

```
<?php

declare(strict_types=1);

namespace Plugin\fp_altcha_spamschutz\src\Handler;

use Plugin\fp_altcha_spamschutz\src\Service\AltchaService;

/**
 * Prueft die ALTCHA-Loesung bei Registrierung und Newsletter-Anmeldung.
 *
 * Registrierung: nutzt den dokumentierten Hook
HOOK_REGISTRIEREN_PAGE_REGISTRIEREN_PLAUSI (41),
 * der genau fuer diesen Zweck (Plausibilitaetspruefung nach
Formularabsendung) vorgesehen ist.
 * Bei fehlender/ungueltiger Pruefung wird ein Eintrag zu "fehlendeAngaben"
hinzugefuegt -- das ist
 * derselbe Mechanismus, den JTL-Shop fuer "Pflichtfeld nicht ausgefuellt"
nutzt, der Kunde bekommt
 * also die normale, vertraute Fehlermeldung des Shops.
 *
 * Newsletter: die Dokumentation zum passenden Hook
(HOOK_NEWSLETTER_PAGE_EMPFAENGEREINTRAGEN) ist
 * nicht eindeutig genug, um sich allein darauf zu verlassen. Deshalb wird
zusaetzlich ganz frueh
 * (direkt in Bootstrap::boot(), vor jeder Shop-Verarbeitung) geprueft:
schlaegt die Pruefung fehl,
 * wird das POST-Feld "abonnieren" entfernt, sodass JTL-Shop die Anmeldung
erst gar nicht verarbeitet.
 * Der Hook-Handler unten ist eine zusaetzliche Absicherung ("defense in
depth"), falls dieser fruehe
 * Filter aus irgendeinem Grund nicht greift.
 */
class ValidationHandler
{
    private AltchaService $altchaService;

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

    /**
     * @param array<string, mixed> $args
     */
    public function checkRegistrationPlausibility(array $args): void
    {
        try {
            if (!$this->altchaService->isEnabledForRegistration()) {
                return;
            }
        }
    }
}
```

```
    }

    if ($this->altchaService->verifyPost()) {
        return;
    }

    if (isset($args['fehlendeAngaben']) &&
    \is_array($args['fehlendeAngaben'])) {
        $args['fehlendeAngaben'][] = 'Sicherheitspruefung';
    }

    if (\array_key_exists('nReturnValue', $args)) {
        $args['nReturnValue'] = 0;
    }
} catch (\Throwable $e) {
    // Bewusste Entscheidung: bei einem unerwarteten Fehler (z. B.
    // Plugin nicht vollstaendig
    // konfiguriert) lieber die Registrierung durchlassen, als die
    // komplette Registrierungs-
    // seite fuer echte Kunden zu blockieren. Ein Bot mehr ist
    // besser als ein Shop, bei dem
    // sich niemand mehr registrieren kann.
    if ($this->altchaService->isDebug()) {
        error_log('[fp_altcha_spamschutz] Fehler bei
    Registrierungspruefung: ' . $e->getMessage());
    }
}

/**
 * Fruehe Pruefung fuer die Newsletter-Anmeldung. Wird direkt aus
 * Bootstrap::boot() aufgerufen,
 * nicht ueber einen Hook, damit sie garantiert vor der Verarbeitung
 * durch newsletter.php laeuft.
 */
public function guardNewsletterSubmission(): void
{
    try {
        if (!$this->altchaService->isEnabledForNewsletter()) {
            return;
        }

        $isNewsletterSubscribeSubmit = ($_SERVER['REQUEST_METHOD'] ??
        '') === 'POST'
            && isset($_POST['abonnieren']);

        if (!$isNewsletterSubscribeSubmit) {
            return;
        }
    }
```

```

        if ($this->altchaService->verifyPost()) {
            return;
        }

        // Feld entfernen, damit JTL-Shop die Anmeldung nicht als
        "abonnieren"-Request erkennt
        // und stattdessen nur das (leere) Formular erneut anzeigt.
        unset($_POST['abonnieren']);
        $_SESSION['fp_altcha_newsletter_error'] = true;
    } catch (\Throwable $e) {
        // Wird direkt in Bootstrap::boot() aufgerufen (nicht ueber
        // einen Hook) -- ein
        // Fehler hier darf auf keinen Fall den kompletten Seitenaufbau
        // verhindern.
        if ($this->altchaService->isDebug()) {
            error_log('[fp_altcha_spamschutz] Fehler bei Newsletter-
            Fruehpruefung: ' . $e->getMessage());
        }
    }
}

/**
 * Zusätzliche Absicherung ueber
HOOK_NEWSLETTER_PAGE_EMPFAENGEREINTRAGEN, falls die fruehe
 * Pruefung in guardNewsletterSubmission() aus irgendeinem Grund nicht
gegriffen hat.
 *
 * @param array<string, mixed> $args Enthält 'oNewsletterEmpfaenger'
(stdClass, per Referenz).
 */
public function checkNewsletterRecipient(array $args): void
{
    try {
        if (!$this->altchaService->isEnabledForNewsletter()) {
            return;
        }

        if (!empty($_SESSION['fp_altcha_newsletter_error'])) {
            // Bereits in guardNewsletterSubmission() als ungueltig
            // erkannt: Empfaenger-E-Mail
            // leeren, damit JTL-Shops eigene Pflichtfeld-Pruefung den
            // Insert verhindert.
            if (isset($args['oNewsletterEmpfaenger']) &&
            \is_object($args['oNewsletterEmpfaenger'])) {
                $args['oNewsletterEmpfaenger']->cEmail = '';
            }
        }
    } catch (\Throwable $e) {
        if ($this->altchaService->isDebug()) {
            error_log('[fp_altcha_spamschutz] Fehler bei Newsletter-
            Absicherung: ' . $e->getMessage());
        }
    }
}

```

```
}  
}  
}  
}
```

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

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

Last update: **2026/09/21 15:39**

