

External Form via InfoCapture

### Overview

It is possible to export a form you've created in InfoCapture into HTML format which you can then embed or publish on your public website. This method allows the general public to submit data just like a regular contact form but the data and processing will be done within InfoCapture.

## Limitations

Please be advised there are several limitations involved in creating a form for external use and its upkeep will require knowledge to manipulate HTML code and CSS.

If you wish to make changes to the form you will have to do so in the Intranet and then re-export the form externally to update this on your website/other.

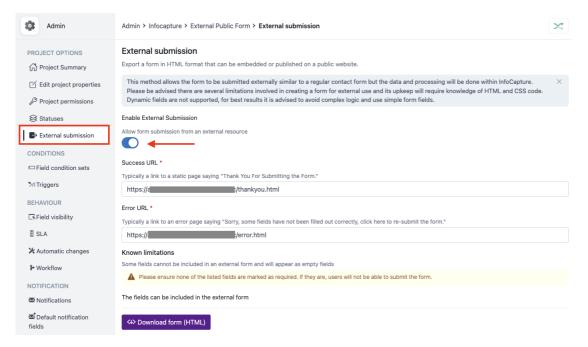
The following advanced InfoCapture features DO NOT work with external forms:

- Dynamic form fields
- Automatic show and hide fields
- Inline form validation
- Automatic field data population such as name and email address
- Any custom InfoCapture plugins

Due to the above limitations, any external form is required to be devoid of complex logic and overall use simple fields.

# How to export an External Form

1. After creating a form, you can export the form by enabling External Submission from the left menu



2. Specify URL for:

Success URL - this is typically a static page saying "Thank You For Submitting the Form." which you may have to recreate on your public website.

Error URL - this is typically an error page saying "Sorry, some fields have not been filled out correctly, click here to re-submit the form."

This will be triggered when there are errors during the form submission e.g. Mandatory fields not completed

- 3. Click Download form (HTML) to download the External form in HTML file format, which you can upload or embed on your public website.
- 4. In cases where the general public only needs to complete a small part of a longer form you can safely remove non-required fields directly in the HTML codes to allow simple initial submission (it's extremely important to only remove fields that are not mandatory so the form logic is not compromised).

Once in the Intranet, the ticket can be put through a more complex workflow if required by your project.

5. Test the form submissions

# Protecting from spam and bots

Public forms are prone to abuse from spammers and bots, we strongly advise implementing solutions such as reCAPTCHA to protect your forms or other solutions such as match captcha or honeypot.



## 1. Adding Google reCAPTCHA to the Exported HTML form

Step 1: Register to Google reCAPTCHA and follow the instruction to get the SITE KEY

Step 2: Place the following code within the <head> section

```
<!-- reCaptcha -->
<script src="https://www.google.com/recaptcha/api.js" async defer></script>
<!-- end reCaptcha -->
```

Place the ReCaptcha code above the submit button. Replace YOUR\_SITE\_KEY with the actual Site Key

```
<!-- reCaptcha -->
<div class="g-recaptcha"
data-sitekey="YOUR_SITE_KEY"
data-callback="enableSubmit"
data-expired-callback="disableSubmit">
</div>
<!-- end ReCaptcha -->
```

Modify the submit button to make it disabled by default and add id="submitButton" for the JavaScript code

```
<input type=submit disabled id="submitButton" class=submit_bttn value="Submit" />
```

Add the JavaScript before the end of the </body> tag

```
!-- reCaptcha JavaScript Functions for reCAPTCHA callbacks -->
<script>
const submitButton = document.getElementById('submitButton');

/**
    * Called by reCAPTCHA when the user successfully checks the box.
    */
function enableSubmit() {
        submitButton.disabled = false;
        console.log("reCAPTCHA verified. Submit button enabled.");
}

/**
    * Called by reCAPTCHA if the user's verification expires.
    */
function disableSubmit() {
        submitButton.disabled = true;
        console.log("reCAPTCHA expired. Submit button disabled.");
}

*/
console.log("reCAPTCHA expired. Submit button disabled.");
}

*/
console.log("reCAPTCHA expired. Submit button disabled.");
}
```

#### Limitation:

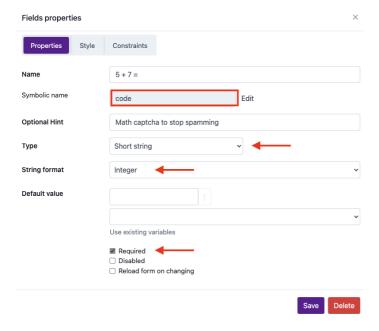
• This JavaScript technique to disable the submit button may not be effective against bots

#### 2. Adding math Captcha

Math Captcha serves as an alternative solution to authenticate human users on websites by employing simple math problems added to the form.

Step 1: Modify the form by adding a new field for the math question

Here is an example for 5+7 = with Symbolic name "code"



Step 2. Add Constraint, which is the answer of the math question.

Fields properties		×
Properties Style	Constraints	
Constraint: reg.exp.		
Constraint: value >=	12	
Constraint: value <=	12	
Constraint: size >=		
Constraint: size <=		
	Save D	elete

Step 3: Add the following code to the HTML of the public form within the <form> tag and position it above the Submit button

Change the value as required. XXX = Project ID in InfoCapture

code is the symbolic name of the field

```
<!-- math captcha -->
<label for=_XXX__field_code>
5 + 7 =
</label>
<input type="text" name="_XXX_field_short_stringcode" class="form-control" id="_XXX__field_code" value="">
<!-- end math captcha -->
```

#### Limitation:

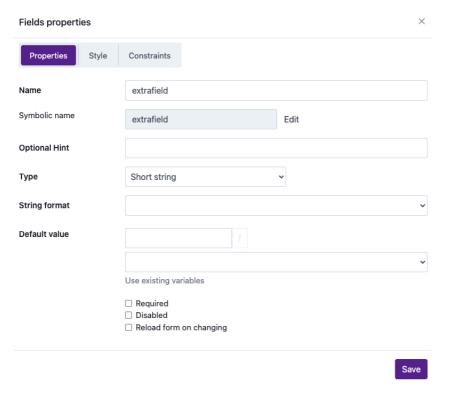
- A sophisticated spam bot may be able to bypass or answer the math question
- It assumes that users have knowledge of mathematics
- You may need to change the valid answer from time to time to be effective

## 3. Adding Honeypot

A honeypot is a hidden form field that uses CSS or JavaScript to make it invisible to real users but detectable by spam bots. The idea is that the user will leave this field blank, but the bot may populate it with some values.

Step 1: Modify the form by adding a new field for the honeypot field

Here is an example: we use the name "extrafield" to make it generic



Step 2. Add a Constraint to expect this form isn't populated, which is a regular expression ^\$



Step 3: Add the following code to the HTML of the public form within the <form> tag and position it above the Submit button

Change the value as required. XXX = Project ID in InfoCapture

extrafield is the symbolic name of the field

We added CSS positioning so that the form isn't visible to the normal user.

```
<!-- extrafield -->
<div style="position:absolute; left:-9999px; top: -9999px;">
<label for=_XXX__field_extrafield>
Comments
</label>
<input type="text" name="_XXX_field_short_stringextrafield" class="form-control" id="_XXX__field_extrafield" value="">
</div>
<!-- end extrafield -->
```

#### Limitations:

- Only effective against a bot, malicious actors can easily bypass honeypot requirements
- $\bullet\,$  A sophisticated spam bot may be able to bypass the CSS techniques to hide the field

Created on 6 September 2022 by Michael Christian. Last modified on 17 October 2025

Tags: Form, public