

oEmbed Verwaltungs-Plugin (für WP) [Opus 4.6]

“ Nicht getestet!

Ein umfassendes WordPress-Plugin zur Verwaltung von oEmbed-Einbettungen. Hier ist der Implementierungsplan:

Implementierungsplan

1. **Haupt-Plugin-Datei** erstellen mit Plugin-Header, Aktivierung/Deaktivierung und Laden der Komponenten
2. **Admin-Einstellungsseite** mit Tabs für:
 - Provider-Verwaltung (aktivieren/deaktivieren einzelner oEmbed-Provider)
 - Globale Einstellungen (Cache-Dauer, max. Breite/Höhe, Lazy Loading)
 - Eigene Provider hinzufügen
 - Blockliste für URLs
3. **Frontend-Filter** die oEmbed-Ausgabe modifizieren (Wrapper, responsive Container, Lazy Loading)
4. **CSS & JS Assets** für Admin und Frontend
5. **Uninstall-Routine** zum sauberen Entfernen

Schritt 1: Haupt-Plugin-Datei

Diese Datei registriert das Plugin, lädt alle Komponenten und definiert Aktivierungs-/Deaktivierungs-Hooks.

```
<?php
/**
 * Plugin Name: oEmbed Manager
 * Plugin URI: https://example.com/oembed-manager
 * Description: Umfassende Verwaltung von oEmbed-Einbettungen – Provider
```

aktivieren/deaktivieren, eigene Provider hinzufügen, Darstellung anpassen, Blockliste und mehr.

```
* Version: 1.0.0
* Author: zcrow
* Author URI: https://example.com
* License: GPL-2.0+
* License URI: https://www.gnu.org/licenses/gpl-2.0.html
* Text Domain: oembed-manager
* Domain Path: /languages
*/
```

```
if ( ! defined( 'ABSPATH' ) ) {
    exit;
}
```

```
define( 'JEPP_OEM_VERSION', '1.0.0' );
define( 'JEPP_OEM_PLUGIN_DIR', plugin_dir_path( __FILE__ ) );
define( 'JEPP_OEM_PLUGIN_URL', plugin_dir_url( __FILE__ ) );
define( 'JEPP_OEM_PLUGIN_BASENAME', plugin_basename( __FILE__ ) );
```

```
/**
 * Standardoptionen bei Aktivierung setzen.
 */
```

```
function jepp_oem_activate() {
    $defaults = array(
        'disabled_providers' => array(),
        'custom_providers'   => array(),
        'blocked_urls'       => '',
        'max_width'          => 800,
        'max_height'         => 0,
        'cache_duration'     => 86400,
        'responsive'         => 1,
        'lazy_loading'       => 1,
        'wrapper_class'      => 'jepp-oembed-wrap',
        'disable_in_feed'    => 0,
        'disable_on_frontend' => 0,
        'strip_inline_styles' => 0,
    );
```

```
if ( false === get_option( 'jepp_oem_settings' ) ) {
```

```

        add_option( 'jepp_oem_settings', $defaults );
    }
}
register_activation_hook( __FILE__, 'jepp_oem_activate' );

/**
 * Aufräumen bei Deaktivierung (Cache leeren).
 */
function jepp_oem_deactivate() {
    global $wpdb;
    $wpdb->query(
        "DELETE FROM {$wpdb->postmeta} WHERE meta_key LIKE '_oembed_%'"
    );
    $wpdb->query(
        "DELETE FROM {$wpdb->options} WHERE option_name LIKE '_transient_oembed_%' OR
option_name LIKE '_transient_timeout_oembed_%'"
    );
}
register_deactivation_hook( __FILE__, 'jepp_oem_deactivate' );

// Komponenten laden
require_once JEPP_OEM_PLUGIN_DIR . 'includes/class-jepp-oem-settings.php';
require_once JEPP_OEM_PLUGIN_DIR . 'includes/class-jepp-oem-providers.php';
require_once JEPP_OEM_PLUGIN_DIR . 'includes/class-jepp-oem-frontend.php';

/**
 * Plugin initialisieren.
 */
function jepp_oem_init() {
    load_plugin_textdomain( 'oembed-manager', false, dirname( JEPP_OEM_PLUGIN_BASENAME ) .
'/languages' );

    new Jepp_OEM_Settings();
    new Jepp_OEM_Providers();
    new Jepp_OEM_Frontend();
}
add_action( 'plugins_loaded', 'jepp_oem_init' );

/**
 * Settings-Link in der Plugin-Liste.

```

```

*/
function jepp_oem_plugin_action_links( $links ) {
    $settings_link = '<a href="' . admin_url( 'options-general.php?page=jepp-oembed-manager' )
    . '">'
        . __( 'Einstellungen', 'oembed-manager' ) . '</a>';
    array_unshift( $links, $settings_link );
    return $links;
}
add_filter( 'plugin_action_links_' . JEPP_OEM_PLUGIN_BASENAME, 'jepp_oem_plugin_action_links'
);

```

Schritt 2: Provider-Verwaltung

Diese Klasse kümmert sich um das Deaktivieren vorhandener und Registrieren eigener oEmbed-Provider.

```

<?php
if ( ! defined( 'ABSPATH' ) ) {
    exit;
}

/**
 * Verwaltet oEmbed-Provider: deaktivieren, eigene hinzufügen, URLs blockieren.
 */
class Jepp_OEM_Providers {

    /** @var array */
    private $settings;

    public function __construct() {
        $this->settings = get_option( 'jepp_oem_settings', array() );

        // Provider deaktivieren
        add_filter( 'oembed_providers', array( $this, 'filter_providers' ), 99 );

        // Eigene Provider registrieren
        add_action( 'init', array( $this, 'register_custom_providers' ), 20 );

        // URLs blockieren

```

```

        add_filter( 'pre_oembed_result', array( $this, 'block_urls' ), 10, 3 );
    }

    /**
     * Deaktivierte Provider aus der Liste entfernen.
     *
     * @param array $providers
     * @return array
     */
    public function filter_providers( $providers ) {
        $disabled = isset( $this->settings['disabled_providers'] ) ? $this->settings['disabled_providers'] : array();

        if ( empty( $disabled ) ) {
            return $providers;
        }

        foreach ( $providers as $pattern => $data ) {
            // $data[0] = endpoint URL, $data[1] = regex flag
            $endpoint = is_array( $data ) ? $data[0] : $data;

            foreach ( $disabled as $disabled_key ) {
                if ( stripos( $pattern, $disabled_key ) !== false || stripos( $endpoint, $disabled_key ) !== false ) {
                    unset( $providers[ $pattern ] );
                }
            }
        }

        return $providers;
    }

    /**
     * Eigene Provider registrieren.
     */
    public function register_custom_providers() {
        $custom = isset( $this->settings['custom_providers'] ) ? $this->settings['custom_providers'] : array();

        if ( empty( $custom ) ) {

```

```

        return;
    }

    foreach ( $custom as $provider ) {
        if ( empty( $provider['pattern'] ) || empty( $provider['endpoint'] ) ) {
            continue;
        }
        $regex = ! empty( $provider['regex'] ) ? true : false;
        wp_oembed_add_provider( $provider['pattern'], $provider['endpoint'], $regex );
    }
}

/**
 * Bestimmte URLs blockieren.
 *
 * @param null|string $result
 * @param string      $url
 * @param array       $args
 * @return null|false|string
 */
public function block_urls( $result, $url, $args ) {
    $blocked_raw = isset( $this->settings['blocked_urls'] ) ? $this->settings['blocked_urls'] : '';

    if ( empty( $blocked_raw ) ) {
        return $result;
    }

    $blocked_lines = array_filter( array_map( 'trim', explode( "\n", $blocked_raw ) ) );

    foreach ( $blocked_lines as $blocked ) {
        if ( empty( $blocked ) ) {
            continue;
        }
        // Unterstützung für Wildcards: * → .*
        $pattern = '#' . str_replace( '\*', '.*', preg_quote( $blocked, '#' ) ) . '#i';
        if ( preg_match( $pattern, $url ) ) {
            return false; // Blockiert
        }
    }
}

```

```

    return $result;
}

/**
 * Gibt alle in WordPress registrierten Standard-Provider zurück.
 *
 * @return array Array mit Schlüssel = Pattern, Wert = Endpoint
 */
public static function get_default_providers() {
    require_once ABSPATH . WPINC . '/class-wp-oembed.php';
    $oembed = _wp_oembed_get_object();
    $providers = $oembed->providers;

    $list = array();
    foreach ( $providers as $pattern => $data ) {
        $endpoint = is_array( $data ) ? $data[0] : $data;
        // Lesbaren Namen aus dem Pattern oder der Endpoint-URL extrahieren
        $name = self::extract_provider_name( $pattern, $endpoint );
        $list[ $pattern ] = array(
            'name' => $name,
            'endpoint' => $endpoint,
            'pattern' => $pattern,
        );
    }

    // Alphabetisch nach Name sortieren
    uasort( $list, function ( $a, $b ) {
        return strcmp( $a['name'], $b['name'] );
    });

    return $list;
}

/**
 * Versucht einen lesbaren Provider-Namen zu extrahieren.
 *
 * @param string $pattern
 * @param string $endpoint
 * @return string

```

```

*/
private static function extract_provider_name( $pattern, $endpoint ) {
    // Aus Endpoint-URL den Hostnamen holen
    $host = wp_parse_url( $endpoint, PHP_URL_HOST );
    if ( $host ) {
        $host = preg_replace( '#^(www\.|publish\.)#i', '', $host );
        $parts = explode( '.', $host );
        if ( count( $parts ) >= 2 ) {
            return ucfirst( $parts[ count( $parts ) - 2 ] );
        }
        return ucfirst( $host );
    }

    // Fallback: Pattern bereinigen
    $clean = preg_replace( '#https?://|[\^a-zA-Z0-9\.]#', '', $pattern );
    $clean = preg_replace( '#^(www\.)#', '', $clean );
    $parts = explode( '.', $clean );
    return ucfirst( $parts[0] ?: 'Unknown' );
}
}

```

Schritt 3: Admin-Einstellungsseite

Die Settings-Klasse baut eine übersichtliche Admin-Seite mit Tabs.

```

<?php
if ( ! defined( 'ABSPATH' ) ) {
    exit;
}

/**
 * Admin-Einstellungsseite für den oEmbed Manager.
 */
class Jepp_OEM_Settings {

    /** @var string */
    private $option_name = 'jepp_oem_settings';
}

```

```

/** @var array */
private $settings;

public function __construct() {
    $this->settings = get_option( $this->option_name, array() );

    add_action( 'admin_menu', array( $this, 'add_menu_page' ) );
    add_action( 'admin_init', array( $this, 'register_settings' ) );
    add_action( 'admin_enqueue_scripts', array( $this, 'enqueue_assets' ) );
}

/**
 * Menüeintrag unter "Einstellungen".
 */
public function add_menu_page() {
    add_options_page(
        __( 'oEmbed Manager', 'oembed-manager' ),
        __( 'oEmbed Manager', 'oembed-manager' ),
        'manage_options',
        'jepp-oembed-manager',
        array( $this, 'render_page' )
    );
}

/**
 * Admin-Assets laden.
 */
public function enqueue_assets( $hook ) {
    if ( 'settings_page_jepp-oembed-manager' !== $hook ) {
        return;
    }
    wp_enqueue_style(
        'jepp-oem-admin',
        JEPP_OEM_PLUGIN_URL . 'assets/css/admin.css',
        array(),
        JEPP_OEM_VERSION
    );
    wp_enqueue_script(
        'jepp-oem-admin',
        JEPP_OEM_PLUGIN_URL . 'assets/js/admin.js',

```

```

        array( 'jquery' ),
        JEPP_OEM_VERSION,
        true
    );
}

/**
 * Settings API registrieren.
 */
public function register_settings() {
    register_setting( 'jepp_oem_group', $this->option_name, array(
        'type'           => 'array',
        'sanitize_callback' => array( $this, 'sanitize_settings' ),
    ) );
}

/**
 * Eingaben bereinigen.
 */
public function sanitize_settings( $input ) {
    $clean = array();

    // Deaktivierte Provider
    $clean['disabled_providers'] = array();
    if ( ! empty( $input['disabled_providers'] ) && is_array( $input['disabled_providers']
) ) {
        $clean['disabled_providers'] = array_map( 'sanitize_text_field',
$input['disabled_providers'] );
    }

    // Eigene Provider
    $clean['custom_providers'] = array();
    if ( ! empty( $input['custom_providers'] ) && is_array( $input['custom_providers'] ) )
{
        foreach ( $input['custom_providers'] as $cp ) {
            if ( empty( $cp['pattern'] ) || empty( $cp['endpoint'] ) ) {
                continue;
            }
            $clean['custom_providers'][] = array(
                'pattern' => sanitize_text_field( $cp['pattern'] ),

```

```

        'endpoint' => esc_url_raw( $cp['endpoint'] ),
        'regex'    => ! empty( $cp['regex'] ) ? 1 : 0,
    );
}

// Blockliste
$clean['blocked_urls'] = '';
if ( ! empty( $input['blocked_urls'] ) ) {
    $clean['blocked_urls'] = sanitize_textarea_field( $input['blocked_urls'] );
}

// Numerische Werte
$clean['max_width']      = isset( $input['max_width'] ) ? absint( $input['max_width']
) : 800;
$clean['max_height']    = isset( $input['max_height'] ) ? absint(
$input['max_height'] ) : 0;
$clean['cache_duration'] = isset( $input['cache_duration'] ) ? absint(
$input['cache_duration'] ) : 86400;

// Checkboxes
$clean['responsive']      = ! empty( $input['responsive'] ) ? 1 : 0;
$clean['lazy_loading']    = ! empty( $input['lazy_loading'] ) ? 1 : 0;
$clean['disable_in_feed'] = ! empty( $input['disable_in_feed'] ) ? 1 : 0;
$clean['disable_on_frontend'] = ! empty( $input['disable_on_frontend'] ) ? 1 : 0;
$clean['strip_inline_styles'] = ! empty( $input['strip_inline_styles'] ) ? 1 : 0;

// Wrapper-Klasse
$clean['wrapper_class'] = ! empty( $input['wrapper_class'] )
    ? sanitize_html_class( $input['wrapper_class'] )
    : 'jepp-oembed-wrap';

// oEmbed-Cache leeren
$this->flush_oembed_cache();

return $clean;
}

/**
 * oEmbed-Cache leeren.

```

```

*/
private function flush_oembed_cache() {
    global $wpdb;
    $wpdb->query(
        "DELETE FROM {$wpdb->postmeta} WHERE meta_key LIKE '_oembed_%'"
    );
    $wpdb->query(
        "DELETE FROM {$wpdb->options} WHERE option_name LIKE '_transient_oembed_%' OR
option_name LIKE '_transient_timeout_oembed_%'"
    );
}

/**
 * Einstellungsseite rendern.
 */
public function render_page() {
    if ( ! current_user_can( 'manage_options' ) ) {
        return;
    }

    $active_tab = isset( $_GET['tab'] ) ? sanitize_key( $_GET['tab'] ) : 'providers';
    $tabs = array(
        'providers' => __( 'Provider', 'oembed-manager' ),
        'display'   => __( 'Darstellung', 'oembed-manager' ),
        'custom'    => __( 'Eigene Provider', 'oembed-manager' ),
        'blocklist' => __( 'Blockliste', 'oembed-manager' ),
    );
    ?>
    <div class="wrap jepp-oem-wrap">
        <h1><?php esc_html_e( 'oEmbed Manager', 'oembed-manager' ); ?></h1>

        <nav class="nav-tab-wrapper jepp-oem-tabs">
            <?php foreach ( $tabs as $slug => $label ) : ?>
                <a href="<?php echo esc_url( add_query_arg( array( 'page' => 'jepp-oembed-
manager', 'tab' => $slug ), admin_url( 'options-general.php' ) ) ); ?>"
                    class="nav-tab <?php echo $active_tab === $slug ? 'nav-tab-active' :
''; ?>">

                    <?php echo esc_html( $label ); ?>
                </a>
            <?php endforeach; ?>

```

```

</nav>

<form method="post" action="options.php">
    <?php
        settings_fields( 'jepp_oem_group' );

        switch ( $active_tab ) {
            case 'display':
                $this->render_tab_display();
                break;
            case 'custom':
                $this->render_tab_custom();
                break;
            case 'blocklist':
                $this->render_tab_blocklist();
                break;
            default:
                $this->render_tab_providers();
                break;
        }

        // Hidden Fields für die anderen Tabs mitgeben, damit sie nicht überschrieben
werden
        $this->render_hidden_fields( $active_tab );

        submit_button( __( 'Einstellungen speichern', 'oembed-manager' ) );
    ?>
</form>
</div>
<?php
}

/**
 * Tab: Provider verwalten.
 */
private function render_tab_providers() {
    $providers = Jepp_OEM_Providers::get_default_providers();
    $disabled = isset( $this->settings['disabled_providers'] ) ? $this-
>settings['disabled_providers'] : array();
    ?>

```

```

<div class="jepp-oem-section">
    <h2><?php esc_html_e( 'oEmbed-Provider aktivieren / deaktivieren', 'oembed-
manager' ); ?></h2>
    <p class="description"><?php esc_html_e( 'Deaktivierte Provider werden nicht mehr
für die automatische Einbettung verwendet.', 'oembed-manager' ); ?></p>

    <div class="jepp-oem-provider-actions" style="margin: 10px 0;">
        <button type="button" class="button jepp-oem-select-all"><?php esc_html_e(
'Alle aktivieren', 'oembed-manager' ); ?></button>
        <button type="button" class="button jepp-oem-deselect-all"><?php esc_html_e(
'Alle deaktivieren', 'oembed-manager' ); ?></button>
    </div>

    <table class="widefat jepp-oem-provider-table">
        <thead>
            <tr>
                <th style="width:50px;"><?php esc_html_e( 'Aktiv', 'oembed-manager' );
?></th>

                <th><?php esc_html_e( 'Provider', 'oembed-manager' ); ?></th>
                <th><?php esc_html_e( 'URL-Pattern', 'oembed-manager' ); ?></th>
                <th><?php esc_html_e( 'Endpoint', 'oembed-manager' ); ?></th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ( $providers as $pattern => $info ) :
                $key          = md5( $pattern );
                $is_disabled = in_array( $key, $disabled, true );
                ?>
                <tr>
                    <td>
                        <input type="checkbox"
                            class="jepp-oem-provider-checkbox"
                            name="<?php echo esc_attr( $this->option_name );
?>[disabled_providers][]"
                            value="<?php echo esc_attr( $key ); ?>"
                            <?php checked( false, $is_disabled ); ?>
                            data-inverted="1"
                        >
                    </td>
                    <td><strong><?php echo esc_html( $info['name'] ); ?></strong></td>
            </tbody>
    </table>

```

```

        <td><code><?php echo esc_html( $pattern ); ?></code></td>
        <td><code><?php echo esc_html( $info['endpoint'] ); ?></code></td>
    </tr>
    <?php endforeach; ?>
</tbody>
</table>
</div>
<?php
}

/**
 * Tab: Darstellungsoptionen.
 */
private function render_tab_display() {
    $s = $this->settings;
    ?>
    <div class="jepp-oem-section">
        <h2><?php esc_html_e( 'Darstellungseinstellungen', 'oembed-manager' ); ?></h2>
        <table class="form-table">
            <tr>
                <th scope="row"><?php esc_html_e( 'Maximale Breite (px)', 'oembed-manager'
); ?></th>
                <td>
                    <input type="number" name="<?php echo esc_attr( $this->option_name );
?>[max_width]"
                        value="<?php echo esc_attr( $s['max_width'] ?? 800 ); ?>"
min="0" step="1" class="small-text">
                    <p class="description"><?php esc_html_e( '0 = keine Beschränkung',
'oembed-manager' ); ?></p>
                </td>
            </tr>
            <tr>
                <th scope="row"><?php esc_html_e( 'Maximale Höhe (px)', 'oembed-manager'
); ?></th>
                <td>
                    <input type="number" name="<?php echo esc_attr( $this->option_name );
?>[max_height]"
                        value="<?php echo esc_attr( $s['max_height'] ?? 0 ); ?>"
min="0" step="1" class="small-text">
                    <p class="description"><?php esc_html_e( '0 = keine Beschränkung',

```

```

'embed-manager' ); ?></p>
        </td>
    </tr>
    <tr>
        <th scope="row"><?php esc_html_e( 'Cache-Dauer (Sekunden)', 'embed-
manager' ); ?></th>
        <td>
            <input type="number" name="<?php echo esc_attr( $this->option_name );
?>[cache_duration]"
                value="<?php echo esc_attr( $s['cache_duration'] ?? 86400 );
?>" min="0" step="1" class="regular-text">
                <p class="description"><?php esc_html_e( '86400 = 1 Tag. 0 = Cache
deaktiviert.', 'embed-manager' ); ?></p>
            </td>
        </tr>
    <tr>
        <th scope="row"><?php esc_html_e( 'Wrapper CSS-Klasse', 'embed-manager'
); ?></th>
        <td>
            <input type="text" name="<?php echo esc_attr( $this->option_name );
?>[wrapper_class]"
                value="<?php echo esc_attr( $s['wrapper_class'] ?? 'jepp-
embed-wrap' ); ?>" class="regular-text">
            </td>
        </tr>
    <tr>
        <th scope="row"><?php esc_html_e( 'Optionen', 'embed-manager' ); ?></th>
        <td>
            <fieldset>
                <label>
                    <input type="checkbox" name="<?php echo esc_attr( $this-
>option_name ); ?>[responsive]" value="1"
                        <?php checked( 1, $s['responsive'] ?? 1 ); ?>>
                    <?php esc_html_e( 'Responsive Container (16:9) für Videos',
'embed-manager' ); ?>
                </label><br>
                <label>
                    <input type="checkbox" name="<?php echo esc_attr( $this-
>option_name ); ?>[lazy_loading]" value="1"
                        <?php checked( 1, $s['lazy_loading'] ?? 1 ); ?>>

```

```

                <?php esc_html_e( 'Lazy Loading für iframes aktivieren',
'embed-manager' ); ?>
                </label><br>
                <label>
                    <input type="checkbox" name="<?php echo esc_attr( $this-
>option_name ); ?>[strip_inline_styles]" value="1"
                        <?php checked( 1, $s['strip_inline_styles'] ?? 0 ); ?>>
                    <?php esc_html_e( 'Inline-Styles aus oEmbed-HTML entfernen',
'embed-manager' ); ?>
                </label><br>
                <label>
                    <input type="checkbox" name="<?php echo esc_attr( $this-
>option_name ); ?>[disable_in_feed]" value="1"
                        <?php checked( 1, $s['disable_in_feed'] ?? 0 ); ?>>
                    <?php esc_html_e( 'oEmbed im RSS-Feed deaktivieren (nur Link
anzeigen)', 'embed-manager' ); ?>
                </label><br>
                <label>
                    <input type="checkbox" name="<?php echo esc_attr( $this-
>option_name ); ?>[disable_on_frontend]" value="1"
                        <?php checked( 1, $s['disable_on_frontend'] ?? 0 ); ?>>
                    <?php esc_html_e( 'oEmbed komplett auf dem Frontend
deaktivieren', 'embed-manager' ); ?>
                </label>
            </fieldset>
        </td>
    </tr>
</table>
</div>
<?php
}

/**
 * Tab: Eigene Provider.
 */
private function render_tab_custom() {
    $custom = isset( $this->settings['custom_providers'] ) ? $this-
>settings['custom_providers'] : array();
    ?>
    <div class="jepp-oem-section">

```

```

<h2><?php esc_html_e( 'Eigene oEmbed-Provider', 'oembed-manager' ); ?></h2>
<p class="description"><?php esc_html_e( 'Hier können Sie eigene oEmbed-Provider
hinzufügen, z.B. für interne Dienste.', 'oembed-manager' ); ?></p>

<table class="widefat jepp-oem-custom-table" id="jepp-oem-custom-table">
  <thead>
    <tr>
      <th><?php esc_html_e( 'URL-Pattern', 'oembed-manager' ); ?></th>
      <th><?php esc_html_e( 'Endpoint-URL', 'oembed-manager' ); ?></th>
      <th style="width:60px;"><?php esc_html_e( 'Regex', 'oembed-manager' );
?></th>
      <th style="width:60px;"><?php esc_html_e( 'Aktion', 'oembed-manager'
); ?></th>
    </tr>
  </thead>
  <tbody>
    <?php if ( ! empty( $custom ) ) : ?>
      <?php foreach ( $custom as $i => $cp ) : ?>
        <tr class="jepp-oem-custom-row">
          <td>
            <input type="text" class="regular-text"
              name="<?php echo esc_attr( $this->option_name );
?>[custom_providers][<?php echo $i; ?>][pattern]"
              value="<?php echo esc_attr( $cp['pattern'] ); ?>"
              placeholder="https://example.com/*">
          </td>
          <td>
            <input type="url" class="regular-text"
              name="<?php echo esc_attr( $this->option_name );
?>[custom_providers][<?php echo $i; ?>][endpoint]"
              value="<?php echo esc_attr( $cp['endpoint'] ); ?>"
              placeholder="https://example.com/oembed">
          </td>
          <td>
            <input type="checkbox"
              name="<?php echo esc_attr( $this->option_name );
?>[custom_providers][<?php echo $i; ?>][regex]"
              value="1" <?php checked( 1, $cp['regex'] ?? 0 );
?>>
          </td>
        </tr>
      </tbody>
    </table>

```

```

                <td>
                    <button type="button" class="button jepp-oem-remove-
row">&times;</button>
                </td>
            </tr>
        <?php endforeach; ?>
    <?php endif; ?>
</tbody>
</table>

<p>
    <button type="button" class="button button-secondary" id="jepp-oem-add-
provider">
        <?php esc_html_e( '+ Provider hinzufügen', 'oembed-manager' ); ?>
    </button>
</p>
</div>
<?php
}

/**
 * Tab: Blockliste.
 */
private function render_tab_blocklist() {
    $blocked = isset( $this->settings['blocked_urls'] ) ? $this->settings['blocked_urls']
: '';
    ?>
    <div class="jepp-oem-section">
        <h2><?php esc_html_e( 'URL-Blockliste', 'oembed-manager' ); ?></h2>
        <p class="description">
            <?php esc_html_e( 'URLs oder URL-Muster (eine pro Zeile), die nicht
eingebettet werden sollen. Wildcards (*) sind erlaubt.', 'oembed-manager' ); ?>
        </p>
        <textarea name="<?php echo esc_attr( $this->option_name ); ?>[blocked_urls]"
            rows="10" class="large-text code"
            placeholder="https://example.com/private/*&#10;https://badsite.com/*"
            ><?php echo esc_textarea( $blocked ); ?></textarea>
        </div>
    <?php
}

```

```

/**
 * Versteckte Felder rendern, damit andere Tabs nicht verloren gehen.
 *
 * @param string $active_tab
 */
private function render_hidden_fields( $active_tab ) {
    $s = $this->settings;

    if ( 'providers' !== $active_tab ) {
        // Deaktivierte Provider als hidden fields
        $disabled = isset( $s['disabled_providers'] ) ? $s['disabled_providers'] :
array();
        foreach ( $disabled as $d ) {
            echo '<input type="hidden" name="' . esc_attr( $this->option_name ) .
'[disabled_providers][ ]' value="" . esc_attr( $d ) . "'>';
        }
    }

    if ( 'display' !== $active_tab ) {
        $fields = array( 'max_width', 'max_height', 'cache_duration', 'wrapper_class' );
        foreach ( $fields as $f ) {
            $val = isset( $s[ $f ] ) ? $s[ $f ] : '';
            echo '<input type="hidden" name="' . esc_attr( $this->option_name ) . '['
. ']' value="" . esc_attr( $val ) . "'>';
        }
        $checkboxes = array( 'responsive', 'lazy_loading', 'strip_inline_styles',
'disable_in_feed', 'disable_on_frontend' );
        foreach ( $checkboxes as $cb ) {
            if ( ! empty( $s[ $cb ] ) ) {
                echo '<input type="hidden" name="' . esc_attr( $this->option_name ) . '['
. $cb . ']' value="1">';
            }
        }
    }

    if ( 'custom' !== $active_tab ) {
        $custom = isset( $s['custom_providers'] ) ? $s['custom_providers'] : array();
        foreach ( $custom as $i => $cp ) {
            echo '<input type="hidden" name="' . esc_attr( $this->option_name ) .

```

```

'[custom_providers][' . $i . '][pattern]" value="" . esc_attr( $cp['pattern'] ) . '">';
    echo '<input type="hidden" name="" . esc_attr( $this->option_name ) .
'[custom_providers][' . $i . '][endpoint]" value="" . esc_attr( $cp['endpoint'] ) . '">';
    if ( ! empty( $cp['regex'] ) ) {
        echo '<input type="hidden" name="" . esc_attr( $this->option_name ) .
'[custom_providers][' . $i . '][regex]" value="1">';
    }
}
}

if ( 'blocklist' !== $active_tab ) {
    $blocked = isset( $s['blocked_urls'] ) ? $s['blocked_urls'] : '';
    echo '<input type="hidden" name="" . esc_attr( $this->option_name ) .
'[blocked_urls]" value="" . esc_attr( $blocked ) . '">';
}
}
}
}

```

Schritt 4: Frontend-Filter

Diese Klasse modifiziert die oEmbed-Ausgabe im Frontend (Wrapper, responsive, Lazy Loading etc.).

```

<?php
if ( ! defined( 'ABSPATH' ) ) {
    exit;
}

/**
 * Modifiziert die oEmbed-Ausgabe im Frontend.
 */
class Jepp_OEM_Frontend {

    /** @var array */
    private $settings;

    public function __construct() {
        $this->settings = get_option( 'jepp_oem_settings', array() );
    }
}

```

```

// oEmbed komplett deaktivieren
if ( ! empty( $this->settings['disable_on_frontend'] ) && ! is_admin() ) {
    add_action( 'init', array( $this, 'disable_oembed_completely' ) );
    return;
}

// Max-Width/Height setzen
add_filter( 'embed_defaults', array( $this, 'set_embed_defaults' ) );

// Cache-Dauer setzen
add_filter( 'oembed_ttl', array( $this, 'set_cache_ttl' ), 10, 4 );

// HTML-Ausgabe modifizieren
add_filter( 'embed_oembed_html', array( $this, 'modify_output' ), 99, 4 );

// Im Feed deaktivieren
if ( ! empty( $this->settings['disable_in_feed'] ) ) {
    add_filter( 'embed_oembed_html', array( $this, 'disable_in_feed' ), 100, 4 );
}

// Frontend-CSS laden
add_action( 'wp_enqueue_scripts', array( $this, 'enqueue_frontend_styles' ) );
}

/**
 * oEmbed komplett deaktivieren.
 */
public function disable_oembed_completely() {
    // Auto-Discovery deaktivieren
    remove_action( 'wp_head', 'wp_oembed_add_discovery_links' );
    remove_action( 'wp_head', 'wp_oembed_add_host_js' );

    // oEmbed REST-API Endpoint deaktivieren
    remove_action( 'rest_api_init', 'wp_oembed_register_route' );

    // oEmbed-Filter entfernen
    remove_filter( 'pre_oembed_result', 'wp_filter_pre_oembed_result', 10 );

    // Auto-Embeds deaktivieren
    remove_filter( 'the_content', array( $GLOBALS['wp_embed'], 'autoembed' ), 8 );
}

```

```

}

/**
 * Standard-Embed-Dimensionen setzen.
 *
 * @param array $defaults
 * @return array
 */
public function set_embed_defaults( $defaults ) {
    $max_w = $this->settings['max_width'] ?? 800;
    $max_h = $this->settings['max_height'] ?? 0;

    if ( $max_w > 0 ) {
        $defaults['width'] = $max_w;
    }
    if ( $max_h > 0 ) {
        $defaults['height'] = $max_h;
    }

    return $defaults;
}

/**
 * Cache-TTL setzen.
 *
 * @param int    $ttl
 * @param string $url
 * @param array  $attr
 * @param int    $post_id
 * @return int
 */
public function set_cache_ttl( $ttl, $url, $attr, $post_id ) {
    $custom_ttl = $this->settings['cache_duration'] ?? 86400;
    return (int) $custom_ttl;
}

/**
 * oEmbed-HTML-Ausgabe modifizieren.
 *
 * @param string $html

```

```

* @param string $url
* @param array  $attr
* @param int    $post_id
* @return string
*/
public function modify_output( $html, $url, $attr, $post_id ) {
    if ( empty( $html ) ) {
        return $html;
    }

    // Inline-Styles entfernen
    if ( ! empty( $this->settings['strip_inline_styles'] ) ) {
        $html = preg_replace( '/\s+style\s*=\s*"[^"]*/i', '', $html );
        $html = preg_replace( "/\s+style\s*=\s*'[^']*'/i", '', $html );
    }

    // Lazy Loading für iframes
    if ( ! empty( $this->settings['lazy_loading'] ) ) {
        $html = $this->add_lazy_loading( $html );
    }

    // Wrapper-Container
    $wrapper_class = $this->settings['wrapper_class'] ?? 'jepp-oembed-wrap';
    $is_video      = $this->is_video_embed( $html, $url );
    $responsive    = ! empty( $this->settings['responsive'] ) && $is_video;

    $classes = esc_attr( $wrapper_class );
    if ( $responsive ) {
        $classes .= ' jepp-oembed-responsive';
    }

    $html = '<div class="' . $classes . '">' . $html . '</div>';

    return $html;
}

/**
 * Lazy Loading Attribut zu iframes hinzufügen.
 *
 * @param string $html

```

```

* @return string
*/
private function add_lazy_loading( $html ) {
    // Nur wenn ein iframe vorhanden ist und noch kein loading-Attribut gesetzt
    if ( stripos( $html, '<iframe' ) !== false && stripos( $html, 'loading=' ) === false )
{
        $html = str_ireplace( '<iframe', '<iframe loading="lazy"', $html );
    }
    return $html;
}

/**
 * Prüft ob es sich um ein Video-Embed handelt.
 *
 * @param string $html
 * @param string $url
 * @return bool
 */
private function is_video_embed( $html, $url ) {
    $video_hosts = array(
        'youtube.com', 'youtu.be', 'vimeo.com', 'dailymotion.com',
        'dai.ly', 'tiktok.com', 'twitch.tv', 'facebook.com/watch',
        'wistia.com', 'videopress.com',
    );

    foreach ( $video_hosts as $host ) {
        if ( stripos( $url, $host ) !== false ) {
            return true;
        }
    }

    // Fallback: iframe mit typischen Video-Attributen prüfen
    if ( preg_match( '/<iframe[^\>]+src=["\'](?:video|embed|player)/i', $html ) ) {
        return true;
    }

    return false;
}

/**

```

```

* Im Feed nur den Link anzeigen.
*
* @param string $html
* @param string $url
* @param array $attr
* @param int $post_id
* @return string
*/
public function disable_in_feed( $html, $url, $attr, $post_id ) {
    if ( is_feed() ) {
        return '<a href="' . esc_url( $url ) . '">' . esc_html( $url ) . '</a>';
    }
    return $html;
}

/**
* Frontend-Styles laden.
*/
public function enqueue_frontend_styles() {
    wp_enqueue_style(
        'jepp-oem-frontend',
        JEPP_OEM_PLUGIN_URL . 'assets/css/frontend.css',
        array(),
        JEPP_OEM_VERSION
    );
}
}

```

Schritt 5: Admin-CSS

```

/* oEmbed Manager – Admin Styles */

.jepp-oem-wrap {
    max-width: 1100px;
}

.jepp-oem-tabs {
    margin-bottom: 20px;
}

```

```
}

.jepp-oem-section {
  background: #fff;
  border: 1px solid #ccd0d4;
  border-radius: 4px;
  padding: 20px 24px;
  margin-top: 15px;
}

.jepp-oem-section h2 {
  margin-top: 0;
  padding-top: 0;
  border-bottom: 1px solid #eee;
  padding-bottom: 10px;
}

/* Provider-Tabelle */
.jepp-oem-provider-table {
  margin-top: 10px;
}

.jepp-oem-provider-table td,
.jepp-oem-provider-table th {
  vertical-align: middle;
  padding: 10px 12px;
}

.jepp-oem-provider-table code {
  font-size: 12px;
  background: #f0f0f1;
  padding: 2px 6px;
  border-radius: 3px;
  word-break: break-all;
}

.jepp-oem-provider-table tbody tr:nth-child(even) {
  background: #f9f9f9;
}
```

```
.jepp-oem-provider-table tbody tr:hover {
    background: #f0f6fc;
}

/* Eigene Provider */
.jepp-oem-custom-table td {
    vertical-align: middle;
    padding: 8px 10px;
}

.jepp-oem-custom-table input[type="text"],
.jepp-oem-custom-table input[type="url"] {
    width: 100%;
}

.jepp-oem-remove-row {
    color: #a00 !important;
    border-color: #a00 !important;
    font-weight: bold;
    font-size: 16px;
    line-height: 1;
    padding: 2px 8px !important;
}

.jepp-oem-remove-row:hover {
    color: #dc3232 !important;
    border-color: #dc3232 !important;
}

/* Provider Actions */
.jepp-oem-provider-actions .button {
    margin-right: 8px;
}
```

Schritt 6: Frontend-CSS

```
/* oEmbed Manager – Frontend Styles */
```

```
.jepp-oembed-wrap {
    max-width: 100%;
    margin: 1.5em 0;
    clear: both;
}

/* Responsive Video Container (16:9) */
.jepp-oembed-responsive {
    position: relative;
    padding-bottom: 56.25%; /* 16:9 */
    height: 0;
    overflow: hidden;
}

.jepp-oembed-responsive iframe,
.jepp-oembed-responsive object,
.jepp-oembed-responsive embed,
.jepp-oembed-responsive video {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    border: 0;
}
```

Schritt 7: Admin-JavaScript

```
/**
 * oEmbed Manager – Admin JavaScript
 */
(function ($) {
    'use strict';

    $(document).ready(function () {

        // --- Provider Tab: Alle aktivieren / deaktivieren ---
```

```

// Invertierte Logik: Checkbox AN = Provider aktiv (NICHT in disabled-Liste)
// Beim Absenden müssen wir die NICHT angehakten als "disabled" senden.

// "Alle aktivieren" → alle Checkboxes abhaken
$('.jepp-oem-select-all').on('click', function () {
    $('.jepp-oem-provider-checkbox').prop('checked', true);
});

// "Alle deaktivieren" → alle Checkboxes abhaken entfernen
$('.jepp-oem-deselect-all').on('click', function () {
    $('.jepp-oem-provider-checkbox').prop('checked', false);
});

// Invertierte Logik: Vor dem Submit die Werte umkehren
$('.form').on('submit', function () {
    $('.jepp-oem-provider-checkbox[data-inverted="1"]').each(function () {
        // Wenn NICHT gecheckt → soll als disabled gesendet werden
        // Wenn gecheckt → NICHT senden (Provider ist aktiv)
        if ($(this).is(':checked')) {
            $(this).prop('disabled', true); // Nicht mitsenden
        } else {
            $(this).prop('checked', true); // Wert mitsenden
            $(this).prop('disabled', false);
        }
    });
});

// --- Custom Provider Tab: Zeilen hinzufügen/entfernen ---

var $table = $('#jepp-oem-custom-table tbody');

$('#jepp-oem-add-provider').on('click', function () {
    var idx = $table.find('tr').length;
    var optName = $('input[name*="custom_providers"]').length > 0
        ?
$('input[name*="custom_providers"]').first().attr('name').replace(/\[custom_providers\]\[\d+\]
.*/ , '[custom_providers]')
        : 'jepp_oem_settings[custom_providers]';

    var row = '<tr class="jepp-oem-custom-row">' +

```

```

        '<td><input type="text" class="regular-text" name="' + optName + '[' + idx +
    '][pattern]" placeholder="https://example.com/*"></td>' +
        '<td><input type="url" class="regular-text" name="' + optName + '[' + idx +
    '][endpoint]" placeholder="https://example.com/oembed"></td>' +
        '<td><input type="checkbox" name="' + optName + '[' + idx + '][regex]"
value="1"></td>' +
        '<td><button type="button" class="button jepp-oem-remove-
row">&times;</button></td>' +
        '</tr>';

    $table.append(row);
});

$(document).on('click', '.jepp-oem-remove-row', function () {
    $(this).closest('tr').remove();
    // Indizes neu nummerieren
    $table.find('tr').each(function (i) {
        $(this).find('input').each(function () {
            var name = $(this).attr('name');
            if (name) {
                $(this).attr('name', name.replace(/\[custom_providers\]\[\d+\]/,
'[custom_providers][' + i + ']'));
            }
        });
    });
});
});
});
})(jQuery);

```

Schritt 8: Uninstall-Routine

```

<?php
/**
 * Wird beim Löschen des Plugins ausgeführt.
 * Entfernt alle gespeicherten Optionen und den oEmbed-Cache.
 */

if ( ! defined( 'WP_UNINSTALL_PLUGIN' ) ) {

```

```

    exit;
}

// Plugin-Option entfernen
delete_option( 'jepp_oem_settings' );

// oEmbed-Cache entfernen
global $wpdb;
$wpdb->query(
    "DELETE FROM {$wpdb->postmeta} WHERE meta_key LIKE '_oembed_%'"
);
$wpdb->query(
    "DELETE FROM {$wpdb->options} WHERE option_name LIKE '_transient_oembed_%' OR option_name
    LIKE '_transient_timeout_oembed_%'"
);

```

Dateistruktur

```

oEmbed Plugin/
├─ oembed-manager.php           ← Haupt-Plugin-Datei
├─ uninstall.php               ← Aufräumen bei Deinstallation
├─ includes/
│   ├─ class-jepp-oem-settings.php ← Admin-Einstellungsseite
│   ├─ class-jepp-oem-providers.php ← Provider-Verwaltung
│   └─ class-jepp-oem-frontend.php ← Frontend-Filter
├─ assets/
│   ├─ css/
│   │   ├─ admin.css           ← Admin-Styles
│   │   └─ frontend.css       ← Frontend-Styles
│   └─ js/
│       └─ admin.js           ← Admin-JavaScript
└─ languages/                 ← Übersetzungsdateien

```

Funktionsübersicht

| Feature | Beschreibung |
|--------------------------------|---|
| Provider an/aus | Jeden WordPress-Standard-Provider einzeln deaktivieren |
| Eigene Provider | Neue oEmbed-Endpunkte mit URL-Pattern hinzufügen |
| URL-Blockliste | Bestimmte URLs/Muster von der Einbettung ausschließen |
| Max. Dimensionen | Maximale Breite und Höhe für Embeds festlegen |
| Responsive Container | Automatisch 16:9-Container für Video-Embeds |
| Lazy Loading | <code>loading="lazy"</code> für iframes aktivieren |
| Inline-Styles entfernen | Unerwünschte Style-Attribute aus dem HTML entfernen |
| Feed-Deaktivierung | Im RSS-Feed nur Links statt Embeds anzeigen |
| Komplett deaktivieren | oEmbed auf dem gesamten Frontend abschalten |
| Cache-Steuerung | Cache-Dauer anpassen, Cache wird bei Änderungen geleert |

Das Plugin ist sofort einsatzbereit – einfach den Ordner `oEmbed Plugin` nach `wp-content/plugins/` kopieren und im WordPress-Backend aktivieren. Die Einstellungen finden sich dann unter **Einstellungen → oEmbed Manager**.

Revision #2

Created 2026-06-05 19:54:30 UTC by art10m

Updated 2026-06-05 19:55:40 UTC by art10m