Joomla! CMS Cheatsheet

A quick reference guide to Joomla!, covering core concepts, administration, development, and security best practices for this popular open-source content management system.

Joomla! Core Concepts

Key Terminology

Article	The basic unit of content in Joomla!. Can be text, images, videos, etc.
Category	Used to organize articles and other content types.
Module	Lightweight extensions used to display content and features in specific positions on the page.
Component	Larger extensions that provide core functionality like content management, user management, or e-commerce.
Plugin	Extensions that modify or enhance Joomla!'s core functionality. They respond to specific events.
Template	Controls the look and feel of the website. Defines the layout and styling.

Admin Interface Sections

Control Panel	Provides an overview of the site and quick access to common tasks.
Content	Manage articles, categories, and media.
Menus	Create and manage website navigation menus.
Users	Manage user accounts and permissions.
Extensions	Install, manage, and configure extensions (components, modules, plugins, templates).
System	Configure global settings, manage system information, and perform maintenance tasks.

Common Tasks & Configuration

Article Management

Creating a New Article:

1. Go to Content > Articles > Add New Article.
2. Enter the title, content, and category.
3. Configure publishing options (status, access level, start/end publishing).
4. Save the article.

Editing an Existing Article:

1. Go to Content > Articles.
2. Find the article you want to edit.
3. Click on the article title.
4. Make your changes and save.

Article Options:

- `Publishing` tab: Set status (Published, Unpublished, Archived, Trashed), start/end publishing dates, and access levels.
- `Options` tab: Control article display settings like title, category, author, and date.

Menu Management

Creating a New Menu Item:

1. Go to Menus > [Your Menu] > Add New Menu Item.
2. Enter the menu item title.
3. Select the Menu Item Type (e.g., Single Article, Category Blog).
4. Configure the required settings for the selected type.
5. Save the menu item.

Editing an Existing Menu Item:

1. Go to Menus > [Your Menu].
2. Find the menu item you want to edit.
3. Click on the menu item title.
4. Make your changes and save.

Menu Item Types:

- `Single Article`: Links to a specific article.
- `Category Blog`: Displays articles from a specific category in a blog format.
- `Category List`: Displays a list of articles from a specific category.
- `External URL`: Links to an external website.

Module Management

Creating a New Module:

1. Go to Content > Site Modules > Add New Module.
2. Select the Module Type (e.g., Custom HTML, Articles - Latest).
3. Configure the module settings.
4. Assign the module to a position on the template.
5. Save the module.

Editing an Existing Module:

1. Go to Content > Site Modules.
2. Find the module you want to edit.
3. Click on the module title.
4. Make your changes and save.

Module Positions:
Module positions are defined in the template. Common positions include `top`, `bottom`, `left`, `right`, `header`, and `footer`. You can create custom positions in your template.

Extension Development

Component Structure

A Joomla! component typically consists of the following files and directories:

- `com_[component_name].xml`: The installation manifest file.
- `admin/`: Contains the administrator-side files.
  - `controller.php`: Handles requests and responses.
  - `models/`: Contains data models.
  - `views/`: Contains view files for displaying data.
  - `tmpl/`: Contains template files for the views.
- `site/`: Contains the front-end files. Similar structure to `admin/`.

Plugin Development

A Joomla! plugin consists of at least one PHP file and an XML manifest file.

- `[plugin_name].xml`: The installation manifest file.
- `[plugin_name].php`: The main plugin file. This file contains the event handlers.

Example:

```
// Plugin Name: plg_system_example

defined('_JEXEC') or die;

class PlgSystemExample extends JPlugin
{
  function onAfterRoute()
  {
    // Your code here
  }
}
```

Module Development

A Joomla! module requires a PHP file and a module XML manifest file.

- `mod_[module_name].xml`: The installation manifest file.
- `mod_[module_name].php`: The main module file.
- `helper.php`: (Optional) Contains helper functions.
- `tmpl/`: Contains the module's template files.

Example:

```
// mod_example.php

defined('_JEXEC') or die;

require_once dirname(__FILE__) .
'/helper.php';

$hello = ModExampleHelper::getHello();
require
JModuleHelper::getLayoutPath('mod_example');
```

Security Best Practices

Core Security Measures

Keep Joomla! Up-to-Date:
Regularly update Joomla! to the latest version to patch security vulnerabilities. Joomla! releases security updates frequently.

Keep Extensions Up-to-Date:
Ensure all installed extensions (components, modules, and plugins) are also up-to-date. Outdated extensions are a common entry point for attackers.

Strong Passwords:
Use strong, unique passwords for all administrator accounts and encourage users to do the same. Use a password manager to generate and store passwords securely.

Two-Factor Authentication (2FA):
Enable 2FA for administrator accounts to add an extra layer of security. This requires a second verification method (e.g., a code from an authenticator app) in addition to the password.

Regular Backups:
Create regular backups of your Joomla! website, including the database and files. Store backups in a secure, off-site location. Use backup extensions like Akeeba Backup.

Configuration Hardening

Rename the `administrator` Directory:
Rename the default `administrator` directory to a unique name to make it harder for attackers to find the login page. Use a tool like Admin Tools to automate this process.

Disable Directory Indexing:
Prevent directory listing by creating an empty `index.html` file in each directory or by modifying your `.htaccess` file.

Secure Configuration.php:
Ensure that your `configuration.php` file has the correct permissions (444 or 644) to prevent unauthorized access.

Enable HTTPS:
Use HTTPS to encrypt communication between the user's browser and the server. Obtain an SSL certificate and configure Joomla! to use HTTPS.

Web Application Firewall (WAF)

Implement a WAF:
Use a web application firewall (WAF) to protect your Joomla! website from common web attacks such as SQL injection, cross-site scripting (XSS), and remote file inclusion (RFI). Popular WAFs include Admin Tools and Cloudflare.

Monitor Logs:
Regularly monitor your server logs for suspicious activity, such as failed login attempts, unusual requests, and error messages. Use log analysis tools to automate the monitoring process.