FORMCENTRIC

Studio User Manual

Version 2506.0.1.0

Formcentric for CoreMedia: Studio User Manual

Schaartor 1, 20459 Hamburg
Germany

Disclaimer

We reserve the right to alter the software and the contents of the manual without notice. We accept no liability for the accuracy of the contents of the manual, nor for any losses that may result from the use of this software.

Trademarks

In the course of this manual, references may be made to trademarks that are not explicitly marked as such. Even if such a mark is not given, the reader should not conclude that the name is free of third-party rights.

Please note: Printed copies are uncontrolled when printed.

Access to documentation

You can always find the latest version of the user manual in the Help centre help.formcentric.com. Older versions and additional information are available in the Formcentric Helpdesk helpdesk.formcentric.com.

1. Introduction

“Formcentric for CoreMedia™” extends the CoreMedia Content Cloud to include a powerful form editor with which you can create and edit any type of web-based form. The CAE integration also included with the product is used to ensure the dynamic display and processing of the forms that you create.

1.1. Overview of functions

Form elements: Formcentric supports all the form elements defined in the HTML standard, such as input fields, drop-down lists or buttons. Other form elements are also provided, such as a *CAPTCHA, Calculated Value* or *Summary*.

Multi-page forms: Detailed or complex forms can be split up over multiple form pages. Form users can then page back and forth between the individual form pages in order to change or add the data they have entered.

Conditions: With the help of conditions, you can design your forms so that the state of individual form elements can be modified dynamically based on the input entered by the user and even entire form pages can be shown or hidden.

Field validators: A range of validators is available to you for validating user input. You specify the validator to use for a form field (if any) directly in the editing area for the corresponding form field. Most validators will also give you the option of configuring additional settings to adjust the validator more precisely to your individual requirements. As a rule, all validation of input takes place on the server, to avoid form data manipulation.

Default field values: You can give input fields default values, which can be fixed, variable or user-specific.

Freely-definable actions: By selecting a processing mode (action) on the Actions screen, you can decide how the form data will be processed after submission. Out of the box, Formcentric comes with the following actions: *Send email*, *Formcentric Analytics*, *Forward to*, *PDF* and *Webhook*. Other actions, customised to suit special requirements, can be developed with the help of an easy-to-use programming interface (API).

Security: Formcentric contains a security servlet filter as a safeguard against cross-site scripting (XSS) attacks and cross-site request forgery (XSRF) attacks. This filter removes illegal HTML tags, CSS and scripts from the form data submitted. The filter also checks to confirm that the form data contains a valid XSRF token.

1.2. Terminology

This manual makes use of the following terms:

Term	Description
Form	An HTML web form displayed in a web browser.
Form element	Elements used when constructing a form (input fields, drop-down lists, check boxes, CAPTCHAs, etc.).
Form field	A specific input field in which the user can enter data such as their name or telephone number.
Form data	The data entered into the form by the user.
Form author	The person that creates and edit forms.
User	The person who fills out a form.
Editor	The form editor in CoreMedia Studio.
Frontend	The website created with the CoreMedia Content Cloud.

2. Form overview

In the main menu on the CoreMedia header bar, you can use the Formcentric logo to access an overview of your forms. This shows you all of the forms in the system and gives you the option of creating a new form or searching/filtering the existing forms. You can also access the Editor directly and view the version history.

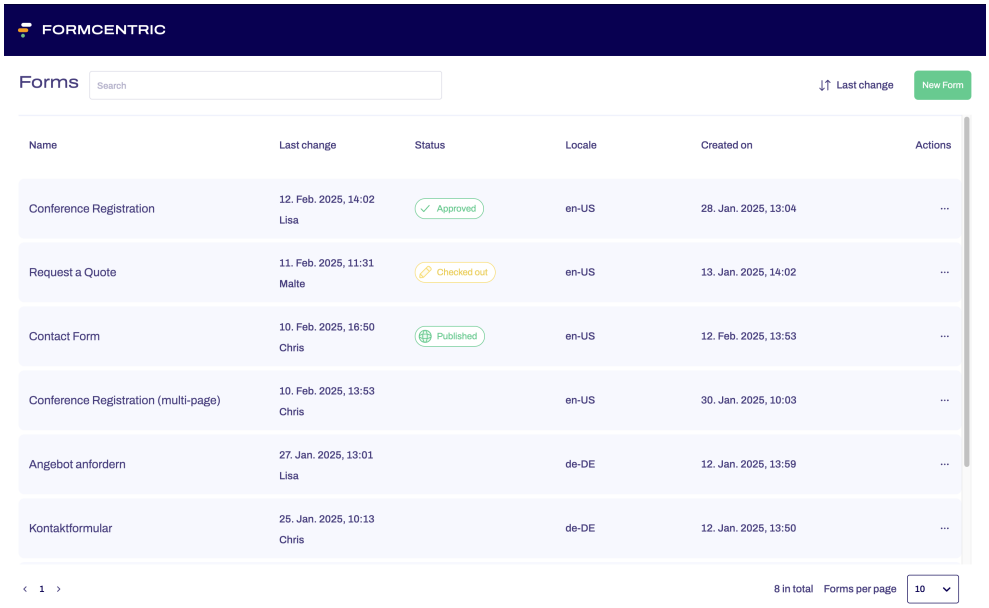


Figure 2.1. Form overview

2.1. Overview layout

The overview lists the forms and provides you with the following information:

Name: The name of the form. Click the form name to open the form in the Editor.

Last change: Displays the date and time of the last change that was made.

Status: Show the current form status of the form. A form can have the status **Checked out**, **Approved** or **Published**.

Locale: Shows the locale that is assigned to the form.

Created on: Records the creation date for the form.

Actions: Use this to carry out other actions for the form.

2.2. Sorting and search

You can sort the list of forms according to the following criteria:

Last change: The forms are sorted chronologically by the date of the last change.

Alphabetical (A–Z or Z–A): The forms are sorted in ascending or descending alphabetical order.

Click to change the sort order.

You can use the search field to search for a specific form by name.

2.3. Creating forms

The forms overview also gives you the option of creating new forms.

Click the **New form** button at the top right. In the dialog that is shown, select the target folder that you want to use and then confirm with **Save**. The Editor opens afterwards automatically, so that you can start working on your form straight away.

2.4. Available actions

You can carry out the following actions from the context menu link or by right-clicking the form name:

Open form: Opens the form in the Editor on a new tab.

Open in Studio : Switches to the CoreMedia workspace and displays the form on a new tab.

Show in library: Navigates to the place where the form is stored in the CoreMedia workspace.

Version history: Enables read-only access to earlier versions of the form. To restore an older version, switch to the CoreMedia workspace.

3. Editing interface

In the Formcentric Editor, the editing interface is split over three screens:

- Form [1]
- Actions [2]
- Settings [3]

You create and manage your forms on the *Form* screen. You specify how the data entered by users should be processed on the *Actions* screen. You define properties that apply to the whole form you are creating on the *Settings* screen. Click the name of the screen to switch to working on that screen.

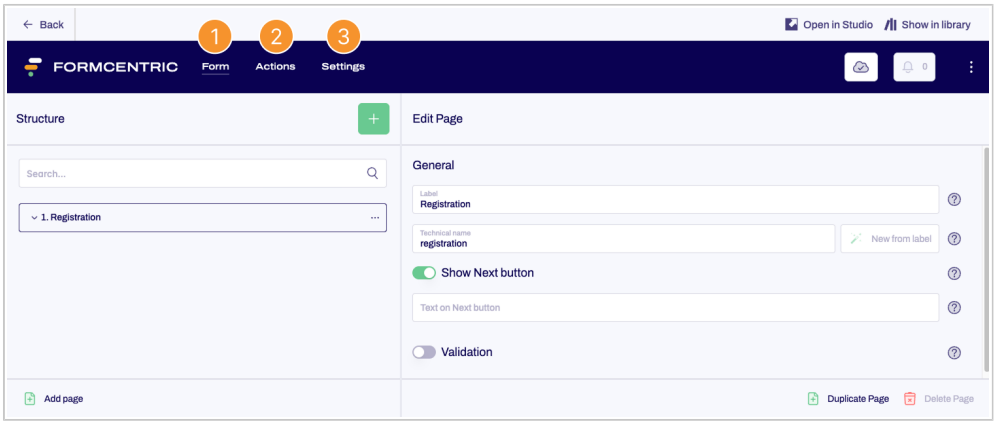


Figure 3.1. Editing interface

Above the editing interface, you have the functions *Back* [4], *Open in Studio tab* [5] and *Show in library tab* [6].

Use *Back* to go back to the CoreMedia Studio user interface. Click *Open in Studio tab* to open the form you are currently editing on a new tab in CoreMedia Studio. Click *Show in library* to open the CoreMedia Studio Library and display the location where the form is currently stored.

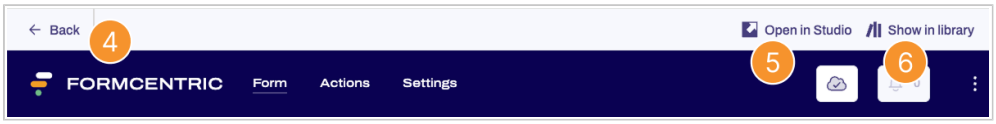


Figure 3.2. Menu tabs

3.1. General

3.1.1. Checking the forms created

The Editor checks the forms you create while their details are being entered. If any form elements are incomplete or have errors, these are marked with a red dot in the form tree.

A bell icon, indicating the current state of the form, is also shown in the Editor on the top right. If the form has errors, the bell turns red and the number of errors is displayed.

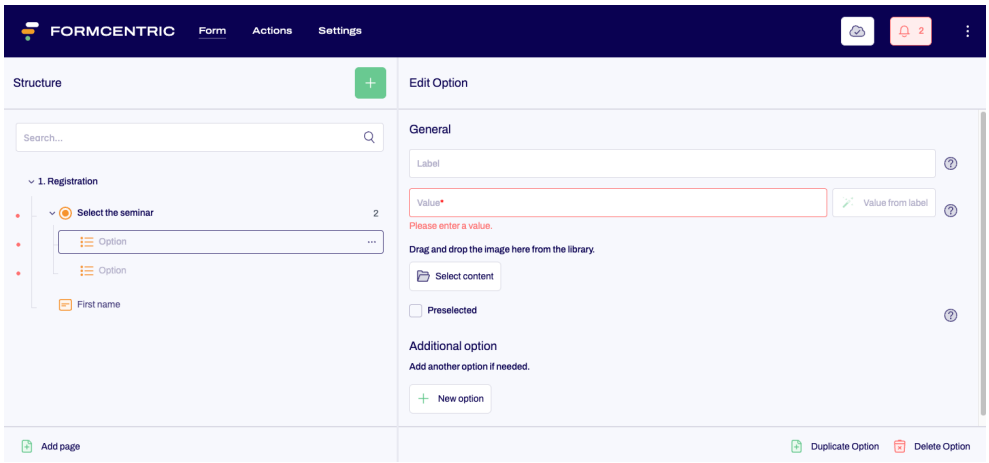


Figure 3.3. An error marked on the screen

Clicking the bell opens a dialogue that lists all of these errors. Click one of the errors in this list to go directly to the corresponding form element.

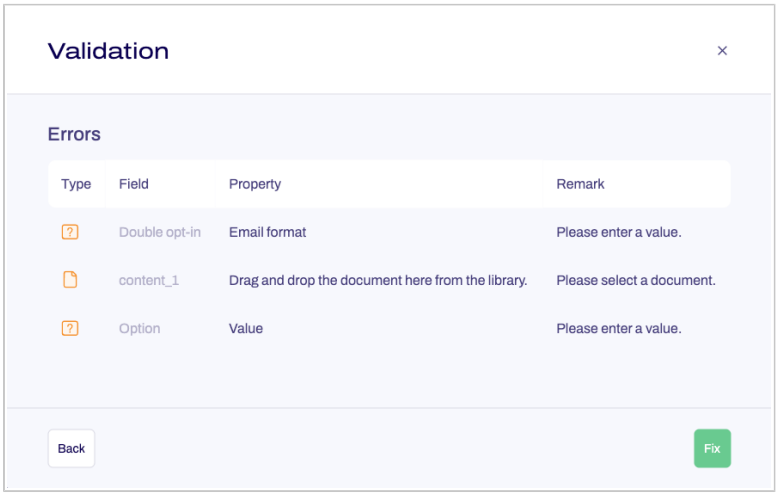


Figure 3.4. Errors in form

3.1.2. Autosave

All changes that you make to a form are saved automatically by Formcentric. This lets you concentrate on your work and means you do not have to worry about making regular saves. The cloud icon shown on the right indicates whether the form is currently saved. You can also click the cloud icon to save the form manually.

3.1.3. Predefined form elements

Predefined form elements – such as phone number, email address, file upload, etc. – give you a way to create your forms more quickly. You can add these form elements

to your form with just a single click. The validation required for each of these form elements is already activated, and you can start adding your details immediately.

3.1.4. Drag-and-drop

To make it easier for you to create your forms, you can use drag-and-drop to move individual form elements and even entire form pages around in the form tree. You can also move form elements from one form page to another. You can change the order of form elements and form pages simply by dragging these elements and pages to a different position in the form tree.

3.1.5. Preview

Within the CoreMedia Studio workspace, the preview shows you the current version of your form, which you can then try out at any time.

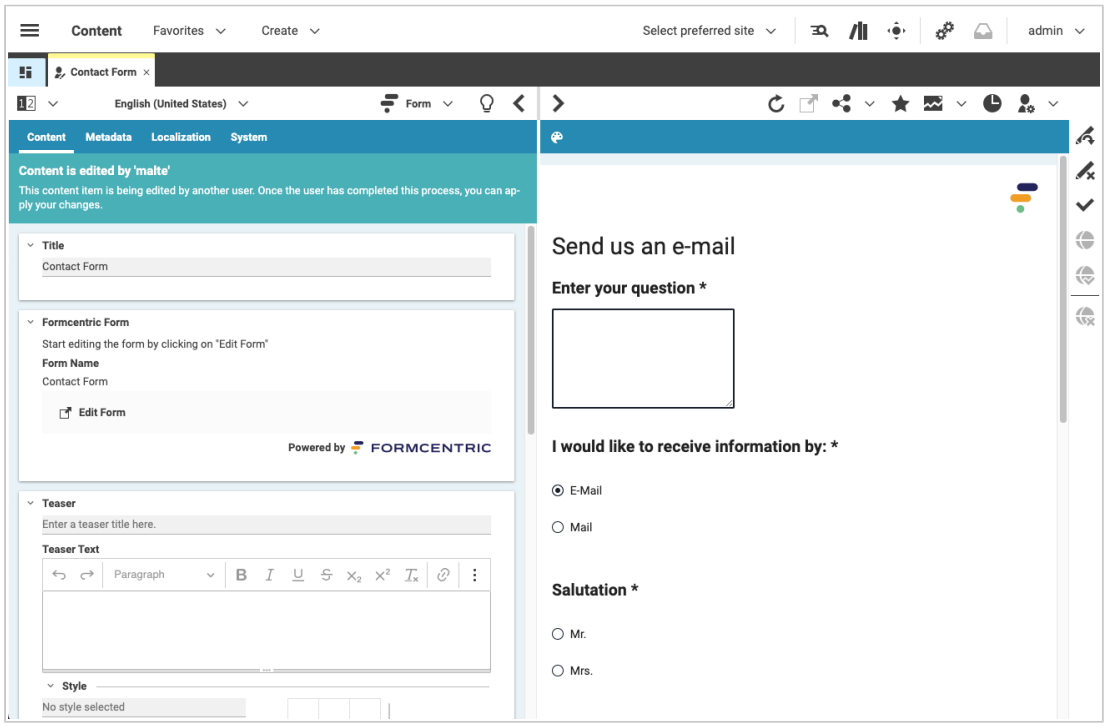


Figure 3.5. CoreMedia Studio preview

4. Form screen

You use the *Form* screen to edit the form with all of its form elements (input fields, drop-down lists, etc.).

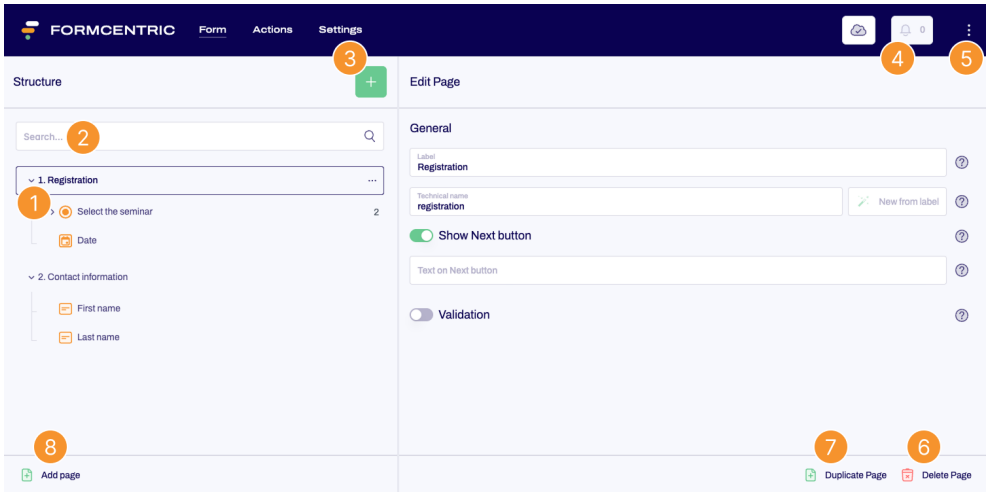


Figure 4.1. Editor

Click the green plus sign [3] in the *structure* area to display a list of all the form elements available to you. Click a form element to add it to your form. This new element is inserted underneath the form element currently selected in the form tree [1]. You can use the search function [2] to search for form elements in the form tree. Click *Add page* [8] at the bottom of the screen to add another page to your form. This is also added to the form tree.

The bell icon [4] on the top right shows you how many form elements in your form (if any) are incomplete or have errors. If conflicts are present, the bell is coloured red. Clicking the bell opens a dialogue that lists all of these errors. You can click one of these errors to jump directly to the affected form element.

Through the three-dot menu (5), you can choose whether the form tree displays the labels or the technical names of the form elements. You can also use this menu to copy a form or, if one has already been copied, to paste it.

When you paste a form, the existing form is completely replaced, including all actions and settings. This action cannot be undone.

You edit the properties for pages and form elements on the right-hand side. If you click *Duplicate* [7] or *Delete* [6] below this editing area, then you will duplicate or delete the form element that you currently have opened.

4.1. Form tree

The form tree that is displayed on the left-hand side of the *form* screen shows the form structure. Each form element and each page is represented here. This lets you

keep an eye on your form structure as you are editing. You can change the order of form elements or form pages whenever you want to. To do so, simply drag and drop the element or page to a different position in the form tree.

Additional actions are available for all elements shown in the form tree. To show available actions, move the mouse over a form element or a page and click the context menu link to open the context menu. You can also open the context menu by right-clicking the element or page.

The following actions are available for *form elements*:

Copy: Copies the form element and places it on the clipboard

Paste: Pastes a form element from the clipboard

Cut: Cuts the form element and places it on the clipboard

Duplicate: Creates a copy of the form element and adds it to the form directly underneath this element

Delete: Deletes the form element

Move to: Moves the form element to a different page

The following actions are available for *form pages*:

Copy: Copies the form page and places it on the clipboard

Paste: Pastes a form page from the clipboard

Cut: Cuts the form page and places it on the clipboard

Insert page after: Adds a new form page after this one

Duplicate: Creates a copy of the form page and adds it to the form directly underneath this page

Delete page content: Deletes all the form elements on this form page

Delete: Deletes the form page

4.2. Page properties

Click the page name in the form tree to display the properties for this form page.

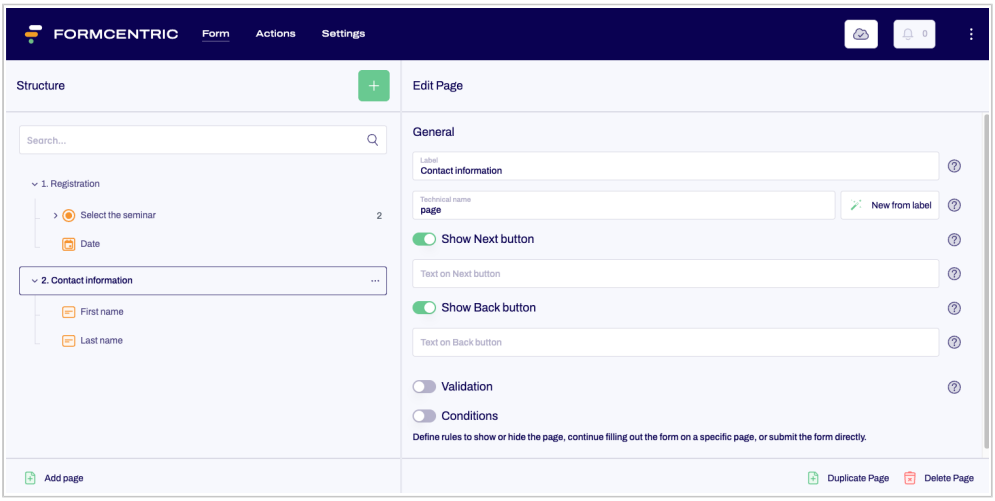


Figure 4.2. Page properties

With multi-page forms, the user is shown “Next” and “Back” buttons. The user can use these buttons to navigate between the individual form pages.

Forms can consist of as many pages as necessary, and even empty form pages are possible.

If a form page should only be displayed if the user has made specific kinds of input on the previous pages of the form, you can define a condition to provide this functionality (see Section 4.2.2, “Conditions”).

The page properties area is split over the following three sections: *General, Validation* and *Conditions*.

4.2.1. General

You define general page settings in the *General* section.

Label: Enter the text of the label to be displayed on the form page.

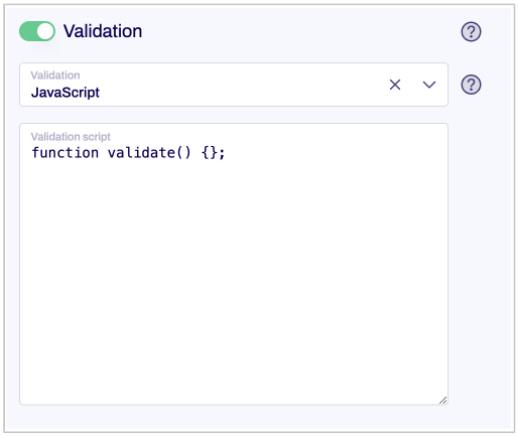
Technical name: Enter a technical name for the form page. This is necessary to ensure that the form page can be uniquely identified internally. You can only use a technical name once within your form. Start the name with a letter and do not use any special characters. The technical name must not be the same as any of the identifiers listed under Appendix A, *Reserved identifiers*.

New from label : Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Text on Next button: Enter the label to be displayed on the Next button for the selected form page. The Next button will not be shown on one-page forms.

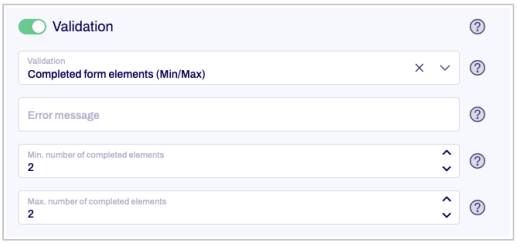
Text on Back button: Enter the label to be displayed on the Back button for the selected form page.

Validation: After activating the validation function, you can choose between various validation options. Validations are executed either when the user leaves the page (if this is the only page in the form) or when the form is submitted.

JavaScript	Unlike the validation that only checks user input for a single form field, the JavaScript validator gives you the option of checking input for multiple form fields at the same time and creating relationships between these fields.
	 
	
	In the *Validation script* text box, you can enter JavaScript that is used to validate the data input into the form. Each new form that you create is provided with the empty *validate()* method. This is a JavaScript function that is executed every time the form is submitted.
	`function validate() {};`
	You can input your own validation logic into this function.
	Please note: if the form data contains errors, the function must return a statement to the form user that clearly describes the error. If no error was determined during validation, then the script must return an empty string (""). In this case, the form data is considered to be correct and is sent for processing.
	When creating the validation script, you can use all the operations and functions available in JavaScript. For handling date values and drop-down lists, the functions *parseDate, parseAge* and *isSelected* are also available.
	The function *parseDate(date format, date)* converts a character string into a JavaScript object of the *Date*type. For the first parameter, specify the underlying date format; for the second parameter, give the date value.
	`parseDate("dd/MM/yyyy", "18/12/1969");`
	The function *parseAge(date format, date)* calculates a person's age based on the birth date that the person enters into the form.

	For the first parameter, specify the underlying date format; for the second parameter, give the date value.
	`parseAge("dd/MM/yyyy", "18/12/1969");`
	You can use the function *isSelected(selection, option)* to check whether the user has chosen a specific option from a selection field (single choice, multiple choice or a drop-down list). For the first parameter, specify the technical name of the selection field; for the second parameter, specify the option value (not its label).
	`isSelected(newsletter, "Yes");`
	Access to form values is provided using variables that are made available to you automatically. If you have defined an input field with the technical name *email*, for example, you can access the user input directly by using the *email* variable. You can access the current page number value with the *pageId* variable.
	If, for example, you want to ensure that the user can only enter a value into the input field for the postcode (technical name *post-code*) if the user has also entered something into the input field for the town (technical name *town*), then you can set this up by using the following function:
	`function validate () { if (town != "" && postcode == "") { return “Please also en`
	The following example shows you a validation script that can be used to ensure that only people aged 16 years or older can subscribe to a newsletter. Younger people can only submit the form without subscribing to the newsletter.
	Alongside the input field for the date of birth (technical name *birthday*), the form also includes a drop-down list (technical name *newsletter*) with the option *Yes* for subscribing to this newsletter.
	`function validate () { var age = parseAge("dd.MM.yyyy", birthday); if (isSelected`
	The validation script is executed as soon as the user moves to a different form page or submits the form.

Completed form elements (Min/Max)	You use this option to specify how many form elements must be filled out (minimum or maximum number) on a page. This validation ensures that users must complete a specified number of fields before they can submit the form.



Error message: Enter the text that is displayed if the minimum or maximum requirements for completed form elements are not satisfied. This message appears either when the user attempts to leave the page or when the form is submitted.

Minimum number of completed elements: Specify the minimum number of elements that must be filled out on the form page. If you specify a minimum of 3 elements, for example, then the user cannot leave the page or submit the form without meeting this requirement.

Maximum number of completed elements: You can apply a limit to prevent too many elements being filled out on a form page. This can be advisable if you want a certain selection or quantity of inputs to be made per page.

This validator helps to structure user input and enforce data quality, both when switching pages and when submitting the form.

4.2.2. Conditions

You use the *Conditions* section to specify that certain form pages will be shown or hidden, or that the form will be submitted, based on what the form user has entered into the form so far. Since the first page in a form cannot be hidden, however, please note that this functionality is available only from the second form page onwards.

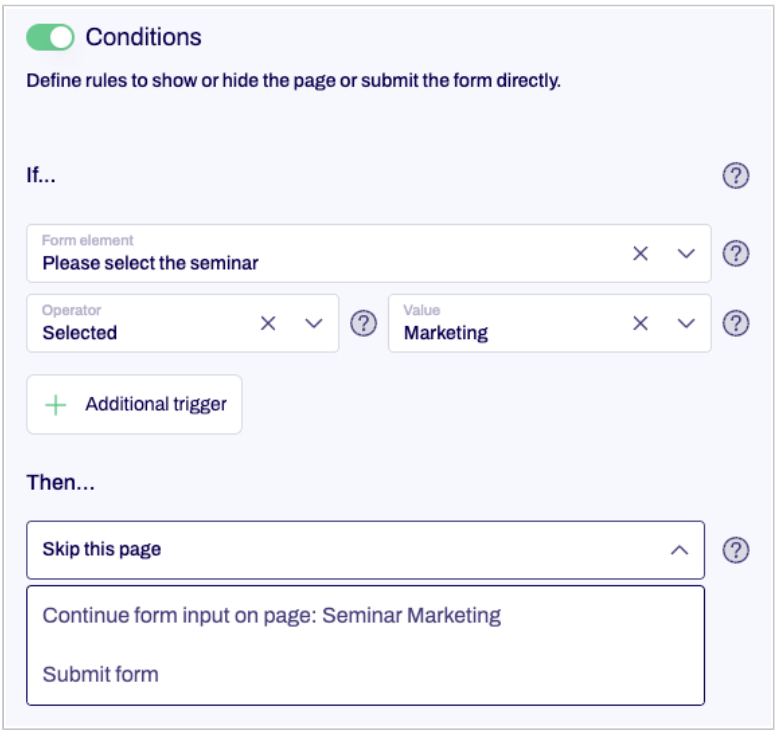


Figure 4.3. Page conditions

To add a new condition, push the slider at *Conditions* to the right.

If...: Specify the trigger for this condition, so that this page will not be displayed when the user clicks the “Next” button on the previous page.

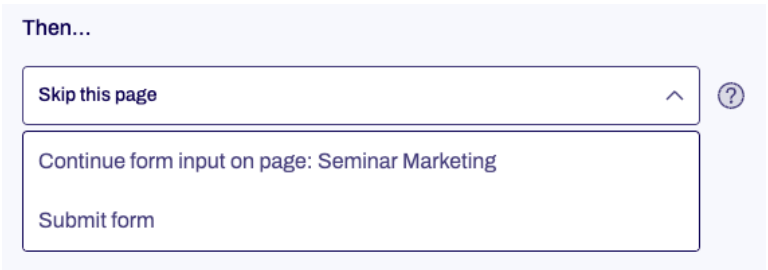
If the page hidden in this way is the last page of the form, then the form will be submitted when the user clicks the “Next” button.

Configuration of the condition follows the same procedure as for Section 4.3.16, “Condition”.

Take care to ensure that *calculated values* always return a value if you use them within a condition. If field input is used to calculate the value, then you must ensure that the form fields involved are set as required fields.

Then...: In this drop-down list, specify the action the system should take if the above mentioned condition is satisfied, and the user clicks the “Next” button on the previous page.

If you select the option *Skip this page* on the last page of the form, the form is submitted automatically when the user clicks *Next*.



4.3. Form elements

Various form elements are available to you for creating your forms.

On the *Form* screen, click the green plus sign. This opens a dialogue, listing all the available form elements. Select a form element to add it to your form. You can adjust the properties of the form elements on the right, in the editing area.

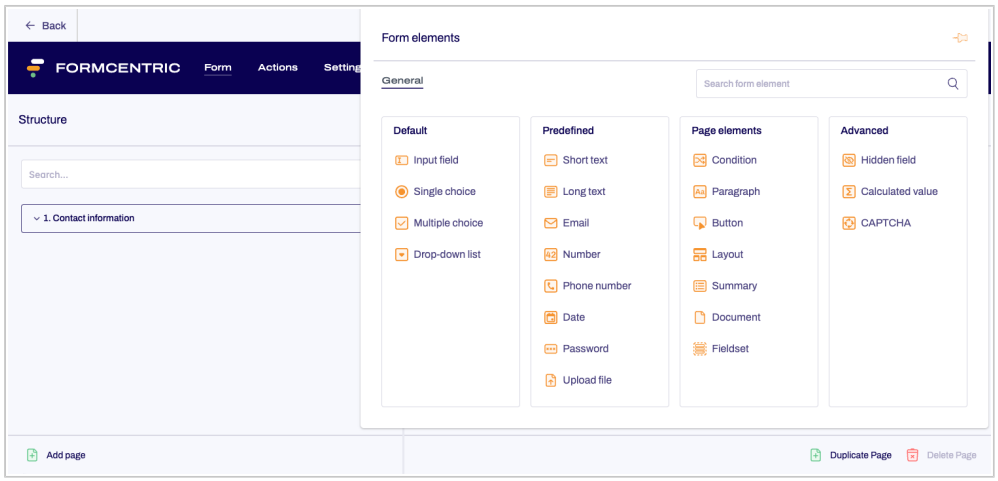


Figure 4.4. Add form element

The following section describes all the form elements in detail.

4.3.1. Input field

You use the *input field* form field to add a single-line input field to your form. This field is suitable for entering short pieces of text, such as name and address details, as well as numbers. You can select all of the validation options available in Formcentric.

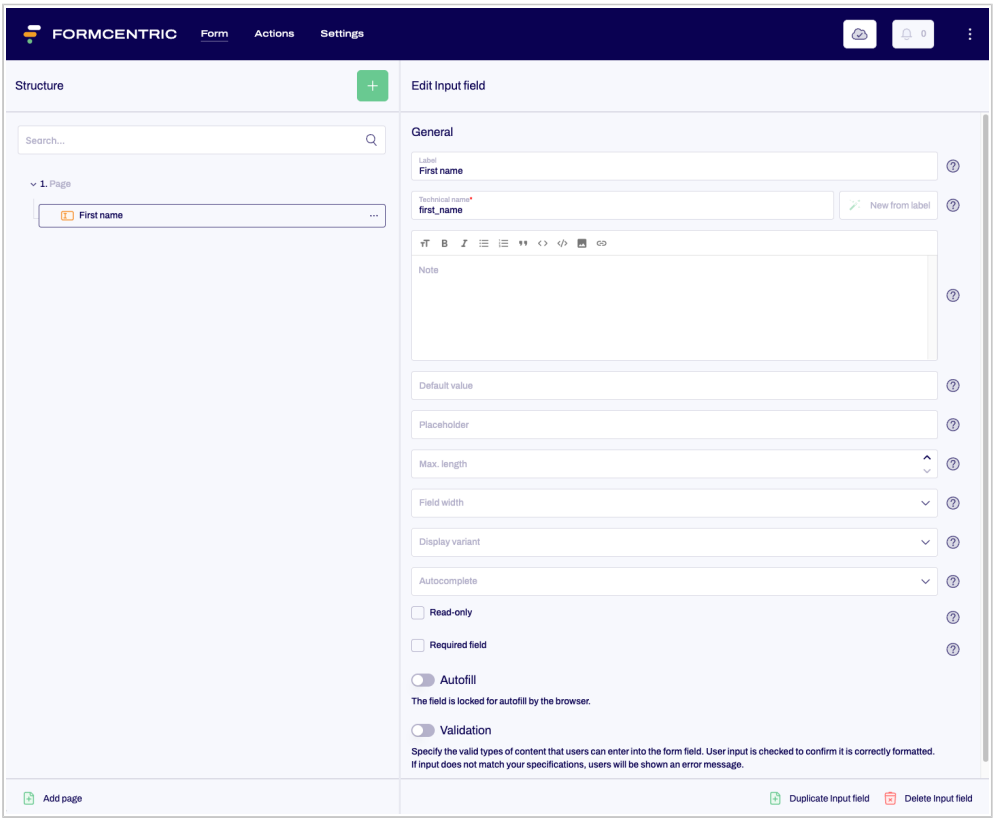


Figure 4.5. Input field

Label: Enter the text of the label that is displayed next to the form field.

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the “technical name” – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Note: Here, you have the option of adding some text to your input field that gives the user additional information. This could be instructions about filling out the field, for example.

Default value: In this field, you can enter a piece of text that is displayed in the input field when the user accesses the form for the first time. Alternatively, you can make use of a variable here. This could insert the date automatically into the input field, for example (see Section 4.5, “Variables”).

Placeholder: In this field, you can enter a piece of placeholder text that is displayed in the input field. This placeholder text disappears as soon as the user starts typing into the form field.

Max. length: Use this to specify the maximum number of characters that the user can enter into the input field.

Field width: Specify how wide the form element should be. Sometimes, it may be useful to display form elements next to one another, e.g. street plus house number.

Display variant: Select one of the available display variants here, so as to specify how the input field is displayed in the form. The variants available are specified on a per-project basis.

Autocomplete: In this field, you can select a data source that will help users fill out the form field by autocompleting text in this field for them. As soon as the user starts typing into the form field, the system displays a list of possible hits from the data source, from which the user can select the entry to use. Formcentric provides you with various default data sources, as *Country names*, *Months*, *Years*, and *Weekdays*.

You also have the option of entering additional parameters to pass to the data source (see also Section 4.4, “Data sources”).

Read-only: If you check *Read-only*, users will not be able to change the text in the form field. Use this setting if you want to enter predefined variables into this form field (see Section 4.5, “Variables”).

Required field: Check *Required field* if this form field must be filled out when completing the form. An “*” will then be added to the end of the label, marking this form field as a required field. The user will then be unable to submit the form or move to the next page (if this is a multi-page form) if they have not completed this form field properly.

Error message: An input field appears when you check the *Required field* box. Enter some text here for the error message that is displayed if the user tries to skip filling out this form field.

Autofill: You can use the *Autofill* function to improve the auto-fill functionality offered by the browser. Browsers save form data entered previously, like addresses and payment details, and then offer these as suggestions for other forms with the same types of fields. To improve the quality of these suggestions, you can tell the browser about the kinds of information that are expected for a specific field. In this way, you can ensure that browsers make appropriate suggestions for auto-completing form fields, which also helps the user to fill out the forms faster.

Shipping or invoice address: Select “Shipping” or “Invoice” if this form field is part of a shipping address or an invoice address.

Contact type: Select an appropriate contact type here if users are expected to input a fixed-line or mobile phone number, fax or pager number, or an email address into this form field. Specifying a contact type will affect the selection offered to you for “Usage” (see below). This field is optional and can be left empty.

Usage: Specify the kinds of information entered into this form field.

If you want to stop the browser saving sensitive data and offering this as an autofill suggestion, then you can deactivate "*Autofill*". To do so, move the slider to the left.

As of this writing, the autofill function is not supported by all web browsers. For an overview of browser compatibility, please see https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/autocomplete.

Validation: If you activate the validation slider (move it to the right), a drop-down list appears. From this list, pick a validator to specify the format to be used for the text that the user enters into the input field. For example, the *email* validator checks to confirm that a valid email address has been entered into the field.

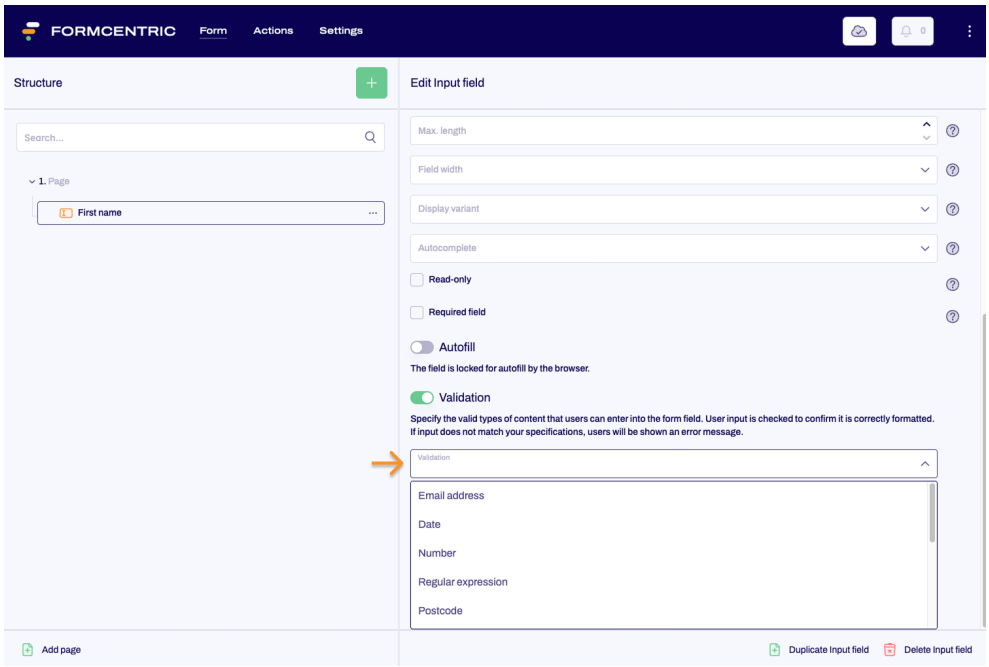
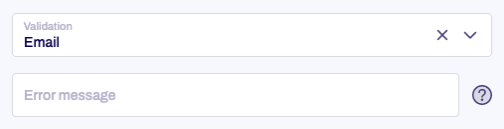


Figure 4.6. Validation

If the user's input does not match the rule(s) you specify here, then an error message is displayed.

Formcentric includes standard validators for *email addresses, dates, numbers, post-codes, numbers of characters, IBANs, EU VAT registration numbers, BICs, telephone numbers* and *matching values*. In addition, you also have the option of using a regular expression to check user input. This is useful for checking data such as customer numbers, etc. The corresponding validator is already activated for predefined form elements.

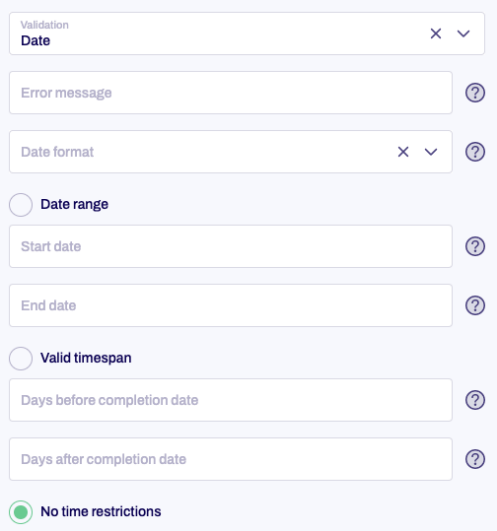
Email address	The email address validator checks to confirm that the user has entered a valid email address.



Error message: Enter some text here for an error message that should be displayed instead of the default error message. This is shown to the user if the form field has not been filled out according to the rule(s) you have set.

Date

This validator checks to see if the value input is a date. In addition, you can also limit the entry of the date to a certain period of time. This is useful when a meeting can only be scheduled between two specific dates, for example.

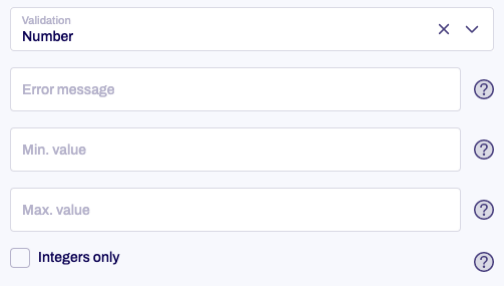


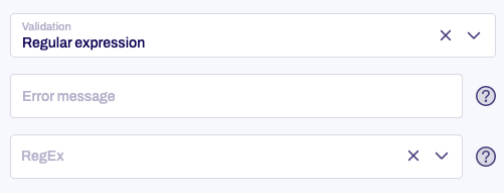
Error message: Enter some text here for an error message that should be displayed instead of the default error message. This is shown to the user if the form field has not been filled out according to the rule(s) you have set.

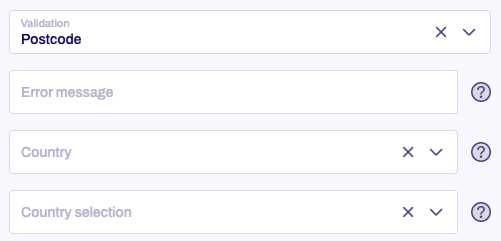
Date format: Select the format that must be used when entering the date.

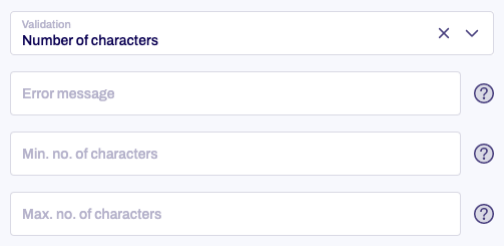
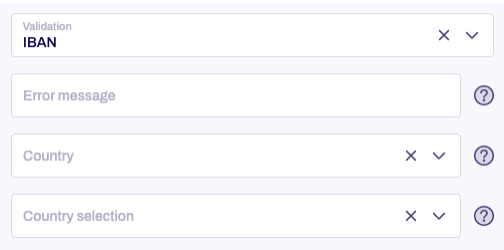
The following placeholders can be used in the format string:

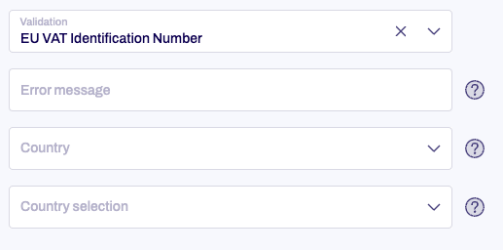
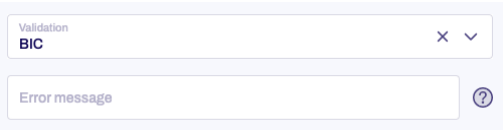
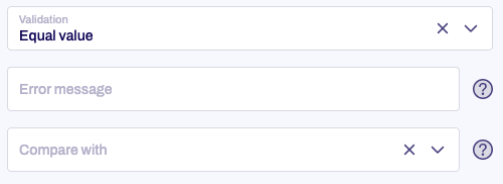
y	Year
MM	Month of the year with leading zero
d	Day of the month
H	Hour of the day (0–23)
m	Minute of the hour
s	Second of the minute

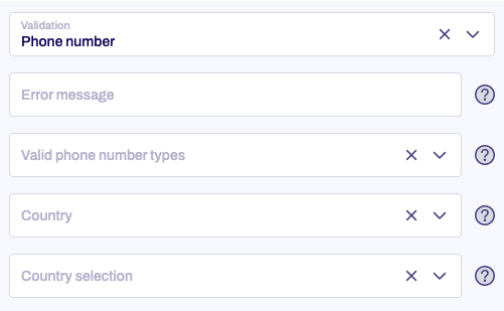
	The drop-down autocomplete list already contains a list of commonly-used data formats. You can either use these directly or modify them to suit your personal requirements. By default, the format *dd/MM/yyyy* is applied. Enter a piece of placeholder text to be displayed in the input field that shows the correct date format to use. In this way, you can avoid unnecessary error messages and help the user fill out the form correctly. **Date range:** Specify a date range here if the user needs to enter a date that lies within a specified period of time. **Date from:** Select the start date for this period of time. **Date to:** Select the end date for this period of time. **Valid timespan:** The values entered here limit the date entered by the user to a number of days before or after the form completion date. **Days before completion date:** Enter the earliest date before the form completion date that can be entered by the user, expressed as the number of days before the completion date. **Days after completion date:** Enter the latest date after the form completion date that can be entered by the user, expressed as the number of days after the completion date. To only allow dates in the past, for example, use the *Days after completion date* parameter and enter either “0” (the user can also enter the completion date itself) or “–1” (the user cannot enter the completion date itself). **No time restrictions:** If you select *No time restrictions*, then the user will be able to enter any date.
Number	This validator checks to see if the value input by the user is a valid number. You can also specify a range of numbers that the value entered by the user must match. 

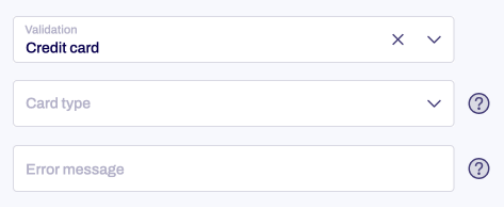
	Error message: Enter some text here for an error message that should be displayed instead of the default error message. This is shown to the user if the form field has not been filled out according to the rule(s) you have set. **Smallest value:** Specify the smallest number that the user is allowed to enter. **Largest value:** Specify the largest number that the user is allowed to enter. **Integers only:** Select *Integers only* if you want to prevent the user from entering decimal numbers.
Regular expression	This validator checks to see whether the character string entered by the user matches a specified pattern. This pattern, which the letters and numbers entered by the user must match, is defined using something called a “regular expression”. A regular expression (which can also be abbreviated as regexp or regex) is a character string that uses syntactical rules to define character string entities.  **Error message:** Enter some text here for an error message that should be displayed instead of the default error message. This is shown to the user if the form field has not been filled out according to the rule(s) you have set. **RegEx:** In this field, you can enter or select a regular expression that defines the input format you require. The drop-down auto-complete list already contains regular expressions suitable for a range of common applications. You can either use these directly or modify them to suit your personal requirements. You will find a short guide to the authoring of regular expressions at the following URL: https://en.wikipedia.org/wiki/Regular_expression
Postcode	This validator checks to see if the user input is a valid postcode.

	 **Error message:** Enter some text here for an error message that should be displayed instead of the default error message. This is shown to the user if the form field has not been filled out according to the rule(s) you have set. **Country:** Select the country for which the postcode should be validated. **Country selection:** If your form contains a drop-down list (see Section 4.3.4, “Drop-down list”) with a country selection, you can select the corresponding form field here. The postcode must then come from the country that the user selected from that list. If both a country and a country selection have been made, then the country selection has priority.  Please note that the *Value* fields for the entries in the drop-down list must contain the valid country codes according to ISO 3166 (e.g. *DE* fur Germany). The countries cannot be validated without valid country codes. The postcode validator supports the formats used in the countries listed below: Albania (AL), Austria (AT), Belgium (BE), Bosnia and Herzegovina (BA), Bulgaria (BG), Croatia (HR), Cyprus (CY), Czech Republic (CZ), Denmark (DK), Estonia (EE), Finland (FI), France (FR), French Guiana (GF), Germany (DE), Greece (GR), Guadeloupe (GP), Hungary (HU), Iceland (IS), Ireland (IE), Italy (IT), Kosovo (RS-KM), Latvia (LV), Liechtenstein (LI), Lithuania (LT), Luxembourg (LU), Macedonia (MK), Malta (MT), Martinique (MQ), Moldavia (MD), Montenegro (ME), Netherlands (NL), Norway (NO), Poland (PL), Portugal (PT), Réunion (RE), Romania (RO), Serbia (RS), Slovakia (SK), Slovenia (SI), Spain (ES), Sweden (SE), Switzerland (CH), Tunisia (TN), Turkey (TR), Ukraine (UA), United Kingdom (UK) If the country code is not listed, then the postcode is not validated.
Number of characters	This validator checks the number of characters entered.

	 **Error message:** Enter some text here for an error message that should be displayed instead of the default error message. This is shown to the user if the form field has not been filled out according to the rule(s) you have set. **Minimum no. of characters:** Enter the minimum number of characters that the user must enter into the input field. **Maximum no. of characters:** Enter the maximum number of characters that the user may enter into the input field.
IBAN	This validator checks to see if the user has input a valid International Bank Account Number (IBAN).  **Error message:** Enter some text here for an error message that should be displayed instead of the default error message. This is shown to the user if the form field has not been filled out according to the rule(s) you have set. **Country:** If the format for the IBAN entered must match the IBAN format for a specific country, select that country here. If you do not select a country, the validator only checks to see if the value entered matches the standard IBAN format. **Country selection:** If your form contains a drop-down list (see Section 4.3.4, “Drop-down list”) with a country selection, you can select the corresponding form field here. In this case, the format of the IBAN entered must match the format that is valid in the country which the user has selected from the drop-down list. If both a country and a country selection have been made, then the country selection has priority.
EU VAT Identification number	This validator checks to confirm that user input matches the structure of an EU VAT registration number.

	 **Error message:** Enter some text here for an error message that should be displayed instead of the default error message. This is shown to the user if the form field has not been filled out according to the rule(s) you have set. **Country:** If the format for the EU VAT registration number entered must match the format for a specific country, select that country here. If you do not select a country, the validator only checks to see if the value entered matches the standard EU VAT registration number format. **Country selection:** If your form contains a drop-down list (see Section 4.3.4, “Drop-down list”) with a country selection, you can select the corresponding form field here. In this case, the format of the EU VAT registration number entered must match the format that is valid in the country which the user has selected from the drop-down list. If both a country and a country selection have been made, then the country selection has priority.
BIC	This validator checks to see if the user has input a valid international Bank Identifier Code (BIC).  **Error message:** Enter some text here for an error message that should be displayed instead of the default error message. This is shown to the user if the form field has not been filled out according to the rule(s) you have set.
Equal value	This validator compares two input fields and checks to confirm that the input in both of these field is identical.  **Error message:** Enter some text here for an error message that should be displayed instead of the default error message. This is

	shown to the user if the form field has not been filled out according to the rule(s) you have set. **Compare with:** Select a second input field whose value will be compared with the first input field. The validator then checks to confirm that the two input fields have matching input.
Phone number	This validator checks to see if the user input is a valid phone number. You can also specify phone number types and/or specify that the phone number must come from a specific country.  **Error message:** Enter some text here for an error message that should be displayed instead of the default error message. This is shown to the user if the form field has not been filled out according to the rule(s) you have set. **Valid phone number types:** Select the valid phone number types from this list. The phone number entered must then match these types. If you do not select any phone number types, the phone number is not validated. The following formats are supported: Fixed line (FIXED_LINE), mobile (MOBILE), fixed line or mobile (FIXED_LINE_OR_MOBILE), toll-free (TOLL_FREE), premium rate (PREMIUM_RATE), shared cost (SHARED_COST), VOIP (VOIP), personal number (PERSONAL_NUMBER), pager (PAGER), universal access numbers (UAN), voicemail (VOICE-MAIL), unknown (UNKNOWN).  For some countries, the *fixed line* and *mobile* types may be ambiguous. In these cases, you may also need to select the *fixed line or mobile* type in order to ensure that validation executes correctly. The *fixed line or mobile* phone number type is not a combination of *fixed line* and *mobile*, but is a separate phone number type. **Country:** Select a country here if the phone number must come from a specific country.

	Country selection: If your form contains a drop-down list (see Section 4.3.4, “Drop-down list”) with a country selection, then you can select the corresponding form field here. The phone number entered must then come from the country that the user selected from that list. If both a country and a country selection have been made, then the country selection has priority. Please note that the *Value* fields for the entries in the drop-down list must contain the valid country codes according to ISO 3166 (e.g. *DE* fur Germany). User input cannot be validated without valid country codes.
Credit card	This validator checks that the entered card number has the correct length, valid checksum, and is associated with a recognized provider. This helps prevent input errors and ensures that only valid credit cards are accepted. Additionally, you can choose to accept only specific providers if needed.  Please note that the validation only checks the entered card number for correct length, valid checksum, and association with a recognized provider. It does not verify whether the card actually exists or is active. **Card type:**Select one or more card types to define which providers are accepted. If no selection is made, all listed types will be accepted. **Error message:** Enter some text here for an error message that should be displayed instead of the default error message. This is shown to the user if the form field has not been filled out according to the rule(s) you have set.

Duplicate Input field: Click *Duplicate Input field* to add a copy of this form field to your form.

Delete Input field: Click *Delete Input field* to delete the form field.

4.3.2. Single choice

A *single choice* field offers your user several options to choose from. The user can select only one of these options, however: if the user picks a second option, then

the option previously chosen is unselected. The individual options are displayed as radio buttons.

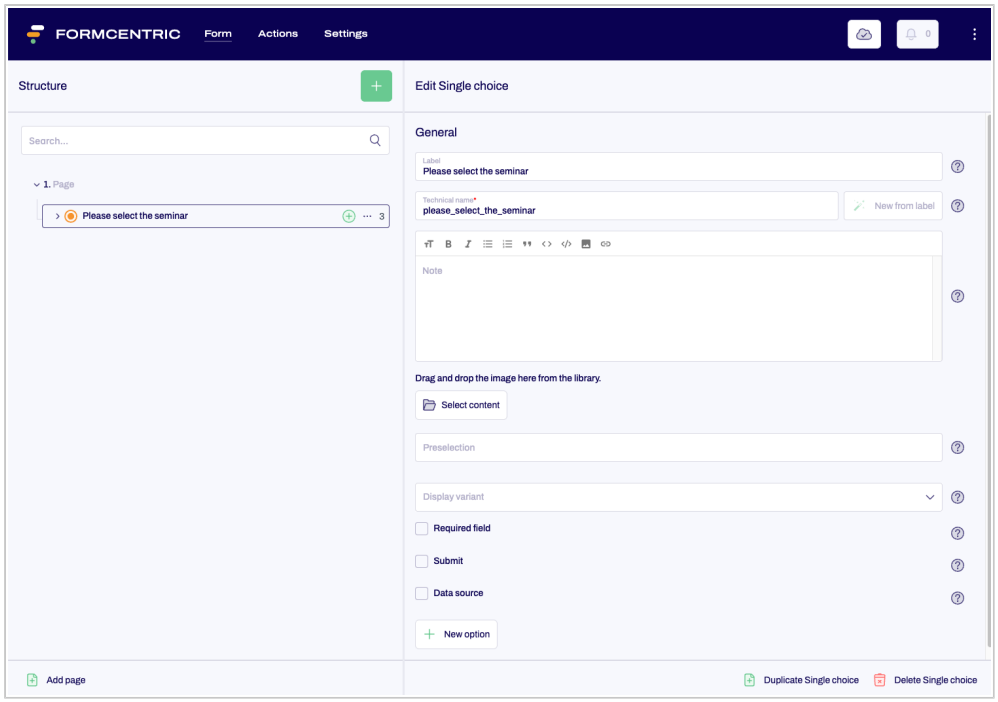


Figure 4.7. Single choice

Label: Enter the text of the label that is displayed next to the form element.

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the "technical name" – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Note: Here, you have the option of adding some text to your form element that gives the user additional information. This could be instructions about filling out the field, for example.

Select content: You can add an image here from the Library that will be shown together with the select field. Simply use drag-and-drop to move it here from the Library.

Preselection: You can specify that one option is preselected when the form is initially accessed. You can also select an option generated from a data source. Enter the *value* for the corresponding option here.

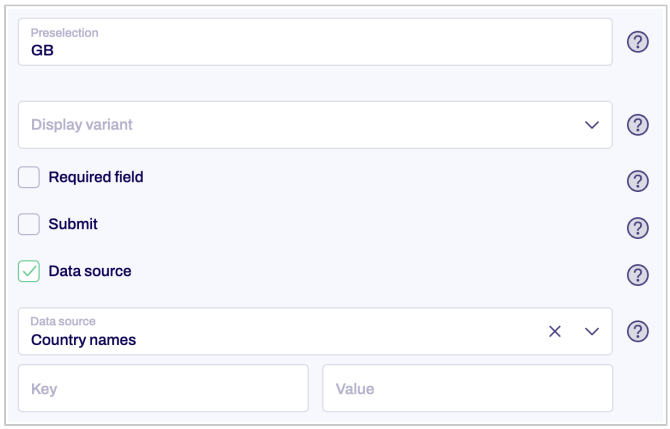


Figure 4.8. Preselection

The data source *Country names* included in the standard delivery package uses country codes according to ISO 3166. For example, if you want Great Britain to be pre-selected, you need to enter the value *GB*.

Display variant: Select one of the available display variants here, so as to specify how the single choice field is displayed in the form.

Required field: Check *Required field* if an option must be selected. An "*" will then be added to the end of the label, marking this form field as a required field. The user will only be able to submit the form or move on to fill out the next page (if this is a multi-page form) if they have selected an option.

Error message: An input field appears when you check the *Required field* box. Enter some text here for the error message that is displayed if the user tries to skip filling out this form field.

Submit: Check *Submit* to forward the user to the next form page once they have made a selection. If the single choice field is placed on the last page of the form, the form is submitted automatically.

Data source: Select a data source that will be used to fill the single choice field with external data at runtime. The available data sources are specified in the *FormcentricSettings* document, which can be found in the sample content of our extension.

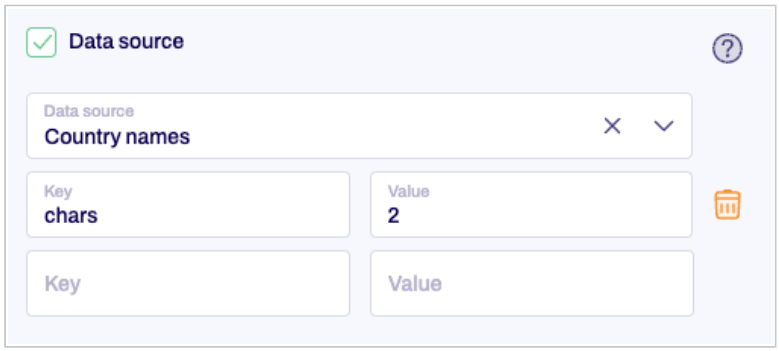


Figure 4.9. Data source

The table gives you the option of entering additional parameters to pass to the data source. The parameters that are accepted depend on the data source selected. The data sources included in the standard delivery package do not require any additional parameters, but they do have optional parameters (see Section 4.4, “Data sources”).

Duplicate Single choice: Click *Duplicate Single choice* to add a copy of this single choice field to your form.

Delete Single choice: Click *Delete Single choice* to delete the single choice field.

Adding options: Once you’ve added your single choice field to the form and defined its corresponding properties, you can then set up the individual options. To do this, click *New option* in the lower part of the editing area for the single choice field. Alternatively, select the single choice in the form tree and click the *green plus sign* shown in a circle.

You then specify the properties for the options in the editing area.

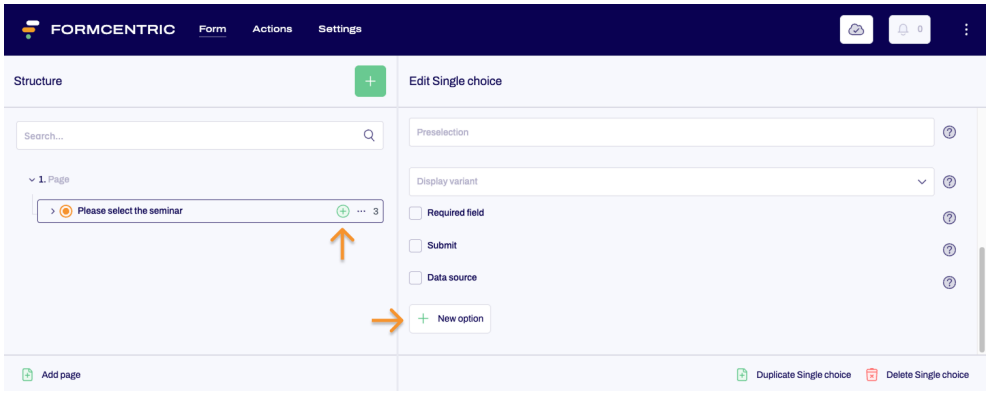


Figure 4.10. Edit option

Label: Enter the text to be shown for the option.

Value: Enter a value for the option. Please note that this identifier must be unique. This value is submitted with the form when the user picks this option.

If you specify “EMPTY_VALUE” as the value here, this option will be ignored in later processing steps, even if has been selected by the user. This function can be used if you want to add a “Please select” option, for example.

Value from label: Click *Value from label* to generate a value from the label.

Image: You can select an image from the Library that will be shown together with the option. Simply use drag-and-drop to move it here from the Library.

Preselected: Click *Preselected* to specify that this option is selected automatically.

Duplicate Option: Click *Duplicate Option* to add a copy of this option to your form.

Delete Option: Click *Delete Option* to delete this option.

4.3.3. Multiple choice

A *multiple choice* field offers your user several options to choose from. The user can pick more than one option. Each of these options is shown as a check box.

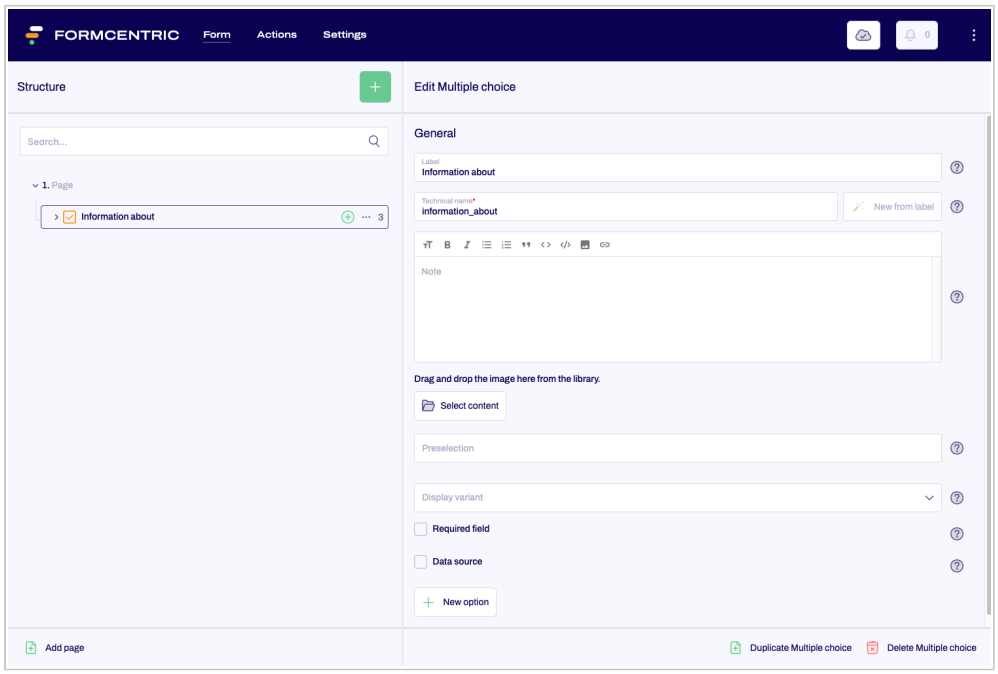


Figure 4.11. Multiple choice

Label: Enter the text of the label that is displayed next to the form element.

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the “technical name” – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Note: Here you have the option of adding some text to your form element that gives the user additional information. This could be instructions about filling out the field, for example.

Select content: You can add an image here from the Library that will be shown together with the select field. Simply use drag-and-drop to move it here from the Library.

Preselection: You can specify that one or more options are preselected when the form is initially accessed. You can also select options generated from a data source. Enter the *values* for the corresponding options here. If there are multiple options, separate them with commas.

Display variant: Select one of the available display variants here, so as to specify how the multiple choice field is displayed in the form.

Required field: Check *Required field* if an option must be selected. An "*" will then be added to the end of the label, marking this form field as a required field. The user will only be able to submit the form or move on to fill out the next page (if this is a multi-page form) if they have selected an option.

Error message: An input field appears when you check the *Required field* box. Enter some text here for the error message that is displayed if the user tries to skip filling out this form field.

Data source: Select a data source that will be used to fill the multiple choice field with external data at runtime. The available data sources are specified in the *Formcentric-Settings* document, which can be found in the sample content of our extension.

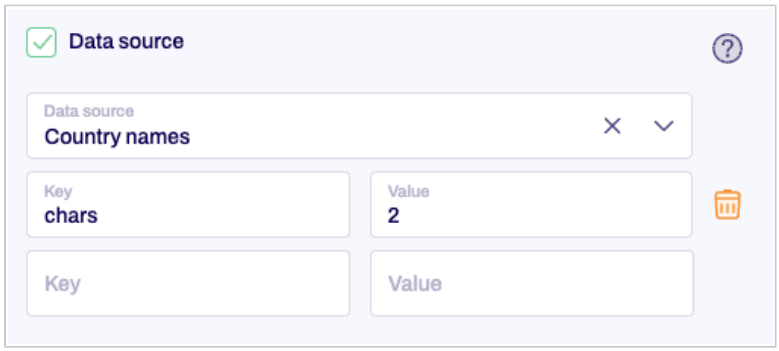


Figure 4.12. Data source

The table gives you the option of entering additional parameters to pass to the data source. The parameters that are accepted depend on the data source selected. The data sources included in the standard delivery package do not require any additional parameters, but they do have optional parameters (see Section 4.4, "Data sources").

Duplicate Multiple choice: Click *Duplicate Multiple choice* to add a copy of this form element to your form.

Delete Multiple choice: Click *Delete Multiple choice* to delete the form element.

Adding options: Once you've added your multiple choice field to the form and defined its corresponding properties, you can then set up the individual options. To do this, click *New option* in the lower part of the editing area for the multiple choice field. Alternatively, select the multiple choice in the form tree and click the *green plus sign* shown in a circle.

You then specify the properties for the options in the editing area.

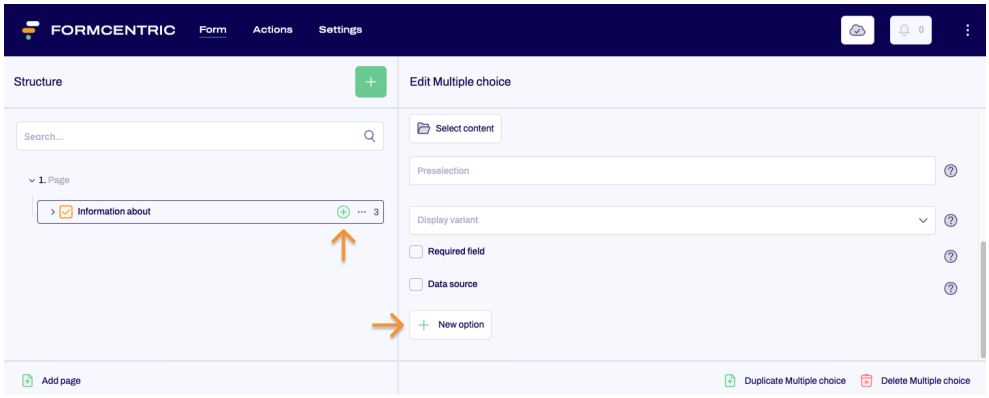


Figure 4.13. Edit option

Label : Enter the text to be shown for the option.

Value : Enter a value for the option. Please note that this identifier must be unique. This value is submitted with the form when the user picks this option.

If you specify "EMPTY_VALUE" as the value here, this option will be ignored in later processing steps, even if has been selected by the user. This function can be used if you want to add a "Please select" option, for example.

Value from label: Click *Value from label* to generate a value from the label.

Image: You can select an image from the Library that will be shown together with the option. Simply use drag-and-drop to move it here from the Library.

Preselected : Click *Preselected* to specify that this option is selected automatically.

Duplicate Option : Click *Duplicate Option* to add a copy of this option to your form.

Delete Option: Click *Delete Option* to delete this option.

4.3.4. Drop-down list

With a *drop-down list*, you offer your user one or more options in the form of a drop-down menu of options. The individual options are not displayed until the user actually

clicks to select the drop-down list. You can allow your user to pick just one or multiple options. You configure this in the list settings.

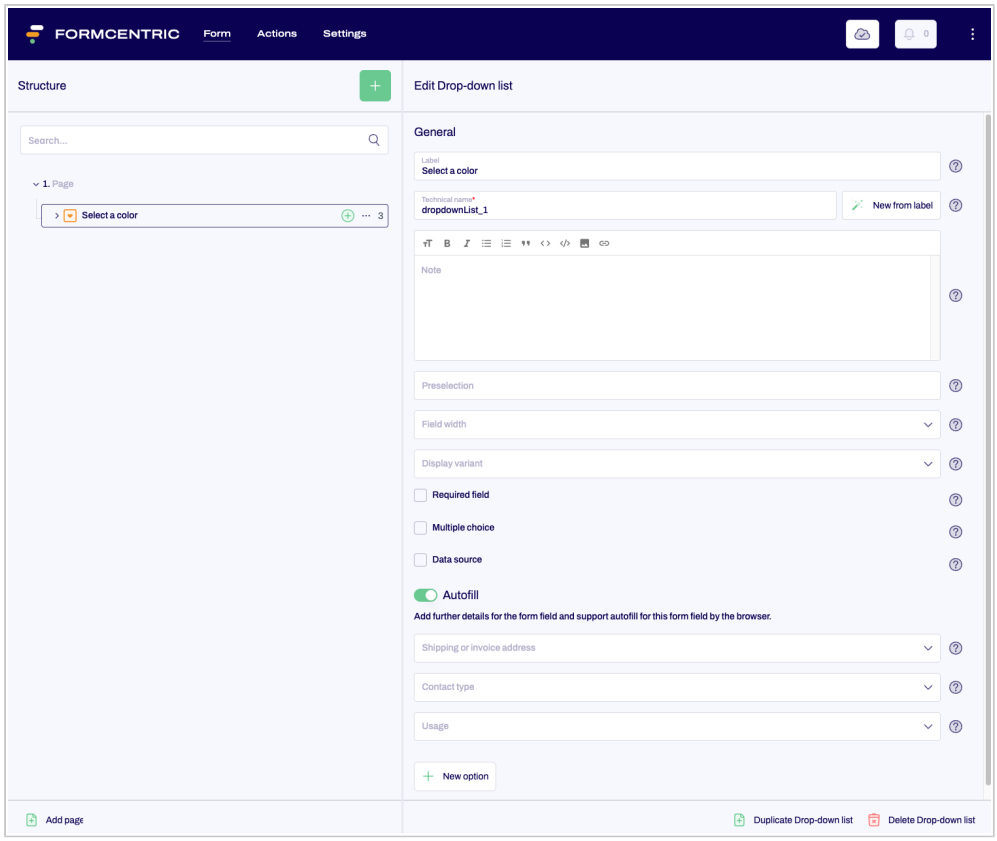


Figure 4.14. Drop-down list

Label: Enter the text of the label that is displayed next to the form element.

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the "technical name" – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Note: Here, you have the option of adding some text to your form field that gives the user additional information. This could be instructions about filling out the field, for example.

Preselection: Here, you can specify that one or more options are preselected when the form is initially accessed. You can also select options generated from a data source. Enter the *values* for the corresponding options here. If there are multiple options, separate them with commas and activate *Multiple choice*.

Field width: Use this to specify a display width for the form element.

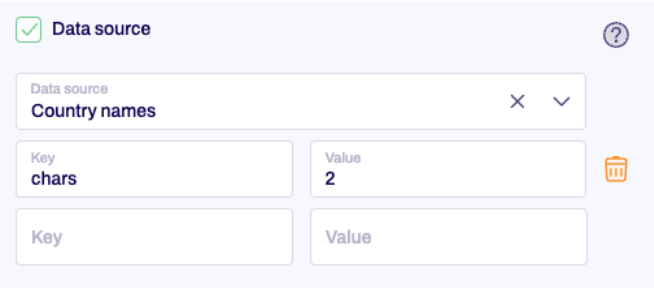
Display variant: Select one of the available display variants here, so as to specify how the drop-down list is displayed in the form.

Required field: Check *Required field* if an option must be selected. An "*" will then be added to the end of the label, marking this form field as a required field. The user will only be able to submit the form or move on to fill out the next page (if this is a multi-page form) if they have selected an option.

Error message: An input field appears when you check the *Required field* box. Enter some text here for the error message that is displayed if the user tries to skip filling out this form field.

Multiple choice: Check *Multiple choice* if the user is allowed to pick multiple options.

Data source: Select a data source that will be used to fill the drop-down list with external data at runtime. The available data sources are specified in the *FormcentricSettings* document, which can be found in the sample content of our extension.



The table gives you the option of entering additional parameters to pass to the data source. The parameters that are accepted depend on the data source selected. The data sources included in the standard delivery package do not require any additional parameters, but they do have optional parameters (see Section 4.4, "Data sources").

Autofill: You can use the *Autofill* function to improve the auto-fill functionality offered by the browser. Browsers save form data entered previously, like addresses and payment details, and then offer these as suggestions for other forms with the same types of fields. To improve the quality of these suggestions, you can tell the browser about the kinds of information that are expected for a specific field. In this way, you can ensure that browsers make appropriate suggestions for auto-completing form fields, which also helps the user to fill out the forms faster.

Shipping or invoice address: Select "Shipping" or "Invoice" if this form field is part of a shipping address or an invoice address.

Contact type: Select an appropriate contact type here if users are expected to input a fixed-line or mobile phone number, fax or pager number, or an email address into

this form field. Specifying a contact type will affect the selection offered to you for “Usage” (see below). This field is optional and can be left empty.

Usage: Specify the kinds of information entered into this form field.

If you want to stop the browser saving sensitive data and offering this as an autofill suggestion, then you can deactivate “*Autofill*”. To do so, move the slider to the left.

As of this writing, the autofill function is not supported by all web browsers. For an overview of browser compatibility, please see https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/autocomplete.

Duplicate Drop-down list: Click *Duplicate Drop-down list* to add a copy of this form element to your form.

Delete Drop-down list: Click *Delete Drop-down list* to delete the form element.

Adding options: Once you’ve added the drop-down list to the form and defined its corresponding properties, you can then set up the individual options. To do this, click *New option* in the lower part of the editing area for the drop-down list. Alternatively, select the drop-down list in the form tree and click the *green plus sign* shown in a circle.

You then specify the properties for the options in the editing area.

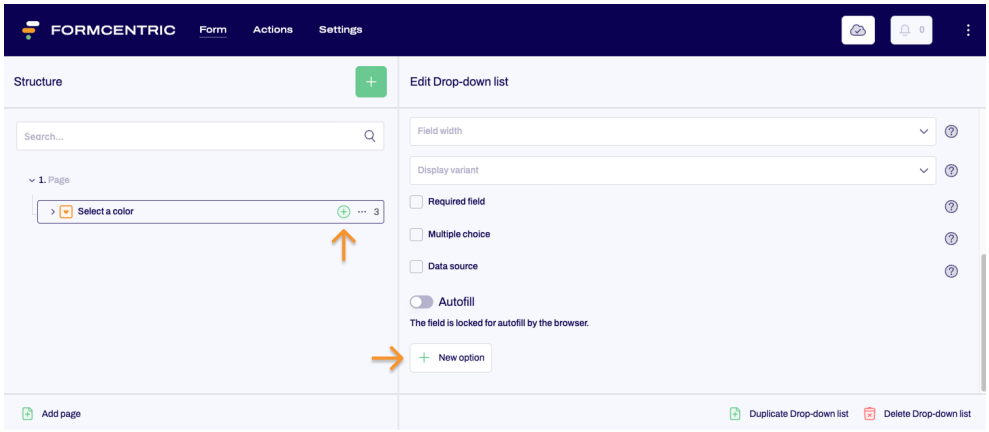


Figure 4.15. Edit option

Label : Enter the text to be shown for the option.

Value : Enter a value for the option. Please note that this identifier must be unique. This value is submitted with the form when the user picks this option.

If you specify “EMPTY_VALUE” as the value here, this option will be ignored in later processing steps, even if has been selected by the user. This function can be used if you want to add a “Please select” option, for example.

Value from label: Click *Value from label* to generate a value from the label.

Image: You can select an image from the Library that will be shown together with the option. Simply use drag-and-drop to move it here from the Library.

Preselected : Click *Preselected* to specify that this option is selected automatically.

Duplicate Option: Click *Duplicate Option* to add a copy of this option to your form.

Delete Option: Click *Delete Option* to delete this option.

4.3.5. Short text

You use the predefined *short text* form element to add a single-line input field to your form that does not require any further validation. This element is ideal for short user responses, such as entering a first and last name.

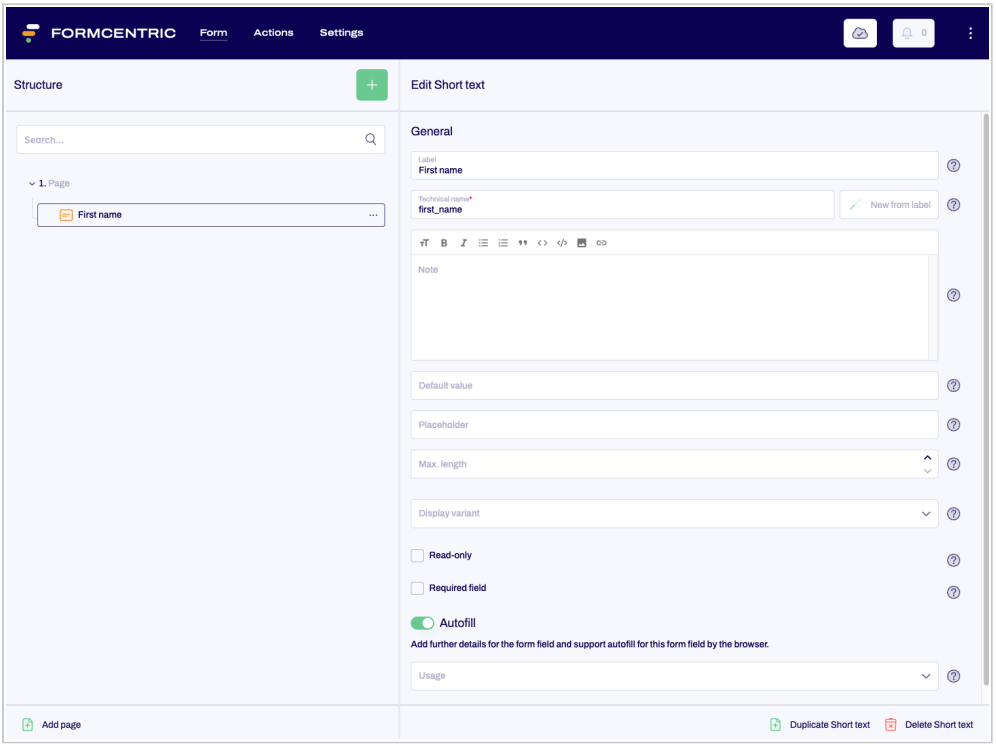


Figure 4.16. Short text

Label: Enter the text of the label that is displayed next to the form field.

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the “technical name” – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Note: Here, you have the option of adding some text to your form field that gives the user additional information. This could be instructions about filling out the field, for example.

Default value: In this field, you can enter a piece of text that is shown in the form field when the user accesses the form for the first time. Alternatively, you can make use of a variable here. This could insert the date automatically into the form field, for example (see Section 4.5, “Variables”).

Placeholder: In this field, you can enter a piece of placeholder text that is displayed in the form field. This placeholder text disappears as soon as the user starts typing into the form field.

Max. length: Use this to specify the maximum number of characters that the user can enter into the input field.

For this form element, you can define display variants in the configuration, as well as prompts that provide the user with help when filling out the form. The prompts and display variants will be shown in the editing area as drop-down lists, in the same way as for the input field form element, for example. Please see the Developer Manual for instructions.

Read-only: If you check *Read-only*, the user will not be able to change the text in the form field. Use this setting if you want to enter predefined variables into this form field (see Section 4.5, “Variables”).

Required field: Check *Required field* if this form field must be filled out when completing the form. An “*” will then be added to the end of the label, marking this form field as a required field. The user will then be unable to submit the form or move to the next page (if this is a multi-page form) if they have not completed this form field properly.

Error message: An input field appears when you check the *Required field* box. Enter some text here for the error message that is displayed if the user tries to skip filling out this form field.

Autofill: You can use the *Autofill* function to improve the auto-fill functionality offered by the browser. Browsers save form data entered previously, like addresses and payment details, and then offer these as suggestions for other forms with the same types of fields. To improve the quality of these suggestions, you can tell the browser about the kinds of information that are expected for a specific field. In this way, you can ensure that browsers make appropriate suggestions for auto-completing form fields, which also helps the user to fill out the forms faster.

Shipping or invoice address: Select “Shipping” or “Invoice” if this form field is part of a shipping address or an invoice address.

Contact type: Select an appropriate contact type here if users are expected to input a fixed-line or mobile phone number, fax or pager number, or an email address into this form field. Specifying a contact type will affect the selection offered to you for “Usage” (see below). This field is optional and can be left empty.

Usage: Specify the kinds of information entered into this form field.

If you want to stop the browser saving sensitive data and offering this as an autofill suggestion, then you can deactivate “*Autofill*”. To do so, move the slider to the left.

As of this writing, the autofill function is not supported by all web browsers. For an overview of browser compatibility, please see https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/autocomplete.

Duplicate Short text: Click *Duplicate Short text* to add a copy of this form field to your form.

Delete Short text: Click *Delete Short text* to delete this form field.

4.3.6. Long text

You use the predefined *long text* form element to add a multi-line input field to your form, which can also contain line breaks. This field is ideal for comments or messages, for example.

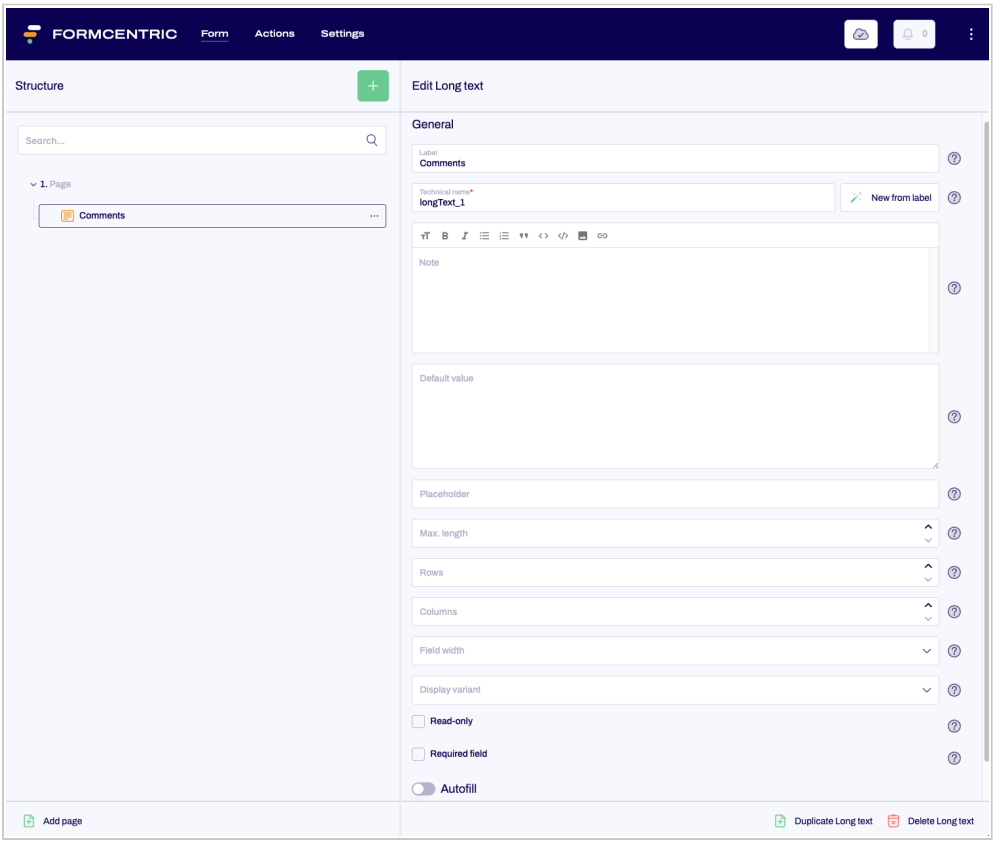


Figure 4.17. Long text

Label: Enter the text of the label that is displayed next to the form field.

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the "technical name" – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Note: Here, you have the option of adding some text to your form field that gives the user additional information. This could be instructions about filling out the field, for example.

Default value: In this field, you can enter a piece of text that is shown in the form field when the user accesses the form for the first time. Alternatively, you can add a variable here. This could insert the date automatically into the form field, for example (see Section 4.5, "Variables").

Placeholder: In this field, you can enter a piece of placeholder text that is displayed in the form field. This placeholder text disappears as soon as the user starts typing into the form field.

Max. length: Use this to specify the maximum number of characters that the user can enter into the input field.

Rows: Use this to set the height of the form field. If you enter the value 5, for example, five lines of text will be visible when multiple lines of text are entered.

Columns: Use this to set the width of the form field. If you enter the value "30", for example, then about 30 characters will fit into one form field line.

Field width: Use this to specify a display width for the form field.

Display variant: Select one of the available display variants here, so as to specify how the form field is displayed in the form. The variants available are specified on a per-project basis.

Read-only: If you check *Read-only*, the user will not be able to change the text in the form field. Use this setting if you want to enter predefined variables into this form field (see Section 4.5, "Variables").

Required field: Check *Required field* if this form field must be filled out when completing the form. An "*" will then be added to the end of the label, marking this

form field as a required field. The user will then be unable to submit the form or move to the next page (if this is a multi-page form) if they have not completed this form field properly.

Error message: An input field appears when you check the *Required field* box. Enter some text here for the error message that is displayed if the user tries to skip filling out this form field.

Autofill: You can use the *Autofill* function to improve the auto-fill functionality offered by the browser. Browsers save form data entered previously, like addresses and payment details, and then offer these as suggestions for other forms with the same types of fields. To improve the quality of these suggestions, you can tell the browser about the kinds of information that are expected for a specific field. In this way, you can ensure that browsers make appropriate suggestions for auto-completing form fields, which also helps the user to fill out the forms faster.

Shipping or invoice address: Select “Shipping” or “Invoice” if this form field is part of a shipping address or an invoice address.

Contact type: Select an appropriate contact type here if users are expected to input a fixed-line or mobile phone number, fax or pager number, or an email address into this form field. Specifying a contact type will affect the selection offered to you for “Usage” (see below). This field is optional and can be left empty.

Usage: Specify the kinds of information entered into this form field.

If you want to stop the browser saving sensitive data and offering this as an autofill suggestion, then you can deactivate “*Autofill*”. To do so, move the slider to the left.

As of this writing, the autofill function is not supported by all web browsers. For an overview of browser compatibility, please see https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/autocomplete.

Duplicate Long text: Click *Duplicate Long text* to add a copy of this form field to your form.

Delete Long text: Click *Delete Long text* to delete this form field.

4.3.7. Email

You use the predefined *email* form element to add a form field to your form that will check the text entered by the user, so as to confirm that the email address entered is valid in terms of its format.

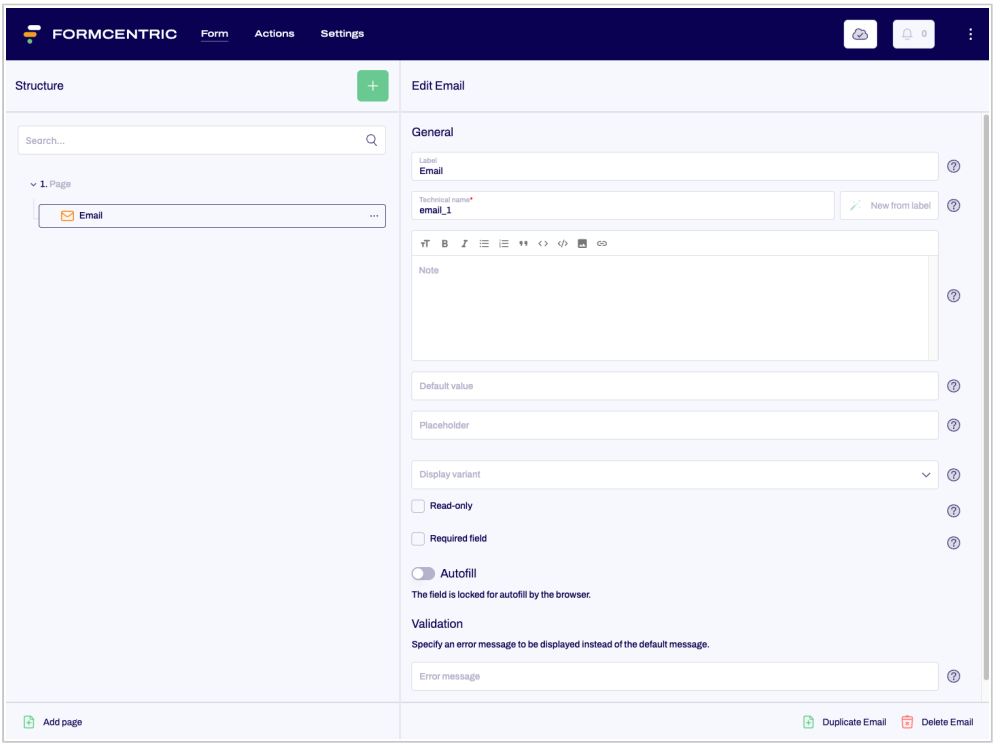


Figure 4.18. Email

Label: Enter the text of the label that is displayed next to the form field.

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the "technical name" – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Note: Here, you have the option of adding some text to your form field that gives the user additional information. This could be instructions about filling out the field, for example.

Default value: In this field, you can enter a piece of text that is shown in the form field when the user accesses the form for the first time. Alternatively, you can add a variable here (see Section 4.5, "Variables").

Placeholder: In this field, you can enter a piece of placeholder text that is displayed in the form field. This placeholder text disappears as soon as the user starts typing into the form field.

Display variant: Select one of the available display variants here, so as to specify how the form field is displayed in the form. The variants available are specified on a per-project basis.

Read-only: If you check *Read-only*, the user will not be able to change the text in the form field. Use this setting if you want to enter predefined variables into this form field (see Section 4.5, “Variables”).

Required field: Check *Required field* if this form field must be filled out when completing the form. An “*” will then be added to the end of the label, marking this form field as a required field. The user will then be unable to submit the form or move to the next page (if this is a multi-page form) if they have not completed this form field properly.

Error message: An input field appears when you check the *Required field* box. Enter some text here for the error message that is displayed if the user tries to skip filling out this form field.

Autofill: You can use the *Autofill* function to improve the auto-fill functionality offered by the browser. Browsers save form data entered previously, like addresses and payment details, and then offer these as suggestions for other forms with the same types of fields. To improve the quality of these suggestions, you can tell the browser about the kinds of information that are expected for a specific field. In this way, you can ensure that browsers make appropriate suggestions for auto-completing form fields, which also helps the user to fill out the forms faster.

Shipping or invoice address: Select “Shipping” or “Invoice” if this form field is part of a shipping address or an invoice address.

Contact type: Select an appropriate contact type here if users are expected to input a fixed-line or mobile phone number, fax or pager number, or an email address into this form field. Specifying a contact type will affect the selection offered to you for “Usage” (see below). This field is optional and can be left empty.

Usage: Specify the kinds of information entered into this form field.

If you want to stop the browser saving sensitive data and offering this as an autofill suggestion, then you can deactivate “*Autofill*”. To do so, move the slider to the left.

As of this writing, the autofill function is not supported by all web browsers. For an overview of browser compatibility, please see https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/autocomplete.

Validation: ***Error message:*** Enter some text here for an error message that should be displayed instead of the default error message. This is shown to the user if their input does not match the format that is required for a valid email address.

Duplicate Email: Click *Duplicate Email* to add a copy of this form field to your form.

Delete Email: Click *Delete Email* to delete this form field.

4.3.8. Number

You use the predefined *number* form element to add a form field to your form that only accepts numerical input from the user. If required, you can define criteria for this number that further restrict user input.

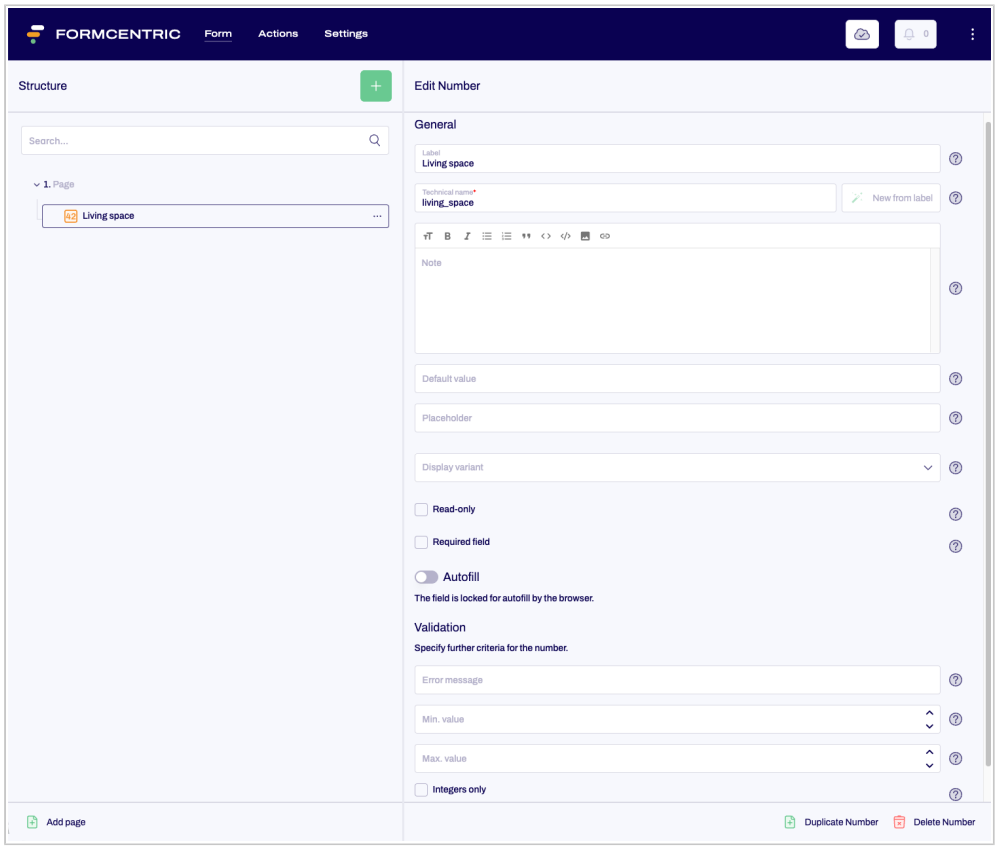


Figure 4.19. Number

Label: Enter the text of the label that is displayed next to the form field.

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the “technical name” – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Note: Here, you have the option of adding some text to your form field that gives the user additional information. This could be instructions about filling out the field, for example.

Default value: In this field, you can enter a piece of text that is shown in the form field when the user accesses the form for the first time. Alternatively, you can add a variable here (see Section 4.5, “Variables”).

Placeholder: In this field, you can enter a piece of placeholder text that is displayed in the form field. This placeholder text disappears as soon as the user starts typing into the form field.

Display variant: Select one of the available display variants here, so as to specify how the form field is displayed in the form. The variants available are specified on a per-project basis.

Read-only: If you check *Read-only*, the user will not be able to change the text in the form field. Use this setting if you want to enter predefined variables into this form field (see Section 4.5, “Variables”).

Required field: Check *Required field* if this form field must be filled out when completing the form. An “*” will then be added to the end of the label, marking this form field as a required field. The user will then be unable to submit the form or move to the next page (if this is a multi-page form) if they have not completed this form field properly.

Error message: An input field appears when you check the *Required field* box. Enter some text here for the error message that is displayed if the user tries to skip filling out this form field.

Autofill: You can use the *Autofill* function to improve the auto-fill functionality offered by the browser. Browsers save form data entered previously, like addresses and payment details, and then offer these as suggestions for other forms with the same types of fields. To improve the quality of these suggestions, you can tell the browser about the kinds of information that are expected for a specific field. In this way, you can ensure that browsers make appropriate suggestions for auto-completing form fields, which also helps the user to fill out the forms faster.

Shipping or invoice address: Select “Shipping” or “Invoice” if this form field is part of a shipping address or an invoice address.

Contact type: Select an appropriate contact type here if users are expected to input a fixed-line or mobile phone number, fax or pager number, or an email address into this form field. Specifying a contact type will affect the selection offered to you for “Usage” (see below). This field is optional and can be left empty.

Usage: Specify the kinds of information entered into this form field.

If you want to stop the browser saving sensitive data and offering this as an autofill suggestion, then you can deactivate “*Autofill*”. To do so, move the slider to the left.

As of this writing, the autofill function is not supported by all web browsers. For an overview of browser compatibility, please see https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/autocomplete.

Validation: Specify further criteria for the number if required.

Error message: Enter some text here for an error message that should be displayed instead of the default error message. This is shown to the user if the form field has not been filled out according to the rule(s) you have set.

Min. value: Specify the smallest number that the user is allowed to enter.

Max. value: Specify the largest number that the user is allowed to enter.

Integers only: Select *Integers only* if you want to prevent the user from entering decimal numbers.

Duplicate Number: Click *Duplicate Number* to add a copy of this form field to your form.

Delete Number: Click *Delete Number* to delete this form field.

4.3.9. Phone number

You use the predefined *phone number* form element to add a form field to your form that only accepts a phone number as input from the user. If required, you can define criteria for this phone number that further restrict user input.

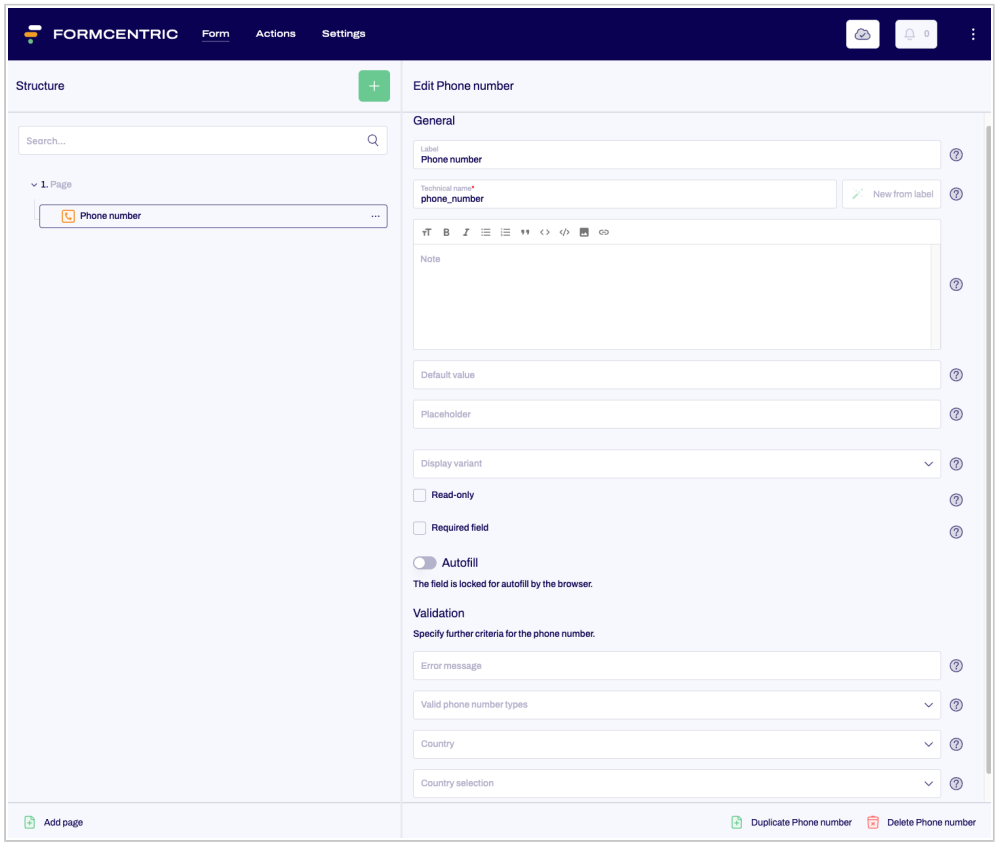


Figure 4.20. Phone number

Label: Enter the text of the label that is displayed next to the form field.

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the "technical name" – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Note: Here you have the option of adding some text to your form field that gives the user additional information, such as instructions about filling out the field.

Default value: In this field, you can enter a piece of text that is shown in the form field when the user accesses the form for the first time. Alternatively, you can add a variable here (see Section 4.5, "Variables").

Placeholder: In this field, you can enter a piece of placeholder text that is displayed in the form field. This placeholder text disappears as soon as the user starts typing into the form field.

Display variant: Select one of the available display variants here, so as to specify how the form field is displayed in the form. The variants available are specified on a per-project basis.

Read-only: If you check *Read-only*, the user will not be able to change the text in the form field. Use this setting if you want to enter predefined variables into this form field (see Section 4.5, "Variables").

Required field: Check *Required field* if this form field must be filled out when completing the form. An "*" will then be added to the end of the label, marking this form field as a required field. The user will then be unable to submit the form or move to the next page (if this is a multi-page form) if they have not completed this form field properly.

Error message: An input field appears when you check the *Required field* box. Enter some text here for the error message that is displayed if the user tries to skip filling out this form field.

Autofill: You can use the *Autofill* function to improve the auto-fill functionality offered by the browser. Browsers save form data entered previously, like addresses and payment details, and then offer these as suggestions for other forms with the same types of fields. To improve the quality of these suggestions, you can tell the browser about the kinds of information that are expected for a specific field. In this way, you can ensure that browsers make appropriate suggestions for auto-completing form fields, which also helps the user to fill out the forms faster.

Shipping or invoice address: Select “Shipping” or “Invoice” if this form field is part of a shipping address or an invoice address.

Contact type: Select an appropriate contact type here if users are expected to input a fixed-line or mobile phone number, fax or pager number, or an email address into this form field. Specifying a contact type will affect the selection offered to you for “Usage” (see below). This field is optional and can be left empty.

Usage: Specify the kinds of information entered into this form field.

If you want to stop the browser saving sensitive data and offering this as an autofill suggestion, then you can deactivate “*Autofill*”. To do so, move the slider to the left.

As of this writing, the autofill function is not supported by all web browsers. For an overview of browser compatibility, please see https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/autocomplete.

Validation: Specify further criteria for the phone number, if required.

Error message: Enter some text here for an error message that should be displayed instead of the default error message. This is shown to the user if the form field has not been filled out according to the rule(s) you have set.

Valid phone number types: Select the valid phone number types from this list. The phone number entered must then match these types. If you do not select any phone number types, the phone number is not validated.

Country: Select a country here if the phone number must come from a specific country.

Country selection: If your form contains a drop-down list (see Section 4.3.4, “Drop-down list”) with a country selection, then you can select the corresponding form field here. The phone number entered must then come from the country that the user selected from that drop-down list. If both a country and a country selection have been made, then the country selection has priority.

Please note that the *Value* fields for the entries in the drop-down list must contain the valid country codes according to ISO 3166 (e.g. *DE* fur Germany).

User input cannot be validated without valid country codes.

Duplicate Phone number: Click *Duplicate Phone number* to add a copy of this form field to your form.

Delete Phone number: Click *Delete Phone number* to delete this form field.

4.3.10. Date

You use the predefined *date* form element to add a form field to your form that only accepts a date as input from the user. You can also specify additional requirements for the date, such as setting a date range: the date entered by the user must then be within this period.

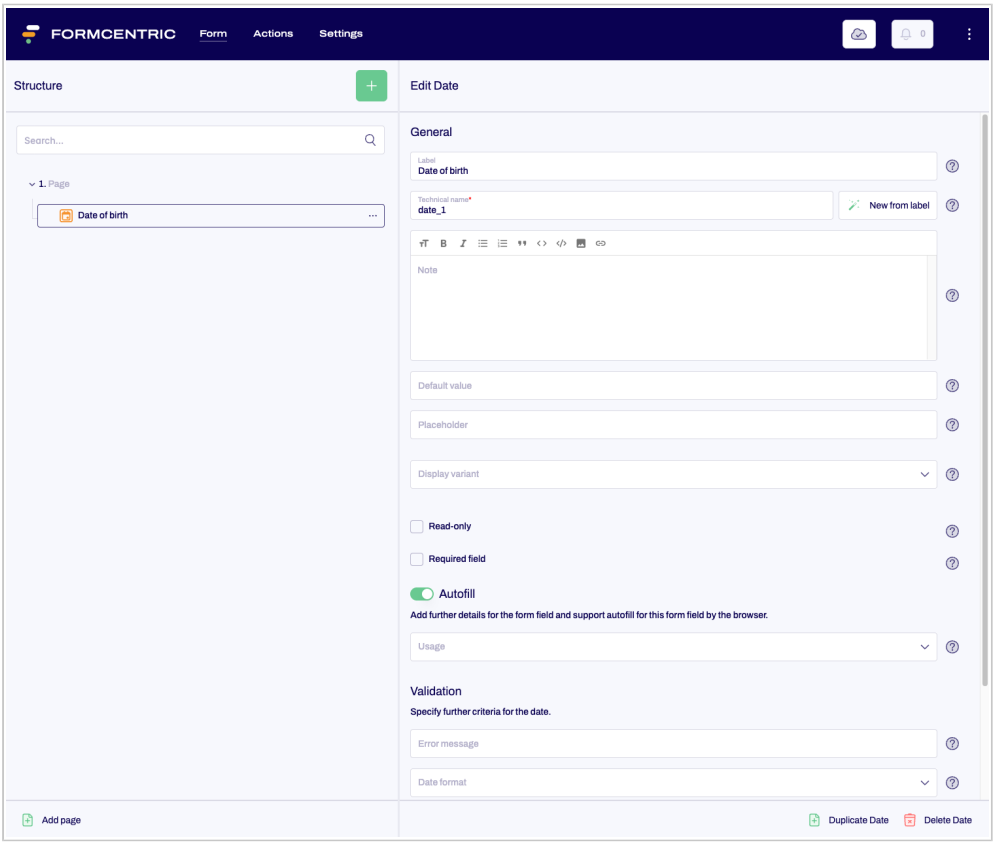


Figure 4.21. Date

Label: Enter the text of the label that is displayed next to the form field.

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the "technical name" – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Note: Here, you have the option of adding some text to your form field that gives the user additional information. This could be instructions about filling out the field, for example.

Default value: In this field, you can enter a piece of text that is shown in the form field when the user accesses the form for the first time. Alternatively, you can add a variable here (see Section 4.5, "Variables").

Placeholder: In this field, you can enter a piece of placeholder text that is displayed in the form field. This placeholder text disappears as soon as the user starts typing into the form field.

Display variant: Select one of the available display variants here, so as to specify how the form field is displayed in the form. The variants available are specified on a per-project basis.

Read-only: If you check *Read-only*, the user will not be able to change the text in the form field. Use this setting if you want to enter predefined variables into this form field (see Section 4.5, “Variables”).

Required field: Check *Required field* if this form field must be filled out when completing the form. An “*” will then be added to the end of the label, marking this form field as a required field. The user will then be unable to submit the form or move to the next page (if this is a multi-page form) if they have not completed this form field properly.

Error message: An input field appears when you check the *Required field* box. Enter some text here for the error message that is displayed if the user tries to skip filling out this form field.

Autofill: You can use the *Autofill* function to improve the auto-fill functionality offered by the browser. Browsers save form data entered previously, like addresses and payment details, and then offer these as suggestions for other forms with the same types of fields. To improve the quality of these suggestions, you can tell the browser about the kinds of information that are expected for a specific field. In this way, you can ensure that browsers make appropriate suggestions for auto-completing form fields, which also helps the user to fill out the forms faster.

Usage: Specify the kinds of information entered into this form field.

If you want to stop the browser saving sensitive data and offering this as an autofill suggestion, then you can deactivate “*Autofill*”. To do so, move the slider to the left.

As of this writing, the autofill function is not supported by all web browsers. For an overview of browser compatibility, please see https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/autocomplete.

Validation: Specify further criteria for the date if required.

Error message: Enter some text here for an error message that should be displayed instead of the default error message. This is shown to the user if the form field has not been filled out according to the rule(s) you have set.

Date format: Specify the format in which the date must be entered.

No time restrictions: If you select *No time restrictions*, then users will be able to enter any date.

Date range: Specify a set of start/end dates: the date entered must be between these dates.

Valid timespan: Enter values here to restrict the date entered by the user to a number of days before or after the form completion date.

Duplicate Date: Click *Duplicate Date* to add a copy of this form field to your form.

Delete Date: Click *Delete Date* to delete this form field.

4.3.11. Password

You use the predefined *password* form element to add a password field to your form. Characters entered by the user are not shown, but are represented by a line of dots. This gives the user a degree of privacy as they enter their password.

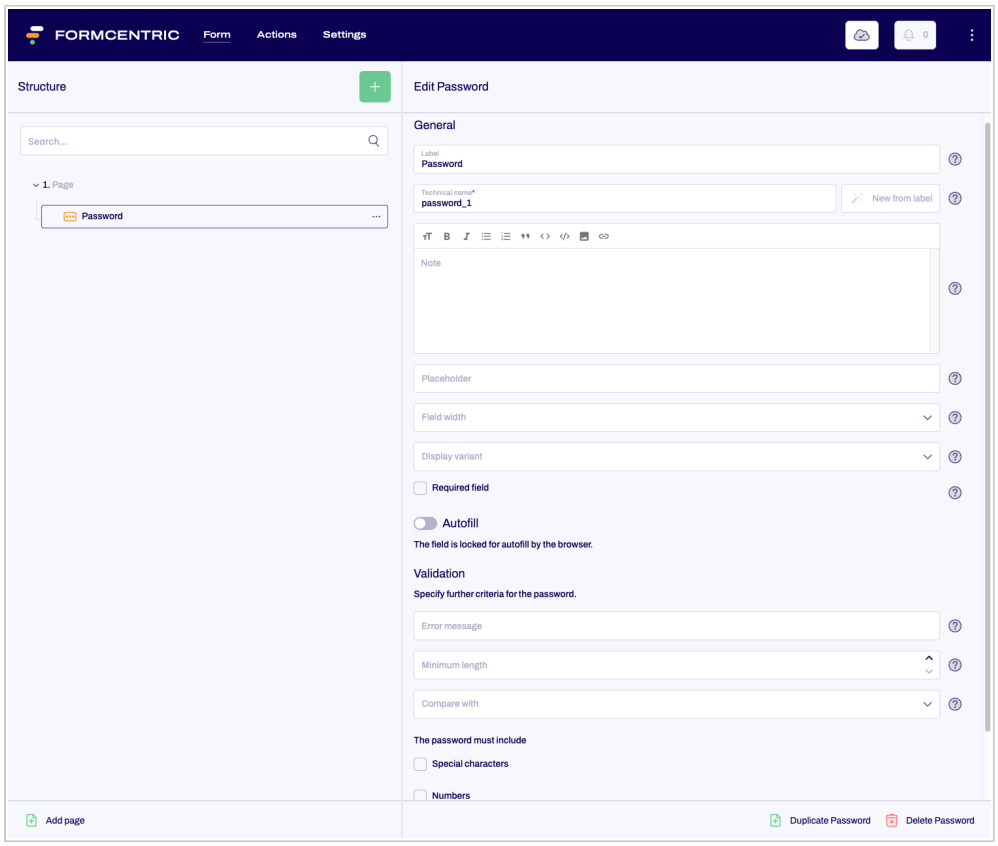


Figure 4.22. Password

Label: Enter the text of the label that is displayed next to the form field.

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the “technical name” – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Note: Here you have the option of adding some text to your form field that gives the user additional information. This could be instructions about how to pick a secure password, for example.

Placeholder: In this field, you can enter a piece of placeholder text that is displayed in the input field. This placeholder text disappears as soon as the user starts typing into the form field.

Field width: Use this to specify a display width for the form field.

Display variant: Select one of the available display variants here, so as to specify how the form field is displayed in the form. The variants available are specified on a per-project basis.

Required field: Check *Required field* if this form field must be filled out when completing the form. An "*" will then be added to the end of the label, marking this form field as a required field. The user will then be unable to submit the form or move to the next page (if this is a multi-page form) if they have not completed this form field properly.

Error message: An input field appears when you check the *Required field* box. Enter some text here for the error message that is displayed if the user tries to skip filling out this form field.

Autofill: You can use the *Autofill* function to improve the auto-fill functionality offered by the browser. Browsers save form data entered previously, like addresses and payment details, and then offer these as suggestions for other forms with the same types of fields. To improve the quality of these suggestions, you can tell the browser about the kinds of information that are expected for a specific field. In this way, you can ensure that browsers make appropriate suggestions for auto-completing form fields, which also helps the user to fill out the forms faster.

Usage: Specify the kinds of information entered into this form field.

If you want to stop the browser saving sensitive data and offering this as an autofill suggestion, then you can deactivate "*Autofill*". To do so, move the slider to the left.

As of this writing, the autofill function is not supported by all web browsers. For an overview of browser compatibility, please see https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/autocomplete.

Validation: Specify further criteria for the password field, if required.

Error message: Enter some text here for an error message that should be displayed instead of the default error message. This is shown to the user if the form field has not been filled out according to the rule(s) you have set.

Minimum length: Specify the minimum number of characters that must be used for the password.

Compare with: Select a second form field to compare with the input from this first form field.

Password must include: Check the relevant checkbox if the password must include special characters, numbers and/or lowercase/uppercase letters.

Duplicate Password: Click *Duplicate Password* to add a copy of this form field to your form.

Delete Password: Click *Delete Password* to delete this form field.

4.3.12. Upload file

You use the *upload file* form element to add an upload area to your form that the user can use to upload one or more files. Depending on the action selected, uploaded files are either sent as email attachments or stored in the web server's file system.

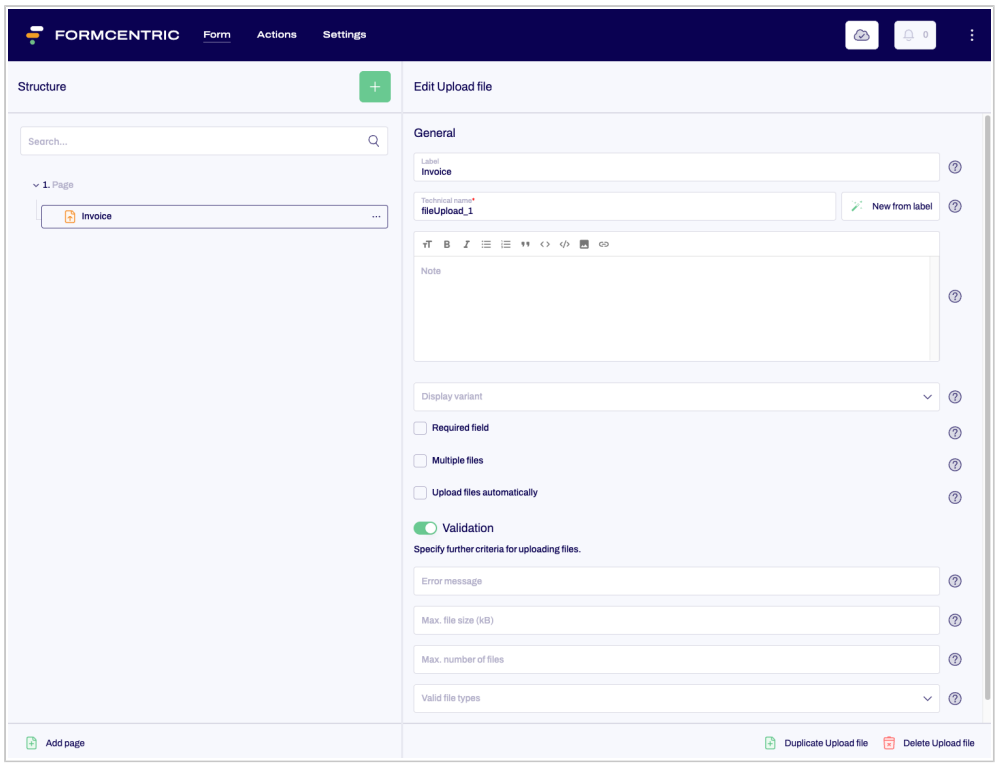


Figure 4.23. Upload file

Label: Enter the text of the label that is displayed next to the form element.

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the "technical name" – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name

must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Note: Here you have the option of adding some text to your upload area that gives the user additional information. This could be information about the maximum file size allowed, for example.

Display variant: Select one of the available display variants here, so as to specify how the form element is displayed in the form. The variants available are specified on a per-project basis.

Required field: Check *Required field* if at least one file must be uploaded when completing the form. An "*" will then be added to the end of the label for the upload file field, marking it as a required field. The user will then be unable to submit the form until they have uploaded a file.

Error message: An input field appears when you check the *Required field* box. Enter some text here for the error message that is displayed if the user does not upload a file.

Multiple files: Check *Multiple files* if you want to allow your user to upload more than one file.

Upload files automatically: Check *Upload files automatically* if files should be uploaded automatically as soon as the user has selected them. If this box is not checked, then selected files are uploaded only when the user actually clicks the *Upload* field.

Validation: Specify further criteria for the upload field, if required.

Error message: Enter some text here for an error message that should be displayed instead of the default error message. This error message is shown to the user if they attempt to upload a file that does not meet the criteria you have specified above.

Max. file size (kB): Specify the maximum file size here. The default maximum file size is set at 50 MB.

We recommend that you always limit the file size. Otherwise, you may run into problems with data traffic handling if several users attempt to upload large files at the same time.

Max. number of files: Use this to specify how many files the user can upload simultaneously.

Valid file types: Specify which kinds of files the user is allowed to upload. If you make no selections here, then any file type is allowed.

Duplicate Upload file: Click *Duplicate Upload file* to add a copy of this form field to your form.

Delete Upload file: Click *Delete Upload file* to delete this form field.

4.3.13. Hidden field

A *hidden field* is a form element that is not displayed on the actual form. Use hidden fields to access additional information about your user. If a hidden field is assigned the variable ${serverDate}, for example, this lets you find out the time and the date when the form was accessed.

Information contained in hidden fields is sent together with the values from the other form fields when the form is submitted.

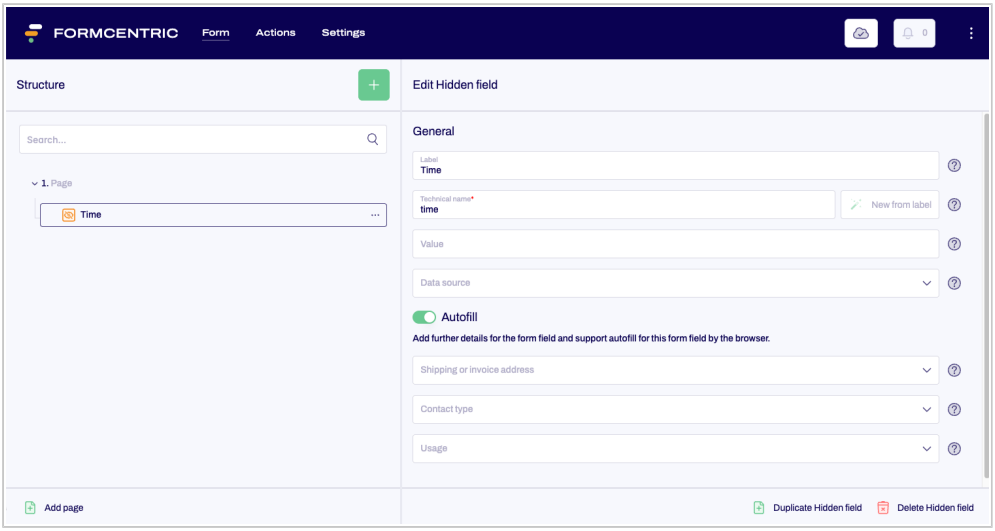


Figure 4.24. Hidden field

Label: Enter a piece of descriptive text that will be sent together with the value from the hidden field. This helps you to distinguish the values sent when checking your submissions.

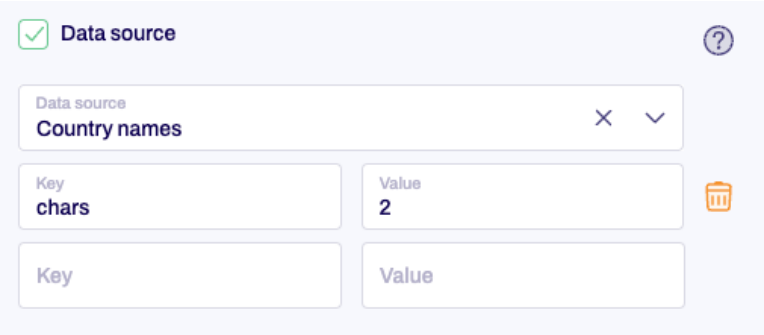
Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the “technical name” – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Value: Enter a value for the information that you want to receive from your hidden field – such as the ${serverDate} variable, for example. You can add one or more variables, depending on the information that you need (see Section 4.5, “Variables”).

Data source: Select a data source here that determines the value of the hidden field dynamically at runtime, i.e. at the exact moment when the user accesses the form.



The table gives you the option of entering additional parameters to pass to the data source. The parameters that are accepted depend on the data source selected. The *Country names* data source does not require any parameters.

Autofill: You can use the *Autofill* function to improve the auto-fill functionality offered by the browser. Browsers save form data entered previously, like addresses and payment details, and then offer these as suggestions for other forms with the same types of fields. To improve the quality of these suggestions, you can tell the browser about the kinds of information that are expected for a specific field. In this way, you can ensure that browsers make appropriate suggestions for auto-completing form fields, which also helps the user to fill out the forms faster.

Shipping or invoice address: Select “Shipping” or “Invoice” if this form field is part of a shipping address or an invoice address.

Contact type: Select an appropriate contact type here if users are expected to input a fixed-line or mobile phone number, fax or pager number, or an email address into this form field. Specifying a contact type will affect the selection offered to you for “Usage” (see below). This field is optional and can be left empty.

Usage: Specify the kinds of information entered into this form field.

If you want to stop the browser saving sensitive data and offering this as an autofill suggestion, then you can deactivate “*Autofill*”. To do so, move the slider to the left.

As of this writing, the autofill function is not supported by all web browsers. For an overview of browser compatibility, please see https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/autocomplete.

Duplicate Hidden field: Click *Duplicate Hidden field* to add a copy of this form element to your form.

Delete Hidden field: Click *Delete Hidden field* to delete this form element.

4.3.14. Calculated value

You use the *calculated value* form element to calculate a value from the input that the user enters into the form. The value is calculated using JavaScript code.

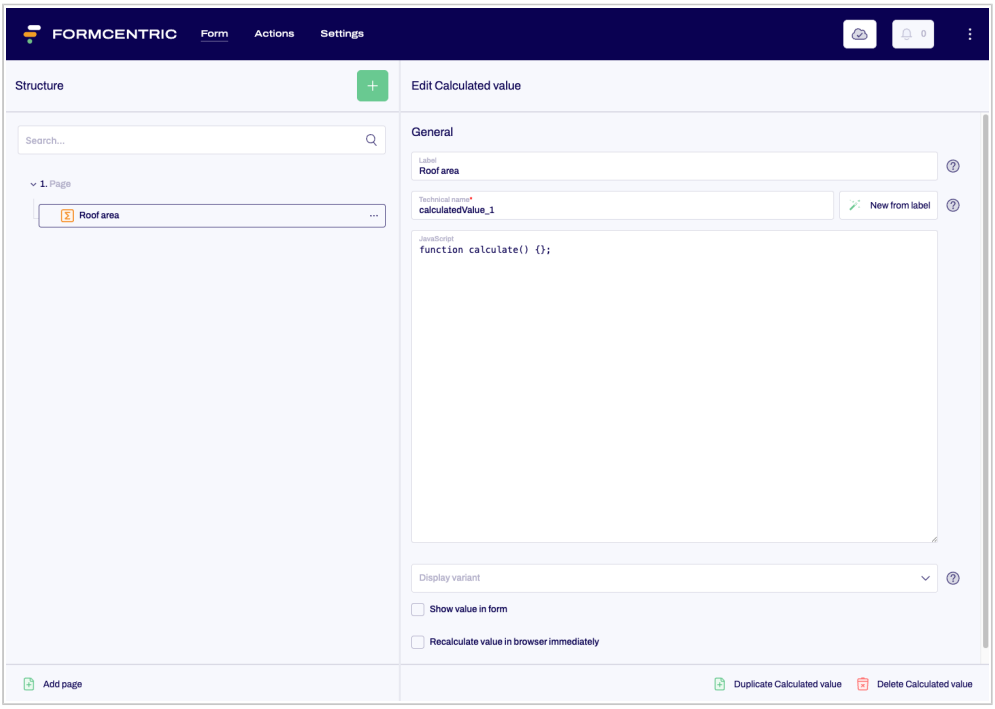


Figure 4.25. Calculated value

Label: Enter a piece of text to use as the label for the value. This label will then be shown in the summary, for example.

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the "technical name" – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

JavaScript: Enter the JavaScript code for calculating the value inside the brackets for the JavaScript "calculate()" function. This JavaScript calculate() function is executed whenever the user moves from one page to another in the form or clicks the submit button.

If you would like to calculate the age of a person for a given date of birth – so as to then be able to use this in a condition, for example – then you can achieve this with the following function. In your form, include an input field (technical name: *birthdate*) in which users are asked to enter their date of birth.

```
function calculate() { return parseAge("yyyy/MM/dd", birthdate); }
```

Display variant: Select one of the available display variants here, so as to specify how the form element is displayed in the form. The variants available are specified on a per-project basis.

Show value in form: Check *Show value in form* to have the result of executing the JavaScript code, i.e. the calculated value, displayed in the form.

Recalculate value in browser immediately: If you activate *Recalculate value in browser immediately*, the value is calculated in real time. The value is then recalculated whenever the user enters input into a form field that is relevant for the calculation made by the JavaScript code. If you do not check this box, then the value is not (re)calculated until the user moves to another page in the form or clicks the submit button.

Tip: Check *Recalculate value in browser immediately* if you are linking the value to a condition. This ensures that the condition will work properly.

Duplicate Calculated value: Click *Duplicate Calculated value* to add a copy of this form element to your form.

Delete Calculated value: Click *Delete Calculated value* to delete this form element.

4.3.15. CAPTCHA

You use this form element to add a “CAPTCHA” (acronym for “Completely Automated Public Turing test to tell Computers and Humans Apart”) to the form. CAPTCHAs are used to ensure that the form is being filled out by a human and not by an automated system.

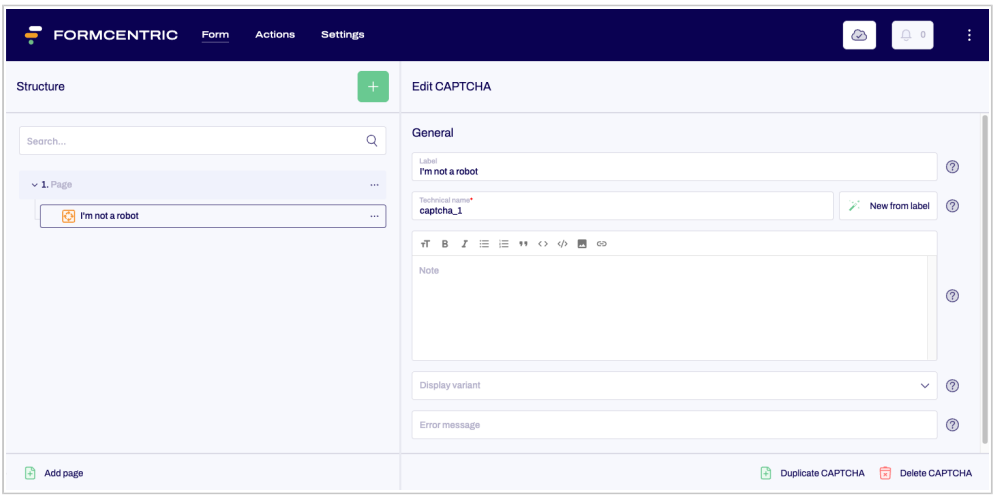


Figure 4.26. CAPTCHA

Label: Enter the text of the label that is displayed next to the CAPTCHA.

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the “technical name” – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly

identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Note: Here you have the option of adding some text to your form field that gives the user additional information. This could be information about why CAPTCHAs are being used, for example.

Display variant: Select one of the available display variants here, so as to specify how the form element is displayed in the form. The variants available are specified on a per-project basis.

Error message: An alternative error message that is output instead of the standard error message.

Duplicate CAPTCHA: Click *Duplicate CAPTCHA* to add a copy of this form element to your form.

Delete CAPTCHA: Click *Delete CAPTCHA* to delete this form element.

4.3.16. Condition

You use the *condition* form element to make dynamic changes to the state of individual form elements based on user input. As one example, you can use this element to ensure that a user only sees relevant form fields while hiding any fields that are not needed for this user. This can also be used to control other kinds of states: optional fields can be changed to required fields, active elements can be deactivated and fields can be set as read-only.

Remember that **implicit conditions** always apply automatically. If you have a form element whose state is changed when the condition is fulfilled, then the **opposite state** – i.e. the state that is opposite to the state specified in the condition – always applies as the initial state.

If a text field should be displayed if the user checks a box, for example, then this text field is hidden by default when the form is first accessed. The text field will only appear when the box is checked and the condition is therefore fulfilled. Accordingly, you do not need to set up a separate condition for the opposite scenario.

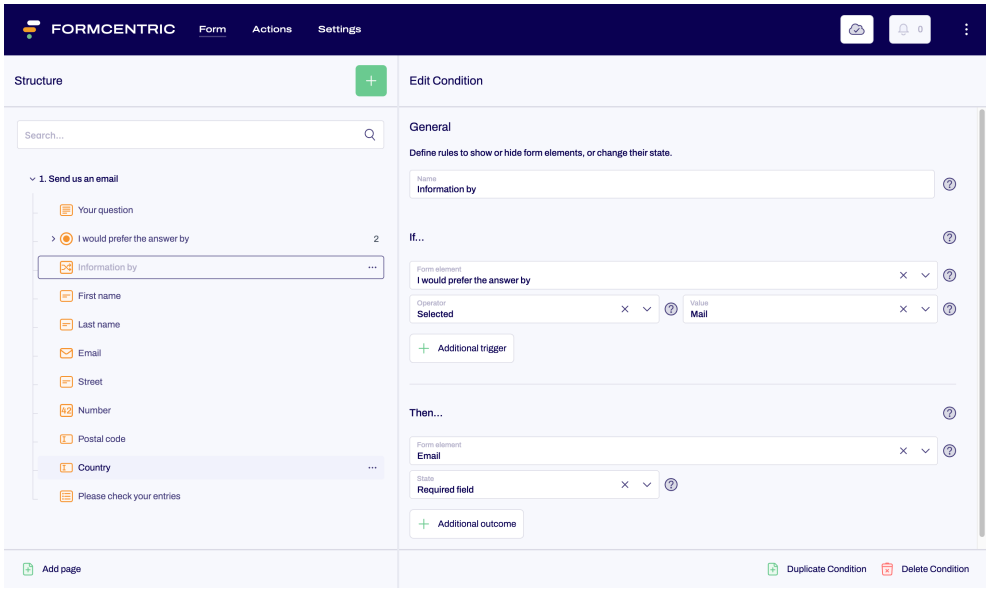
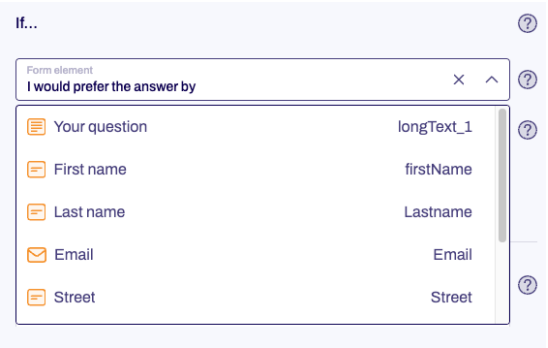


Figure 4.27. Condition

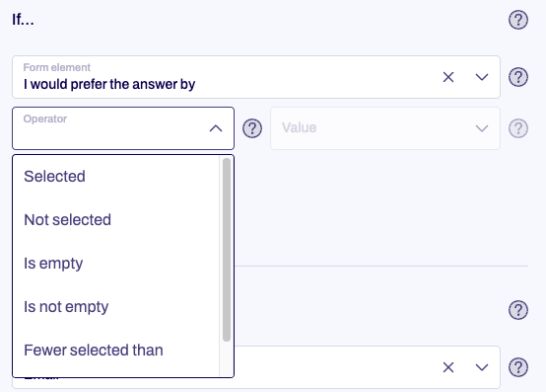
Name: Enter a name for the condition to help you identify it later. The name is shown only in the form tree.

If...: Specify the trigger for the condition so that the outcome takes place as specified in the *Then...* section.

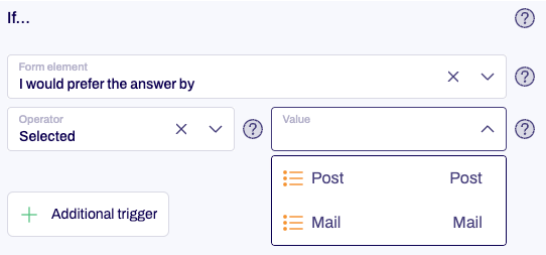
Select the form element that will be used as the trigger for the condition.



You use the *Operator* column to specify the logical operator for the condition. This is then applied to the comparison value specified in the *Value* column. Various operators will be available here, depending on whether you have selected a selection field or a text input field for the form element.



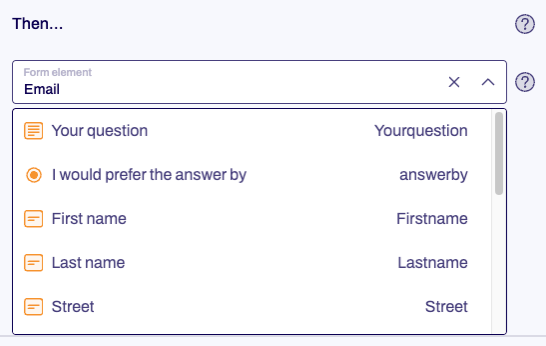
In the *Value* column, enter the comparison value or, in the case of a drop-down list, select the corresponding option.



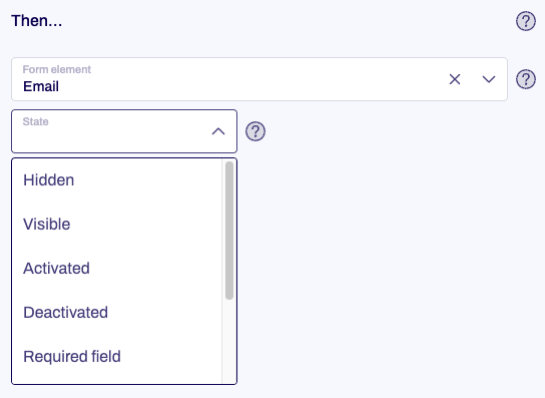
If you define multiple triggers, then you need to specify whether one or all triggers must be activated for the outcome to happen specified in *Then...*.

Then...: Under *Then...*, you specify which action is carried out when the criterion specified under *If...* has been met.

Select the form element to be referenced by the trigger that has been specified under *If...*.



In *State*, select the outcome that should happen when the criteria for the trigger are fulfilled. Various states will be available here, depending on whether you have selected a selection field or a text input field for the form element.



While the condition remains unfulfilled, the form element will always be in the opposite state.

You never need to set up a separate condition for the opposite scenario! By default, the opposite logic always applies to the form element until the condition is fulfilled. If a form element is to be shown when a specific option is selected from a drop-down list, for example, then this form element will be hidden by default. The element is only revealed when the user selects the specified option and thereby fulfils the condition.

Extra conditions for hiding the element when other options are selected are therefore not necessary. Multiple conditions aimed at achieving the same state can overlap or contradict one another, which can have unexpected effects on your form's behaviour. Accordingly, you only need to define the state you require in the event of the condition being fulfilled. While the condition remains unfulfilled, the opposite logic will automatically apply.

Please note: You can only specify the states *Optional*/*Required field* and *Editable*/*Read-only* for form elements that do not have their *Required field* or *Read-only* checkboxes checked.

Hidden form elements are also hidden on the summary pages and in any emails that are sent. Values from deactivated form elements are ignored.

Duplicate Condition: Click *Duplicate Condition* to add a copy of this form element to your form.

Delete Condition: Click *Delete Condition* to delete this form element.

4.3.17. Paragraph

You use the *paragraph* form element to add a block of text anywhere in your form. This is a read-only piece of text that cannot be changed by the user. This can be used to offer advice or give explanations, for example.

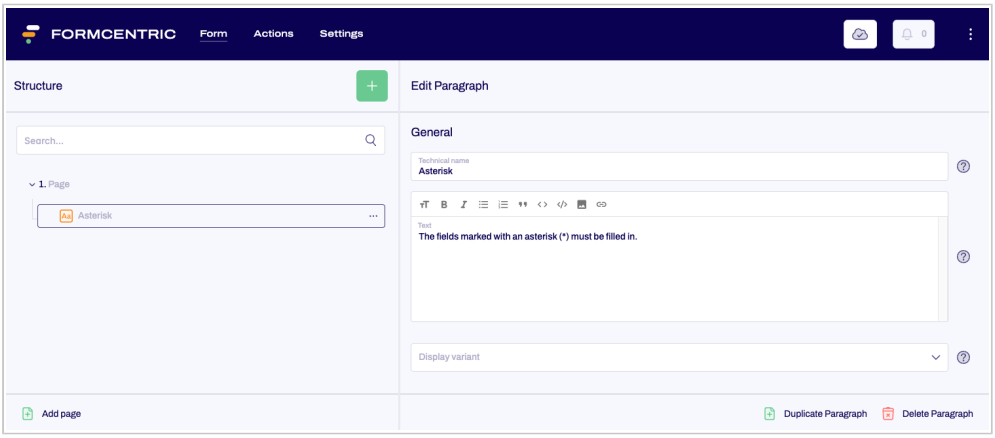


Figure 4.28. Paragraph

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the "technical name" – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

Text: Enter a piece of text here to be displayed in your form. You can use Section 4.6, "Markdown" to format the text.

Display variant: Select one of the available display variants here, so as to specify how the form element is displayed in the form. The variants available are specified on a per-project basis.

Duplicate Paragraph: Click *Duplicate Paragraph* to add a copy of this form element to your form.

Delete Paragraph: Click *Delete Paragraph* to delete this form element.

4.3.18. Button

You use the *button* form element to include a JavaScript action in your form. This action is executed when the user clicks the button.

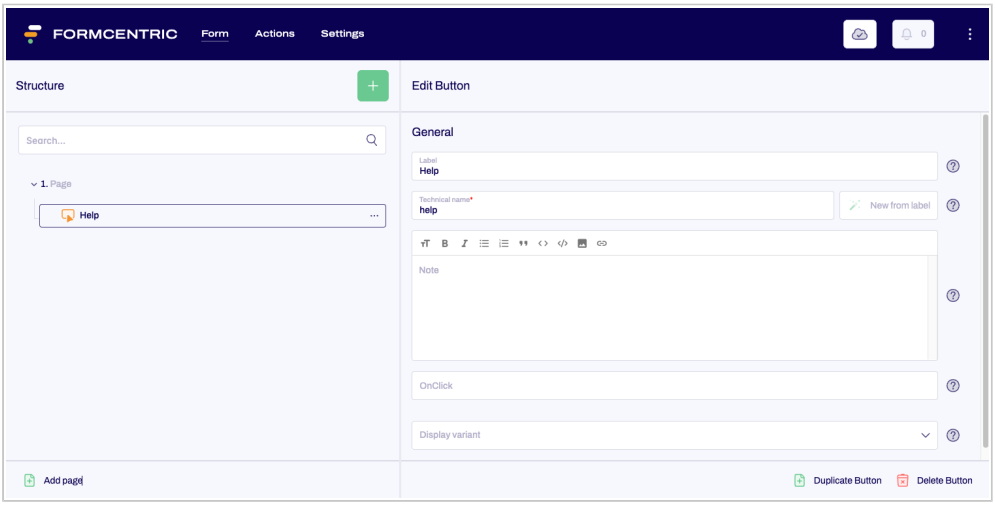


Figure 4.29. Button

Label: Enter a piece of text to be displayed on the button.

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the "technical name" – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Note: Here you have the option of adding some text to your button that gives the user additional information.

OnClick: To specify what should happen when the button is clicked, use the event handler *onclick* here, so as to respond to the click with JavaScript.

You can define display variants for this form element in the configuration. Please see the Developer Manual for instructions. These will be shown as drop-down lists, in the same way as for the input field form element, for example. Please see the Developer Manual for instructions.

Display variant: Select one of the available display variants here, so as to specify how the form element is displayed in the form. The variants available are specified on a per-project basis.

Duplicate Button: Click *Duplicate Button* to add a copy of this form element to your form.

Delete Button: Click *Delete Button* to delete this form element.

4.3.19. Layout

You use the *layout* form element to combine multiple form elements into a single group. You can then assign a display variant to this group, so as to create a two-column layout, for example.

To define a group, add the layout form element to the form tree. You then add the form elements you want to combine into a group to the layout element.

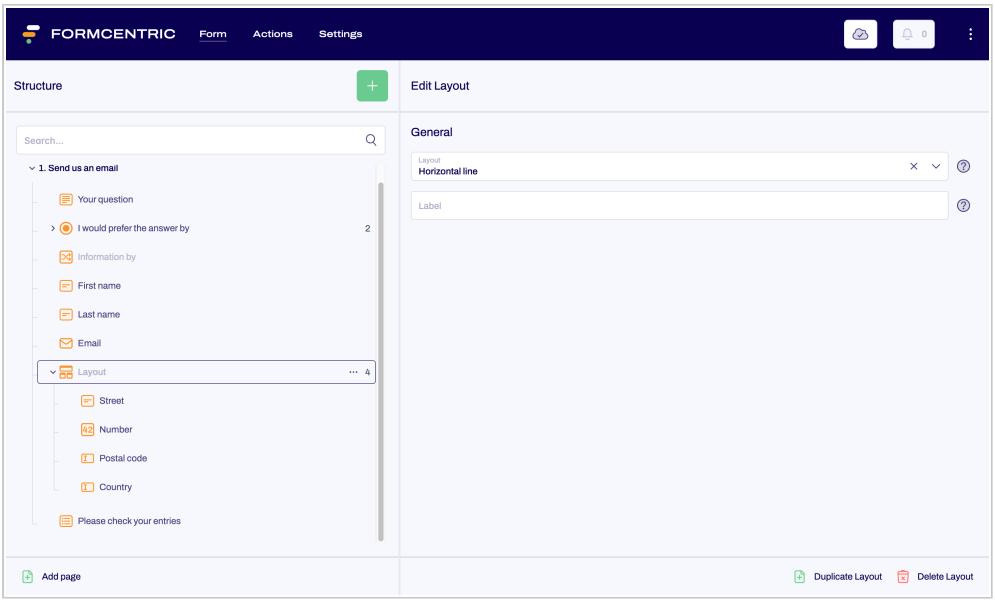


Figure 4.30. Layout

Layout: You use this field to select the available display variants and therefore specify how the group should be displayed within your form. The variants available are specified on a per-project basis.

Label: You can use this field to enter an optional piece of label text. Whether (and where) the label is displayed in the form depends on the layout selected.

Duplicate Layout: Click *Duplicate Layout* to add a copy of this form element to your form.

Delete Layout: Click *Delete Layout* to delete this form element.

4.3.20. Summary

The *summary* form element presents an overview of all of the items of data that the user has entered into the form.

You can use the summary as the last page of the form, for example, so as to give the user an overview of the data that they have entered before the form is finally submitted. If the user finds something that they want to change, they can use the “Back” button to go back and make those changes. Once everything is correct, they can then submit the form.

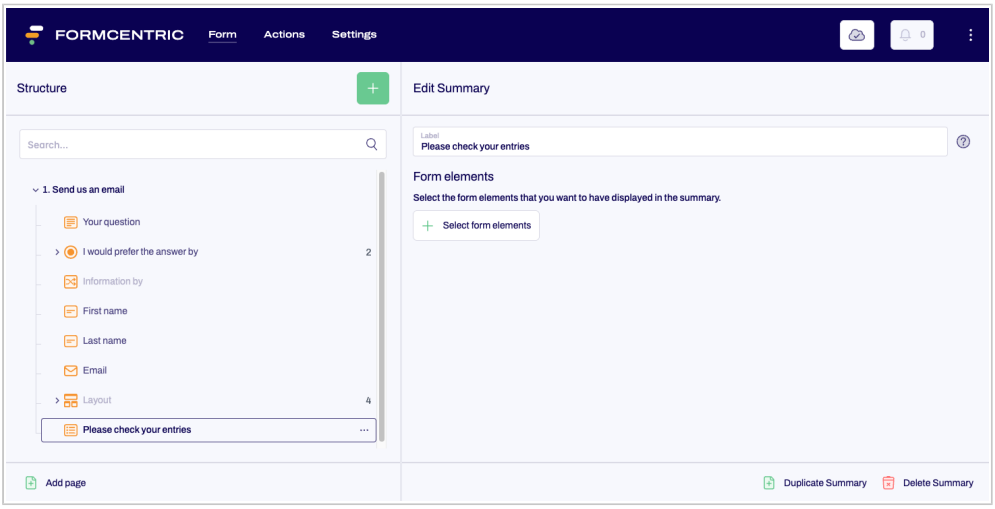


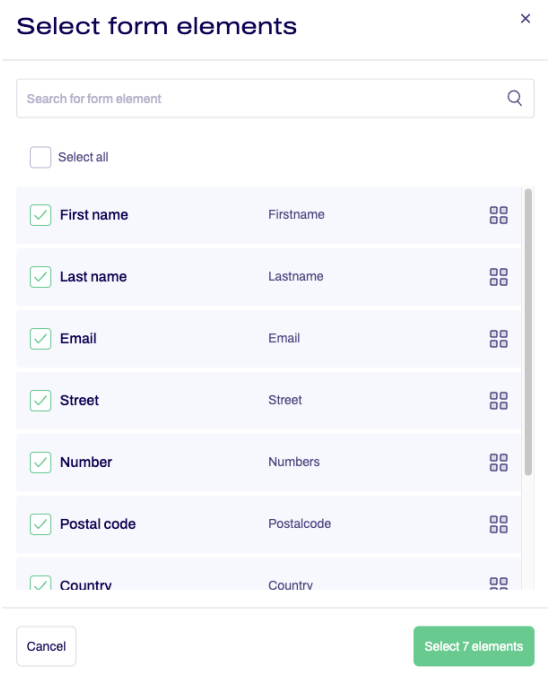
Figure 4.31. Summary

Label: Enter the text of the label that is displayed together with the summary.

Select form elements: Click *Select form elements* and choose the form elements that you want to have displayed in the summary. In the summary, the form elements will be displayed in the order in which they are listed here. If you want to use a different order, you can use drag-and-drop to rearrange your form elements.

Use *Select all* to select all of the elements in one go. Use *Unselect all* to cancel this selection. Click the green button at the bottom right to confirm your selection.

Confirm your selection to close the dialogue field. You can now see the form elements that you have selected in the editing area. You can click the recycle bin on the right to delete individual form elements from your selection.



If you select a form element for the summary and then change its technical name later, this element is automatically removed from the summary. Accordingly, you will need to select the form element again to have it included in the summary.

Hide empty fields: If you check *Hide empty fields*, then the summary will only show the form fields where the user has entered something into the field or selected something from the field. Empty form fields will not be shown.

Duplicate Summary: Click *Duplicate Summary* to add a copy of this form element to your form.

Delete Summary: Click *Delete Summary* to delete this form element.

4.3.21. Document

You use the *document* form element to embed CoreMedia documents into your form. You can embed an image between two input fields, for example.

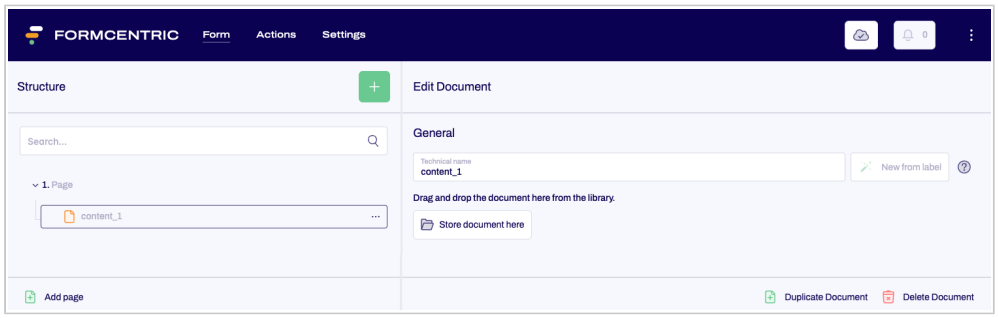


Figure 4.32. Document

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the “technical name” – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Store document here: Simply use drag-and-drop to move the CoreMedia document from the Library and insert it into your form.

Duplicate Document: Click *Duplicate Document* to add a copy of this form element to your form.

Delete Document: Click *Delete Document* to delete this form element.

4.3.22. Fieldset

You use the fieldset form element to group multiple form elements together under a single heading.

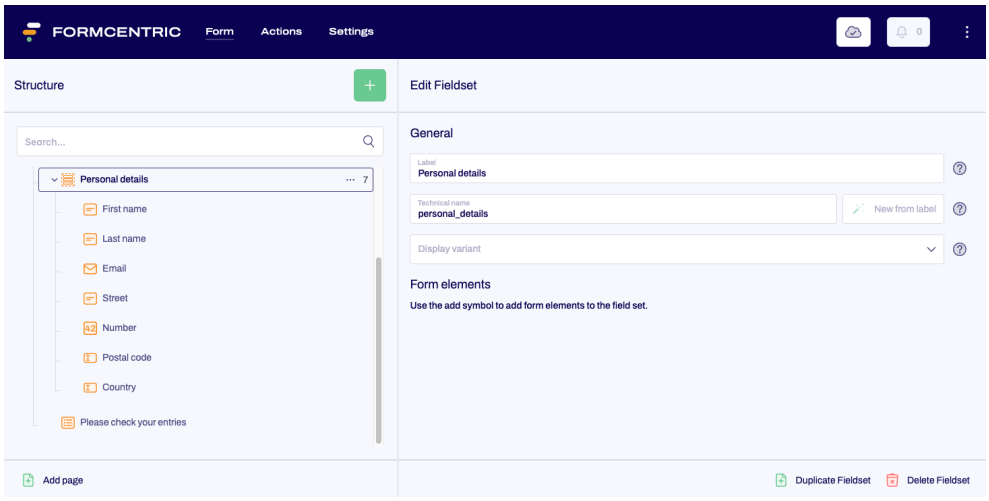


Figure 4.33. Fieldset

Label: Enter the text of the label that is displayed next to the fieldset.

Technical name: Each form element that is added to a form is automatically assigned a unique name – known as the "technical name" – by Formcentric. This is necessary for technical reasons, to ensure that the form element can be properly identified and processed. The name is based on the designator for the form element and can be changed if required.

The technical name is a required field. The technical name must start with a letter and must not contain any special characters. In addition, the name must not match an identifier that is listed under Appendix A, *Reserved identifiers*. A technical name can be used only once within a form.

New from label: Alternatively, you can generate a technical name from the label. To do this, click *New from label*.

Display variant: Select one of the available display variants here, so as to specify how the fieldset is displayed in the form. The variants available are specified on a per-project basis.

4.3.23. Formatting labels

You can format form element labels by using Markdown in the corresponding input field.

Bold: To display text in bold, place the desired text between double asterisks (**Fett**) or underscores (__Fett__).

For example: **First name** appears as: **First name**

Italic: For italic text, use single asterisks (*Italic*) or underscores (_Italic_).

For example: *Last name* appears as: *Last name*

Links: You can insert links to additional content: [Linktext](https://example.com)

For example: I agree to the [privacy policy](https://beispiel.com).

Underlined: To underline text, place double plus signs before and after the desired text: ++Underlined++

For example: ++First name++ appears as: First name

Strikethrough: To display text with a strikethrough, place double tilde characters before and after the desired text: : ~~Strikethrough text~~

Using special characters in text: If you want to display these special characters as regular text without triggering formatting, you need to escape them. To do this, insert a backslash (\) directly in front of the character.

For example: Password (minimum **8 characters**) appears as: Password (minimum **8 characters**)

4.4. Data sources

A typical requirement when putting together forms is creating lists that offer the selection of a large number of options or a range of variable selection options. To help with this process, Formcentric provides data sources that let you create selection lists or input fields at runtime that are fed with data from external systems. This data can be static, dynamic or user-specific.

To make the data sources available for selection in the editor, they must be defined in the *FormcentricSettings* document, which is located in the sample content of our extension.

When selecting a data source for a form field, you also have the option of specifying additional configuration parameters. These parameters let you set the language for the data source output, for example. The configuration parameters available will depend on the specific data source that you select.

Data sources provided as standard are listed below, together with their configuration parameters:

4.4.1. Country names

This data source creates a list of country names. By specifying a region, the country data included in the list can be restricted to a geographical or organisational subset.

Key	Description
chars	The minimum number of characters that users must enter into the field before an autocomplete entry is shown.
lang	Language in which the country names should be output in the list. The following languages are supported:

Key	Description
	• *de* – German • *en* – English • *fr* – French • *es* – Spanish • *it* – Italian • *ru* – Russian
region	Use this to restrict the autocomplete to a specific region: • *global* – all countries (default setting) • *emea* – Europe, Middle East and Africa • *apac* – Asia-Pacific • *australia* – Australasia • *north-america* – North America • *south-america* – South America • *central-america* – Central America • *asia* – Asia • *africa* – Africa • *oceania* – Oceania You can select the following regions when using the postcode validator: • *europe* – European countries • *eu* – Member states of the European Union • *dach* – Austria, Germany and Switzerland • *efta* – Member states of the European Free Trade Association • *zip* – All countries whose postcodes (zip codes) can be validated by the postcode validator

4.4.2. Weekdays

This data source creates a list of weekdays. The key *first* determines which weekday appears first in the list.

Key	Description
first	• *mon* - Monday • *tue* - Tuesday • *wed* - Wednesday • *thu* - Thursday • *fri* - Friday • *sat* - Saturday • *sun* - Sunday

4.4.3. Months

This data source creates a list of months.

It is possible to preselect a month in a *single choice*, *multiple choice*, or *drop-down list*.

To do this, the corresponding parameter for the desired month is entered in the *Preselection* field: January (jan), February (feb), March (mar), April (apr), May (may), June (jun), July (jul), August (aug), September (sep), October (oct), November (nov), December (dec).

4.4.4. Years

This data source creates a list of years, with a maximum span of 100 years between them. The *from* and *to* keys can be used to limit the range of the result list.

4.5. Variables

You can draw on a range of variables when setting default values for input fields. These variables are replaced with a value when the form is displayed. As one example, you can set a field to have the current date as a default value by entering the variable *${clientDate}* into the field's default value setting. Variables must always be specified using the format *${name-of-the-variable}*.

All variables can be combined with additional text or other variables. The following variables are available to you as standard.

Variable	Description
date	The date, in the time zone UTC±0, on which the user completed the form. The output format is determined by the browser language configured by the user (example: UK English = 30/03/2014).
time	The time, in the time zone UTC±0, at which the user completed the form. The output format is determined by the browser language configured by the user (example: UK English = 18:36).
serverDate	The date, in the server's time zone, on which the user completed the form. The date references the server time zone. The output format is determined by the browser language configured by the user (example: UK English = 30/05/2013).
serverTime	The time, in the server's time zone, at which the user completed the form. The time references the server time zone. The output format is determined by the browser language configured by the user (example: UK English = 17:33).
clientDate	The date, in the user's time zone, on which the user completed the form. The output format is determined by the browser language configured by the user (example: UK English = 30/05/2013).
clientTime	The time, in the client's time zone, at which the user completed the form. The output format is determined by the browser language configured by the user (example: UK English = 17:33).
timezone	The time zone that the user has configured for their browser (example: Europe/London).
language	The language configured for the user's browser. This is given in the form of the country code (de, en, etc.).
ip	The IP address assigned to the user's computer by their provider.
remoteUser	Name with which the user logged in to your website. Please note: This variable is available only if the login was handled by the deployed Application Server.
principal	Name of the authenticated user (from the principal object). Please note: This variable is available only if the login was handled by the deployed Application Server or JAAS is being used.
userAgent	Identification string supplied by the user's browser.
referer	The URL used to access the web page containing the form (also known as the referrer page or the history page in browser jargon). A relative or absolute URL is given as the value.

As an example, if you want to use the date and time to set a default value for a field, then you can achieve this by entering the following in the *Value* field:

Input: *${clientDate} ${clientTime}*

Output: *30/03/2014 18:36*

4.6. Markdown

Markdown is a simple mark-up language that you can use to add formatting and links to pieces of plain text. Markdown is available as standard in the following areas:

- Paragraph
- Note text
- Confirmation text

Examples of common kinds of Markdown formatting are shown in the following table:

Formatting	**Text as input**	**Text as displayed**
Bold	Example for text in **bold type**	Example for text in **bold type**
Italics	Example for text in _italic type_	Example for text in *italic type*
Ordered lists	1. Element 1 2. Element 2	1. Element 1 2. Element 2
Bulleted lists	* Element 1 * Element 2	• Element 1 • Element 2
Headings	# Heading level 1 ## Heading level 2 ### Heading level 3	Heading level 1 Heading level 2 Heading level 3
Links	[Link text](http://sample-url.com) {param1=value1,...,paramN=valueN}	Link text

For full details of the formatting options available with Markdown, please visit https://commonmark.org/.

5. Actions screen

On the *Actions* screen, you specify how the data entered by the user should be processed. Various actions are available to you, which you can add to your form either individually or as combinations.

If you want to send the data that has been entered as an email, for example, you simply select the *send as email* action.

If there is no condition present that would prevent the action being executed, then the action is executed when the user clicks the submit button. For multi-page forms, the submit button is located on the last page of the form.

On the *Actions* screen, click the green plus sign in the *Structure* area. A dialogue opens, listing all of the available actions. Click an action to add it to the form.

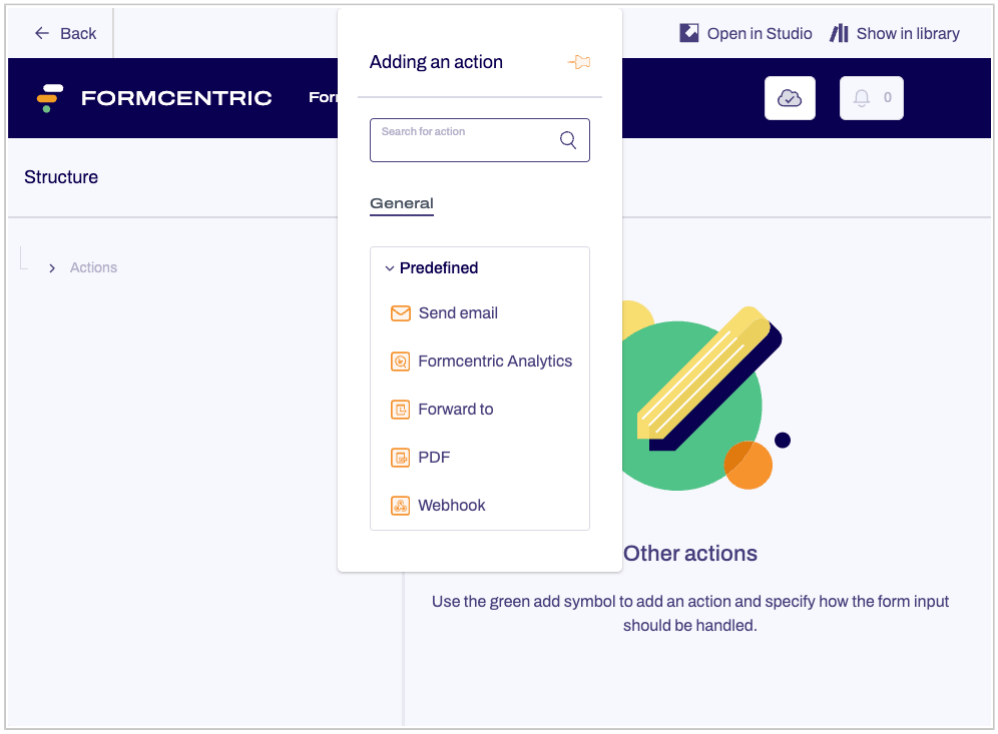


Figure 5.1. Adding an action

The following section describes all of the actions in detail.

5.1. Send email

The *Send email* action sends data entered into the form as an email attachment to any number of recipients.

You can individually select which of the entered data is sent.

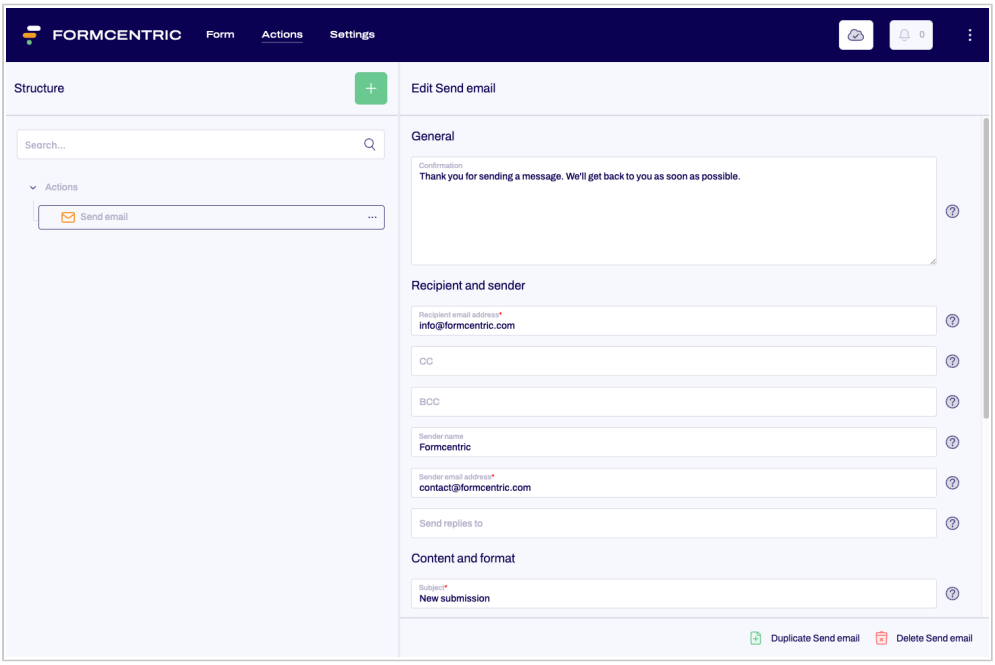


Figure 5.2. Send email action

5.1.1. General

Confirmation: Enter the confirmation message that is shown to the user on the web page after the form has been submitted.

5.1.2. Recipient and sender

Recipient email address: Enter one or more comma-separated email addresses to which the data entered by the user will be sent.

CC: Enter one or more comma-separated email addresses to which a copy of the data entered by the user will be sent.

BCC: Enter one or more comma-separated email addresses to which a blind copy of the data entered by the user will be sent. These email addresses are not visible to other recipients.

Sender name: Enter the name to be displayed as the sender.

Sender email address: Enter the email address to be used as the sender.

Send replies to: If answers to the email from this action should be sent to an address that is different to the sender email address, then you can enter this email address here. Use commas to separate multiple email addresses.

5.1.3. Content and format

Subject: Enter the subject here that is displayed to the user when they receive the mail.

Message: Enter a piece of text to be added to the body of the email, in addition to the form data.

Select form elements: Click *Select form elements* and choose the form elements that you want to have displayed in the mail. In the mail, the form elements will be displayed in the order in which they are listed here. If you want to use a different order, you can use drag-and-drop to rearrange your form elements.

Use *Select all* to select all of the elements in one go. Use *Unselect all* to cancel this selection. Click the green button at the bottom right to confirm your selection.

Confirm your selection to close the dialogue field. You can now see the form elements that you have selected in the editing area. You can click the recycle bin on the right to delete individual form elements from your selection.

Hide empty fields: Click *Hide empty fields* to ensure that only completed form fields are included in the email.

Email format: This field lets you specify whether the email should be sent in HTML format or as a plain text message.

Format	Description
HTML	Creates an HTML mail with the specified message text and the selected form values. The form values are added automatically as a simple list (label : value) at the end of the message text.
Text	Creates a plain text mail with the specified message text and the selected form values. The form values are added automatically as a simple list (label : value) at the end of the message text.
FreeMarker (text)	Creates a plain text email. When this format option is selected, the message text is interpreted and executed as a FreeMarker template. With this format option, the form values must be added manually to the message text.
FreeMarker (HTML)	Creates an HTML email. When this format option is selected, the message text is interpreted and executed as a FreeMarker template. With this format option, the HTML code and form values must be added manually to the message text.

5.1.4. Conditions

Add conditions: Use this field to specify the user input for which the action should be executed/not be executed. To do this, proceed as described in Section 4.3.16, “Condition”.

If you define multiple triggers, then you need to specify whether only one or all triggers must be fulfilled in order for the action to be triggered/not be triggered.

5.2. Formcentric Analytics

Use the *Formcentric Analytics* action if you want to store and analyse the form data with the help of Formcentric Analytics.

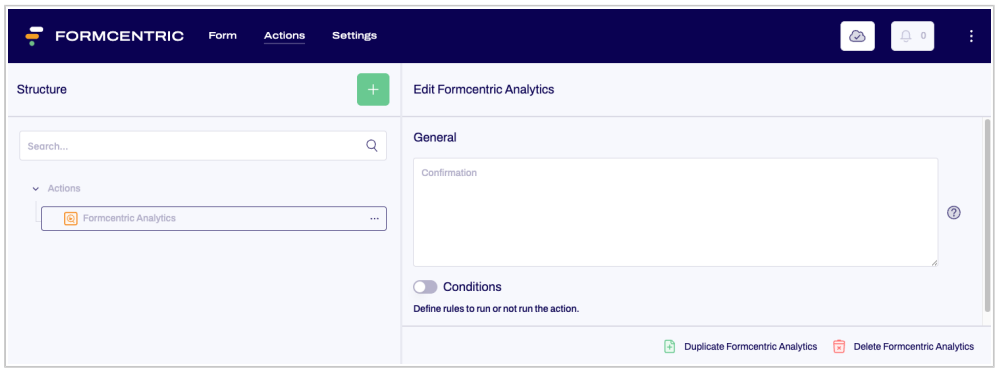


Figure 5.3. Formcentric Analytics action

Confirmation: Enter the confirmation message that is shown to the user on the web page after the form has been submitted. You can use Section 4.6, “Markdown” to format the text.

Conditions: Use this field to specify the user input for which the action should be executed/not be executed. To do this, proceed as described in Section 4.3.16, “Condition”.

If you define multiple triggers, then you need to specify whether only one or all triggers must be fulfilled in order for the action to be triggered/not be triggered.

You can test the action while you are still editing the form and then use Formcentric Analytics to view the entries submitted. However, please note that these entries will be removed once you have finished editing the form. Entries from previous versions of the form are not deleted, of course.

5.3. Forward to

Use the *Forward to* action if you want to forward the user to a different page after the form has been submitted. You can reference an external address or a CoreMedia document here.

Please note that actions are executed in the order in which they are added. The *Forward to* action must always be placed last.

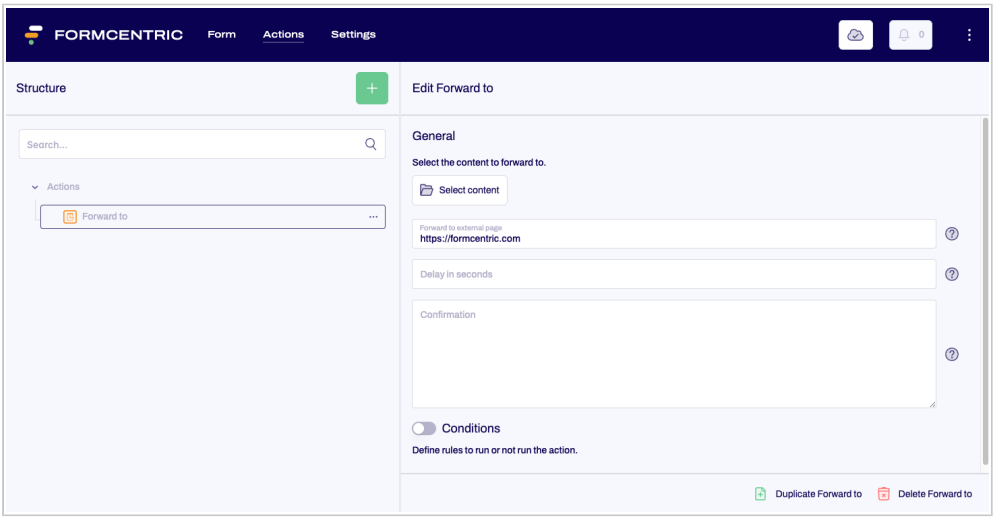


Figure 5.4. Forward to action

Select content: If you want to forward the user to a CoreMedia document after submitting the form, then you can specify the corresponding document here. Simply use drag-and-drop to move it to *Store here* from the Library.

Forward to external page: If you want to forward the user to an external website after submitting the form, then you can specify the URL for the corresponding website here. You can also use Formcentric variables (see Section 4.5, “Variables”).

Delay in seconds: Specify how long to wait in seconds before forwarding the user to the target address.

Confirmation: Enter the confirmation message that is shown to the user on the web page after the form has been submitted. Apart from the form data, the variables *_url* and *_delay* are also available, which can be used to display the target address or the delay time.

Conditions: Use this field to specify the user input for which the action should be executed/not be executed. To do this, proceed as described in Section 4.3.16, “Condition”.

If you define multiple triggers, then you need to specify whether only one or all triggers must be fulfilled in order for the action to be triggered/not be triggered.

5.4. PDF

The *PDF* action lets you fill a PDF document containing interactive or editable form fields with the form data from your web form. This can be used to let the user download a PDF file that is filled out with their own data.

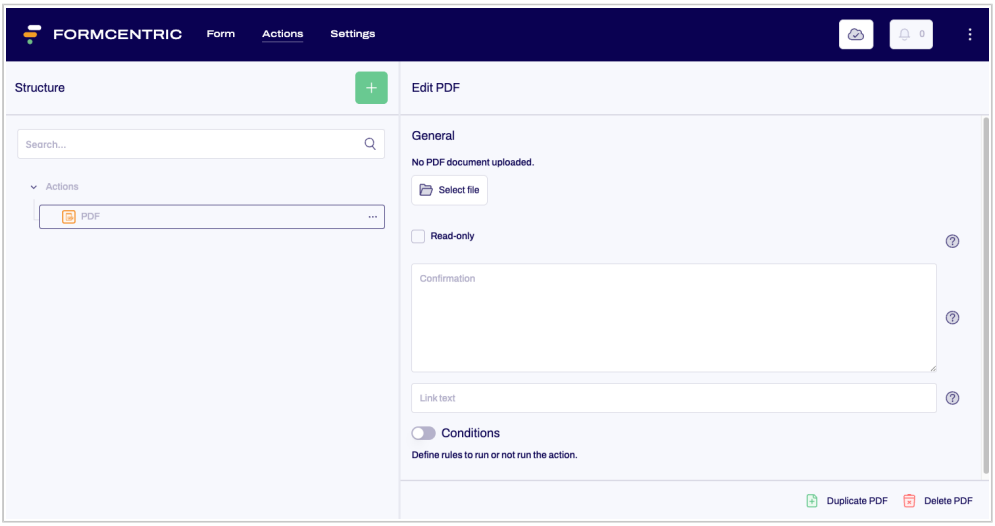


Figure 5.5. PDF action

Select file: Store your PDF template that has been prepared for filling out here. You can then map this PDF to your web form. Simply use drag-and-drop to move it here from the Library.

Assignment: Once you have stored a PDF document, you then need to map the form elements from the Formcentric web form to the form elements in the PDF document.

Read-only: Check this check box to specify that the data items mapped to the PDF form are read-only and cannot be edited.

Confirmation: Enter the confirmation message that is shown to the user on the web page after the form has been submitted.

Link text: Enter the text for the download link, which the user can click to download the generated PDF document. Leave this field empty if the PDF filename should be used as the link text.

Conditions: Use this field to specify the user input for which the action should be executed/not be executed. To do this, proceed as described in Section 4.3.16, “Condition”.

If you define multiple triggers, then you need to specify whether only one or all triggers must be fulfilled in order for the action to be triggered/not be triggered.

5.5. Webhook

You use the *Webhook* action to send form input directly to a specified URL or compatible web application as soon as a form has been completed.

This offers you a way to integrate third-party services such as Slack, Zapier or your own backend system.

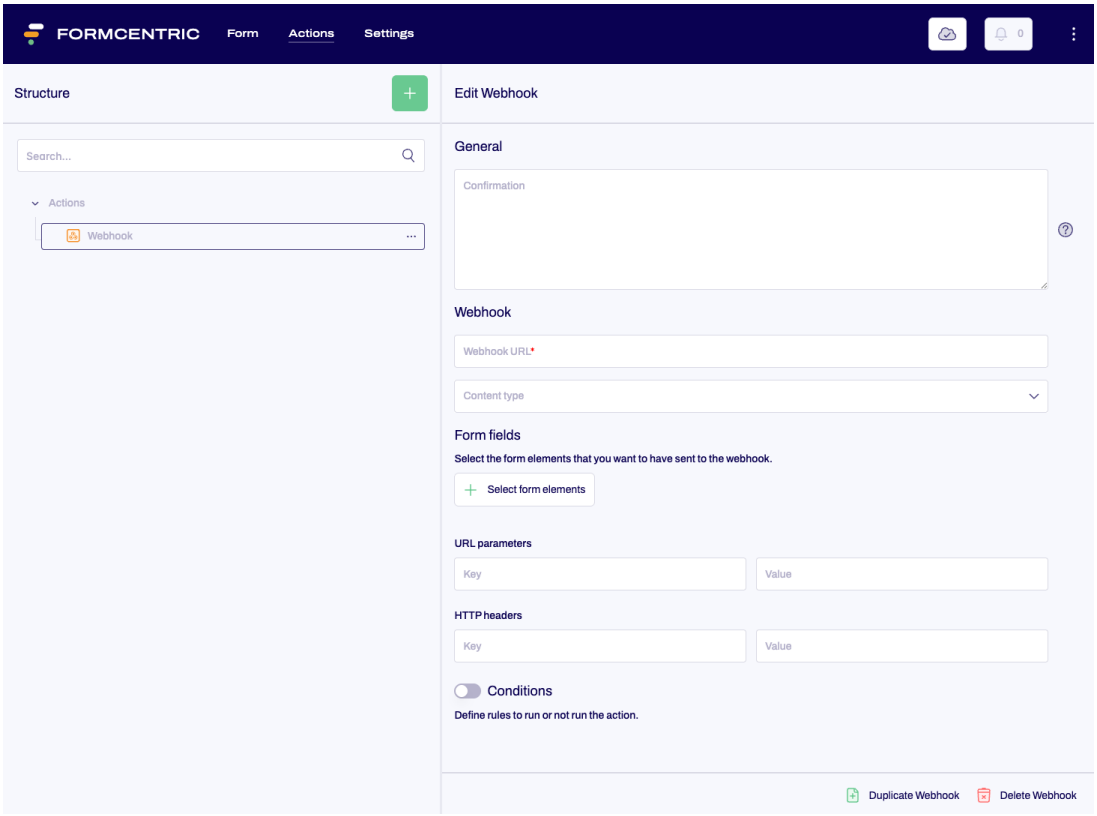


Figure 5.6. Webhook

Confirmation: Enter the confirmation message that is shown to the user on the web page after the form has been submitted.

Webhook URL: Enter the URL to which the request should be sent.

The Webhook URL utilises the following format:

```
https://www.mydomain.com/path
```

The use of secure HTTP (HTTPS) is not mandatory but is strongly recommended. For security reasons, a local address (such as *localhost, 127.0.0.1*, etc.) cannot be specified.

Content type: Select the format for the Webhook request. The following formats are supported:

Format	Description
application-json	Send the form data in JSON format in the body of the HTTP request.
application-x-www-form-urlencoded	Sends the form data as a URL-encoded data record separated by *&* characters in the body of the HTTP request.
multipart-form-data	Sends the form data as a multipart HTTP request. Use this content type if the form data to be sent contains file attachments.

Select form elements: Select the form fields whose data is to be sent to the Webhook endpoint.

URL parameters: Any additional parameters you want to append to the Webhook URL. When specifying parameter values, you can make use of form values and form variables by specifying the value as a placeholder with the format *${element-name}* or *${variable-name}* .

HTTP header: You can specify user-defined HTTP headers that are to be used when sending the data to the specified Webhook endpoint. When specifying a header value, you can again use form values and form variables in the same way as when specifying the URL parameters.

Conditions: Use this field to specify the user input for which the action should be executed/not be executed. To do this, proceed as described in Section 4.3.16, “Condition”.

If you define multiple triggers, then you need to specify whether only one or all triggers must be fulfilled in order for the action to be triggered/not be triggered.

6. Settings screen

On the *Settings* screen, you can give your form a name, and also specify the labels that will be used for the cancel and submit buttons. You can also activate saving for form input on this tab, as well as configuring the settings for the double opt-in function.

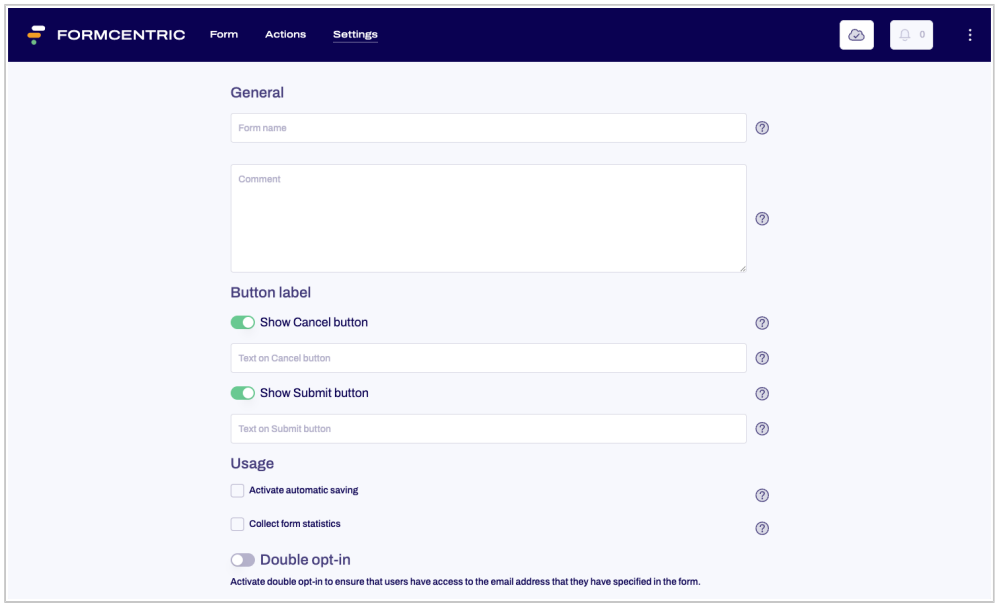


Figure 6.1. Settings screen

6.1. General

Form name: Enter a memorable name for your form that you can use to identify it later. This name is used when displaying the form within the Formcentric Analytics web interface.

Comment: You can also include a comment here. This comment is for internal use only and is not displayed on the form itself.

Text on Cancel button: Enter a piece of text here that will be displayed on the cancel button for your form.

Text on Submit button: Enter a piece of text here that will be displayed on the submit button for your form.

Activate automatic saving: If you check *Activate automatic saving*, then the user can stop filling out the form at any time and continue with the form later on, without losing any of the data already entered. Form input is saved until the user has filled out the form in full and submitted it. This option is especially useful for multi-page forms.

Collect form statistics: If you activate *Collect form statistics* and have a Formcentric licence that includes Formcentric Analytics, then statistical metrics for this form will be collected and processed for reports. You can view these reports in the Formcentric Analytics Reporting application and drill down further into reports on form user behaviour.

When you activate *Collect form statistics*, then statistics will start to be collected and sent to Formcentric Analytics even while you are editing the form. However, please note that this data will be deleted once you have finished editing the form. Data from previous versions and data collected after form completion is not deleted, of course.

6.2. Double opt-in

In the *Double opt-in* section, you can enable and configure double opt-in for users. This function lets you verify that the user has access to the email address that was entered into the form. To enable this verification, an additional step is added to the form that sends the user an email with a confirmation link. This link must be accessed in order to complete form entry successfully. Please note: the actions you have configured are not executed until the user has accessed the link that was sent in the email.

Activate double opt-in by moving the slider to the right: this activates email confirmation for this form. Then complete the fields as specified. All of the input that you enter here will be validated. Please note: no validation takes place if the double opt-in is deactivated. However, the data already entered remains stored and can be used if the feature is activated at a later date.

Please note that the double opt-in feature can only be used in conjunction with Formcentric Analytics.

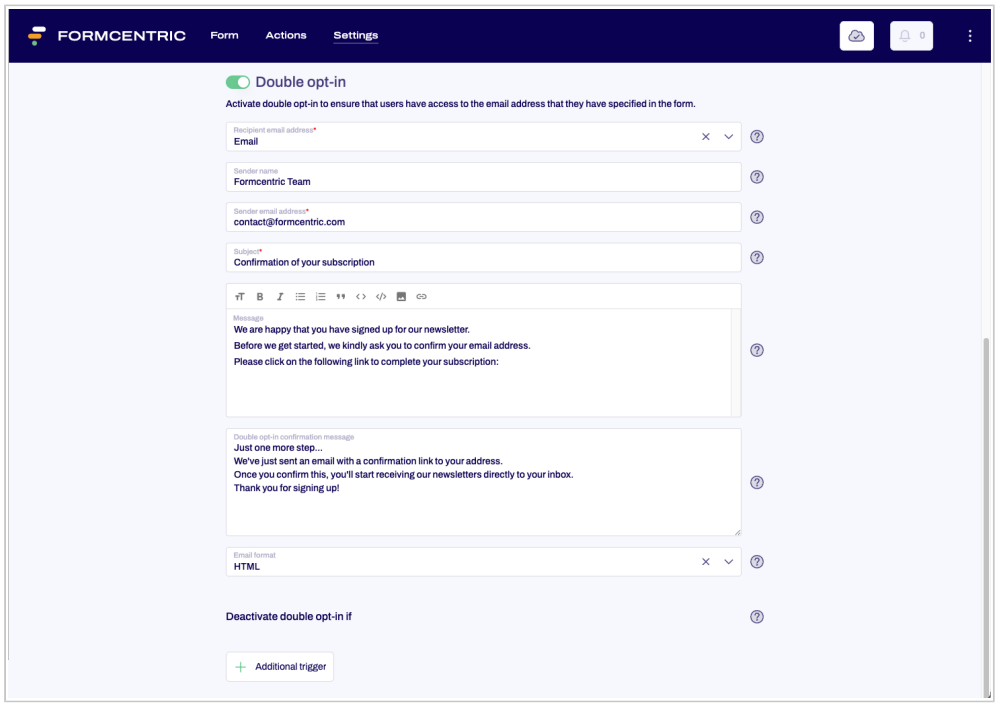


Figure 6.2. Double opt-in

Recipient email address: Select the form element into which the user must enter their email address. Please note that the drop-down list only includes the predefined “Email address” form element, as well as input fields for which email validation has been activated. Input fields are only accepted if these fields are defined as required fields.

Sender name: Enter the name to be displayed as the sender.

Sender email address: Enter the email address to be used as the sender for the confirmation mail. Note that the email address must be valid, otherwise an error will be generated.

Subject: Enter the subject here that is displayed to the user when they receive the mail.

Message: Enter the text of the email here. Include the *${url}* variable anywhere in your email text to add the confirmation link that the user needs to click. If you do not use the variable, the link is appended automatically to the end of the email message you have entered. Regardless of the *Email format* that you select, you can always format your message using FreeMarker markup.

Double opt-in confirmation message: When you activate double opt-in, the user will be shown a new intermediate page when the form has been filled out in full. The user must now respond to the email that they have been sent. Enter a piece of text here that will be displayed on this intermediate page. You can use Section 4.6, “Mark-down” to format the text.

Email format: Specify whether the email is sent to the user in HTML format or as a plain text message.

Deactivate double opt-in if: Use this field to specify any user input for which the double opt-in feature will not be used.

The condition can be created as described in Section 4.3.16, “Condition”.

If you define multiple triggers, then you need to specify whether only one or all triggers must be fulfilled in order to deactivate double opt-in.

A. Reserved identifiers.

For technical reasons, the technical name given to a form element must not match any of the following reserved identifiers:

abstract, action, arguments, array, await, boolean, break, byte, case, catch, char, constructor, currentpage, currentpagenode, date, else, enum, eval, export, extends, false, final, finally, float, for, form, formdata, formvariables, function, goto, hasown-property, if, implements, import, in, infinity, instanceof, int, interface, isfinite, isnan, isprototypeof, let, long, math, nan, native, new, nil, null, number, object, package, pagecount, pageelements, private, propertyisenumerable, protected, prototype, resolvedcaptchas, return, selectedelements, self, short, static, string, super, switch, synchronized, target, this, throw, throws, tolocalestring, tosource, tostring, transient, true, try, typeof, undef, undefined, unwatch, valueof, var, view, void, volatile, watch, while, with, yield