**EU CAPTCHA**

Managed Service documentation and implementation manual

**Contents**

[1. Introduction 3](#_Toc73446732)

[1.1. Who is this document for? 3](#_Toc73446733)

[1.2. What is a CAPTCHA 3](#_Toc73446734)

[1.3. Features 3](#_Toc73446735)

[2. Integration 4](#_Toc73446736)

[2.1. URL 4](#_Toc73446737)

[2.2. Required files 4](#_Toc73446738)

[2.3. Website integration 4](#_Toc73446739)

[2.3.1. Required for both EU CAPTCHAs 4](#_Toc73446740)

[2.3.2. Alphanumeric CAPTCHA 5](#_Toc73446741)

[2.3.3. Image rotation CAPTCHA 7](#_Toc73446742)

[2.4. Style 8](#_Toc73446743)

[3. Integration (Angular) 8](#_Toc73446744)

[4. EU CAPTCHA process 9](#_Toc73446745)

[4.1. REST and JSON 9](#_Toc73446746)

[4.2. Calls and requests 9](#_Toc73446747)

[4.2.1. Alphanumeric CAPTCHA 9](#_Toc73446748)

[**4.2.1.1.** **Getting the CAPTCHA** 10](#_Toc73446749)

[**4.2.1.2.** **Reload the CAPTCHA** 13](#_Toc73446750)

[**4.2.1.3.** **Validate the CAPTCHA** 14](#_Toc73446751)

[4.2.2. Image rotation CAPTCHA 14](#_Toc73446752)

[**4.2.2.1.** **Adding Images** 14](#_Toc73446753)

[**4.2.2.2.** **Getting the CAPTCHA** 15](#_Toc73446754)

[**4.2.2.3.** **Reload the CAPTCHA** 15](#_Toc73446755)

[**4.2.2.4.** **Validate the CAPTCHA** 16](#_Toc73446756)

# Introduction

## Who is this document for?

This CAPTCHA Managed Service Implementation Manual is for the reference of developers wanting to use the EU CAPTCHA Managed Service.

## 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](https://github.com/pwc-technology-be/eu-captcha) 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 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 EU CAPTCHA Managed Service 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 via integration configuration.

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.

# Integration

## URL

To integrate with test instance of EU CAPTCHA Managed Service we will be using the following URL:

**QA environment:** [**https://testapi.eucaptcha.eu/**](https://testapi.eucaptcha.eu/)

**PRD environment:** [**https://api.eucaptcha.eu/**](https://api.eucaptcha.eu/)

## Required files

The files required for client site EU CAPTCHA Managed Service integration can be found in the following folder in the EU CAPTCHA GitHub repository.

[eu-captcha/clientfiles/html-javascript\_client/](https://github.com/pwc-technology-be/eu-captcha/tree/master/clientfiles/html-javascript_client)

As of the time of the creation of this document, the EU CAPTCHA Managed Service supports two types of CAPTCHA:

* Alphanumeric CAPTCHA
* Image rotation CAPTCHA

Depending on which CAPTCHA you want to integrate you may need just a subset of the client files in [GitHub](https://github.com/pwc-technology-be/eu-captcha/tree/master/clientfiles/html-javascript_client) repository. For your convenience, we provided a sample page for each version

* index.html (Alphanumeric CAPTCHA)
* rotate.html (Image rotation CAPTCHA)

## Website integration

### Required for both EU CAPTCHAs

To include the alphanumeric or image rotation CAPTCHA in your web page and integrate with the EU CAPTCHA managed service, you should add the following code in the head section of your web page.

<head>  
 <meta http-equiv="content-type" content="text/html; charset=utf-8">

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>  
 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>  
 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">  
 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI" crossorigin="anonymous"></script>  
 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<link rel="stylesheet" href="***<path to your css files>***/font-awesome-4.7.0/css/font-awesome.min.css">  
 <link rel="stylesheet" href="***<path to your css files>***/jquery-ui.min.css">

<script src="***<path to your javascript files>***/jquery-1.2.min.js"></script>  
</head>

### Alphanumeric CAPTCHA

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

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

restCaptcha.js file provides every functionality that is needed to use the Alphanumeric CAPTCHA version:

* variable *captchaLength*: you can use this variable to configure the length of 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.
  + ***Please note that, per CAPTCHA, as soon as the audio is played once, the case sensitivity will not be checked during validation for that specific CAPTCHA.***
* 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.

Please update the “url:” parameter in each function with the URL specified in section 2.1. For example:

function getcaptcha(){  
…  
QA <url:'https://testapi.eucaptcha.eu/api/captchaImg?captchaLength='+captchaLength>,

PRD

<url:'https://api.eucaptcha.eu/api/captchaImg?captchaLength='+captchaLength>,   
  
…  
}

function reloadCaptcha(){  
…  
QA

url: 'https://testapi.eucaptcha.eu/api/reloadCaptchaImg/' + $("#captchaImg").attr("captchaId") + '/?lang=' + sessionStorage.getItem("language") + '&captchaLength='+ captchaLenght,

PRD

url: 'https://api.eucaptcha.eu/api/reloadCaptchaImg/' + $("#captchaImg").attr("captchaId") + '/?lang=' + sessionStorage.getItem("language") + '&captchaLength='+ captchaLenght,

…  
}

function validateCaptcha(){  
…  
QA

url: ‘https://testapi.eucaptcha.eu/api/validateCaptcha/’ + $("#captchaImg").attr("captchaId"),

PRD

url: ‘https://api.eucaptcha.eu/api/validateCaptcha/’ + $("#captchaImg").attr("captchaId"),  
…  
}

In the included sample index.html page 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 have the behavior you probably want for the application. This means there will be changes required when the validation is successful or has failed. The EU CAPTCHA solution is designed so that, on successful validation of the CAPTCHA, 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 show 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 index.html:

<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  
}

### Image rotation CAPTCHA

To complete the Image rotation CAPTCHA integration, you should also make the eu-captcha-style.css and restWhatsUpCaptcha.js file available on the web page by adding the following line in the head section of your web page.

<head>

<link rel="stylesheet" href="<path to your css files>/eu-captcha-style.css">

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

</head>

To solve the rotation CAPTCHA there are 2 possibilities, one with buttons and another with a slider. It is not possible to use them alternately because the one with buttons is controlled by jQuery and the one with the slider is controlled by the restWatsUpCaptcha.js. To offer a user both possibilities there is a toggle between them, on start up it will be always the buttons that are shown and above the image there will be a button named ‘Slider’ to toggle to the slider validation. When the slider is shown there will be a button above the image named ‘Arrows’ to switch back to the button solution. Every time a user toggles between the 2 possibilities the image will be reloaded.

! For mobile users we recommend the use of the buttons because the slider will manipulate the whole screen and not only the slider !

restWhatsUpCaptcha.js file provides every functionality that is needed to use the Image rotation CAPTCHA version:

* function *getWhatsUpcaptcha()*: this function is called to rotate and load the CAPTCHA image to show in the image area on the web page.
* function *reloadCaptcha()*: this function is called to reload a new CAPTCHA image.
* function *validateCaptcha()*: this function is called to validate the CAPTCHA.
* function *rotate()*: this function is called to rotate the CAPTCHA image.

To integrate with EU CAPTCHA Managed Service please update url: in mentioned above functions to include https://testapi.eucaptcha.eu/

Example:

QA

url: 'https://testapi.eucaptcha.eu/api/captchaImg?captchaType=WHATS\_UP',

PRD

url: 'https://api.eucaptcha.eu/api/captchaImg?captchaType=WHATS\_UP',

QA

url: ' https://testapi.eucaptcha.eu/api/reloadCaptchaImg/' + $("#captchaImage").attr("captchaId")+ "?captchaType=WHATS\_UP",

PRD

url: ' https://api.eucaptcha.eu/api/reloadCaptchaImg/' + $("#captchaImage").attr("captchaId")+ "?captchaType=WHATS\_UP",

QA

url: " https://testapi.eucaptcha.eu/api/validateCaptcha/" + $("#captchaImage").attr("captchaId"),

PRD

url: " https://api.eucaptcha.eu/api/validateCaptcha/" + $("#captchaImage").attr("captchaId"),

## Style

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

The example below is for Alphanumeric CAPTCHA but styling can be applied to image rotation captcha as well

The jsp code snippet contains:

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

# Integration (Angular)

Prerequisites:

A complete local setup overview can be found at ‘https://angular.io/guide/setup-local’

These are the keypoints:

* Download Node.js (<https://nodejs.org/en/>)
* Install Angular CLI: open a terminal window and run npm install -g @angular/cli

To run the Angular example application navigate to ‘clientfiles\Eu-Captcha-Angular’ in a terminal window.

Run npm install

Run npm serve

The application will be available at: ‘localhost:4200’

# 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

QA[*https://testapi.eucaptcha.eu/api/captchaImg*](https://testapi.eucaptcha.eu/api/captchaImg)*.*

PRD[*https://api.eucaptcha.eu/api/captchaImg*](https://api.eucaptcha.eu/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

### Alphanumeric CAPTCHA

To make a basic call to the Alphanumeric 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.
* Previous 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. *https://testapi.eucaptcha.eu/?lang=bg*). Please refer to Language table 1 to find the correct parameter for the language of interest.
* The default length is set to 10. If you would like to use an other length you must add a length parameter (e.g. *https://testapi.eucaptcha.eu/?* *captchaLength=4*)
* In the example UI screens there is a checkbox ‘Capitalized’ to switch between either **a mixture of UPPER and lower case characters** and **only lower case characters**. When the checkbox is checked, the backend will return a mixture of characters. This can be overwritten (and left out of the UI) by adding a parameter to the URL (e.g. *https://testapi.eucaptcha.eu/?* *Capitalized=false*) where the value ‘true’ returns a mixture and ‘false’ returns only lower case characters.

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

#### **Getting the CAPTCHA**

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

GET: api/captchaImg

Example:   
QA: curl -X GET "https://testapi.eucaptcha.eu/api/captchaImg" -H "accept: \*/\*"

PRD: curl -X GET "https://api.eucaptcha.eu/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.

*QA -* [*https://testapi.eucaptcha.eu/api/captchaImg?lang=en*](https://testapi.eucaptcha.eu/api/captchaImg?lang=en)

*PRD -* [*https://api.eucaptcha.eu/api/captchaImg?lang=en*](https://api.eucaptcha.eu/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' |

#### **Reload the CAPTCHA**

The request for reloading CAPTCHA may look like this:

GET: /api/reloadCaptchaImg/{previousCaptchaId}

! ‘previousCaptchaId’ should be the same ID that is returned by the getCaptcha execution.

Example:   
QA - curl -X GET "https://testapi.eucaptcha.eu/api/reloadCaptchaImg/h2e7ofq3htb3efkptarlj889ht" -H "accept: \*/\*"

PRD - curl -X GET "https://api.eucaptcha.eu/api/reloadCaptchaImg/h2e7ofq3htb3efkptarlj889ht" -H "accept: \*/\*"

The Response will be:   
{

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

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

}

#### **Validate the CAPTCHA**

POST: api/validateCaptcha/{captchaId}

! ‘captchaId’ should be the same ID that is returned by the getCaptcha or reloadCaptcha execution.

Example:   
QA - curl -X POST "https://testapi.eucaptcha.eu/api/validateCaptcha/h2e7ofq3htb3efkptarlj889ht" -H "accept: \*/\*"

PRD - curl -X POST "https://api.eucaptcha.eu/api/validateCaptcha/h2e7ofq3htb3efkptarlj889ht" -H "accept: \*/\*"

In the body of the validateCaptcha you will need 3 values as x-www-form-urlencoded:

* captchaAnswer
* useAudio
* x-jwtString (this can be found in the header of the getCaptchacall)

The Response will be:

{"responseCaptcha":"success"} or

{"responseCaptcha":"fail"}

### Image rotation CAPTCHA

#### **Adding Images**

With the example implantation there are 7 images included which are randomly chosen and the application logic calculates an angle to show them to the user.

There is always a possibility to add extra images, the folder used to choose a random image is: ‘eu-captcha/src/main/resources/captchaImages/’

Every image can be added but we recommend using png images with a maximum of 600kb for a high performance.

The image does not have to be circle shaped, it is possible to use every image and shape them with ‘<https://crop-circle.imageonline.co/>’ which is an online free to use tool.

#### **Getting the CAPTCHA**

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

GET: api/captchaImg?captchaType=WHATS\_UP

Example:   
QA - curl -X GET "https://testapi.eucaptcha.eu/api/captchaImg?captchaType=WHATS\_UP" -H "accept: \*/\*"

PRD - curl -X GET "https://api.eucaptcha.eu/api/captchaImg?captchaType=WHATS\_UP" -H "accept: \*/\*"

The Response will be:   
{

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

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

}



#### 

#### **Reload the CAPTCHA**

The request for reloading CAPTCHA may look like this:

GET: /api/reloadCaptchaImg/{previousCaptchaId}?captchaType=WHATS\_UP"

! ‘previousCaptchaId’ should be the same ID that is returned by the getCaptcha execution.

Example:   
QA curl -X GET "https://testapi.eucaptcha.eu/api/reloadCaptchaImg/h2e7ofq3htb3efkptarlj889ht?captchaType=WHATS\_UP" -H "accept: \*/\*"

PRD curl -X GET "https://api.eucaptcha.eu/api/reloadCaptchaImg/h2e7ofq3htb3efkptarlj889ht?captchaType=WHATS\_UP" -H "accept: \*/\*"

The Response will be:   
{

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

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

}

#### **Validate the CAPTCHA**

The request for validation CAPTCHA may look like this:

POST: api/validateCaptcha/{captchaId}

! ‘captchaId’ should be the same ID that is returned by the getCaptcha or reloadCaptcha execution.

Example:

QA - curl -X POST "https://testapi.eucaptcha.eu/api/validateCaptcha/h2e7ofq3htb3efkptarlj889ht" -H "accept: \*/\*"

PRD - curl -X POST "https://api.eucaptcha.eu/api/validateCaptcha/h2e7ofq3htb3efkptarlj889ht" -H "accept: \*/\*"

The Response will be:

{"responseCaptcha":"success"} or

{"responseCaptcha":"fail"}