EU CAPTCHA

Documentation and implementation manual

# TABLE OF CONTENT

Contents

[1. Introduction 4](#_Toc46935628)

[1.1. Who is this document for? 4](#_Toc46935629)

[1.2. What is a CAPTCHA 4](#_Toc46935630)

[1.3. Features 4](#_Toc46935631)

[2. Website integration 5](#_Toc46935632)

[2.1. EUCAPTCHA.jsp 6](#_Toc46935633)

[2.2. restCaptcha.js 7](#_Toc46935634)

[3. EU CAPTCHA process 10](#_Toc46935635)

[3.1. REST and JSON 10](#_Toc46935636)

[3.2. Calls and requests 10](#_Toc46935637)

[3.2.1. URL 11](#_Toc46935638)

[3.2.2. Getting the CAPTCHA 11](#_Toc46935639)

[3.2.3. Reload the CAPTCHA 15](#_Toc46935640)

[3.2.4. Validate the CAPTCHA 15](#_Toc46935641)

[4. Deployment 15](#_Toc46935642)

[4.1. Deploy by copying archive: 15](#_Toc46935643)

[4.2. Deploy from Tomcat Manager 15](#_Toc46935644)

[4.2.1. Deploy Directory or WAR File Located on Server 16](#_Toc46935645)

[4.2.2. WAR File to Deploy 16](#_Toc46935646)

# Introduction

## Who is this document for?

This CAPTCHA API Implementation Manual is for the reference of developers wanting to use the REST API gateway.

## What is a CAPTCHA

A CAPTCHA is a test intended to distinguish human from machine input in order to thwart spam and automatic submission or extraction of data. It is a short online typing test that is easy for humans to pass but difficult for robotic software programs to complete—hence the test's actual name, Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA). The user is typically challenged to solve a puzzle that relies on expected capacities of the human brains but whose resolution is complex to automate.

## Features

The objective of this solution is to offer to the Member States an open source CAPTCHA released under the EUPL (European Union Public License) that is maintained, secure, user-friendly and multilingual. This component can be operated as a service. The solution is published on GitHub so that it can be reviewed and maintained by the open source community.

A CAPTCHA is an essential component of information systems dealing specifically with human users, such as citizens. Having an open source secure and user-friendly CAPTCHA prevents from having to acquire or implement a specific one in several information systems. It allows for Member States to offer a consistent user experience throughout public services in Europe for a step-in administrative process that are perceived as complicated by many users. The EU CAPTCHA solution solves a common problem in a consistent and cost-effective way and solves a common problem that is not related to any particular sector or EU Member State.

EU CAPTCHA is multilingual with support for all official languages from the European Union. It allows you to make a request to the included REST API using the desired language. You can include the unique language *code* of the desired language as a query parameter (see also Language table 1). By default, the configured language is English. If the CAPTCHA solution is used on an internationalized page, further configuring EU CAPTCHA can be helpful for the user. The user can select the preferred language themselves from a drop-down list, or the developer pre-configures this during deployment.

The textual CAPTCHAs are case sensitive. Users will have to insert the exact upper- and lower-case letters that the CATPCHA image is showing them.

The selected language has an impact on the alphabet used in the CAPTCHA image. For example, if EU CAPTCHA is configured to use Bulgarian, the Bulgarian alphabet will be used:

'А', 'а', 'Б', 'б', 'В', 'в', 'Г', 'г', 'Д', 'д', 'Е', 'е', 'Ж', 'ж', 'З', 'з', 'И', 'и',  
'Й', 'й', 'К', 'к', 'Л', 'л', 'М', 'м', 'Н', 'н', 'О', 'о', 'П', 'п', 'Р', 'р', 'С', 'с',  
'Т', 'т', 'У', 'у', 'Ф', 'ф', 'Х', 'х', 'Ц', 'ц', 'Ч', 'ч', 'Ш', 'ш', 'Щ', 'щ',  
'Ю', 'ю', 'Я', 'я'

The EU CAPTCHA also supports users with visual impairments. The images that contain the textual CAPTCHA combination only use font colour combinations that minimize complications for users with colour blindness (red-green and blue-yellow). Moreover, EU CAPTCHA has support for spoken CAPTCHAs for all official languages from the European Union. Please note that the spoken CAPTCHAs do *not* support checks for upper- or lower-case letters during validation.

# Website integration

To include the EU CAPTCHA component in your web page, you should use the html code snippet in your web page.

EUCAPTCHA.jsp:  
<head>  
 <meta http-equiv="content-type" content="text/html; charset=utf-8">  
 <link rel="stylesheet" href="css/jquery-ui.min.css">  
 <script src="js/jquery-1.2.min.js"></script>  
 <script src="js/restCaptcha.js"></script>  
 <title>EU Captcha</title>  
</head>

To complete the integration, you should also make the restCaptcha.js file (indicated in red) available on the web page by adding the following line in the head section of your web page.

<script src="<path to your javascript files>/restCaptcha.js"></script>

## EUCAPTCHA.jsp

The interface of the EU CAPTCHA can be styled to match the rest of the website.

The jsp code snippet contains:

* <img alt="Captcha Loading"…> for the section where the CAPTCHA image is shown.
* <audio controls autostart="1" …> for the section to facilitate audio support.
* <button class="btn btn-primary btn-lg " …> for button to reload the CAPTCHA image.
* <input type="text" class="form-control" …> for the input area where the user can enter the CAPTCHA.

<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>  
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>  
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>  
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>  
  
<!DOCTYPE html>  
<html>  
  
<head>  
 <meta http-equiv="content-type" content="text/html; charset=utf-8">  
 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">  
 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  
 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>   
 <link rel="shortcut icon" href="#" />  
 <link rel="stylesheet" href="css/font-awesome-4.7.0/css/font-awesome.min.css">  
 <link rel="stylesheet" href="css/jquery-ui.min.css">  
 <link rel="stylesheet" href="css/restCaptcha.css">  
 <script src="js/jquery-1.2.min.js"></script>  
 <script src="js/restCaptcha.js"></script>  
  
 <title>EU Captcha</title>  
</head>  
  
<body>

……………..

<div>  
 <div>  
 <img alt="Captcha Loading" class="img-fluid img-thumbnail" src="" id="captchaImg" captchaId="">  
 <hr>  
 <audio controls autostart="1" src="" id="audioCaptcha" onplay="*onPlayAudio*()"></audio>  
 </div>  
 <div>  
 <button class="btn btn-primary btn-lg " id="captchaReload"> <i class="fa fa-refresh"></i> </button>  
 </div>  
  
</div>  
<hr>  
<div>  
 <div>  
 <input type="text" class="form-control" id="captchaAnswer" placeholder="Captcha Text">  
 </div>  
 <div>  
 <button class="btn btn-primary btn-lg " id="captchaSubmit"> <i class="fa fa-check"

aria-hidden="true"></i> </button>  
 </div>  
</div>

………….

## restCaptcha.js

var useAudio = false;  
var EuCaptchaToken;  
function *onPlayAudio*(){  
 useAudio = true;  
}  
$(function(){  
 function getcaptcha(){  
 getCaptchaUrl = $.ajax({  
 type: "GET",  
 url: 'http://localhost:8080/api/captchaImg',  
 success: function (data) {  
 EuCaptchaToken = getCaptchaUrl.getResponseHeader("token");  
 var jsonData=JSON.parse(data);  
 $("#captchaImg").attr("src","data:image/png;base64,"+jsonData.captchaImg);  
 $("#captchaImg").attr("captchaId",jsonData.captchaId);  
 $("#audioCaptcha").attr("src","data:audio/wav;base64,"+jsonData.audioCaptcha);  
 }  
 });  
 }  
 function reloadCaptcha(){  
 reloadCaptchaUrl = $.ajax({  
 type: "GET",  
 url: 'http://localhost:8080/api/reloadCaptchaImg/'+$("#captchaImg").attr("captchaId"),  
 beforeSend: function(xhr) {  
 xhr.setRequestHeader("Accept", "application/json");  
 xhr.setRequestHeader("Content-Type", "application/json");  
 xhr.setRequestHeader("jwtString", EuCaptchaToken);  
 },  
 success: function (data) {  
 var jsonData=JSON.parse(data);  
 $("#captchaImg").attr("src","data:image/png;base64,"+jsonData.captchaImg);  
 $("#captchaImg").attr("captchaId",jsonData.captchaId);  
 $("#audioCaptcha").attr("src","data:audio/wav;base64,"+jsonData.audioCaptcha);  
 $("#captchaAnswer").val("");  
 useAudio = false;  
 }  
 });  
 }  
 function validateCaptcha(){  
 $.ajax({  
 type: "POST",  
 contentType: 'application/json; charset=utf-8',  
 url: "http://localhost:8080/api/validateCaptcha/"+$("#captchaImg").attr("captchaId"),  
 beforeSend: function(xhr) {  
 xhr.setRequestHeader("Accept", "application/json");  
 xhr.setRequestHeader("Content-Type", "application/json");  
 xhr.setRequestHeader("jwtString", EuCaptchaToken);  
 },  
 data: jQuery.param({ captchaAnswer: $("#captchaAnswer").val() ,  
 useAudio : useAudio}) ,  
 contentType: 'application/x-www-form-urlencoded; charset=UTF-8',  
 cache: false,  
 timeout: 600000,  
 success: function (data) {  
 $("input").css({"border": ""});  
 obj = JSON.parse(data);  
 if ('success' == obj.responseCaptcha)  
 {  
 $("#success").css("visibility", "visible");  
 $("#fail").css("visibility", "hidden");  
 }   
  
 {  
 $("#fail").css("visibility", "visible");  
 $("#success").css("visibility", "hidden");  
 reloadCaptcha();  
 }  
 },  
 error: function (e) {  
 $("input").css({"border": "2px solid red"});  
 }  
 });  
 }  
  
 $("#captchaReload").click(function(){  
 $("#fail").css("visibility", "hidden");  
 $("#success").css("visibility", "hidden");  
 reloadCaptcha();  
 });  
   
 $("#captchaSubmit").click(function(){  
 validateCaptcha();  
 });  
  
 $('#captchaAnswer').keypress(function(e) {  
 if (e.keyCode == 13) {  
 validateCaptcha();  
 return false; *// prevent the button click from happening* }  
 });  
  
 $(document).ready(function () {  
 $("#dropdown-language").change(function () {  
 var selectedOption = $('#dropdown-language').val();  
 languageSelected = selectedOption;  
 if (selectedOption != '') {  
 window.location.replace('?lang=' + selectedOption);  
 }  
 });  
 });  
 getcaptcha();  
   
 });

else {  
 $("#fail").css("visibility", "visible");  
 $("#success").css("visibility", "hidden");  
 reloadCaptcha();  
 }  
 },  
 error: function (e) {  
 $("input").css({"border": "2px solid red"});  
 }  
 });  
 }  
  
 $("#captchaReload").click(function(){  
 $("#fail").css("visibility", "hidden");  
 $("#success").css("visibility", "hidden");  
 reloadCaptcha();  
 });  
   
 $("#captchaSubmit").click(function(){  
 validateCaptcha();  
 });  
  
 $('#captchaAnswer').keypress(function(e) {  
 if (e.keyCode == 13) {  
 validateCaptcha();  
 return false; *// prevent the button click from happening* }  
});  
  
$(document).ready(function () {  
 $("#dropdown-language").change(function () {  
 var selectedOption = $('#dropdown-language').val();  
 languageSelected = selectedOption;  
 if (selectedOption != '') {  
 window.location.replace('?lang=' + selectedOption);  
 }  
 });  
 });  
 getcaptcha();  
   
});

This JavaScript file provides every functionality that is needed to use the EU CAPTCHA:

* function *onPlayAudio()*: this function is called when the user uses the audio to solve the CAPTCHA, the application needs this to eliminate the case sensitive validation.
* function *getcaptcha()*: this function is called to generate the CAPTCHA combination to show in the image area on the web page.
* function *reloadCaptcha()*: this function is called to reload the CAPTCHA.
* function *validateCaptcha()*: this function is called to validate the CAPTCHA.
* *#captchaReload*: links to the <button class="btn btn-primary btn-lg " …> id in the web page and calls the reloadCaptcha function.
* *#captchaSubmit*: links to the <button class="btn btn-primary btn-lg " …> id in the web page and calls the validateCaptcha function.
* *#captchaAnswer*: prevents the call to validate the CAPTCHA when the enter button is pressed but there are less than 8 characters in the answer.

In the example the URL 'http://localhost:8080/’ is used. For a production environment this URL should be changed into the correct server URL and server port corresponding to your server setup and infrastructure. The server URL should be for the server where the application is deployed on. the server port should be the same as for which the application is listening for incoming messages (e.g. https//www.myserver.com: 8082).

In the example shown above, the code for this working example can be found in the CAPTCHA package, the following application logic is applied: if the CAPTCHA has succeeded, a green background is made visible and the message ‘EU CAPTCHA validation successful’ is shown to the user in the selected language. If the CAPTCHA has failed, a red background is made visible and the message ‘The text you have entered does not match, please try again.’ is shown to the user in the selected language. The default language used in the CAPTCHA package for messages and text is English.

When the CAPTCHA application is used in a production environment these messages will not be the behaviour you probably want for the application so there will be needed changes on the action’s success or failure. The EU CAPTCHA solution is designed so that, on success, the user will be able to be granted access to an application or will be forwarded to another webpage. For this to work properly, the JavaScript function validateCaptcha() must be further configured in the section ‘success’:

From the code snippet in the sample js file:

success: function (data) {  
 $("input").css({"border": ""});  
 obj = JSON.parse(data);  
 if ('success' == obj.responseCaptcha)  
 {  
 $("#success").css("visibility", "visible"); //makes the success message visible  
 $("#fail").css("visibility", "hidden"); // hides the validation message  
 }

As an example, this can be changed to:

Success: function(data) {

$("input").css({"border": ""});  
 obj = JSON.parse(data);  
 if ('success' == obj.responseCaptcha)  
 {  
 window.location.href= https://www.example.com/thankyou.html;  
 }

If the validation fails, the included sample code will reload the CAPTCHA image and shows a message that validation has failed. This message will be displayed on the top of the page.

The user can retry the CAPTCHA or reload the CAPTCHA image with a different combination of characters and numbers.

The place where the sample code shows the failure message may not be a good solution, so the position of this error message can be changed by moving the divider (*div*) with *id=”fail”* to any desired place in the web page. The ‘else’ part of the JavaScript function allows for the error message to become visible.

The inline CSS can be ignored because in a production environment this should be a separate file.

Snippet of the HTML code in the example from EUCAPTCHA.jsp:

<div class=" col-md-12 alert alert-danger" id="fail" role="alert" style="visibility : hidden">  
 <i class="fa fa-exclamation-triangle fa-3x" aria-hidden="true"></i>

<fmt:message key="euCaptcha.invalid" />  
</div>

Snippet of the restCaptcha.js:

else  
{  
$("#fail").css("visibility", "visible"); //makes the fail message visible  
$("#success").css("visibility", "hidden"); //hides the success message  
reloadCaptcha(); //reloads the CAPTCHA  
}

# EU CAPTCHA process

EU CAPTCHA [REST](https://en.wikipedia.org/wiki/Representational_state_transfer) API will allow you to generate, get, reload and validate [CAPTCHA](https://b2evolution.net/man/post) through the HTTP protocol (GET / POST methods). Responses are returned in JSON format.

Operations on resources of this CAPTCHA API are implemented with the standard methods of HTTP: *GET* to get or reload the CAPTCHA, *POST* to send the answer and get as a response the server validation of the user input. Each resource is represented as an URL, such as *http://server\_name:server\_port/api/captchaImg*.

## REST and JSON

REST is an acronym for “[REpresentational State Transfer](http://en.wikipedia.org/wiki/Representational_state_transfer)”. REST adopts a fixed set of operations on named resources, where the representation of each resource is the same for retrieving and setting information. In other words, you can retrieve (read) data in an JSON/XML format and send data back to the server in similar JSON/XML format in order to set (write) changes to the system.

JSON (JavaScript Object Notation) is a notation style to represent complex object structures in a serialized manner (i.e. transferable over the internet). It has the same role as XML but is much less verbose and therefore faster.

Please note that JSON is the only data format that is used in the EU CAPTCHA solution.

## Calls and requests

To make a basic call to the EU CAPTCHA REST API, a message must be sent to the API with the following information:

* The API URL
* The HTTP verb GET/POST
* The Content-Type (application/json, application/x-www-form-URL encoded; charset=UTF-8).
* CAPTCHA answer as parameter for validating the CAPTCHA.
* Pervious CAPTCHA ID as parameter in case of reloading the CAPTCHA.
* The default language is English. If you would like to use a different alphabet and audio language, you must add a language parameter (e.g. *http://localhost:8080/euCaptcha?lang=da*). Please refer to Language table 1 to find the correct parameter for the language of interest.

The API will then return a JSON-formatted response body.

* + 1. URL

The base URL of the API is *http://server\_name:server\_port*

* + 1. Getting the CAPTCHA

A basic HTTP request to the REST API may look like this:



Example:   
curl -X GET "http://localhost:8080/api/captchaImg" -H "accept: \*/\*"

The Response will be:   
{

"captchaId ": "the ID of the generated CAPTCHA",

"captchaImg “: **data: image/png;base64**

}



EU CAPTCHA is multilingual with support for all official languages from the European Union, in order to accomplish that you can make a request to the REST API with the language desired, the default language is English. However, you can send the code of the language as a query parameter. The request may look like the following example.

*http://server\_name:server\_port/api/captchaImg?lang=en*

To get the CAPTCHA for each language, an API URL should have the following format:

Language table 1

|  |  |
| --- | --- |
| **Language** | **URL** |
| Bulgarian | http://server\_name:server\_port/api/captchaImg?lang=bg |
| Croatian | http://server\_name:server\_port/api/captchaImg?lang=hr |
| Czech | http://server\_name:server\_port/api/captchaImg?lang=cs |
| Danish | http://server\_name:server\_port/api/captchaImg?lang=da |
| Dutch | http://server\_name:server\_port/api/captchaImg?lang=nl |
| English | http://server\_name:server\_port/api/captchaImg?lang=en |
| Estonian | http://server\_name:server\_port/api/captchaImg?lang=et |
| Finnish | http://server\_name:server\_port/api/captchaImg?lang=fi |
| French | http://server\_name:server\_port/api/captchaImg?lang=fr |
| German | http://server\_name:server\_port/api/captchaImg?lang=de |
| Greek | http://server\_name:server\_port/api/captchaImg?lang=el |
| Hungarian | http://server\_name:server\_port/api/captchaImg?lang=hu |
| Irish | http://server\_name:server\_port/api/captchaImg?lang=ga |
| Italian | http://server\_name:server\_port/api/captchaImg?lang=it |
| Latvian | http://server\_name:server\_port/api/captchaImg?lang=lv |
| Lithuanian | http://server\_name:server\_port/api/captchaImg?lang=lt |
| Maltese | http://server\_name:server\_port/api/captchaImg?lang=mt |
| Polish | http://server\_name:server\_port/api/captchaImg?lang=pl |
| Portuguese | http://server\_name:server\_port/api/captchaImg?lang=pt |
| Romanian | http://server\_name:server\_port/api/captchaImg?lang=ro |
| Slovak | http://server\_name:server\_port/api/captchaImg?lang=sk |
| Slovenian | http://server\_name:server\_port/api/captchaImg?lang=sl |
| Spanish | http://server\_name:server\_port/api/captchaImg?lang=es |
| Swedish | http://server\_name:server\_port/api/captchaImg?lang=sv |

The choice of the language has an impact on the alphabet that is used in the CAPTCHA image. The EU CAPTCHA solution uses the official alphabet for the chosen language. In other words, the CAPTCHA solution supports the extended Latin script (which is the script most languages use), the Greek script, and the Cyrillic script.

|  |  |
| --- | --- |
| **Language** | **Alphabet** |
| English, Dutch, French, Italian, Portuguese | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T','U', 'V', 'W', 'X', 'Y', 'Z' |
| Bulgarian | А', 'а', 'Б', 'б', 'В', 'в', 'Г', 'г', 'Д', 'д', 'Е', 'е', 'Ж', 'ж', 'З', 'з', 'И', 'и', 'Й', 'й', 'К', 'к', 'Л', 'л', 'М', 'м', 'Н', 'н', 'О', 'о', 'П', 'п', 'Р', 'р', 'С', 'с', 'Т', 'т', 'У', 'у', 'Ф', 'ф', 'Х', 'х', 'Ц', 'ц', 'Ч', 'ч', 'Ш', 'ш', 'Щ', 'щ', 'Ю', 'ю', 'Я', 'я', '0', '1', '2', '3', '4', '6', '7', '8', '9' |
| Croatian | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'r', 's', 't', 'u', 'v', 'z', 'A', 'B', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'R', 'S', 'T','U', 'V', 'Z', 'Č', 'Ć', 'Đ', 'Š', 'Ž', 'č', 'ć', 'đ', 'š', 'ž' |
| Czech | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T','U', 'V', 'W', 'X', 'Y', 'Z', 'Á', 'á', 'Č', 'č', 'Ď', 'ď', 'É', 'é', 'Í', 'í', 'Ň', 'ň', 'Ó', 'ó', 'Ř', 'ř', 'Š', 'š', 'Ť', 'ť', 'Ú', 'ú', 'Ý', 'ý', 'Ž', 'ž' |
| Danish | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T','U', 'V', 'W', 'X', 'Y', 'Z', 'Æ', 'æ', 'Ø', 'ø', 'Å', 'å' |
| Estonian | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T','U', 'V', 'W', 'X', 'Y', 'Z', 'š', 'ž', 'õ', 'ä', 'ö', 'ü', 'Š', 'Ž', 'Õ', 'Ä', 'Ö', 'Ü' |
| Finnish | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T','U', 'V', 'W', 'X', 'Y', 'Z', 'Å', 'å', 'Ä', 'ä', 'Ö', 'ö' |
| German | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T','U', 'V', 'W', 'X', 'Y', 'Z', 'Ä', 'Ö', 'Ü', 'ß', 'ä', 'ö', 'ü', 'ß' |
| Greek | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','Α', 'α', 'Β', 'β', 'Γ', 'γ', 'Δ', 'δ', 'Ε', 'ε', 'Ζ', 'ζ', 'Η', 'η', 'Θ', 'θ', 'Ι', 'ι', 'Κ', 'κ', 'Λ', 'λ', 'Μ', 'μ', 'Ν', 'ν', 'Ξ', 'ξ', 'Ο', 'ο', 'Π', 'π', 'Ρ', 'ρ', 'Σ', 'σ', 'Τ', 'τ', 'Υ', 'υ', 'Φ', 'φ', 'Χ', 'χ', 'Ψ', 'ψ', 'Ω', 'ω' |
| Hungarian | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T','U', 'V', 'W', 'X', 'Y', 'Z', 'Á', 'á', 'É', 'é', 'Í', 'í', 'Ó', 'ó', 'Ö', 'ö', 'Ő', 'ő', 'Ú', 'ú', 'Ü', 'ü', 'Ű', 'ű' |
| Irish | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T','U', 'V', 'W', 'X', 'Y', 'Z', 'á', 'é', 'í', 'ó', 'ú', 'Á', 'É', 'Í', 'Ó', 'Ú' |
| Latvian | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'r', 's', 't', 'u', 'v', 'z', 'A', 'B', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'R', 'S', 'T','U', 'V', 'Z', 'Ā', 'Č', 'Ē', 'Ģ', 'Ī', 'Ķ', 'Ļ', 'Ņ', 'Š', 'Ū', 'Ž', 'ā', 'č', 'ē', 'ģ', 'ī', 'ķ', 'ļ', 'ņ', 'š', 'ū', 'ž', 'C', 'c' |
| Lithuanian | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'r', 's', 't', 'u', 'v', 'z', 'A', 'B', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'R', 'S', 'T','U', 'V', 'Z', 'Ą', 'Č', 'Ę', 'Ė', 'Į', 'Š', 'Ų', 'Ū', 'Ž', 'ą', 'č', 'ę', 'ė', 'į', 'š', 'ų', 'ū', 'ž', 'c', 'C' |
| Maltese | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'r', 's', 't', 'u', 'v', 'z', 'A', 'B', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'R', 'S', 'T','U', 'V', 'Z', 'Ċ', 'Ġ', 'Ħ', 'Ż', 'ċ', 'ġ', 'ħ', 'z', 'ż' |
| Polish | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'r', 's', 't', 'u', 'v', 'z', 'A', 'B', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'R', 'S', 'T','U', 'V', 'Z', 'Ą', 'Ć', 'Ę', 'Ł', 'Ń', 'Ó', 'Ś', 'Ź', 'Ż', 'ą', 'ć', 'ę', 'ł', 'ń', 'ó', 'ś', 'Y', 'y', 'ź', 'ż' |
| Romanian | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T','U', 'V', 'W', 'X', 'Y', 'Z', 'Ă', 'ă', 'Â', 'â', 'Î', 'î', 'Ș', 'ș', 'Ț', 'ț' |
| Slovakian | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T','U', 'V', 'W', 'X', 'Y', 'Z', 'Ä', 'Č', 'Ď', 'Í', 'Ľ', 'Ô', 'Š', 'Ť', 'Ž', 'ä', 'č', 'ď', 'í', 'ľ', 'ô', 'š', 'ť', 'ž' |
| Slovenian | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'r', 's', 't', 'u', 'v', 'z', 'A', 'B', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'R', 'S', 'T','U', 'V', 'Z','Č', 'č', 'c', 'C', 'Š', 'š', 'Ž', 'ž' |
| Spanish | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T','U', 'V', 'W', 'X', 'Y', 'Z', 'Ñ', 'ñ' |
| Swedish | '0', '1', '2', '3', '4', '5', '6', '7', '8', '9','a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T','U', 'V', 'W', 'X', 'Y', 'Z', 'Å', 'å', 'Ä', 'ä', 'Ö', 'ö' |

* + 1. Reload the CAPTCHA

The request for reloading CAPTCHA may look like this:



Example:   
curl -X GET "http://localhost:8080/api/reloadCaptchaImg/h2e7ofq3htb3efkptarlj889ht " -H "accept: \*/\*"

The Response will be:   
{

"captchaId ": "the ID of the new CAPTCHA",

"captchaImg “: **data: image/png;base64**

}

* + 1. Validate the CAPTCHA



# Deployment

EU CAPTCHA is a lightweight application and easy to deploy. An Apache Tomcat server gives us multiple ways to deploy a WAR (Web Archive). This is the default file when the code is compiled from the EU CAPTCHA project. This can be downloaded from:

## Deploy by copying archive:

To deploy, it is recommended to drop the WAR file into the *$CATALINA\_HOME\webapps* directory of any Tomcat instance. If the instance is running, the deployment will start instantly as Tomcat unpacks the archive and configures its context path.

If the instance is not running, then the server will deploy the project the next time it is started.

## Deploy from Tomcat Manager

Before continuing, make sure to have obtained the WAR file. To continue deploying EU CAPTCHA, it is recommended to use the manager dashboard. You can access the manager dashboard by visiting http://localhost:8080/manager.

The dashboard has five different sections: Manager, Applications, Deploy, Diagnostics, and Server Information. If you go to the Deploy section, you will find two subsections. These are explained below.

### ****Deploy Directory or WAR File Located on Server****

Assuming the WAR file is located on the server where the Tomcat instance is running, please fill in the required *Context Path* field preceded by a forward slash “/”. To be able to access the web application from a browser using the URL *http://localhost:8080/EU-CAPTCHA*, the context path field should be */EU-CAPTCHA.*

Next, skip the XML Configuration file URL field and head over to the WAR or Directory URL field. In this section, it is required to enter the absolute URL to the WAR (Web ARchive) file as it appears on the server. For example, if the file's location is *C:/apps/EU-CAPTCHA.war*, you should enter this location. Do not forget the .war extension.

Afterwards, please click the *deploy* button. The page will reload, and you should see the following message at the top of the page.

*OK – Deployed application at context path /EU-CAPTCHA*

Additionally, the application should also appear in the *Applications* section of the page.

### ****WAR File to Deploy****

Click the choose file button, then navigate to the location of the WAR file and select it. To finish, click the deploy button.

This concludes the deployment of EU CAPTCHA. In both situations, the Tomcat console will inform us that the deployment has been successful with the following message.

*INFO: Deployment of web application archive \path\to\ EU-CAPTCHA.war has finished in 4,833 ms*