

Verwendung von Captcha

Um zu verhindern, dass Bots ein Formular absenden, werden Captchas verwendet. Hierbei handelt es sich um kleine Bilder, die meist eine Folge von Zahlen, Buchstaben und Zeichen oder sogar einfachen mathematischen Rechnungen beinhalten. Dieser Code wird bei jedem Aufruf zufällig generiert und ist für Maschinen oft gar nicht oder nur schwer lesbar. Daher wird er dazu verwendet, einen echten Benutzer und eine maschinelle Eingabe zu unterscheiden.

Eine solche Captcha-Funktion wird im System von EGOTEC angeboten.

Die Captcha-Funktion benötigt das GD-Lib Modul mit integrierter Free-Type Bibliothek (<http://de2.php.net/imagettfbbox>).

Vorbereitungen

Im ersten Schritt wird die Captcha-Klasse eingebunden. Sobald das geschehen ist, steht sowohl im Skript als auch als Smarty-Variable automatisch das Objekt **\$captcha** zur Verfügung:

QuelltextPHP Code:

```
1. // Captcha einbinden
2. require_once('captcha/Ego_Captcha.php');
```

```
// Captcha einbinden
require_once('captcha/Ego_Captcha.php');
```

Überprüfung (im Skript über PHP)

Mit Hilfe der Methode *check()* kann nun einfach überprüft werden, ob die Eingabe korrekt war:

QuelltextPHP Code:

```
1. if($captcha->check($modus,$name))
2. {
3.   echo 'Captcha wurde korrekt ausgefüllt';
4. }
```

```
if($captcha->check($modus,$name))
{
    echo 'Captcha wurde korrekt ausgefüllt';
}
```

Hierbei existieren zwei Parameter:

Parameter	Beschreibung
-----------	--------------

modus	Art der Captcha Prüfung ("string" oder "number"). Der Wert "string" wird ausdrücklich empfohlen.
name	Prüfung, ob Captcha-Funktion aktiviert ist. Beim Wert current wird für die aktuellen Seite geprüft, ob im Backend die Captcha-Funktion aktiviert wurde. (Der Seitentyp muss entsprechend um eine [optional] Checkbox im Adminbereich erweitert werden). Wird der Parameter weggelassen, erfolgt keine Prüfung auf Aktivierung: Die Captcha-Funktionalität ist automatisch aktiv.

Einige Fallbeispiele:

- Kann man bei der aktuellen Seite Spam-Schutz einstellen und ist dieser deaktiviert, liefert diese Funktion immer **true** zurück.

- Ist der Spam-Schutz aktiviert, liefert diese Funktion **false** zurück wenn die Überprüfung fehlgeschlagen ist.
- Kann man Spam-Schutz nicht einstellen, ist dieser automatisch immer aktiviert und diese Funktion liefert **false** zurück wenn die Überprüfung fehlgeschlagen ist.

Überprüfung (im Template über Smarty)

Die check-Methode kann über das \$captcha-Objekt ebenso im Template verwendet werden.

QuelltextSmarty Code:

1. `{* Prüfen, ob für die aktuelle Seite Captcha aktiviert wurde (diese Prüfung kann auch weg gelassen werden) *}`
2. `{if $captcha.active.current}`
- 3.
4. `{* Auf Eingabe prüfen *}`
5. `{if !$captcha.check}`
6. `Die Captcha Prüfung ist fehlgeschlagen.`
7. `{/if}`
- 8.
9. `{/if}`

```
{* Prüfen, ob für die aktuelle Seite Captcha aktiviert wurde (diese Prüfung kann auch weg gelassen werden) *}
{if $captcha.active.current}
```

```
{* Auf Eingabe prüfen *}
{if !$captcha.check}
    Die Captcha Prüfung ist fehlgeschlagen.
{/if}
```

```
{/if}
```

Damit die Überprüfung auf korrekte Eingabe vom \$captcha-Objekt überhaupt durchgeführt werden kann, müssen im Template die einzelnen Elemente wie Captcha-Bild, Eingabe-Feld, reload-Button usw. positioniert werden:

QuelltextSmarty Code:

1. `{* Über die Smarty-Funktion get_captcha werden HTML-Elemente generiert *}`
2. `{get_captcha var_text="cap_text" var_image="cap_image" var_reload="cap_reload" height=60}`
3. `<table style="width:500px;">`
4. `<tr>`
5. `<td valign="top">`
6. `{ $cap_image }`
7. `</td>`
8. `<td class="{if $error_captcha}error{/if}" valign="top" width="100%" style="padding-left: 10px;">`
9. `{t}Sicherheitsprüfung:{/t}`
10. `{ $cap_text }`
11. `{ $cap_reload }`
12. `</td>`
13. `</tr>`
14. `</table>`

```
{* Über die Smarty-Funktion get_captcha werden HTML-Elemente generiert *}
{get_captcha var_text="cap_text" var_image="cap_image" var_reload="cap_reload" height=60}
<table style="width:500px;">
```

```

<tr>
  <td valign="top">
    {$cap_image}
  </td>
  <td class="{if $error_captcha}error{/if}" valign="top" width="100%" style="padding-left: 10px;">
    <span style="display:block; padding-bottom:5px;">{t}Sicherheitsprüfung:{/t}</span>
    {$cap_text}
    <span style="display:block; padding-top:5px;">{$cap_reload}</span>
  </td>
</tr>
</table>

```

Weitere Informationen finden Sie im Designerhandbuch unter [{get_captcha}](#).

3. Spam-Schutz einstellbar machen

Möchten Sie im Adminbereich der Seite den Captcha-Schutz aktivierbar machen, erstellen Sie auf dem gewünschten Reiter eine entsprechende Checkbox. Diese muss zwingend den Namen **captcha_active** tragen, da auf dieses Extrafeld in der **Ego_Captcha** Klasse geprüft wird. Gültige Werte für dieses Feld im Falle einer Aktivierung sind **1** oder **true**.

Den Text "neu laden" für das Captcha können Sie in /skin/mandant/locale/sprache übersetzen. Lesen Sie dazu bitte auch [Mehrsprachigkeit](#). Ergänzen Sie die translation.ini einfach um den Zeile neu laden = reload (z.B. für die Englische Sprache)

Erstellt mit
EGOTEC®
Internet:
www.egotec.com
© EGOTEC
GmbH

EGOTEC GmbH
Alte Neckarelzer Straße 24
D-74821 Mosbach

Tel: +49 (0)6261 /
6743-0
Fax: +49 (0)6261
/ 6743-29
E-Mail:
info@egotec.com