

Docs

Markdown Project

Derafu Markdown

Anonymous · 21/08/2026 · #php

Table of Contents

Markdown Project 3

Introduction 4

Guide 8

Docs » Views and Templates Category » Markdown Project

Markdown Project

Derafu Markdown

Anonymous · 21/08/2026 · #php

Derafu Markdown

Last updated on 21/08/2026

Docs » Views and Templates Category » Markdown Project » Introduction

Introduction

PHP Markdown Rendering Library

Anonymous · 21/08/2026

PHP Markdown Rendering Library

last commit

march

 CI

passing

code size

62.2 KiB

open issues

0

downloads

2.7 k

downloads

165 this month

Derafu Markdown is a PHP library that provides a powerful Markdown rendering engine with support for advanced extensions. It leverages `league/commonmark` and additional features to enhance Markdown processing for documentation, blogs, and dynamic content.

Features

- **Full Markdown Support:** Standard CommonMark and GitHub Flavored Markdown (GFM).
- **Extended Capabilities:** TOC, footnotes, mentions, permalinks, embeds, and more.
- **Custom Attributes:** Add CSS classes and IDs to elements.
- **External Link Handling:** Open in new tabs, add `nofollow`, etc.
- **Highly Configurable:** Fine-tune Markdown behavior with options.
- **Easy Integration:** Works standalone or within any PHP project.
- **MIT Licensed:** Open-source and free to use.

Admonition Extension

This library includes an admonition extension to process common messages. For example: tip, info and warning.

Installation

Install via Composer:

```
composer require derafu/markdown
```

Usage

Basic Rendering

```
use Derafu\Markdown\Service\MarkdownCreator;
use Derafu\Markdown\Service\MarkdownService;

$markdownService = new MarkdownService(new MarkdownCreator());

echo $markdownService->render('example.md');
```

Rendering with Layout

```
$markdownService->render('example.md', [
    '__view_layout' => 'layout.php',
    '__view_title' => 'My Markdown Page'
]);
```

Available Extensions

□ GitHub Flavored Markdown (GFM)

- Task lists:

```
- [x] Completed
- [ ] Pending
```

- Tables:

```
| Name | Age |
|-----|-----|
| John | 25  |
| Alice | 30  |
```

□ Table of Contents (TOC)

```
## Section 1
## Section 2
```

□ Header Permalinks

```
#### Important Header
```

Generates an anchor link like `#important-header`.

Footnotes

Here is a reference^[^1].

^[^1]: Footnote text.

Custom Attributes

```
### Title {.custom-class}
```

External Links Handling

```
[Google](https://www.google.com)
```

Adds attributes like `rel="noopener noreferrer"`.

Mentions & Issues

Hello @user, check issue #123.

Links to GitHub profiles and issues.

Embeds

```
https://www.youtube.com/watch?v=dQw4w9WgXcQ
```

Automatically embeds videos.

Advanced Configuration

You can customize the behavior of Markdown processing by passing an options array to `MarkdownCreator`:

```
$options = [
  'environment' => [
    'mentions' => [
      '@' => ['generator' => 'https://github.com/%s'],
      '#' => ['generator' => 'https://github.com/derafu/markdown/issues/%d']
    ]
  ]
];

$creator = new MarkdownCreator($options);
```

Template Metadata Support

Markdown templates can include metadata in YAML format:

```
---
__view_title: "Markdown Template Guide"
---
```

Last updated on 21/08/2026

Docs » Views and Templates Category » Markdown Project » Guide

Guide

Template Usage Guide

Anonymous · 21/08/2026

Template Usage Guide

This document is designed to help you maximize the capabilities of Markdown within your application. By default, various extensions are configured to expand Markdown functionalities, allowing you to create dynamic and well-styled content easily.

1. Introduction to Markdown

Markdown is a lightweight markup language that allows you to write plain text documents and easily convert them into HTML. Its simple syntax makes it a popular choice for documentation, blogs, and other web applications. Below are some basic Markdown examples:

Headings

```
# Level 1 Heading

## Level 2 Heading

### Level 3 Heading
```

Emphasis

```
**Bold Text**

*Italic Text*

~~Strikethrough Text~~
```

Lists

```
- Unordered list item
- Another item

1. Ordered list item
2. Another item
```


Links and Images

```
[Link to Google](https://www.google.com)
```

```
![Alternative Image Text](https://www.example.com/image.jpg)
```

2. Markdown Extensions

You can extend Markdown capabilities with various extensions that enhance your content. Below is a description of the included extensions and how to use them.

2.1. GitHub Flavored Markdown (GFM)

GFM is an extended version of Markdown used on GitHub, adding features such as tables, task lists, and syntax-highlighted code blocks. To use GFM, simply write your Markdown content, and the extension will handle the rest.

Examples:

Task Lists:

- [x] Completed task
- [] Pending task

Tables:

```
| Header 1 | Header 2 |  
| ----- | ----- |  
| Cell 1   | Cell 2   |
```

Code Blocks:

```
```php  
echo "This is PHP code";
```

### 2.2. Automatic Table of Contents (TOC)

The Table of Contents (TOC) extension allows you to automatically generate an index based on the document's headings. To include a TOC, simply place `[TOC]` where you want it to appear.

#### Example:

```
Section 1
Content of section 1.
```

```
Section 2
Content of section 2.
```

## 2.3. Header Permalinks

Permalinks allow headers to have permanent links that users can copy and share easily. This extension automatically generates permalinks for all headers.

### Example:

```
Important Header
```

This will automatically generate the permalink `#content-important-header`.

## 2.4. Footnotes

Footnotes allow you to add references or clarifications without disrupting the text flow. Use the syntax `[^1]` to mark a footnote and define its content at the bottom of the document.

### Example:

Here is a phrase with a footnote<sup>[^1]</sup>.

<sup>[^1]</sup>: This is the footnote, appearing at the end of the document.

## 2.5. Definitions

Definition lists allow you to create glossaries or descriptive lists easily.

### Example:

Term 1  
: Definition for term 1.

Term 2  
: Definition for term 2.

## 2.6. Custom Attributes

This extension allows you to add custom HTML attributes to specific Markdown elements.

### Example:

```
Header with Custom ID {.my-css-class}
```

## 2.7. External Links

You can define how external links are handled. You can configure them to open in a new window, add attributes like `nofollow`, and more.

### Example:

```
[External Link](https://www.google.com)
```

This will generate a link that opens in a new window with additional attributes like `rel="noopener noreferrer"`.

## 2.8. Mentions

Mentions allow you to link, by default, to GitHub profiles or issues directly from Markdown.

### Example:

```
Hello @user, please check issue #123.
```

This will generate links to `https://github.com/user` and `https://github.com/derafu/markdown/issues/123`.

## 2.9. Embeds

The embed extension allows you to insert content from sites like YouTube directly into your document.

### Example:

```
https://www.youtube.com/watch?v=dQw4w9WgXcQ
```

This will embed the YouTube video instead of just displaying the link.

## 3. Advanced Options

---

Check the `MarkdownCreator` service to see the extensive customization options available for handling Markdown in your application. You can modify aspects such as:

- ID prefixes for permalinks.
- Custom styles for footnotes and tables.
- Advanced settings for mentions and embeds.

## Assigning Options

You must assign options when creating the `MarkdownCreator` service by passing an array with the options you want to modify.

Example:

```
$options = [
 'environment' => [
 'mentions' => [
 '@' => [
 'prefix' => 'https://github.com/',
 'pattern' => '[a-z\d](?:[a-z\d]|-(?=[a-z\d])){0,38}(?!\\w) ',
 'generator' => 'https://github.com/%s',
],
 '#' => [
 'prefix' => '#',
 'pattern' => '\\d+',
 'generator' => 'https://github.com/derafu/markdown/issues/%d',
],
],
],
];
$creator = new MarkdownCreator($options);
```

## 4. Template Metadata Block

Markdown templates allow the inclusion of a metadata block placed at the beginning of the file. These metadata entries are in YAML format, and all defined keys will be passed to the rendered layout and available as extracted variables in it.

Example:

```

__view_title: "Markdown Template Usage Guide"

```

With this metadata block, the `__view_title` index will be assigned to the layout data and can be used later as a variable within it.

## 5. Conclusion

The Markdown renderer used by this library, along with the preconfigured extensions, provides a powerful Markdown template rendering engine that allows you to create rich and dynamic documents effortlessly. With support for various extensions and advanced settings, you can fully customize the

Markdown experience in your application.

---

Last updated on 21/08/2026