызова

```

<?= \skeeks\cms\cmsWidgets\breadcrumbs\BreadcrumbsCmsWidget::widget([
'viewFile' => '@app/views/widgets/BreadcrumbsCmsWidget/default',
]); ?>

```

Содержимое шаблона

```
<?php
/* @var $this yii\web\View */
/* @var $widget \skeeks\cms\cmsWidgets\breadcrumbs\BreadcrumbsCmsWidget */
?>
<? if (\Yii::$app->breadcrumbs->parts) : ?>
    <? $count = count(\Yii::$app->breadcrumbs->parts); ?>
    <? $counter = 0; ?>
    <? if ($count > 1) : ?>
        <ul class="breadcrumb">
            <? foreach (\Yii::$app->breadcrumbs->parts as $data) : ?>
                <? $counter ++; ?>
                <? if ($counter == $count): ?>
                    <li class="active"><? = $data['name']; ?></li>
                <? else : ?>
                    <li><a href="<? = $data['url']; ?>" title="<? = $data['name']; ?>"><? = $data[
↪ 'name']; ?></a></li>
                <? endif;?>
            <? endforeach; ?>
        </ul>
    <? endif;?>
<? endif;?>
```

2.3.5.4 Элементы контента

\skeeks\cms\cmsWidgets\contentElements\ContentElementsCmsWidget — виджет для выборки и вывода элементов контента на сайт.

Пример вызова

```
<? = \skeeks\cms\cmsWidgets\contentElements\ContentElementsCmsWidget::widget([
    'namespace' => 'home-news',
    'label'      => 'Новости компании',
    'viewFile'   => '@app/views/widgets/ContentElementsCmsWidget/home-news',
]); ?>
```

Содержимое шаблона

```
<?php
/* @var $this yii\web\View */
/* @var $widget \skeeks\cms\cmsWidgets\contentElements\ContentElementsCmsWidget */
/* @var $element \skeeks\cms\models\CmsContentElement */
?>
<? if ($widget->dataProvider->query->count() > 1) : ?>
    <h5><? = $widget->label; ?></h5>
    <div class="news__posts">
        <? foreach ($widget->dataProvider->query->all() as $element) : ?>
            <div class="news__post">
                <span class="date"><? = \Yii::$app->formatter->asDate($element->published_at); ?></
↪ span>
                <span><a href="<? = $element->url ?>" title="<? = $element->name; ?>"><? = $element-
↪ name; ?></a></span>
```

(continues on next page)

```
        </div>
        <? endforeach; ?>
    </div>
    <? if($element->cmsTree) : ?>
        <div class="more">
            <a class="more" href="<?= $element->cmsTree->url; ?>">остальные новости</a>
        </div>
    <? endif; ?>
<? endif; ?>
```

2.4 Javascript

2.4.1 Общее

Мы развиваем и поддерживаем свой js фреймворк.

Он позволяет легко писать собственные js классы и наследовать их друг от друга.

Используется глобальный namespace `sx`

2.4.2 Установка

В SkeekS CMS уже используется данный фреймворк.

Для установки в любой yii2 проект, можно использовать `skeeks/yii2-sx` пакет.

2.4.3 Базовые классы

Вот так можно получить ссылку на `resize` изображения.

```
sx.classes.Entity
sx.classes.Component
sx.classes.AjaxQuery
sx.classes.Cookie
```

2.4.4 Пространство видимости

Используется глобальный namespace `sx`

Если же нужно создать класс внутри области видимости нужно использовать следующую конструкцию.

```
sx.createNamespace('test.classes', sx);
```

Это позволит создавать классы внутри пространства `sx.test.classes`

Например:

```
sx.test.classes.Entity = sx.classes.Entity.extend({
});
```

2.4.5 Наследование

Пример конструкции наследования:

```

sx.classes.Product = sx.classes.Entity.extend({
  //Тут код расширяющий возможности класса Entity

  getCustomName: function()
  {
    return this.get('name') + " (" + this.get('article') + ")";
  }
});

```

2.4.6 Использование классов

```

var Product = new sx.classes.Product({
  "name" : "Подушка",
  "article" : "A15226",
});

Product.get('name'); //Подушка
Product.get('article'); //A15226
Product.getCustomName(); //Подушка (A15226)

Product.set('name', 'Новое название');
Product.get('name'); //Новое название

//объект продукт можно положить так же в пространство sx
sx.Product = new sx.classes.Product({
  "name" : "Подушка",
  "article" : "A15226",
});

```

2.4.7 Переопределение родительского конструктора

```

sx.classes.ProductCustom = sx.classes.Product.extend({

  construct: function (name, opts)
  {
    opts = opts || {};
    this.set('name', name);
    this.applyParentMethod(sx.classes.Product, 'construct', [opts]);
  }
});

//Тогда продукт нужно создавать так

new sx.classes.ProductCustom("Подушка", {
  "article" : "A15226",
});

```

2.4.8 Базовый компонент

```

sx.classes.Demo = sx.classes.Component.extend({
    _init: function()
    {
        //Тут код который исполняется сразу же в момент создания класса
    },
    _onDomReady: function()
    {
        //Тут написать код которые выполнятся когда сработает событие domReady
    },
    _onWindowReady: function()
    {
        //Тут написать код которые выполнятся когда сработает событие windowReady
    }
});

```

2.4.9 Подключение библиотеки к проекту на yii2

В шаблоне:

```

<?
//Минимум
skeeks\sx\assets\Core::register($this);
//Или более полную библиотеку
skeeks\sx\assets\Custom::register($this);
?>

```

Или через asset bundle:

```

namespace frontend\assets;

class AppAsset extends AssetBundle
{
    public $basePath = '@webroot';
    public $baseUrl = '@web';
    public $css = [
        'css/app.css',
    ];
    public $js = [
        'js/app.js',
    ];
    public $depends = [
        'skeeks\sx\assets\Custom',
    ];
}

```

2.5 CMS Admin

Административная часть сайта реализована компонентом `skeeks/cms-admin`. По умолчанию CMS уже содержит этот компонент.

2.5.1 Настройка и конфигурирование

Компонент админки подключается и настраивается стандартным способом в проект.

2.5.1.1 Пример конфигурирования

В файле конфига проекта `frontend/config/main.php` отредактировать секцию `components`

```
'backendAdmin' =>
[
    'class'          => '\skeeks\cms\admin\AdminComponent',
    'controllerPrefix' => 'admin',
    'urlRule'        => [
        'urlPrefix' => '~sx' //Префикс админки, то есть путь к админке сайта может быть любой
    ],
    'allowedIPs' => [
        '91.219.167.252',
        '93.186.50.*'
    ],
    'on beforeRun' => function($event) {
        \Yii::$app->httpBasicAuth->verify();
    }
],
```

2.5.1.2 Доступные настройки

`allowedIPs`

Разрешенный массив ip адресов, по умолчанию ['*']

2.5.2 Меню

Административное меню формируется путем слияния конфигов всех установленных расширений и конфига проекта.

- `@skeeks/cms/config/admin/menu.php`
- `@skeeks/cms-admin/config/admin/menu.php`
- `@all-other-extensions/config/admin/menu.php`
- `@common/config/admin/menu.php`

2.5.2.1 Формат

В конечном виде конфиг меню представляет собой один большой массив с элементами

```
[
  'users' =>
  [
    'label'      => \Yii::t('skeeks/cms', 'Users'),
    'priority'   => 200,
    'enabled'    => true,
    'img'        => ['\skeeks\cms\modules\admin\assets\AdminAsset', 'images/icons/user.png'],

    'items' =>
    [
      [
        'label'    => \Yii::t('skeeks/cms', "User management"),
        'url'      => ["cms/admin-user"],
        'img'      => ['\skeeks\cms\modules\admin\assets\AdminAsset', 'images/icons/user.
↵png'],
        'priority' => 0
      ],
      //....
    ]
  ],
],
```

Каждый элемент массива может содержать следующие опции:

- **label** — Название пункта меню
- **priority** — Порядок чем меньше тем выше пункт
- **enabled** — Показывается или не показывается
- **img** — Картинка (массив [Asset, „путь к файлу“])
- **url** — URL массив который будет передан в `yii\helpers\Url::to()`;

2.5.3 Создание контроллера

2.5.3.1 В проекте

Ссылка на контроллер: `Url::to(['/admin-competition'])`

Создать файл контроллера: `frontend/controllers/AdminCompetitionController.php`

```
namespace frontend\controllers;
use frontend\modules\competition\models\Competition;

class AdminCompetitionController extends \skeeks\cms\modules\admin\controllers\AdminModelEditorController
{
    public function init()
    {
        $this->name                = \Yii::t('app', 'Конкурсы');
        $this->modelShowAttribute  = "name";
        $this->modelClassName     = Competition::className();

        parent::init();
    }
}
```


Создать файл шаблона: frontend/views/admin-competition/index.php

```
<? $pjax = \skeeks\cms\modules\admin\widgets\Pjax::begin(); ?>

<?php echo $this->render('_search', [
    'searchModel' => $searchModel,
    'dataProvider' => $dataProvider
]); ?>

<?= \skeeks\cms\modules\admin\widgets\GridViewStandart::widget([
    'dataProvider' => $dataProvider,
    'filterModel' => $searchModel,
    'adminController' => $controller,
    'pjax' => $pjax,
    'columns' => [
        'name',
    ],
]); ?>

<? $pjax::end(); ?>
```

Создать файл для фильтров и поиска: frontend/views/admin-competition/_search.php

```
<?
    $filter = new \yii\base\DynamicModel([
        'id',
    ]);
    $filter->addRule('id', 'integer');

    $filter->load(\Yii::$app->request->get());

    if ($filter->id)
    {
        $dataProvider->query->andWhere(['id' => $filter->id]);
    }
?>
<? $form = \skeeks\cms\modules\admin\widgets\filters\AdminFiltersForm::begin([
    'action' => '/' . \Yii::$app->request->pathInfo,
]); ?>

<?= $form->field($searchModel, 'name')->setVisible(); ?>

<? $form::end(); ?>
```

Создать файл для редактирования элементов: frontend/views/admin-competition/_form.php

```
<?php $form = ActiveForm::begin(); ?>
<?php ?>

<?= $form->fieldSet( \Yii::t('skeeks/form2/app', 'General information'))?>
    <?= $form->field($model, 'name')->textInput(); ?>
    <?= $form->field($model, 'description')->textInput(); ?>
    <?= $form->field($model, 'is_active')->checkbox(); ?>
<?= $form->fieldSetEnd(); ?>

<?= $form->buttonsCreateOrUpdate($model); ?>
<?php ActiveForm::end(); ?>
```

2.5.3.2 Создание контроллера

2.6 Расширения

Дополнительные решения, расширяющие стандартные возможности SkeekS CMS

2.6.1 Shop (skeeks/cms-shop)

Полноценный интернет магазин.

2.6.1.1 Виджеты

Этапы оформления заказа

Потребуется установка skeeks/cms-shop-cart-steps-widget

```
<?= \skeeks\cms\shopCartStepsWidget\ShopCartStepsWidget::widget(); ?>
```

Товары в корзине

Потребуется установка skeeks/cms-shop-cart-items-widget

```
<?= \skeeks\cms\shopCartItemsWidget\ShopCartItemsListWidget::widget([
    'dataProvider' => new \yii\data\ActiveDataProvider([
        'query' => \Yii::$app->shop->shopFuser->getShopBaskets(),
        'pagination' =>
            [
                'defaultPageSize' => 100,
                'pageSizeLimit' => [1, 100],
            ]
    ]),
]); ?>
```

Оформление заказа

Потребуется установка skeeks/cms-shop-checkout-widget

```
<? $checkout = \skeeks\cms\shopCheckout\ShopCheckoutWidget::begin([]); ?>
<? \skeeks\cms\shopCheckout\ShopCheckoutWidget::end(); ?>
```

Скидочные купоны

Потребуется установка skeeks/cms-shop-discount-coupon-widget

```
<?= \skeeks\cms\shopDiscountCoupon\ShopDiscountCouponWidget::widget(); ?>
```

2.6.1.2 Примеры

Страница полной корзины

Шаблон находится по адресу `default\modules\shop\cart\cart.php`

```

<?
\frontend\assets\CartAsset::register($this);
\skeeks\cms\shop\widgets\ShopGlobalWidget::widget();
$this->registerJs(<<<JS
(function(sx, $, _)
{
    new sx.classes.shop.FullCart(sx.Shop, 'sx-cart-full');
})(sx, sx.$, sx._);
JS
);
?>
<!--==== Content Part ====>
<section class="sx-cart-layout bg-printair">
    <div class="row">
        <div class="container sx-border-block">
            <? \skeeks\cms\modules\admin\widgets\Pjax::begin([
                'id' => 'sx-cart-full',
            ]) ?>

            <? if (\Yii::$app->shop->shopFuser->isEmpty()) : ?>
            <!-- EMPTY CART -->
            <div class="panel panel-default">
                <div class="panel-body">
                    <strong>Ваша корзина пуста!</strong><br/>
                    В вашей корзине нет покупок.<br/>
                    Кликните <a href="/" data-pjax="0">сюда</a> для продолжения покупок. <br/>
                    <!--<span class="label label-warning">this is just an empty cart example</
=>span>-->

                </div>
            </div>
            <!-- /EMPTY CART -->
            <? else: ?>
            <? = \skeeks\cms\shopCartStepsWidget\ShopCartStepsWidget::widget(); ?>
            <hr/>
            <!-- LEFT -->
            <div class="col-lg-9 col-sm-8">
                <? = \skeeks\cms\shopCartItemsWidget\ShopCartItemsListWidget::widget([
                    'dataProvider' => new \yii\data\ActiveDataProvider([
                        'query' => \Yii::$app->shop->shopFuser->getShopBaskets(),
                        'pagination' =>
                        [
                            'defaultPageSize' => 100,
                            'pageSizeLimit' => [1, 100],
                        ]
                    ]),
                ]),
            </div>
            <!-- RIGHT -->
            <div class="col-lg-3 col-sm-4">
                <? $url = \yii\helpers\Url::to(['/shop/cart/checkout']); ?>

```

(continues on next page)

(продолжение с предыдущей страницы)

```

                <?= $this->render("_result", [
                    'submit' => <<<HTML
<a href="{ $url}" class="btn btn-primary btn-lg btn-block size-15" data-pjax="0">
    <i class="fa fa-mail-forward"></i> Оформить
</a>
HTML
                ]); ?>
            </div>
        <? endif; ?>

        <? \skeeks\cms\modules\admin\widgets\Pjax::end() ?>
    </div>
</div>
</section>

```

Страница оформления заказа

Шаблон находится по адресу `default\modules\shop\cart\checkout.php`

```

<?
\frontend\assets\CartAsset::register($this);
\skeeks\cms\shop\widgets\ShopGlobalWidget::widget();

$this->registerJs(<<<JS
(function(sx, $, _)
{
    new sx.classes.shop.FullCart(sx.Shop, 'sx-cart-full');
})(sx, sx.$, sx._);
JS
);
?>

<!--==== Content Part ====-->
<section class="sx-cart-layout bg-printair">
    <div class="row">
        <div class="container sx-border-block">
            <? \skeeks\cms\modules\admin\widgets\Pjax::begin([
                'id' => 'sx-cart-full',
            ]) ?>

            <? if (\Yii::$app->shop->shopFuser->isEmpty()) : ?>
                <!-- EMPTY CART -->
                <div class="panel panel-default">
                    <div class="panel-body">
                        <strong>Ваша корзина пуста!</strong><br />
                        В вашей корзине нет покупок.<br />
                        Кликните <a href="/" data-pjax="0">сюда</a> для продолжения покупок. <br />
                        <!--<span class="label label-warning">this is just an empty cart example</
<span>-->
                    </div>
                </div>
                <!-- /EMPTY CART -->
            <? else: ?>

```

(continues on next page)

(продолжение с предыдущей страницы)

```

<?= \skeeks\cms\shopCartStepsWidget\ShopCartStepsWidget::widget(); ?>

<hr />

<!-- LEFT -->
<div class="col-lg-9 col-sm-8">

    <!-- CART -->

    <!-- cart content -->
    <div id="cartContent">

<?
$this->registerCss(<<<CSS
.radio input[type=radio]
{
    left: 0px;
    margin-left: 0px;
}
.checkbox label, .radio label
{
    padding-left: 0px;
}
);
?>
CSS

    <? $checkout = \skeeks\cms\shopCheckout\ShopCheckoutWidget::begin([
        'btnSubmitWrapperOptions' =>
        [
            'style' => 'display: none;'
        ]
    ]); ?>
    <? \skeeks\cms\shopCheckout\ShopCheckoutWidget::end(); ?>

    <div class="clearfix"></div>
</div>
<!-- /cart content -->

<!-- /CART -->

</div>

<!-- RIGHT -->
<div class="col-lg-3 col-sm-4">

    <? $url = \yii\helpers\Url::to(['/shop/cart/payment']) ; ?>
    <?= $this->render("_result", [
        'submit' => <<<HTML
    <a href="#" onclick="$({'#{$checkout->formId}').submit(); return false;" class="btn btn-primary
    ↪btn-lg btn-block size-15" data-pjax="0">
        <i class="fa fa-mail-forward"></i> Оформить
    </a>
HTML

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        ]); ?>

        </div>
    <? endif; ?>

    <? \skeeks\cms\modules\admin\widgets\Pjax::end() ?>
</div>
</div>
</section>

```

Финальная страница заказа

Шаблон находится по адресу `default\modules\shop\order\finish.php`

```

<section>
    <div class="row">
        <div class="col-sm-12">

<? = \skeeks\cms\shopCartStepsWidget\ShopCartStepsWidget::widget(); ?>
<hr />
<div class="box-light">
    <!--==== Content Part =====>
    <div class="row">
        <div class="col-lg-12 col-md-10">
            <h4>Заказ №<? = $model->id; ?> от <? = \Yii::$app->formatter->asDatetime($model->
<-created_at); ?> </h4>

            <div class="table-responsive">
                <? = \yii\widgets\DetailView::widget([
                    'model' => $model,
                    'template' => "<tr><th>{label}</th><td style='width:50%;>{value}</td></tr>",
                    'attributes' => [
                        /*[
                            // the owner name of the model
                            'label' => 'Номер заказа',
                            'format' => 'raw',
                            'value' => $model->id,
                        ],*/
                        /*[
                            // the owner name of the model
                            'label' => 'Создан',
                            'format' => 'raw',
                            'value' => \Yii::$app->formatter->asDatetime($model->created_at),
                        ],*/
                        [
                            // the owner name of the model
                            'label' => 'Сумма заказа',
                            'format' => 'raw',
                            'value' => \Yii::$app->money->convertAndFormat($model->moneyOriginal),
                        ],
                        [
                            // the owner name of the model
                            'label' => 'Способ оплаты',
                            'format' => 'raw',
                            'value' => $model->paySystem->name,
                        ],
                        [
                            'label' => 'Доставка',
                            'format' => 'raw',

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        'value' => 'Курьер',
    ],
    [
        // the owner name of the model
        'label' => 'Статус',
        'format' => 'raw',
        'value' => Html::tag('span', $model->status->name, ['style' => 'color:
↪ ' . $model->status->color]),
    ],
    [
        // the owner name of the model
        'label' => 'Оплата',
        'format' => 'raw',
        'value' => $model->payed == 'Y' ? "<span style='color: green;'">Оплачен
↪ </span>" : "<span style='color: red;'">Не оплачен</span>",
    ],
    [
        // the owner name of the model
        'attribute' => 'Заказ отменен',
        'label' => 'Заказ отменен',
        'format' => 'raw',
        'value' => $model->reason_canceled,
        'visible' => $model->canceled == 'Y',
    ],
    ],
    ])?>
</div>
<h4>Данные покупателя: </h4>

<div class="table-responsive">
    <?= \yii\widgets\DetailView::widget([
        'model' => $model->buyer->relatedPropertiesModel,
        'template' => "<tr><th style='width: 50%;'">{label}</th><td style='width:50%;'">
↪ {value}</td></tr>",
        'attributes' => array_keys($model->buyer->relatedPropertiesModel->toArray())
    ])?>
</div>
<h4>Содержимое заказа: </h4>
<!-- cart content -->
<?= \skeeks\cms\shopCartItemsWidget\ShopCartItemsListWidget::widget([
    'dataProvider' => new \yii\data\ActiveDataProvider([
        'query' => $model->getShopBaskets(),
        'pagination' =>
        [
            'defaultPageSize' => 100,
            'pageSizeLimit' => [1, 100],
        ],
    ]),
    'footerView' => false,
    'itemView' => '@skeeks/cms/shopCartItemsWidget/views/items-list-order-item',
]);?>
<!-- /cart content -->
<div class="toggle-transparent toggle-bordered-full clearfix">
    <div class="toggle active" style="display: block;">
        <div class="toggle-content" style="display: block;">

            <span class="clearfix">
                <span
                    class="pull-right"><?= \Yii::$app->money->convertAndFormat(
↪ $model->moneyOriginal);?></span>

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        <strong class="pull-left">Товаров:</strong>
    </span>
    <? if ($model->moneyDiscount->getValue() > 0) : ?>
    <span class="clearfix">
        <span
            class="pull-right"><?= \Yii::$app->money->convertAndFormat(
↳$model->moneyDiscount); ?></span>
            <span class="pull-left">Скидка:</span>
        </span>
    <? endif; ?>

    <? if ($model->moneyDelivery->getValue() > 0) : ?>
    <span class="clearfix">
        <span
            class="pull-right"><?= \Yii::$app->money->convertAndFormat(
↳$model->moneyDelivery); ?></span>
            <span class="pull-left">Доставка:</span>
        </span>
    <? endif; ?>

    <? if ($model->moneyVat->getValue() > 0) : ?>
    <span class="clearfix">
        <span
            class="pull-right"><?= \Yii::$app->money->convertAndFormat(
↳$model->moneyVat); ?></span>
            <span class="pull-left">Налог:</span>
        </span>
    <? endif; ?>

    <? if ($model->weight > 0) : ?>
    <span class="clearfix">
        <span class="pull-right"><?= $model->weight; ?> г.</span>
        <span class="pull-left">Вес:</span>
    </span>
    <? endif; ?>
</hr>

    <span class="clearfix">
        <span
            class="pull-right size-20"><?= \Yii::$app->money->
↳convertAndFormat($model->money); ?></span>
        <strong class="pull-left">ИТОГ:</strong>
    </span>
</hr>
    <? if ($model->allow_payment == \skeeks\cms\components\Cms::BOOL_Y) : ?>
    <? if ($model->paySystem->paySystemHandler && $model->payed == 'N') : ?
↳>
        <?= Html::a("Оплатить", \yii\helpers\Url::to(['/shop/order/finish-
↳pay', 'key' => $model->key]), [
            'class' => 'btn btn-lg btn-primary'
        ]); ?>
    <? else : ?>

    <? endif; ?>
    <? else : ?>
    <? if ($model->paySystem->paySystemHandler) : ?>

```

(continues on next page)


```

        <span class="clearfix">
            <span class="pull-right"><?= \Yii::$app->money->convertAndFormat(\Yii::$app->
↵shop->shopFuser->moneyVat); ?></span>
            <span class="pull-left">Налог:</span>
        </span>
    <? endif; ?>

    <? if (\Yii::$app->shop->shopFuser->weight > 0) : ?>
        <span class="clearfix">
            <span class="pull-right"><?= \Yii::$app->shop->shopFuser->weight; ?> р.</span>
            <span class="pull-left">Всего:</span>
        </span>
    <? endif; ?>

    <hr />

    <span class="clearfix">
        <span class="pull-right size-20"><?= \Yii::$app->money->convertAndFormat(\Yii::
↵$app->shop->shopFuser->money); ?></span>
        <strong class="pull-left">ИТОГ:</strong>
    </span>

    <hr />

    <?= $submit; ?>
</div>
</div>
</div>

```

2.7 FAQ

2.7.1 Разные вопросы

Тут временно публикуются различные полезные примеры, которые можно использовать на своих сайтах. Куча разных несвязанных примеров, которые могут быть полезны. Позже это будет структурировано и разнесено

2.7.1.1 Как правильно сделать resize изображений?

Вот так можно получить ссылку на resize изображения.

```

echo \Yii::$app->imaging->thumbnailUrlOnRequest($model->image ? $model->image->src : null,
    new \skeeks\cms\components\imaging\filters\Thumbnail([
        'w' => 0,
        'h' => 200,
    ]), $model->code
);

```

2.7.1.2 Как можно загрузить файл в хранилище?

Скачать с удаленного сервера и загрузить в хранилище

```

$imageUrl = 'http://test.ru/test.jpg';
$element = \skeeks\cms\models\CmsContentElement::find(10);

$file = \Yii::$app->storage->upload($imageUrl, [
    'name' => $element->name
]);

$element->link('image', $file);

```

2.7.1.3 Как правильно поймать событие отправки формы в конструкторе

Чаще всего, возникает необходимость отлавливать события успешной отправки формы, и сообщать о них в yandex или google метрику. Для этого предлагается использовать следующий код:

```

<?
\yii\bootstrap\Modal::begin([
    'header' => 'Обратная связь',
    'id' => 'sx-feedback',
    'toggleButton' => false,
    'size' => \yii\bootstrap\Modal::SIZE_DEFAULT
]);
?>
<?= \skeeks\modules\cms\form2\cmsWidgets\form2\FormWidget::widget([
    'form_code' => 'feedback',
    'namespace' => 'FormWidget-feedback',
    'viewFile' => 'with-messages',
    'successJs' => new \yii\web\JsExpression(<<<JS
console.log('successJs');
//yaCounter17836507.reachGoal('btn-send');
JS
    ),
    'errorJs' => new \yii\web\JsExpression(<<<JS
console.log('errorJs');
JS
    ),
    //'viewFile' => '@app/views/widgets/FormWidget/fiz-connect'
]); ?>
<?
\yii\bootstrap\Modal::end();
?>

```

2.7.1.4 Как отметить обязательные поля в формах *

Для того чтобы добавить во все формы генерируемые стандартными средствами yii2, звездочки * обязательных полей. Глобально на странице можно подключить js и css.

```

$(function()
{
    $('.form-group.required label').each(function()
    {
        $(this).append('<span class="sx-from-required">').text(' *');
    });
});

```

```
.sx-from-required
{
  color: red;
  font-weight: bold;
}
```

2.7.1.5 Как изменить timeout рjax?

Для того, чтобы изменить timeout рjax, глобально во всех виджетах рjax проекта, можно добавить код javascript:

```
$(function()
{
  $.pjax.defaults.timeout = 30000;
});
```

2.7.1.6 Как вызвать событие в момент окончания работы рjax-скрипта с определенным id?

Для того, чтобы изменить вызвать событие в момент окончания работы рjax-скрипта, можно добавить код javascript:

```
$(document).on('pjax:complete', function (e) {
//любой рjax на странице закончился
  if (e.target.id == self.get('id')) {
    //рjax с конкретным id закончился $(e.target)
  }
});
```

2.7.1.7 Оптимальный robots.txt

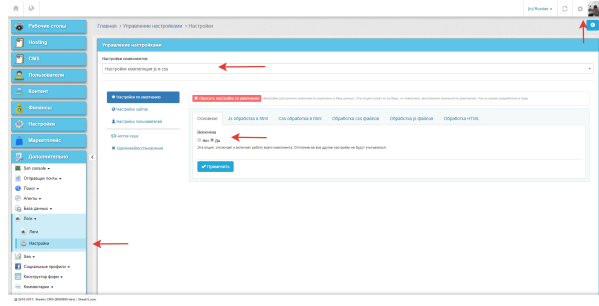
```
User-agent: *
Disallow: /*~*
Disallow: /search*
Disallow: *&SearchProductsModel*
Disallow: *?SearchProductsModel*
Disallow: *&SearchRelatedPropertiesModel*
Disallow: *?SearchRelatedPropertiesModel*
Disallow: *&ProductFilters*
Disallow: *?ProductFilters*
Clean-Param: from&_openstat&utm_source&utm_medium&utm_campaign&utm_content&utm_term&pm_source&pm_
↪block&pm_position&clid&yclid&ymlclid&frommarket&text
Host: https://your-site.com
Sitemap: https://your-site.com/sitemap.xml
```

2.7.1.8 Как включить js, css и html оптимизацию?

Для этих целей существуют дополнительное расширение, которое обычно уже стоит в базовых проектах.

<https://github.com/skeeks-cms/cms-assets-auto-compress>

Включение и настройка оптимизаций, производится через систему управления сайтом:



2.7.2 Перенос проекта на другой хостинг

2.7.2.1 Архивация

Создать актуальный архив базы данных

```
php yii dbDumper/mysql/dump
```

Создать архив вашего проекта

2.7.2.2 Восстановление

1. Скачать файлы проекта

Развернуть архив, или клонировать проект из git репозитория

2. Установка composer и зависимостей

```
# Download latest version of composer in project
curl -sS https://getcomposer.org/installer | COMPOSER_HOME=.composer php
# Download dependency
COMPOSER_HOME=.composer php composer.phar install -o
```

3. Configuring the database

Прописать коннект к базе данных *common/config/db.php*

4. Installation of migrations

```
#Installation of ready-dump
php yii dbDumper/mysql/restore
```