

```
<?php

declare(strict_types=1);

namespace Plugin\fp_altcha_spamschutz\src\Handler;

use JTL\phpQuery\phpQueryObject;
use JTL\Plugin\PluginInterface;
use JTL\Smarty\JTLSmarty;
use Plugin\fp_altcha_spamschutz\src\Service\AltchaService;

/**
 * Fuegt das ALTCHA-Widget (Container + Skript) in Registrierungs- und
 * Newsletter-Formular ein.
 * Wird ueber HOOK_SMARTY_OUTPUTFILTER aufgerufen, also nach dem Rendern der
 * Seite, auf dem
 * fertigen HTML-Dokument (phpQuery, jQuery-aehnliche PHP-DOM-API).
 */
class TemplateHandler
{
    private PluginInterface $plugin;
    private AltchaService $altchaService;

    public function __construct(PluginInterface $plugin, AltchaService
$altchaService)
    {
        $this->plugin = $plugin;
        $this->altchaService = $altchaService;
    }

    /**
     * @param array<string, mixed> $args Enthaelt 'smarty' (JTLSmarty) und
     * 'document' (phpQueryObject).
     */
    public function generalTemplateIntegration(array $args): void
    {
        // WICHTIG: Dieser Hook laeuft bei JEDEM Seitenaufruf im gesamten
        Shop. Ein Fehler hier
        // darf niemals die Anzeige irgendeiner Shop-Seite verhindern --
        deshalb komplett
        // defensiv mit try/catch umschlossen. Im Zweifel wird einfach kein
        Widget angezeigt,
        // statt die Seite kaputt zu machen.
        try {
            $this->doTemplateIntegration($args);
        } catch (\Throwable $e) {
            if ($this->altchaService->isDebug()) {
                error_log('[fp_altcha_spamschutz] Fehler bei Template-
Integration: ' . $e->getMessage());
            }
        }
    }
}
```

```
    }  
}  
  
/**  
 * @param array<string, mixed> $args  
 */  
private function doTemplateIntegration(array $args): void  
{  
    /** @var phpQueryObject $document */  
    $document = $args['document'];  
  
    $injected = false;  
  
    if ($this->altchaService->isEnabledForRegistration()) {  
        $registerForm = $document->find('form.register-form');  
        if (\count($registerForm) > 0) {  
            $this->injectWidget($document, $registerForm);  
            $injected = true;  
        }  
    }  
  
    if ($this->altchaService->isEnabledForNewsletter()) {  
        $newsletterForm =  
$document->find('input[name="abonnieren"]')->closest('form');  
        if (\count($newsletterForm) > 0) {  
            $this->injectWidget($document, $newsletterForm);  
            $injected = true;  
        }  
    }  
  
    if ($injected) {  
        $this->includeScript($document);  
    }  
}  
  
private function injectWidget(phpQueryObject $document, phpQueryObject  
$form): void  
{  
    // Nicht doppelt einfuegen (z. B. wenn die Seite mehrfach gefiltert  
wird).  
    if (\count($form->find('.fp-altcha')) > 0) {  
        return;  
    }  
  
    $challenge = $this->altchaService->createChallengeArray();  
    $challengeJson = htmlspecialchars(  
        (string) json_encode($challenge, \JSON_UNESCAPED_SLASHES),  
        \ENT_QUOTES,  
        'UTF-8'  
    );  
}
```

```

        $isGerman = empty($_SESSION['cISOSprache']) ||
$_SESSION['cISOSprache'] === 'ger';
        $labelPreparing = $isGerman ? 'Sicherheitsprüfung wird vorbereitet
...' : 'Preparing security check ...';
        $labelPreparing = htmlspecialchars($labelPreparing, \ENT_QUOTES,
'UTF-8');

        $html = '<div class="fp-altcha form-group" data-fp-altcha="' .
$challengeJson . '">'
            . '<span class="fp-altcha-status text-muted small"><i class="fa
fa-shield" aria-hidden="true"></i> '
            . $labelPreparing . '</span>'
            . '<input type="hidden" name="altcha" class="fp-altcha-input"
value="">'
            . '</div>';

        // Direkt vor dem Absende-Button einfügen.
        $submitButton = $form->find('button[type="submit"]',
input[type="submit"]);
        if (\count($submitButton) > 0) {
            $submitButton->first()->before($html);
        } else {
            $form->append($html);
        }
    }

    private function includeScript(PHPQueryObject $document): void
    {
        if (\count($document->find('script[data-fp-altcha-script]')) > 0) {
            return;
        }

        // Bewusst analog zum bewährten Muster anderer, quelloffener JTL5-
        Plugins direkt ueber
        // die globale URL_SHOP-Konstante aufgebaut, statt eine Plugin-Pfad-
        API zu erraten.
        $assetUrl = \URL_SHOP . '/plugins/fp_altcha_spamschutz/assets/fp-
altcha.js';
        $assetUrl = htmlspecialchars($assetUrl, \ENT_QUOTES, 'UTF-8');

        $document->find('body')->append(
            '<script data-fp-altcha-script defer src="' . $assetUrl .
' "></script>'
        );
    }
}

```

From:
<https://guide.falk.plus/> - **Best Practice Guide**

Permanent link:
https://guide.falk.plus/doku.php?id=gitea_code:jtl-shop-altcha-spamschutz:templatehandler_php

Last update: **2026/09/21 16:16**

