

```
<?php

declare(strict_types=1);

namespace Plugin\fp_altcha_spamschutz;

use JTL\Events\Dispatcher;
use JTL\Plugin\Bootstrapper;
use JTL\Plugin\PluginInterface;
use JTL\Shop;
use Plugin\fp_altcha_spamschutz\src\Handler\TemplateHandler;
use Plugin\fp_altcha_spamschutz\src\Handler\ValidationHandler;
use Plugin\fp_altcha_spamschutz\src\Service\AltchaService;

require_once __DIR__ .
    '/src/Vendor/AltchaOrg/Altcha/V1/Hasher/HasherInterface.php';
require_once __DIR__ .
    '/src/Vendor/AltchaOrg/Altcha/V1/Hasher/Algorithm.php';
require_once __DIR__ . '/src/Vendor/AltchaOrg/Altcha/V1/Hasher/Hasher.php';
require_once __DIR__ .
    '/src/Vendor/AltchaOrg/Altcha/V1/BaseChallengeOptions.php';
require_once __DIR__ .
    '/src/Vendor/AltchaOrg/Altcha/V1/ChallengeOptions.php';
require_once __DIR__ .
    '/src/Vendor/AltchaOrg/Altcha/V1/CheckChallengeOptions.php';
require_once __DIR__ . '/src/Vendor/AltchaOrg/Altcha/V1/Challenge.php';
require_once __DIR__ . '/src/Vendor/AltchaOrg/Altcha/V1/Payload.php';
require_once __DIR__ . '/src/Vendor/AltchaOrg/Altcha/V1/Solution.php';
require_once __DIR__ . '/src/Vendor/AltchaOrg/Altcha/V1/Obfuscator.php';
require_once __DIR__ .
    '/src/Vendor/AltchaOrg/Altcha/V1/ServerSignaturePayload.php';
require_once __DIR__ .
    '/src/Vendor/AltchaOrg/Altcha/V1/ServerSignatureVerificationData.php';
require_once __DIR__ .
    '/src/Vendor/AltchaOrg/Altcha/V1/ServerSignatureVerification.php';
require_once __DIR__ . '/src/Vendor/AltchaOrg/Altcha/V1/Altcha.php';
require_once __DIR__ . '/src/Service/AltchaService.php';
require_once __DIR__ . '/src/Handler/TemplateHandler.php';
require_once __DIR__ . '/src/Handler/ValidationHandler.php';

/**
 * Bootstrap-Klasse fuer das Plugin fp_altcha_spamschutz.
 *
 * Setzt einen selbst gehosteten, quelloffenen Proof-of-Work-Spamschutz
(ALTCHA, MIT-Lizenz) fuer
 * die Kundenregistrierung und die Newsletter-Anmeldung ein. Keine Daten
verlassen den eigenen
 * Server -- bewusst ohne Google reCAPTCHA oder aehnliche externe Dienste
und ohne Umleitung
 * des Datenverkehrs ueber Cloudflare oder vergleichbare Drittanbieter.

```

```
*
* Hintergrund/Anlass: ein Bot-Problem bei einem Kunden (massenhafte Fake-
Registrierungen und
* Newsletter-Anmeldungen, u. a. "Test"/"Test User" in
Vorname/Nachname/Ort/Strasse, aber echte
* Drittanbieter-E-Mail-Adressen). Andere getestete Ansätze konnten das
Muster nicht
* zuverlässig erkennen; dieses Plugin prüft stattdessen direkt beim
Absenden von
* Registrierung und Newsletter-Anmeldung selbst.
*/
class Bootstrap extends Bootstrapper
{
    public function boot(Dispatcher $dispatcher): void
    {
        parent::boot($dispatcher);

        if (!Shop::isFrontend()) {
            return;
        }

        // Die komplette Einrichtung ist defensiv umschlossen: boot() läuft
bei JEDEM
        // Frontend-Aufruf, noch bevor irgendeine Seite gerendert wird. Ein
Fehler hier (z. B.
        // weil das Plugin gerade erst installiert und noch nicht
konfiguriert wurde) darf
        // niemals den gesamten Shop lahmlegen.
        try {
            /** @var PluginInterface $plugin */
            $plugin = $this->getPlugin();

            $altchaService = new AltchaService($plugin);
            $templateHandler = new TemplateHandler($plugin, $altchaService);
            $validationHandler = new ValidationHandler($altchaService);

            // Früheste mögliche Prüfung der Newsletter-Anmeldung: läuft
direkt hier, noch
            // bevor JTL-Shop das $_POST verarbeitet. Siehe
ValidationHandler::guardNewsletterSubmission().
            $validationHandler->guardNewsletterSubmission();

            // Widget (Container + Skript) in Registrierungs- und
Newsletter-Formular einfügen.
            $dispatcher->listen(
                'shop.hook.' . \HOOK_SMARTY_OUTPUTFILTER,
                [$templateHandler, 'generalTemplateIntegration']
            );
        }
    }
}
```

```
// Registrierung: Plausibilitaetspruefung nach
Formularabsendung.
$dispatcher->listen(
    'shop.hook.' . \HOOK_REGISTRIEREN_PAGE_REGISTRIEREN_PLAUSI,
    [$validationHandler, 'checkRegistrationPlausibility']
);

// Newsletter: zusaetzliche Absicherung kurz vor dem Speichern
des Empfaengers.
if (\defined('HOOK_NEWSLETTER_PAGE_EMPFAENGEREINTRAGEN')) {
    $dispatcher->listen(
        'shop.hook.' .
        \HOOK_NEWSLETTER_PAGE_EMPFAENGEREINTRAGEN,
        [$validationHandler, 'checkNewsletterRecipient']
    );
}
} catch (\Throwable $e) {
    error_log('[fp_altcha_spamschutz] Fehler beim Initialisieren des
Plugins: ' . $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:bootstrap_php

Last update: **2026/09/21 16:12**

