

CaptchaAd with Java

Last updated: 17.09.2011

So that the integration of CaptchaAd can be made even easier, we've arranged all necessary steps right within this guide. With a little programming knowledge, CaptchaAd can be up and running on your website in no time.

Detailed steps toward integrating the module into CaptchaAd

0. Keys and necessary info
1. Downloading the files
2. Unzipping and Uploading
3. Configuration
4. Inclusion of header scripts
5. Integration into a form
6. Validating input

Online-Integration Guide (PHP version)

<http://www.captchaad.com/captchaad-client>

CaptchaAd Demo (PHP demo)

<http://www.captchaad.com/captchaad-client/demo.php>

Keys and necessary info

Before you begin integration, make sure you've read the information listed below. In general it deals with information regarding CaptchaAd and reCAPTCHA, all available per request. This information is absolutely necessary for the integration of CaptchaAd.

Information regarding CaptchaAd

- ▶ CAPTCHAAD_API_KEY
- ▶ Publisher ID
- ▶ CAPTCHAAD_ZONE_ID

All of this information can be obtained from your representative at CaptchaAd.

The Keys and IDs are used to identify and secure communication between the CaptchaAd on your end and the ad-server CaptchaAd.

Information regarding reCAPTCHA

- ▶ reCAPTCHA public key
- ▶ reCAPTCHA private key

The reCAPTCHA keys are available at the [reCAPTCHA](#) website.

If you have an account at reCAPTCHA, you can of course use your existing key.

reCAPTCHA is automatically used as a fallback for CaptchaAd in the event that the viewer's browser has disabled JavaScript and Flash. This occurs in approximately 1% of cases.

Link to reCAPTCHA: <http://www.google.com/recaptcha/whyrecaptcha>

Download the files

The first step is to download the latest version of CaptchaAd files from our website:

- ▶ http://www.captchaad.com/captchaad-client/CaptchaAd_Java_v.110527.zip

The ZIP file contains all necessary files to integrate CaptchaAd into your website.

Unzip and Upload

Unzip the file and upload the CaptchaAd directory to your webserver.

Configuration

see config.cfg below (or in the configuration file /config.cfg)

Open the configuration directory and edit CaptchaAd config.cfg file into the folder. This file is the configuration file.

Important:

- ▶ Each of the subsequent rows with a red CaptchaAd arrow **needs** to be manually configured.
- ▶ The lines with the gray arrows are **optional**.

The red-highlighted text entries are examples only. Please replace these with your own personal information (see above "Keys and necessary info").

- ▶ **Line 2:** (API key / CAPTCHAAD_API_KEY)
After the colon please enter your previously-received CaptchaAd API key.

For example:

CAPTCHAAD_API_KEY:**85219e7febbe1918c2fd83cf0eff9a1f**

- ▶ **Line 3:** (publisher-ID / CAPTCHAAD_PUBLISHER_ID)
Enter the received CaptchaAd Publisher ID as an integer.

For example:

CAPTCHAAD_PUBLISHER_ID:**999**

- ▶ **Line 5:** (path to the data / CAPTCHAAD_BASE_URL)
Enter the absolute URL in the CaptchaAd directory. This is the location of the CaptchaAd directory on your webserver.

For example:

CAPTCHAAD_BASE_URL:**<http://localhost:8080/CaptchaAdTest/>**

Do not forget the trailing "/".

- ▶ **Line 6:** (voice / CAPTCHAAD_LANGUAGE)
Here, enter the abbreviation for the language as a string that represents the presentation of the contents of CaptchaAd. This affects the language of the question, the answers and the text on the home page.

German (**de**) and English (**en**) are currently available.

For example:

CAPTCHAAD_LANGUAGE:**en**

- ▶ **Line 7:** (width / CAPTCHAAD_WIDTH)
Enter the width of the CaptchaAd as an integer. Default is 300 pixels.

For example:

CAPTCHAAD_WIDTH:**300**

- ▶ **Line 8:** (Height / CAPTCHAAD_HEIGHT)
Enter the amount shown as the CaptchaAd integer. Default is 300 pixels.

For example:

`CAPTCHAAD_HEIGHT:250`

- ▶ Line 11: (automatic start / CAPTCHAAD_AUTOSTART)
Here you can set whether the video will play automatically within the CaptchaAd, or wait for user interaction. Default is false (off).

For example:

`CAPTCHAAD_AUTOSTART:false`

- ▶ Line 12: (Sound / CAPTCHAAD_SOUND)
Here you can set whether the sound within the CaptchaAd begins on or off. Default is false (off).

For example:

`CAPTCHAAD_SOUND:false`

- ▶ Line 14: (reCAPTCHA public key / CAPTCHAAD_3RDPARTY_RECAPTCHA_PUBLIC_KEY)
Enter the reCAPTCHA public key that was given to you upon receipt of reCAPTCHA.

For example:

`CAPTCHAAD_3RDPARTY_RECAPTCHA_PUBLIC_KEY:6LcDLgkAAAAAPYhUh88_ryL5a0WMOOrtMYg-OJ`

- ▶ Line 15: (reCAPTCHA private key / CAPTCHAAD_3RDPARTY_RECAPTCHA_PRIVATE_KEY)
Enter the reCAPTCHA private key that was given to you upon receipt of reCAPTCHA.

For example:

`CAPTCHAAD_3RDPARTY_RECAPTCHA_PRIVATE_KEY:6LcDLgkAAAAAPYhUh88_ryL5a0WMOOrtMYg-OJ`

The CaptchaAd object comes with a path to configuration.

The class then instantiates a configuration object to be passed to the path, plus the name of the configuration file "config.cfg" used to read it.

For example: /path/to/my/config/file/config.cfg

The necessary configuration data is then read and cast to the required data type. For an example of the configuration file, see later in this document.

It should be noted that no quotes are used. Spaces at the beginning and end are deleted when read.

ATTENTION

If you can't define any absolute paths in your runtime environment, you can also directly instantiate a `CaptchaAd_Conf` and skip the instantiation of the `CaptchaAd` object. The configuration data must then be placed over the corresponding setter methods.

For example

1. `CaptchaAd_Conf cc = new CaptchaAdConf();`
2. `cc.setCAPTCHAAD_API_KEY("XXXXXXXXXX");`
3. `// must be set so that translation files`
4. `// are found`
5. `cc.setCAPTCHAAD_BASE_PATH("PATH_TO_CONFIG");`

Please ensure that the setter methods are of the correct type.

CaptchaAd constructors

1. Instantiation with path specifications (absolute)

```
CaptchaAd ca = new CaptchaAd("PATH_TO_CONFIG");
```

2. Instantiation with `CaptchaAd_Conf` Objekt

```
CaptchaAd_Conf cc = new CaptchaAdConf();
cc.setCAPTCHAAD_API_KEY("XXXXXXXXXX");
// must be set so that translation files
// are found
cc.setCAPTCHAAD_BASE_PATH("PATH_TO_CONFIG");
CaptchaAd ca = new CaptchaAd(cc);
```

3. Instantiation with `CaptchaAd_Conf` und `CaptchaAd_Translation` Objekt

```
// Instantiate Conf object
CaptchaAd_Conf cc = new CaptchaAdConf();
cc.setCAPTCHAAD_API_KEY("XXXXXXXXXX");

// Instantiate translation object
CaptchaAd_Translation ct = new CaptchaAd_Translation();
// Translation Setting Properties
ct.setALTTEXT("Meine Demo!");
ct.setCHECKBUTTON_LABEL("Prüfen");
ct.setInput_LABEL("Welcome to my demo");
ct.setZONE_ID(81);
// Create instance CaptchaAd
CaptchaAd ca = new CaptchaAd(cc,ct);
```

Examples for the instantiation (from demo.jsp)

```
8. // New instance config
9. // Here the analogue can also be used without the path constructor
10. CaptchaAd_Conf cf = new CaptchaAd_Conf("/PATH/TO/CONFIG");
11.
12. // New instance of translation
13. CaptchaAd_Translation ct = new CaptchaAd_Translation();
14. // Translation Setting Properties
15. ct.setALTTEXT("My Demo!");
16. ct.setCHECKBUTTON_LABEL("test");
17. ct.setInput_LABEL("Welcome to my demo");
18. ct.setZONE_ID(81);
19.
20. // New instance CaptchaAd
21. CaptchaAd c;
22. c = new CaptchaAd(cf,ct);
```

Inclusion of Header Scripts

Integration into an HTML page is possible as follows.

The first step is for the CaptchaAd object to be instantiated via the path to the configuration file (or one of the other two paths). Then, by calling up the method **getHtmlHeader ()**, can you return the necessary information as a string.

In the HTML, header scripts are included for the swfobject.js script tags from the Resources folder, and the OpenX tag integrated. Here it's important that the files from the resources folder on your Web server be available at the address "CAPTCHAAD_BASE_URL".

The sub-folder structure of resource **must** be preserved.

For example: Creating HTML header scripts

1. `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" 'http://www.w3.org/TR/html4/loose.dtd'>`
2. `<html>`
3. `<head>`
4. `<meta http-equiv='Content-Type' content='text/html; charset=UTF-8'>`
5. `<title>Welcome to Spring WEB MVC project</title>`
6. `<% out.print(c.getHtmlHeader()); %>`
7. `</head>`
8. `<body>`

Please make sure that an integrated display is indeed shown in the HTML head.

Integration in a form

Analogous to the instantiation and display in in HTML head, the integration of user input into the HTML-body can easily be carried out.

By invoking the method `getHtmlBody ()` a `CaptchaAd_Translation` object is created (if it has not passed through the constructor). Here, the abbreviation of the configuration file is applied as recognition for the language.

By default, the German translation should be pre-loaded and not missing.

The path for the translation files is the path to the configuration file. In addition, the translation files must be found in in a subdirectory named "translation".

For example: / path / to / my / config / file / translation / **de**.cfg

The file `de.cfg` in the "translation" directory can serve as a template for other translations. The corresponding name of the translation must be entered only in the configuration file of the `CaptchaAd_Objekts - CaptchaAd_Lanugage` field.

For example: Integrating the Body

1. `<form method='post' id='testForm' action='URL_TO_THIS_PAGE'>`
2. `<%`
3. `// Zone-ID = Zone ID of translation takes`
4. `// Zone-ID > 0 this`
5. `// Incorrect zone ID will fallback reCAPTCHA displays`
6. `out.print(c.getHtmlBody(0));`
7. `%>`
8. `<input type='submit' name='isPost' value='submit' />`
9. `</form>`

Validation of input

To review the entry of a user a `CaptchaAd_Request` object must first be created. This requires the following data:

CaptchaAd fields

<code>CaptchaAdInput</code>	// input from users
<code>CaptchaAdSession</code>	// The session ID of the user
<code>CaptchaAdSwf</code>	// If the event that the SWF file was used
<code>RemoteAddr</code>	// is the remote address of the user

Recaptcha fields for fallback, in normal cases zero

<code>ReCaptchaChallengeField</code>	// The Recaptcha Challenge Field
<code>ReCaptchaResponseField</code>	// The Recaptcha Response Field

This object is then called with the method `CheckAnswer ()` from `CaptchaAd_Objekt`. As a result, `True` or `False` will be returned as a Boolean data type.

Example: Checking the input

```

1. // Check whether form was sent
2. if(request.getParameter('isPost') != null && request.getParameter('isPost').equals('submit')){
3.     CaptchaAd_Request cr = new CaptchaAd_Request();
4.     cr.setReCaptchaChallengeField(request.getParameter('recaptcha_challenge_field'));
5.     cr.setReCaptchaResponseField(request.getParameter('recaptcha_response_field'));
6.     cr.setReCaptchaInput(request.getParameter('CaptchaAdInput'));
7.     cr.setReCaptchaSession(request.getParameter('CaptchaAdSession'));
8.     cr.setReCaptchaSwf(request.getParameter('CaptchaAdSwf'));
9.     cr.setRemoteAddr(request.getRemoteAddr());
10.
11.     boolean result = c.checkAnswer(cr);
12.
13.     if(result == true) {
14.         check = '<span>Die Prüfung war erfolgreich</span>';
15.     } else {
16.         check = '<span>The test failed</span>';
17.     }
18. }
```

If the request "**CaptchaAdSwf**" does not adhere and the Recaptcha fields aren't zero, then the fallback is performed and evaluated.

Special Notes

Languages (homepage text)

Thus far the standard CaptchaAd package includes only German and English. If you need further language versions, they can be prepared quickly and without much effort.

If you want to make adjustments, you can find the language translation files in the folder. In the PHP files stored there, you will also find the texts that appear on the home page of CaptchaAd.

Automatic Restart

ATTENTION: You may only configure CaptchaAds to automatically play after you have spoken with CaptchaAd! An incorrect use can seriously reduce the value of an advertising medium. This can happen, for example, when a clip is running in places that are not immediately visible to the user.

When installed inappropriately and without our explicit acknowledgement, we reserve the right to block the corresponding publisher ID.

Sound on/off

CAUTION: You may only configure CaptchaAds to automatically play sound after you have discussed this with CaptchaAd! Under certain circumstances, this is a violation of applicable guidelines.

When installed inappropriately and without agreement, we reserve the right to block the corresponding publisher ID.

Scripts in overview

Configuration

config.cfg

1. # CaptchaAd publsher settings
2. CAPTCHAAD_API_KEY:**85219e7febbe1918c2fd83cf0eff9a1f**
3. CAPTCHAAD_PUBLISHER_ID:**34**
4. # Your environment settings
5. CAPTCHAAD_BASE_URL:<http://localhost:8080/CaptchaAdTest/>
6. CAPTCHAAD_LANGUAGE:**de**
7. CAPTCHAAD_WIDTH:300
8. CAPTCHAAD_HEIGHT:250
9. CAPTCHAAD_SWF_VERSION:20110527a
10. #special settings
11. CAPTCHAAD_AUTOSTART:**false**
12. CAPTCHAAD_SOUND:**false**
13. # reCaptcha settings
14. CAPTCHAAD_3RDPARTY_RECAPTCHA_PUBLIC_KEY:**6LcDLgkAAAAAAPYhU88_ryL5a0WMoOrtMYg-OJ**
15. CAPTCHAAD_3RDPARTY_RECAPTCHA_PRIVATE_KEY:**6LcDLgkAAAAAGoVKxysw0infpBwHP88xtZqj6vb**
16. # -----
17. # don't change the lines behind this commentline
18. # -----
19. CAPTCHAAD_3RDPARTY_OPENX_URL:<http://master.captchaad.com/www/delivery/spcjs.php?id=>
20. CAPTCHAAD_HOST:server.captchaad.com
21. CAPTCHAAD_URL:<http://server.captchaad.com/>

FAQ

Network Communications

The main part of the communication network for displaying the CaptchaAd runs client-side.

On the server side, when during a session one of the of the servers is queried by CaptchaAd, the corresponding will be applied to client-server communication. Server-side communications will be directed through the standard HTTP port 80. For the firewall, CAPTCHAAD_HOST can be released through connections to port 80.

RollBack options

Rollback options are generally at the discretion of the publisher.

A simple way to perform a rollback on an integrated CaptchaAd is to give an invalid value (API key, zone ID or publisher ID) in the CaptchaAd call. In such a cases an automatically integrated reCAPTCHA is displayed.

TimeOut treatment

With HTTP communication the default of 5 seconds will be used.

To set the timeout via the configuration is not a problem and can be implemented without much effort. The timeouts are intercepted as exceptions and have as a result the delivery of reCAPTCHA's due to an invalid SessionID.

HTTP communication

Standard IO and NET libraries are applied.

»too many open files«

The shortcomings in the first versions regarding this point have been resolved. Closing the sockets is also possible. *stdin* and *stdout* in the HTTP client have been implemented.