

Handbuch – Joomla Content Plugin „Jlightbox“

Stand: 10.02.2026

Paket: Joomla.Plugin / content

Namespace: Joomla\Plugin\Content\Jlightbox\Extension

Autor: jumbo125

Lizenz: GNU GPL v2+

Deckblatt

Jlightbox – Plugin-Handbuch

Dieses Dokument beschreibt Aufbau, Konfiguration und Nutzung des Joomla Content Plugins **Jlightbox**. Das Plugin bindet die **GLightbox**-Library ein und stellt Optionen zur Steuerung der Lightbox-Aktivierung bereit (Wrapper-Klasse & Exclude-Selektoren).

Drittanbieter / Third-party: - GLightbox – MIT License, Copyright (c) 2018 Biati Digital

1. Quickguide

1.1 Installation

1. Plugin-Paket als Joomla-Erweiterung installieren (ZIP).
2. Plugin unter **Erweiterungen** → **Plugins** suchen: Content – Jlightbox.
3. Plugin aktivieren.

1.2 Minimale Konfiguration

- **Wrapper-Klasse** (Standard: `lightbox_wrapper`)
- **Exclude-Selektoren** (Standard: `.slides img,.delete,.edit-icon img,.no-lightbox`)

Empfehlung:

Bilder/Galerien, die in die Lightbox sollen, in einen gemeinsamen Wrapper setzen (z. B. `lightbox_wrapper`) und Ausnahmen mit `.no-lightbox` markieren.

1.3 Beispiel (HTML im Beitrag)

```
<div class="lightbox_wrapper">
  <a href="/images/full/1.jpg"></a>
  <a href="/images/full/2.jpg"></a>
</div>

<!-- ausgeschlossen -->

```

2. Überblick

Das Plugin basiert auf der GLightbox-Library (Biati Digital, MIT). Es nutzt Joomla 4/5 typische Muster: - Event-Subscriber (`SubscriberInterface`) - WebAssetManager + Registry-Datei (`joomla.asset.json`) - Übergabe von Parametern via `addScriptOptions()`

2.1 Ablauf in Kurzform

1. Joomla feuert `onBeforeCompileHead`.
 2. Plugin lädt Asset-Registry (`joomla.asset.json`).
 3. Plugin aktiviert Styles/Scripts via `useStyle()` / `useScript()`.
 4. Plugin übergibt Konfiguration an JS via `addScriptOptions('jlightbox', ...)`.
-

3. Installation & Aktivierung

3.1 Erwartete Verzeichnisstruktur (typisch)

```
plugins/content/jglightbox/  
  src/Extension/Jglightbox.php  
  services/provider.php  
  jglightbox.xml (Manifest, nicht im Snippet enthalten)  
  
media/plg_content_jglightbox/  
  joomla.asset.json  
  css/...  
  js/...
```

Wichtig:

Die Asset-Registry muss unter `media/plg_content_jglightbox/joomla.asset.json` existieren und die Handles müssen exakt stimmen: - `lightbox_style` - `lightbox_original` - `lightbox_original_inject`

4. Konfiguration (Parameter)

Parameter	Standard	Bedeutung
<code>wrapper_classes</code>	<code>lightbox_wrapper</code>	CSS-Klasse(n), die als „Container“ für Lightbox-Inhalte dienen. Das JS sucht typischerweise innerhalb dieses Wrappers nach passenden Links/Bildern.
<code>exclude_selectors</code>	<code>.slides</code> <code>img,.delete,.edit-icon</code> <code>img,.no-lightbox</code>	CSS-Selektoren, die von der Lightbox ausgenommen werden (z. B. Admin-Icons, Slider-Images, manuelle Ausnahmen).

Praxis-Tipp:

Wenn unerwartete Lightbox-Auslösungen passieren, `exclude_selectors` um Selektoren aus deinem Template erweitern (z. B. Slider/Galerie-Module).

5. Funktionsweise & Events

5.1 Event-Subscription

Das Plugin implementiert `SubscriberInterface` und registriert sich für: - `onBeforeCompileHead`

Dadurch werden Assets sehr früh in den Dokument-Head eingebunden, bevor Joomla den Head final kompiliert.

5.2 Was passiert in `onBeforeCompileHead`?

1. `WebAssetManager` beziehen:
 - `Factory::getApplication()->getDocument()->getWebAssetManager()`
2. Asset-Registry hinzufügen:
 - `addRegistryFile('media/plg_content_jglightbox/joomla.asset.json')`
3. Assets aktivieren:
 - `useStyle('lightbox_style')`
 - `useScript('lightbox_original')`
 - `useScript('lightbox_original_inject')`
4. Parameter lesen und via `ScriptOptions` an JS übergeben:
 - `addScriptOptions('jglightbox', ['wrapper' => ..., 'exclude' => ...])`

Hinweis:

In der aktuellen Implementierung werden Assets auf jeder Seite geladen, auf der das Event ausgeführt wird.

Wenn du das nur auf bestimmten Seiten möchtest, wäre eine Erweiterung sinnvoll (z. B. nur laden, wenn Content bestimmte Marker/Wrapper enthält).

6. Assets & WebAssetManager

Joomla verwaltet CSS/JS über den **WebAssetManager**. Die Asset-Definitionen liegen in einer Registry-Datei (`joomla.asset.json`). Das Plugin aktiviert dann die dort registrierten Handles.

6.1 Verwendete Handles

- `lightbox_style` – CSS für GLightbox
- `lightbox_original` – Original-GLightbox Script
- `lightbox_original_inject` – Zusatzscript (vermutlich Initialisierung + Lesen der ScriptOptions)

Wichtig für JS:

Die Konfiguration steht clientseitig typischerweise unter `Joomla.getOptions('jglightbox')` oder über den Joomla ScriptOptions-Mechanismus zur Verfügung (abhängig von deiner Initialisierungsdatei).

7. Verwendung im Content

7.1 Standard-Workflow

1. Wrapper um den Bereich setzen, der Lightbox-fähig sein soll: `lightbox_wrapper`.
2. Anklickbare Links auf große Bilder verwenden.
3. Unerwünschte Elemente per `exclude_selectors` oder `.no-lightbox` ausschließen.

7.2 Beispiel: Galerie

```
<div class="lightbox_wrapper">
  <figure>
    <a href="/images/large/a.jpg">
      
    </a>
    <figcaption>Bild A</figcaption>
  </figure>

  <figure>
    <a href="/images/large/b.jpg">
      
    </a>
    <figcaption>Bild B</figcaption>
  </figure>
</div>
```

8. Troubleshooting

8.1 Assets werden nicht geladen

- Prüfen, ob `media/plg_content_jglightbox/joomla.asset.json` vorhanden ist.
- Prüfen, ob die Handles exakt passen:
 - `lightbox_style`, `lightbox_original`, `lightbox_original_inject`
- Browser-Konsole prüfen: 404/500 auf CSS/JS?

8.2 Lightbox reagiert auf „falsche“ Bilder

- Wrapper enger setzen (nicht um den kompletten Artikel, sondern gezielt um die Galerie).
- `exclude_selectors` erweitern (Slider/Module/Editor-Icons).
- Einzelne Elemente mit `.no-lightbox` kennzeichnen.

8.3 Konflikte mit Template / JS

- Test ohne JS-Minifier/Optimizer (Cache leeren).
 - Prüfen, ob andere Lightbox-Skripte parallel geladen werden.
 - Reihenfolge/Abhängigkeiten in `joomla.asset.json` definieren.
-

9. Drittanbieter-Lizenzen

- **GLightbox:** MIT License, Copyright (c) 2018 Biati Digital
(laut Header-Kommentar im Code)

Stelle sicher, dass Lizenztexte der Drittanbieter-Bibliotheken in deinem Distributionspaket bzw. in der Dokumentation berücksichtigt werden.

10. Anhang (Code aus dem Projekt)

10.1 Plugin-Klasse Jglightbox (src/Extension/Jglightbox.php)

```
<?php
/**
 * @package      Joomla.Plugin
 * @subpackage    [PLUGIN_NAME]
 * @author        jumbo125
 * @copyright     Copyright (C) 2025 jumbo125. All rights reserved.
 * @license       GNU General Public License version 2 or later; see LICENSE.txt
 *
 * Fremde Skripte / Third-party libraries:
 * - Original library: MIT License Copyright (c) 2018 Biati Digital https://www.biati.digital
 */

// Sicherheitscheck
namespace Joomla\Plugin\Content\Jglightbox\Extension;

\defined('_JEXEC') or die;

use Joomla\CMS\Plugin\CMSPlugin;
use Joomla\CMS\Factory;
use Joomla\Event\Event;
use Joomla\Event\SubscriberInterface;
use Joomla\Event\DispatcherInterface;

class Jglightbox extends CMSPlugin implements SubscriberInterface
{
    public static function getSubscribedEvents(): array
    {
        return [
            'onBeforeCompileHead' => 'onBeforeCompileHead',
        ];
    }

    public function onBeforeCompileHead(Event $event): void
    {
        $wa = Factory::getApplication()->getDocument()->getWebAssetManager();
        $wa->getRegistry()->addRegistryFile('media/plg_content_jglightbox/joomla.asset.json');

        $wa->useStyle('gllightbox_style');
        $wa->useScript('gllightbox_original');
        $wa->useScript('gllightbox_original_inject');

        $wrapper = $this->params->get('wrapper_classes', 'lightbox_wrapper');
        $exclude = $this->params->get('exclude_selectors', '.slides img,.delete,.edit-icon
        ↪ img,.no-lightbox');
```

```

        Factory::getDocument()->addScriptOptions('jlightbox', [
            'wrapper' => $wrapper,
            'exclude' => $exclude,
        ]);
    }
}

```

10.2 Service Provider (services/provider.php)

```

<?php
/**
 * @package      Joomla.Plugin
 * @subpackage   [PLUGIN_NAME]
 * @author       jumbo125
 * @copyright    Copyright (C) 2025 jumbo125. All rights reserved.
 * @license      GNU General Public License version 2 or later; see LICENSE.txt
 *
 * Fremde Skripte / Third-party libraries:
 * - Original library: MIT License Copyright (c) 2018 Biati Digital https://www.biatidigital.com
 */

// Sicherheitscheck
defined('_JEXEC') or die;

use Joomla\CMS\Factory;
use Joomla\CMS\Extension\PluginInterface;
use Joomla\CMS\Plugin\PluginHelper;
use Joomla\Event\DispatcherInterface;
use Joomla\DI\Container;
use Joomla\DI\ServiceProviderInterface;
use Joomla\Plugin\Content\Jlightbox\Extension\Jlightbox;
use Joomla\Registry\Registry;

return new class implements ServiceProviderInterface {
    public function register(Container $container) {
        $container->set(
            PluginInterface::class,
            function (Container $container) {
                $config = (array) PluginHelper::getPlugin('content', 'jlightbox');
                $dispatcher = $container->get(DispatcherInterface::class);

                // Plugin erzeugen
                $plugin = new Jlightbox($dispatcher, $config);

                // Plugin-Parameter manuell setzen
                $plugin->params = new Registry($config['params'] ?? []);

                // Joomla Application zuweisen
                $plugin->setApplication(Factory::getApplication());

                return $plugin;
            }
        );
    }
};

```

Hinweis zur DI:

Hier wird das Plugin manuell instanziiert und Parameter werden explizit in ein Registry-Objekt überführt. Dadurch sind `$this->params->get(...)` Aufrufe in der Plugin-Klasse möglich.

Manual – Joomla Content Plugin “Jglightbox”

Status: 2026-02-10

Package: Joomla.Plugin / content

Namespace: Joomla\Plugin\Content\Jglightbox\Extension

Author: jumbo125

License: GNU GPL v2+

Cover page

Jglightbox – Plugin Manual

This document describes the structure, configuration, and usage of the Joomla content plugin **Jglightbox**. The plugin integrates the **GLightbox** library and provides options to control Lightbox activation (wrapper class & exclude selectors).

Third-party: - GLightbox – MIT License, Copyright (c) 2018 Biati Digital

1. Quick guide

1.1 Installation

1. Install the plugin package as a Joomla extension (ZIP).
2. Find the plugin under **Extensions** → **Plugins: Content - Jglightbox**.
3. Enable the plugin.

1.2 Minimal configuration

- **Wrapper class** (default: `lightbox_wrapper`)
- **Exclude selectors** (default: `.slides img,.delete,.edit-icon img,.no-lightbox`)

Recommendation:

Wrap images/galleries that should open in the lightbox in a common container (e.g. `lightbox_wrapper`) and mark exceptions with `.no-lightbox`.

1.3 Example (HTML inside an article)

```
<div class="lightbox_wrapper">
  <a href="/images/full/1.jpg"></a>
  <a href="/images/full/2.jpg"></a>
</div>

<!-- excluded -->

```

2. Overview

The plugin is based on the GLightbox library (Biati Digital, MIT). It uses typical Joomla 4/5 patterns: - Event subscriber (`SubscriberInterface`) - WebAssetManager + registry file (`joomla.asset.json`) - Passing parameters via `addScriptOptions()`

2.1 High-level flow

1. Joomla triggers `onBeforeCompileHead`.
 2. The plugin loads the asset registry (`joomla.asset.json`).
 3. The plugin enables styles/scripts via `useStyle()` / `useScript()`.
 4. The plugin passes configuration to JS via `addScriptOptions('jglightbox', ...)`.
-

3. Installation & activation

3.1 Expected directory structure (typical)

```
plugins/content/jglightbox/  
  src/Extension/Jglightbox.php  
  services/provider.php  
  jglightbox.xml (manifest, not included in the snippet)  
  
media/plg_content_jglightbox/  
  joomla.asset.json  
  css/...  
  js/...
```

Important:

The asset registry must exist at `media/plg_content_jglightbox/joomla.asset.json` and the handles must match exactly: - `lightbox_style` - `lightbox_original` - `lightbox_original_inject`

4. Configuration (parameters)

Parameter	Default	Meaning
<code>wrapper_classes</code>	<code>lightbox_wrapper</code>	CSS class(es) that define the “container” for lightbox content. The JS typically searches within this wrapper for matching links/images.
<code>exclude_selectors</code>	<code>.slides</code> <code>img,.delete,.edit-icon</code> <code>img,.no-lightbox</code>	CSS selectors that should be excluded from the lightbox (e.g. admin icons, slider images, manual exclusions).

Practical tip:

If lightbox triggers unexpectedly, extend `exclude_selectors` with selectors from your template (e.g. slider/gallery modules).

5. How it works & events

5.1 Event subscription

The plugin implements `SubscriberInterface` and subscribes to: - `onBeforeCompileHead`

This injects assets early into the document head, before Joomla compiles the final head output.

5.2 What happens in `onBeforeCompileHead`?

1. Get the `WebAssetManager`:
 - `Factory::getApplication()->getDocument()->getWebAssetManager()`
2. Add the asset registry:
 - `addRegistryFile('media/plg_content_jglightbox/joomla.asset.json')`
3. Enable assets:
 - `useStyle('lightbox_style')`
 - `useScript('lightbox_original')`
 - `useScript('lightbox_original_inject')`
4. Read parameters and pass them to JS via `ScriptOptions`:
 - `addScriptOptions('jglightbox', ['wrapper' => ..., 'exclude' => ...])`

Note:

In the current implementation, assets are loaded on every page where this event runs. If you only want this on certain pages, you could extend the plugin (e.g. load only when the content contains a specific marker/wrapper).

6. Assets & WebAssetManager

Joomla manages CSS/JS through the **WebAssetManager**. Asset definitions live in a registry file (`joomla.asset.json`). The plugin then enables the registered handles.

6.1 Handles used

- `lightbox_style` – CSS for GLightbox
- `lightbox_original` – original GLightbox script
- `lightbox_original_inject` – additional script (likely initialization + reading ScriptOptions)

Important for JS:

Client-side configuration is typically available via `Joomla.getOptions('jglightbox')` or Joomla's ScriptOptions mechanism (depending on your initialization file).

7. Usage in content

7.1 Standard workflow

1. Add a wrapper around the area that should be lightbox-enabled: `lightbox_wrapper`.
2. Use clickable links to the large images.
3. Exclude unwanted elements via `exclude_selectors` or `.no-lightbox`.

7.2 Example: gallery

```
<div class="lightbox_wrapper">
  <figure>
    <a href="/images/large/a.jpg">
      
    </a>
    <figcaption>Image A</figcaption>
  </figure>

  <figure>
    <a href="/images/large/b.jpg">
      
    </a>
    <figcaption>Image B</figcaption>
  </figure>
</div>
```

8. Troubleshooting

8.1 Assets are not loading

- Check that `media/plg_content_jglightbox/joomla.asset.json` exists.
- Check that the handles match exactly:
 - `lightbox_style`, `lightbox_original`, `lightbox_original_inject`
- Check the browser console: 404/500 for CSS/JS?

8.2 Lightbox reacts to the “wrong” images

- Use a tighter wrapper (not around the entire article, but specifically around the gallery).
- Extend `exclude_selectors` (slider/modules/editor icons).
- Mark individual elements with `.no-lightbox`.

8.3 Conflicts with template / JS

- Test without JS minifier/optimizer (clear caches).
 - Check whether other lightbox scripts are loaded in parallel.
 - Define order/dependencies in `joomla.asset.json`.
-

9. Third-party licenses

- **GLightbox**: MIT License, Copyright (c) 2018 Biati Digital (according to the header comment in the code)

Make sure that license texts of third-party libraries are considered in your distribution package and/or documentation.

10. Appendix (code from the project)

10.1 Plugin class Jlightbox (src/Extension/Jlightbox.php)

```
<?php
/**
 * @package      Joomla.Plugin
 * @subpackage    [PLUGIN_NAME]
 * @author        jumbo125
 * @copyright     Copyright (C) 2025 jumbo125. All rights reserved.
 * @license       GNU General Public License version 2 or later; see LICENSE.txt
 *
 * Third-party libraries:
 * - Original library: MIT License Copyright (c) 2018 Biati Digital https://www.biati.digital
 */

// Security check
namespace Joomla\Plugin\Content\Jlightbox\Extension;

\defined('_JEXEC') or die;

use Joomla\CMS\Plugin\CMSPlugin;
use Joomla\CMS\Factory;
use Joomla\Event\Event;
use Joomla\Event\SubscriberInterface;
use Joomla\Event\DispatcherInterface;

class Jlightbox extends CMSPlugin implements SubscriberInterface
{
    public static function getSubscribedEvents(): array
    {
        return [
            'onBeforeCompileHead' => 'onBeforeCompileHead',
        ];
    }

    public function onBeforeCompileHead(Event $event): void
    {
        $wa = Factory::getApplication()->getDocument()->getWebAssetManager();
        $wa->getRegistry()->addRegistryFile('media/plg_content_jlightbox/joomla.asset.json');

        $wa->useStyle('lightbox_style');
        $wa->useScript('lightbox_original');
        $wa->useScript('lightbox_original_inject');

        $wrapper = $this->params->get('wrapper_classes', 'lightbox_wrapper');
        $exclude = $this->params->get('exclude_selectors', '.slides img,.delete,.edit-icon
        ↪ img,.no-lightbox');
```

```

        Factory::getDocument()->addScriptOptions('jglightbox', [
            'wrapper' => $wrapper,
            'exclude' => $exclude,
        ]);
    }
}

```

10.2 Service provider (services/provider.php)

```

<?php
/**
 * @package      Joomla.Plugin
 * @subpackage   [PLUGIN_NAME]
 * @author       jumbo125
 * @copyright    Copyright (C) 2025 jumbo125. All rights reserved.
 * @license      GNU General Public License version 2 or later; see LICENSE.txt
 *
 * Third-party libraries:
 * - Original library: MIT License Copyright (c) 2018 Biati Digital https://www.biatidigital.com
 */

// Security check
defined('_JEXEC') or die;

use Joomla\CMS\Factory;
use Joomla\CMS\Extension\PluginInterface;
use Joomla\CMS\Plugin\PluginHelper;
use Joomla\Event\DispatcherInterface;
use Joomla\DI\Container;
use Joomla\DI\ServiceProviderInterface;
use Joomla\Plugin\Content\Jglightbox\Extension\Jglightbox;
use Joomla\Registry\Registry;

return new class implements ServiceProviderInterface {
    public function register(Container $container) {
        $container->set(
            PluginInterface::class,
            function (Container $container) {
                $config = (array) PluginHelper::getPlugin('content', 'jglightbox');
                $dispatcher = $container->get(DispatcherInterface::class);

                // Create plugin
                $plugin = new Jglightbox($dispatcher, $config);

                // Set plugin parameters manually
                $plugin->params = new Registry($config['params'] ?? []);

                // Assign Joomla application
                $plugin->setApplication(Factory::getApplication());

                return $plugin;
            }
        );
    }
};

```

Note on DI:

Here the plugin is instantiated manually and parameters are explicitly converted into a Registry object. This enables `$this->params->get(...)` calls in the plugin class.

End