

No-CMS User, Designer, and Developer Guide 0.6.5 stable



Go Frendi Gunawan, 2013-08-17

User guide

This section is intended for everyone. Some basic concept will be covered in this part.

Installation

1. If you use windows xampp, copy No-CMS into c:\xampp\htdocs, start up apache and MySQL, open your browser and go to <http://localhost/No-CMS/>
If you use Linux or UNIX, copy No-CMS into /var/www, or your public_html directory, change the access into 777, open <http://your-domain/No-CMS/>
2. Follow on-screen instruction

Themes

Like many other CMS, No-CMS provide some fancy default themes. You can change the theme by accessing CMS Management | Change Theme. To make your own themes, you should read “Designer guides” section.

Authorization

Unlike other CMS there is no such a specific “control-panel” or “admin-dashboard” in No-CMS, You have a full control about “who can do what”.

The authorization is used to ensure that only certain users can access some pages.

No-CMS provide 4 authorization types. This is already covers everything.

1. **Everyone**
The most common authorization. Every page with “Everyone” authorization can be viewed by all visitor, regardless of their login information. Some pages such a home-page or blog that can be viewed by all visitor should have “Everyone” authorization type.
2. **Authenticated**
Once a user logged in into the system, he/she will be authenticated. Every page with “Authenticated” authorization can only be viewed by logged-in user. Some pages that require login (e.g: change profile) should have this type of authorization type
3. **Unauthenticated**
Before a user logged in into the system (or once he/she log-out), he/she will be unauthenticated. This type of authorization is rarely used. Some pages that usually require such a authorization types are “forgot-password” and “register”.
4. **Authorized**
This is the most strict authorization type. Only user that has logged-in and already registered in certain user-group can access it. Some crucial pages such a “CMS-Management page” should has such a privilege.

PS : if a user is a member of Admin user-group, he/she will always be authorized to view any page.

User And Group

CMS Management >> User Management

Edit User

User Name* :

Go

E mail :

gofrendiasgard@gmail.com

Real Name :

Active :

☒ active
☐ inactive

Groups :

Admin x Employee x |

Update changes

Update and go back to list

Cancel

Figure 1: User management

Users are registered visitor. To be a user of No-CMS system, one need to register. The registration can be done via registration page or by using third-party authentication (if activated). You can manage users by accessing CMS Management | User Management.

A user can be a member of several user group, and a user group can contains many user. Group represent user's role. Since a user can has several role, therefore he/she can also be member of several group. User group is important for page with “authorized” authorization type.. You can manage group by accessing CMS Management | Group Management.

You can manage user and user-group relation by using both, group management or user management.

PS : if a user is a member of Admin user-group, he/she will always be authorized to view any page.

Page And Navigation

Website is a collection of web pages. You can add, edit, activate, deactivate, or even delete pages by accessing CMS Management | Navigation Management.

Page has several common properties including

- Navigation Code (required, and should be unique)
This is the identifier of a page
- Navigation Title
This will be appeared in Navigation Menu
- Page Title
The title of the page. It will be appeared in your browser's tab. This is also useful for SEO, so it is recommended to give an appropriate title to make your website search-engine friendly.

- **Page Keyword**
Some old web-crawler still use meta tag. This is not very useful now a day, since several search engine use title and page content to do crawl activities.

The screenshot shows a web application window titled "Edit Navigation (Page)". It contains several form fields for editing a page's navigation information:

- Navigation Code :** A text input field containing "main_index".
- Parent :** A dropdown menu with "Select Parent" as the current selection.
- Navigation Title (What visitor see) :** A text input field containing "Home".
- Page Title :** A text input field containing "Home page".
- Page Keyword (Comma Separated) :** A text input field containing "No-CMS, CodeIgniter CMS".
- Description :** A larger text area containing "There is no place like home :D".
- Active :** Two radio buttons labeled "active" (which is selected) and "inactive".

Figure 2: Page Management

There are two types of pages:

1. **Static page**

Static pages are only contains of HTML and CSS without any need of server-side programming. This is the easiest one to create. You can create new static page without even know any programming knowledge. If you are already familiar with MS word, Libre-Office write or any word-processor program, then you already have everything to make static page. To make a new static page, you need to change the activate “static” option. Once you activate static option, a word-editor-like input will be appeared. Do your editing there.

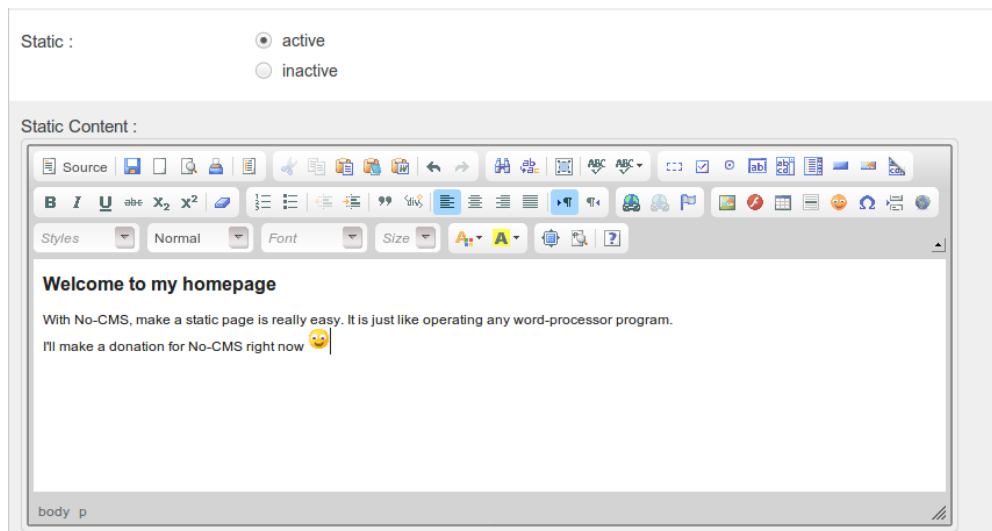


Figure 3: Static Page

2. Dynamic page

Dynamic pages are more than just static page. There are some server-side programming involved. A normal user might don't need to mess up with this. Navigation Management, user management and group management are some example of dynamic page. When you deactivate static option, you will be asked for the URL.

In both, static and dynamic page, you can have custom theme. To do this, you can choose an appropriate “Default Theme”. This will overwrite No-CMS default theme setting.

Normally you can access a page by using navigation menu. It is natural to have some grouped pages. You can change “Navigation Parent Code” to make a page nested in another page.

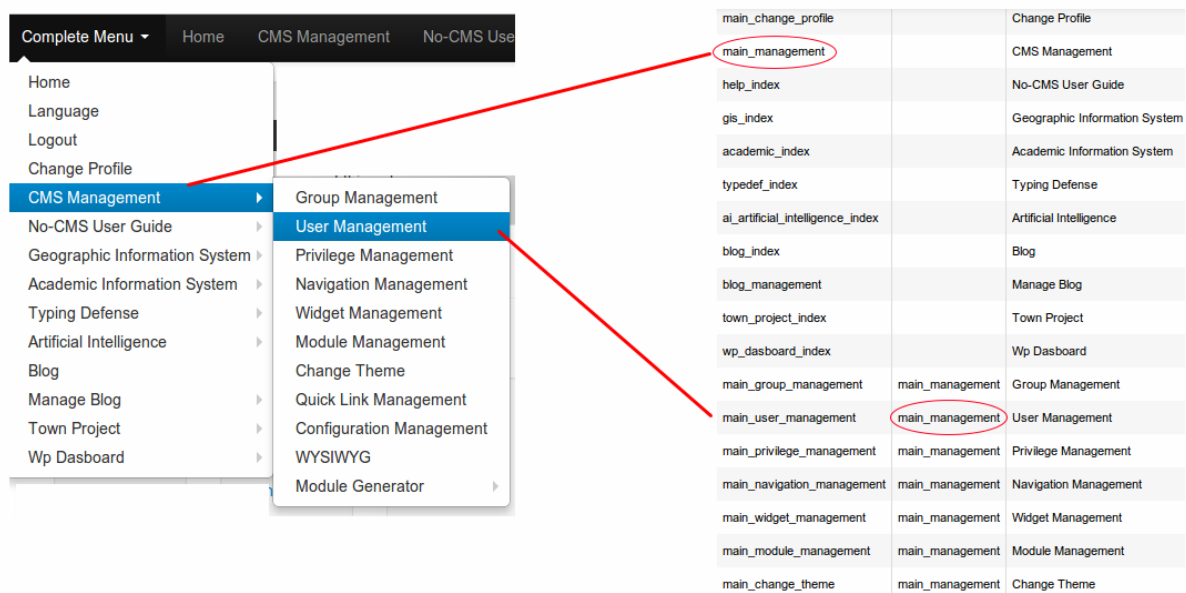


Figure 4: How Navigation Parent Works

One of No-CMS best feature is page authorization. You have full control of what page can be accessed by certain user group. This can be easily done by changing “authorization” value of a page in Navigation management. Four authorization type is provided for this purpose: Everyone, authenticated, unauthenticated and authorized (See Authorization section). If you choose authorized, you should also provide user-group.

Authorization : Authorized ✕ ▼

Groups : Employee ✕

Figure 5: Page Authorization

PS: Please remember that “Admin” group will always be authorized.

Quicklink

Sometime, frequently accessed pages are hidden inside complex navigation structure (i.e: have a lot of navigation parent). Quicklink is used to make a shortcut for those kind of pages. You can add quicklink by accessing CMS Management | Quicklink Management

Widget

Widget is everything that can be injected into your page. It can be a simple javascript, or even another page. You can manage widget by accessing CMS Management | Widget Management.

As pages, you can also define widget's authorization type.

A widget can has several slugs (separated by comma). The slugs are used to group and identify widgets.

➕ Add Widget 📄 Export 🖨 Print								
Widget Code	Title (What visitor see)	Active	Static	Description	Authorization	Slug	Groups	Actions
login	Login	Active	inactive	Visitor need to login for authentication	Unauthenticated	sidebar,user_widget		
logout	User Info	Active	active	Logout	Authenticated	sidebar,user_widget		
social_plugin	Share This Page !!	Active	active	Addthis	Everyone	sidebar		
google_search	Search	Inactive	active	Search from google	Everyone	sidebar		
google_translate	Translate !!	Inactive	active	The famous google translate	Everyone	sidebar		
calendar	Calendar	Inactive	active	Indonesian Calendar	Everyone	sidebar		
google_map	Map	Inactive	active	google map	Everyone	sidebar		
donate_nocms	Donate No-CMS	Active	active	No-CMS Donation	Everyone	advertisement		
blog_widget_newest	Article	Inactive	inactive		Everyone	sidebar		

🔍
25
⏪
⏩
Page 1 of 1
🔄
Displaying 1 to 9 of 9 items

Figure 6: Widget and Slug

By using `{{ widget_slug:slug }}` tag (further about it in designer guide), you can show any widget by

their slugs or code anywhere (in themes, layout, or even static/dynamic page content).

For example, `{{ widget_slug:user_widget }}` will be rendered and replaced by widgets with “user_widget” slug (in this case, login and logout widget).

To be more specific, you can also use `{{ widget_name:widget_code }}`, e.g:
`{{ widget_name:donate_nocms }}` will be rendered as No-CMS donation widget.

Layout Editor

For convenience, No-CMS also provide layout editor. You can use it by accessing CMS Management | Layout Management. Here you can change the site title, site jargon, site logo, navigation & widget order, and also default language of your site. For more complete configuration control, you can use Configuration Management.

Configuration Management

There are a lot of “hidden” configuration to be managed here. If you are familiar with windows OS, configuration management is similar to regedit. It is a place when you can manage various configuration. Normally you don't need to do anything with configuration management, but when you want to mess up with it, please make sure you know what you do. Some configuration keys are vital. Delete it or give a wrong value will make your entire website doesn't work. With configuration management you can manage several things, including automatic e mail sending, google analytic account, etc.

Designer guide

This section is intended for designers. If you are just a “normal” user without any CSS-HTML-PHP technical skill, and you are happy with currently available themes (or any other themes which is already provided by your designer), then you don't need to read this section (except the layout's tag section). Some design concept will be covered in this part. Some php skill might help you do things, but it is not mandatory.

Theme

Making theme is No-CMS is quiet easy if you are already familiar with Phil Sturgeon's template library. More about template library can be found at `template_user_guide` directory in your No-CMS installation folder. Basically a theme can consists of several layout and partials. No-CMS themes are located at `/themes` directory.

Layout

The basic idea of layout is that most of your pages will share the same header, footer, wrapping navigation, etc and only the actual body of the page will change. Using this logic you can avoid repeating yourself and having lots of similar HTML.

This code might give you the idea what layout really like is :

```
<!DOCTYPE html>
<html>
    <head>
        <title><?php echo $template['title']; ?></title>
        <?php echo $template['metadata']; ?>
    </head>
    <body>
        <h1><?php echo $template['title']; ?></h1>
        <div class="nav-collapse in collapse" id="main-menu" style="height: auto; ">
            {{ widget_name:top_navigation }}
        </div>
        <?php echo $template['body']; ?>
    </body>
    <footer>{{ site_footer }}</footer>
</html>
```

Code 1: default layout (/themes/your_theme_name/views/layouts/default.php)

Almost all your pages will share the same footer, the same site slogan, the same site logo, and even the same widgets. In this case, the only thing changed is the content of the page. As a designer, you can retrieve the content of the page by using `$template['body']` variable.

The default layout is located in `/themes/your_theme_name/views/layouts/default.php`. Some php variables and tags can be accessed here. The usage of tags, whenever possible is recommended, since it can also be applied for static pages too.

Layout tags:

Tag pattern	Description
<code>{{ user_id }}</code>	Show user id or empty string if user not logged in
<code>{{ user_name }}</code>	Show user name or empty string if user not logged in
<code>{{ user_real_name }}</code>	Show user's real name or empty string if user not logged in
<code>{{ user_email }}</code>	Show user's email or empty string if user not logged in
<code>{{ site_url }}</code>	Show the site's url with trailing slash (e.g: http://localhost/No-CMS/index.php/) Usually used if you want to show a page <code>Go to home</code>
<code>{{ base_url }}</code>	Show the base url with trailing slash (e.g: http://localhost/No-CMS/) Usually used if you want to include an image, javascript or css <code></code>
<code>{{ module_path }}</code>	Show the module path relative to the /modules directory (e.g: main, blog etc)
<code>{{ module_site_url }}</code>	Show the module site url (e.g: http://localhost/No-CMS/index.php/blog)
<code>{{ module_base_url }}</code>	Show the module site url (e.g: http://localhost/No-CMS/blog)
<code>{{ widget_slug:slug }}</code>	Show widgets with certain slug (e.g: <code>{{ widget:advertisement }}</code> will show all widget that has <i>advertisement</i> slug)
<code>{{ widget_name:widget_name }}</code>	Show widget with certain code (e.g: <code>{{ widget_name:login }}</code> will show a widget named <i>login</i>)
<code>{{ language }}</code>	Show current language used
<code>{{ language:some_word }}</code>	Show translation of word in current language. The language setting file is located in /assets/nocms/languages and /modules/module_name/assets/languages . Just add a new entry and it will works
<code>{{ if_language:a_language }}</code> something that will be appeared for a language <code>{{ elif_language:another_language }}</code> something that will be appeared for another language <code>{{ else }}</code> something that will be appeared for another case	This tag make No-CMS prevail in multi-language content. Let's take a look at this simple example: <pre>{{ if_language:english }} <p>Nice to meet you, {{ user_real_name }}</p> {{ elif_language:japanese }} <p>始めまして, {{ user_real_name }} さん</p> {{ elif_language:indonesia }} <p>Senang bertemu dengan anda, {{ user_real_name }}</p> {{ else }}</pre>

{{ end_if }}	<p><p>Hi, {{ user_real_name }}</p> {{ end_if }}</p> <p>If your language setting is “english”, then there will be “Nice to meet you, Go Frendi Gunawan” shown. If your language setting is “japanese”, then there will be “始めまして, Go Frendi Gunawan さん” shown. Such a thing will also be applied if your language setting is “indonesia” or “other” language.</p>
{{ configuration_value }}	<p>Show configuration value. Example: {{ site_name }}, {{ site_slogan }}, {{ site_favicon }}</p> <p>For complete configuration value, take a look at CMS Management Configuration Management</p>
{{ navigation_path }}	Create navigation path

Layout PHP variables:

Variable	Description	Example of possible value (type) value
\$template['title'], \$template['metadata'], \$template['partials'], \$template['body']	Related to template user guide, Please have a visit at your_no_cms_installation_directory/template_user_guide for more information	

Partials

For modularity sake, you might want to separate your layout into several part (e.g: header, footer, etc). Phil Sturgeon's template library (which is used in No-CMS) already support this. You can make as many partials as you need, and access it by using **\$template['partials']['your_partial_name']** variable. Partial files should be located at **/themes/your_theme_name/views/partials/default/**. The previous example layout can be separated into several partials:

```
<!DOCTYPE html>
<html>
    <head>
        <title><?php echo $template['title']; ?></title>
        <?php echo $template['metadata']; ?>
    </head>
    <body>
        <?php echo $template['partials']['header']; ?>
        <?php echo $template['body']; ?>
    </body>
    <?php echo $template['partials']['footer']; ?>
</html>
```

Code 2: default layout with several partials (/themes/your_theme_name/views/layouts/default.php)

```
<h1><?php echo $template['title']; ?></h1>
<div class="nav-collapse in collapse" id="main-menu" style="height: auto; ">
    {{ navigation_top_quicklink }}
</div>
```

Code 3: header partial (/themes/your_theme_name/views/partials/default/header.php)

```
<footer>{{ site_footer }}</footer>
```

Code 4: footer partial (/themes/your_theme_name/views/partials/default/footer.php)

Developer guide

This section is intended for developers. Some programming and API stuff will be covered in this part. If you are a just a user, or designer, this section might be not suitable for you. This section is intended to module developer or everyone who want to make a change in No-CMS's already exist modules. To know the concept of module, you should already be familiar with CodeIgniter MVC pattern and HMVC plugin. If you want a basic tutorial, please visit <https://github.com/goFrendiAsgard/No-CMS/blob/master/readme.md>

Modules

Modules are used to extend No-CMS capability. No-CMS blog and Nordrassil module-generator is an example of a module. All modules should be located at /modules/ directory and consists of models/, controllers/ and views/ directory. For more information about module, please visit <https://github.com/goFrendiAsgard/No-CMS/tree/master#tutorial-04-modules>

Module Generator

The module generator (also known as Nordrassil) is one of No-CMS best feature. With module generator you can easily make a skeleton of a module. To use module generator, follow these instructions:

- Go to CMS Management | Module Generator
- Make a new project. (You can make a project based on already exists database). It will be better if your database has some certain table prefix.
- Edit tables and columns of your project
- Click generate, and it is.

Classes & API

Not available yet, but having a look at /core/MY_Controller and /core/My_Model might give you a clue. The source code is well documented.

ERD

Below is No-CMS ERD:

