
SkeekS CMS Documentation

Release 3.2.2

Semenov Aleksander

February 23, 2017

1	Links	3
2	Info	5
2.1	Overview	5
2.1.1	Requirements	5
2.1.2	Installation	6
2.1.3	Update	8
2.1.4	Configuring Web Servers	8
2.1.5	Reporting a security vulnerability	10
2.1.6	Work with documents	10

SkeekS CMS - modern generic content management system based on yii2 php framework.

Links

- Web site
- Author
- ChangeLog

Info

It provides capabilities to manage site structure and content elements (news, publications, products, etc.).

Opportunities for management and dynamic creation of additional properties of the content sections and elements through the administration system.

Opportunities to manage users, privileges, roles and their purpose.

Supports single query entry point (one index.php), for backend and frontend parts of the project. By then, it is more convenient to configure nginx and 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.

Overview

Requirements

Software

- apache
- mod_rewrite apache module
- php >= 5.5
- DB mysql ~ 5.5
- SSH access

Php modules

- mbstring
- xml

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

Php settings

- short_open_tag on

Installation

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 | php
```

Note: 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. 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 | php

# Installing the base project SkeekS CMS
COMPOSER_HOME=.composer php composer.phar create-project --no-install --prefer-dist skeeks/app-basic
# Going into the project folder
cd demo.ru
# Download latest version of composer in project
curl -sS https://getcomposer.org/installer | php

# Extra plug-ins
COMPOSER_HOME=.composer php composer.phar global require fxp/composer-asset-plugin --no-plugins
# Enter your github api key in composer.json
# Download dependency
COMPOSER_HOME=.composer php composer.phar install -o
# Run the command to initialize the project, the installer executable file and the necessary rights
php yii cms/init
```

3. Configuring the database

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

4. Installation of migrations

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

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 \(web-server\)](#)

6. Authorization system

Default management system is available at the following address (if desired, it can be reconfigured)

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

root (login)

skeeks (password)

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).

Attention: It is important to remember to check the setting of php: **short_open_tag on**

Update

Standart update

```
# Composer update to the latest stable version
COMPOSER_HOME=.composer php composer.phar self-update
# Extra plug-ins
COMPOSER_HOME=.composer php composer.phar global require fxp/composer-asset-plugin --no-plugins
# Download dependency
COMPOSER_HOME=.composer php composer.phar update -o
# Clear all caches (Just in case)
php yii cms/cache/flush-all
# Installation of migration
php yii cms/migrate --interactive=0
# Init privilages. If the component is installed skeeks/cms-rbac (optionality)
php yii rbac/init
# Init agents. If the component is installed skeeks/cms-agent (optionality)
php yii cmsAgent/init
# Clear all caches (Just in case)
php yii cms/cache/flush-all
```

Fast update

Or all of these commands in one line

```
COMPOSER_HOME=.composer php composer.phar self-update && COMPOSER_HOME=.composer php composer.phar g:
```

Custom update

Or mount it in your settings file composer.json

```
"scripts": {
    "post-install-cmd": [
        "skeeks\\cms\\console\\Composer::postInstall"
    ],
    "post-update-cmd": [
        "skeeks\\cms\\console\\Composer::postUpdate",
        "php yii cms/cache/flush-all",
        "php yii cms/migrate --interactive=0",
        "php yii rbac/init",
        "php yii cmsAgent/init",
        "php yii cms/cache/flush-all"
    ]
},
```

Exemple: <https://github.com/skeeks-cms/app-basic/blob/master/composer.json>

Configuring Web Servers

Note: 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.

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

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

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.

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.

Work with documents

```
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
```