FORMCENTRIC

SiteArchitect User Manual

Version 7.2.9

Formcentric for FirstSpirit: SiteArchitect User Manual

Copyright © 2026 Formcentric GmbH
Schaartor 1, 20459 Hamburg
Germany

The contents of this document – whether in whole or in part – may not be reproduced, conveyed, disseminated or stored in any form whatsoever without obtaining prior written permission from Formcentric GmbH.

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.

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 1
1.1. Overview of functions 1
1.2. Terminology 1
2. Editing interface 3
2.1. General 4
2.1.1. Checking the forms created 4
2.1.2. Predefined form elements 5
2.1.3. Drag-and-drop 5
2.1.4. Preview 5
2.2. Form elements 6
2.2.1. Form 7
2.2.2. Input field 14
2.2.3. Single choice 25
2.2.4. Multiple choice 28
2.2.5. Drop-down list 30
2.2.6. Short text 33
2.2.7. Long text 35
2.2.8. Email address 38
2.2.9. Number 40
2.2.10. Phone number 43
2.2.11. Date 45
2.2.12. Password 48
2.2.13. Upload file 50
2.2.14. Hidden field 53
2.2.15. Page break 54
2.2.16. Page break condition 55
2.2.17. Calculated value 56
2.2.18. Captcha 58
2.2.19. Condition 59
2.2.20. Paragraph 62
2.2.21. Button 63
2.2.22. Fieldset 65
2.2.23. Image 66
2.2.24. Layout 67
2.2.25. Summary 67
2.3. Variables 69
2.4. Actions 70
2.4.1. Send as Email 71
2.4.2. PDF 77
2.4.3. Data source 80
2.4.4. Media Management 81
2.4.5. Formcentric Analytics 83
2.4.6. Forward to 83
2.4.7. Webhook 84
2.4.8. Sequence 86

2.5. Data sources 87
2.5.1. Country names 87
2.5.2. Weekdays 88
2.5.3. Months 88
2.5.4. Year numbers 89
2.6. Markdown 89
A. Reserved identifiers 90

1. Introduction

“Formcentric for FirstSpirit™” extends the FirstSpirit Digital Experience Platform to include a powerful form editor with which you can create and edit any type of web-based form. The web app components also included with the product are used to ensure the dynamic display and processing of the forms that you create.

This manual will show you how to use Formcentric to create and edit forms in FirstSpirit SiteArchitect.

1.1. Overview of functions

Form elements: Formcentric supports all of 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 are 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) in the Editor, you can decide how the form data will be processed after submission. Out of the box, the software includes the actions *Send as email*, *Webhook*, *Forward to*, *Formcentric Analytics*, *PDF*, *Data source* and *Media Management* (see Section 2.4, “Actions”). 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 elements	All of the elements used when constructing a form (input fields, drop-down lists, check boxes, captchas, etc.).
Form data	The data entered into the form by the user.
Form author	The person that creates and edit forms.
User	The person that fills out a form.
Editor	The form editor in FirstSpirit SiteArchitect.
Frontend	The web page created with the FirstSpirit Digital Experience Platform.

2. Editing interface

The editing interface used for the Formcentric Editor is split into two main areas. You use the left-hand area to work on your overall form with an easy-to-use tree structure. You use the right-hand area to edit the specific properties of the element that is currently selected.

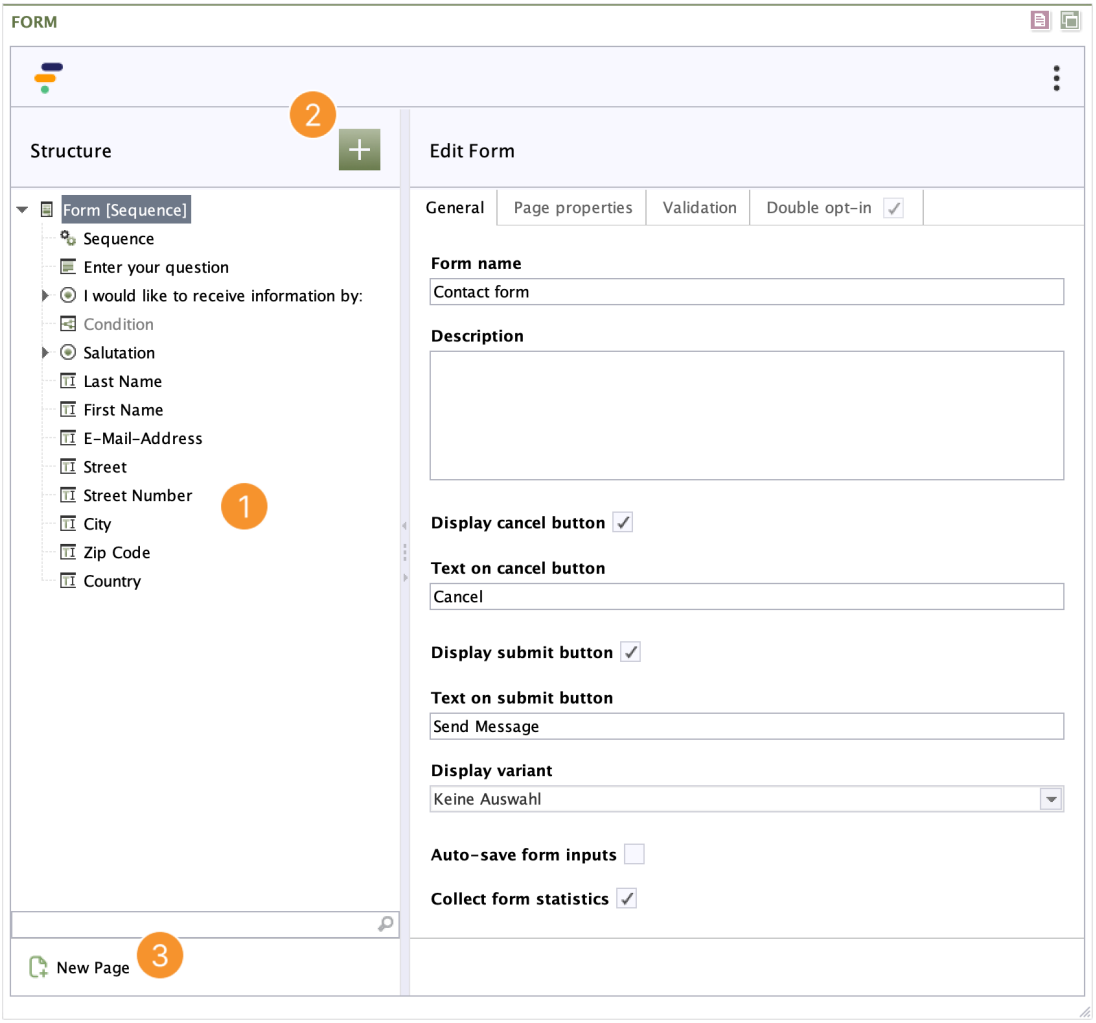


Figure 2.1. Editing interface

Click the green plus sign [2] in the *structure* area to display a list of all of 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]. Click *New page* [3] at the bottom of the screen to add another page to your form. This is also shown in the form tree as a *page break*.

The form tree represents 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 access them, move the cursor over the form tree. Then right-click to open the context menu with the actions.

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

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 start: Moves the form element to the top start of the form

Move to end: Moves the form element to the end of the form

Move up: Moves the form element up one position towards the start of the form

Move down: Moves the form element down one position towards the end of the form

Show additional information: Activates the display of useful information about the form elements in the form tree

2.1. General

2.1.1. 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 exclamation mark in the form tree. You can find a detailed error message in the properties area for the affected form element.

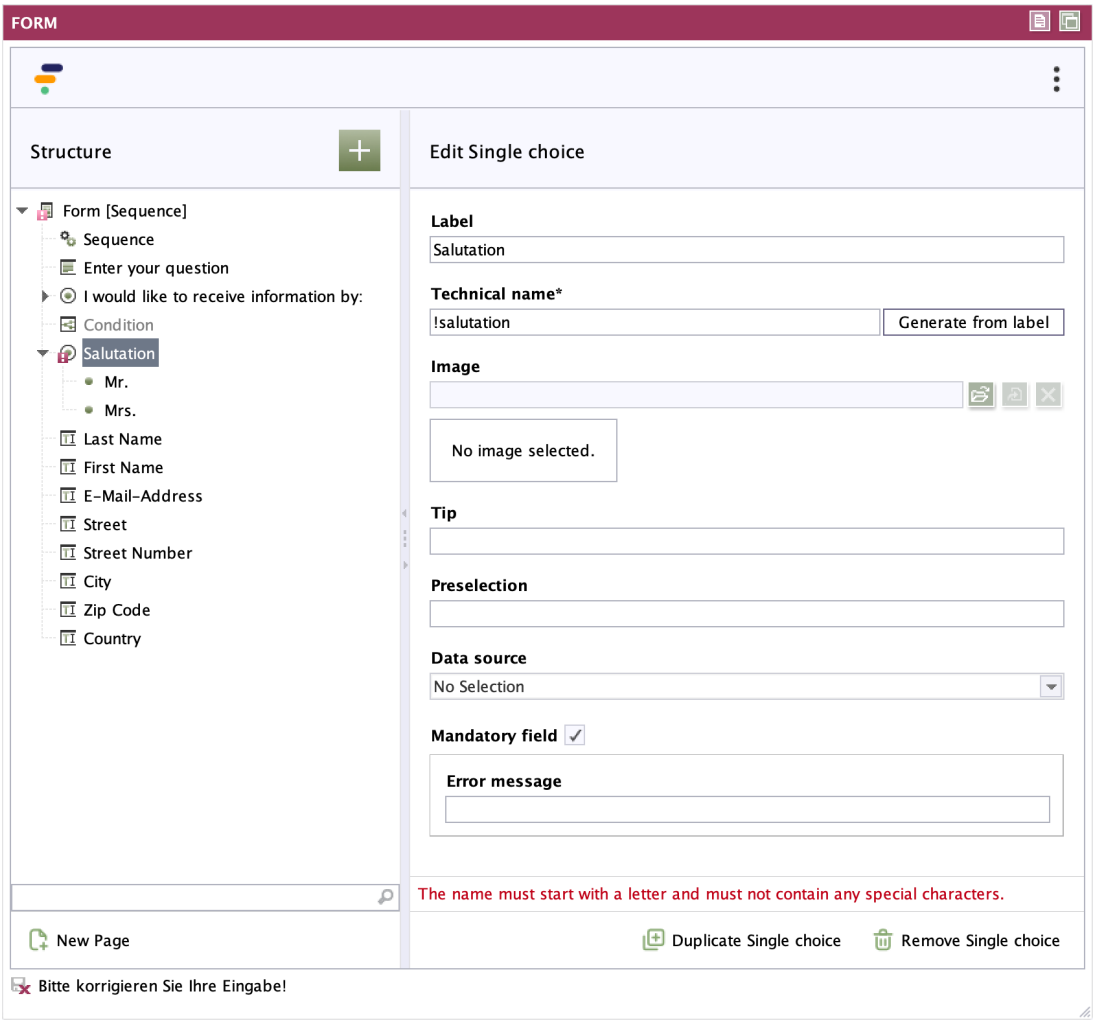


Figure 2.2. An error marked on the screen

Please note: You cannot save a form if it contains errors.

2.1.2. 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.

2.1.3. Drag-and-drop

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

2.1.4. Preview

The preview area in SiteArchitect shows you the current version of your form, which you can try out at any time.

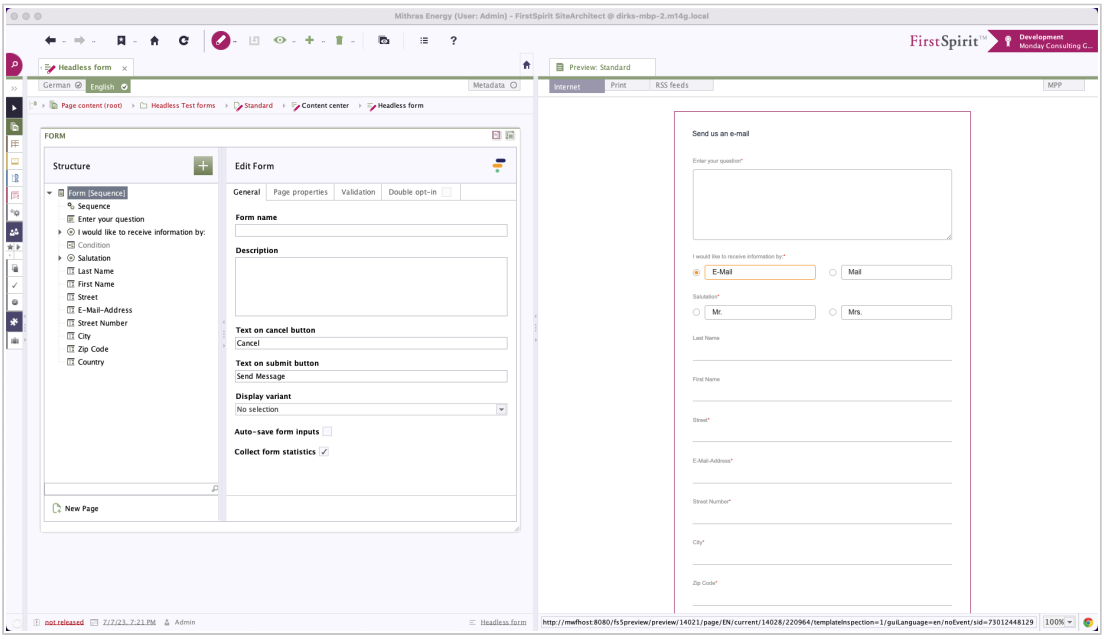


Figure 2.3. SiteArchitect preview

2.2. Form elements

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

In the *Structure* area, click the green plus sign. A dialog opens, listing all of the available form elements. Click a form element to add it to your form. You edit the properties for form elements on the right-hand side.

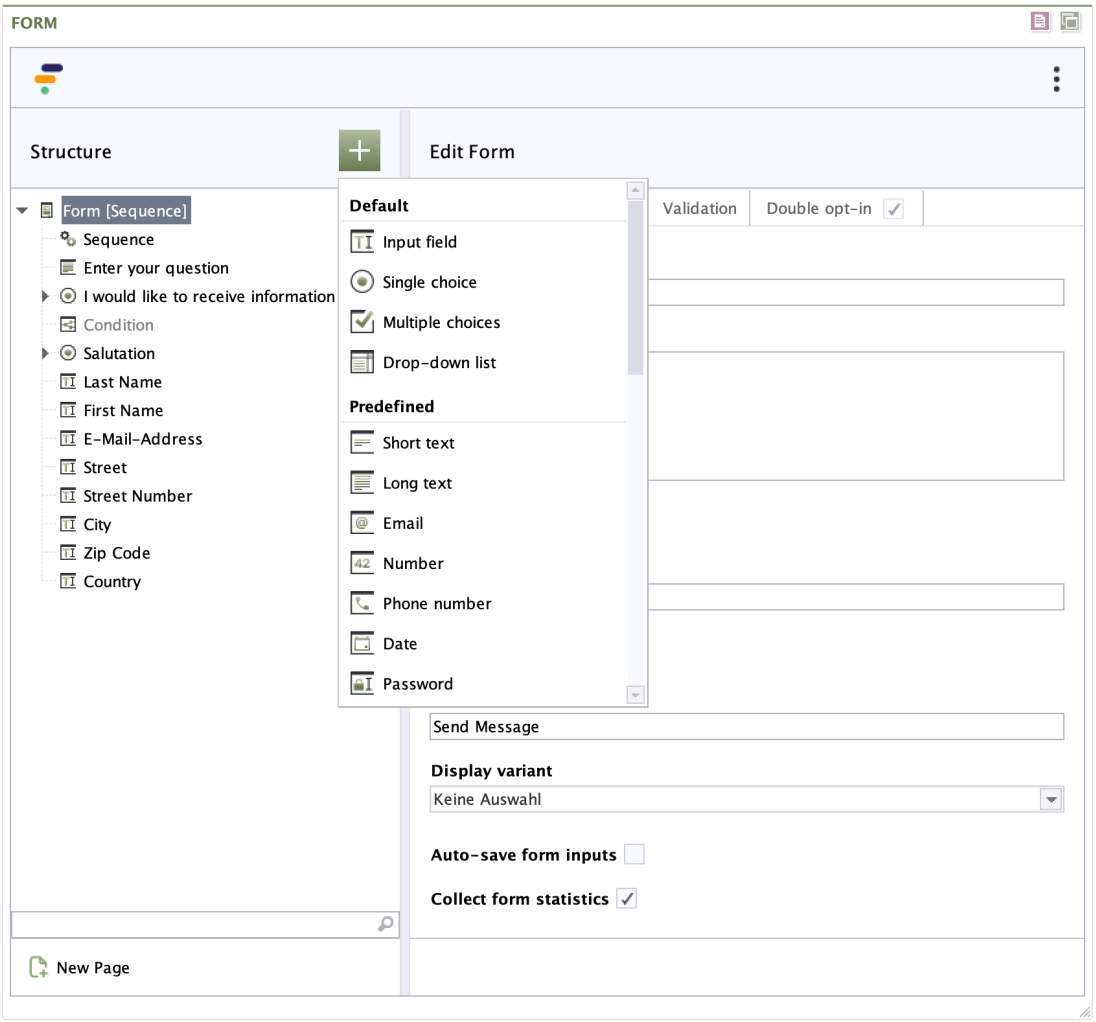


Figure 2.4. Add form element

The following section describes all of the form elements in detail.

2.2.1. Form

The *Form* element represents the form itself. This is always shown right at the top in the form tree. The system arranges all of the other elements underneath this element.

For multi-page forms, the form element is the first form page. You can identify the other pages by the *page break* element (see Section 2.2.15, “Page break”).

The form elements are output in the order in which they are displayed in the form tree.

The detail view of the form element is split over the tabs *General, Page properties* and *Double opt-in*. The various configuration options are explained in the following sections.

General tab

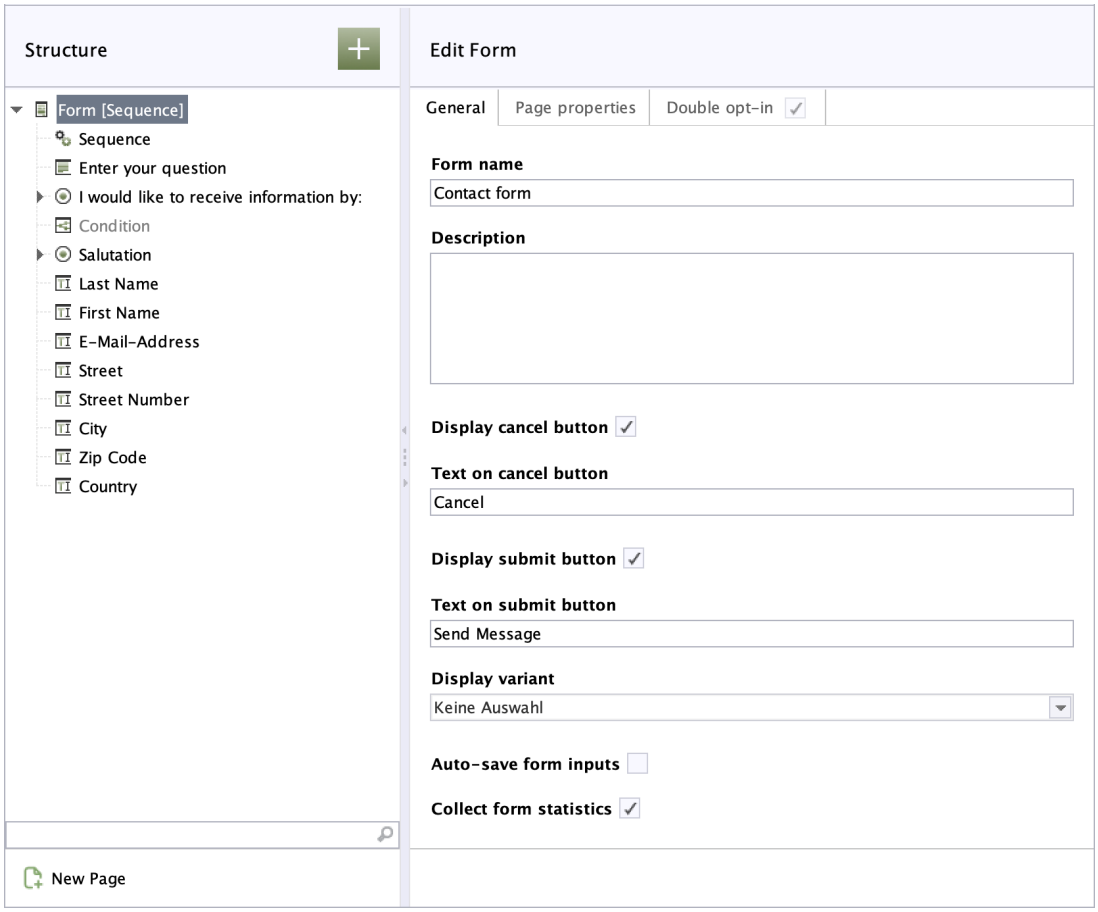


Figure 2.5. Form element – "General" tab

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.

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

Save form input automatically: If you activate the *Save form input automatically* function, this lets the user interrupt form completion and then continue filling out the form later on without losing any of the data that they have already entered. All of the

form data is stored until the form has been completed and submitted by the user. This function is especially useful for large, multi-page forms.

Collect form statistics: If you activate the *Collect form statistics* function, statistical data is then collected about this form. You can view this data in the Formcentric Analytics Reporting application and investigate user behaviour. Please note: this function requires a Formcentric licence that includes Formcentric Analytics.

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.

Page properties tab

You use the *Page properties* tab to specify the properties for the first page of your form. The properties for other pages in your form can then be specified on the corresponding *Page Break* element (see Section 2.2.15, “Page break”).

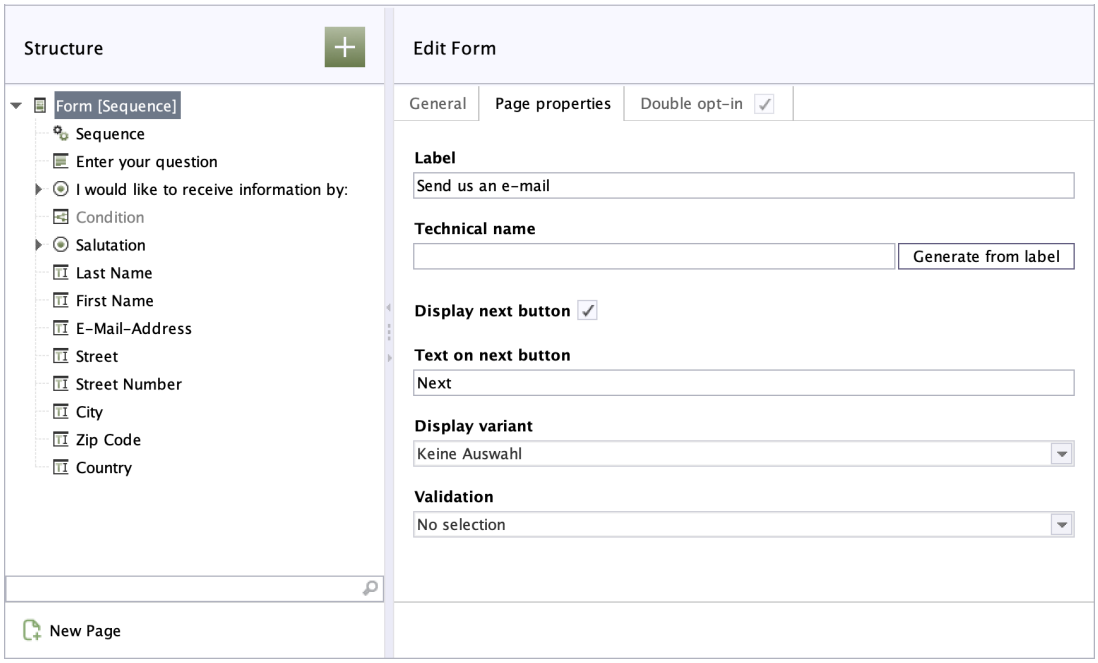


Figure 2.6. Form element – “Page properties” tab

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

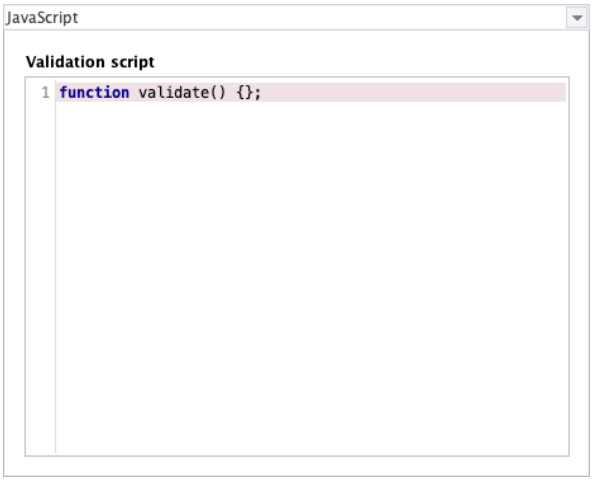
Technical name: Enter the technical name of the first page in the form here.

Text on next button: This text is displayed on the “Next” button that is shown on the first page of a multi-page form. This text is not used for forms that have only one page.

Display variant: Use this to pick an alternative presentation style for the first form page.

Validation: Optionally, you can choose between various validation options on each form page. 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. Initially, the text box contains the empty JavaScript function

```
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 of 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 enter a postcode.";
  } else {
    return "";
  }
}
```

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.

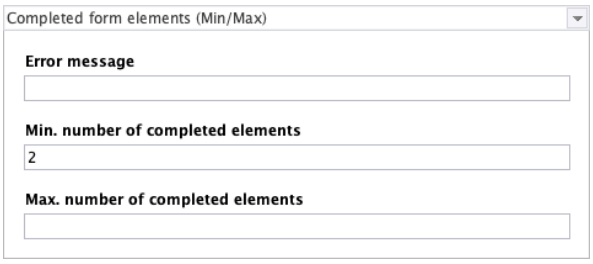
```
function validate () {
  var age = parseAge("dd.MM.yyyy", birthday);

  if (isSelected(newsletter, "ja") && age < 16) {
    return "You must be at least 16 years old to subscribe
         to the newsletter.";
```

```
  } else {
    return "";
  }
}
```

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.

Double opt-in tab

On the *Double opt-in* tab, you can activate and configure the double opt-in feature (also known as “email confirmation”) for users. This functionality 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 in which the user is sent a confirmation link. This link must be accessed in order to complete form entry successfully. This also means that the actions you have configured are not executed until the user has accessed the link that was sent in the email.

Check the double opt-in checkbox on the tab page to activate email confirmation for this form. Complete the fields as listed here. All input entered here will be validated. Please note: this validation will not take place if you then deactivate double opt-in. All input that you have already made is saved and can be used at a later point in time.

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

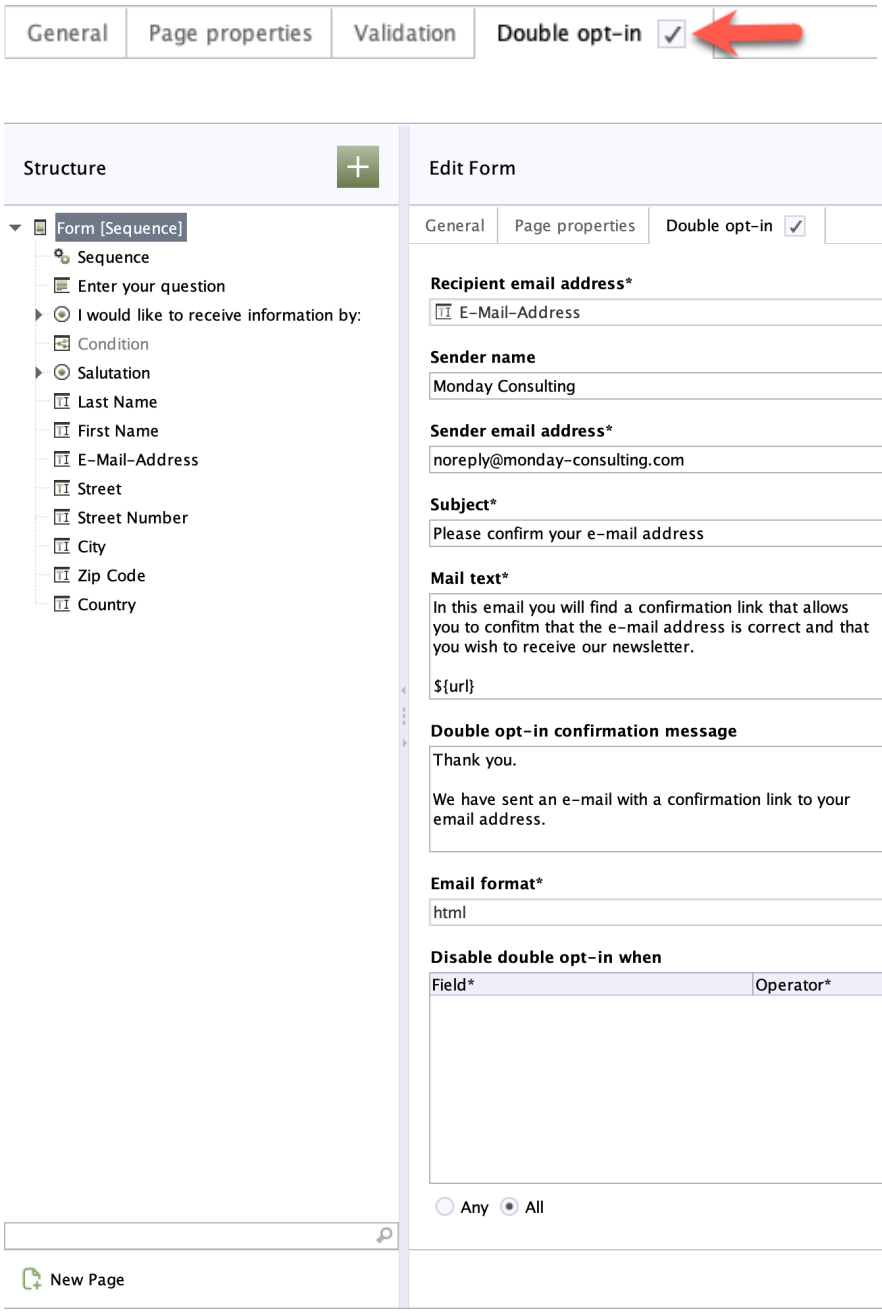


Figure 2.7. Form element – "Double opt-in" tab

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.

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 2.6, “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 2.2.19, “Condition”.

2.2.2. 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.

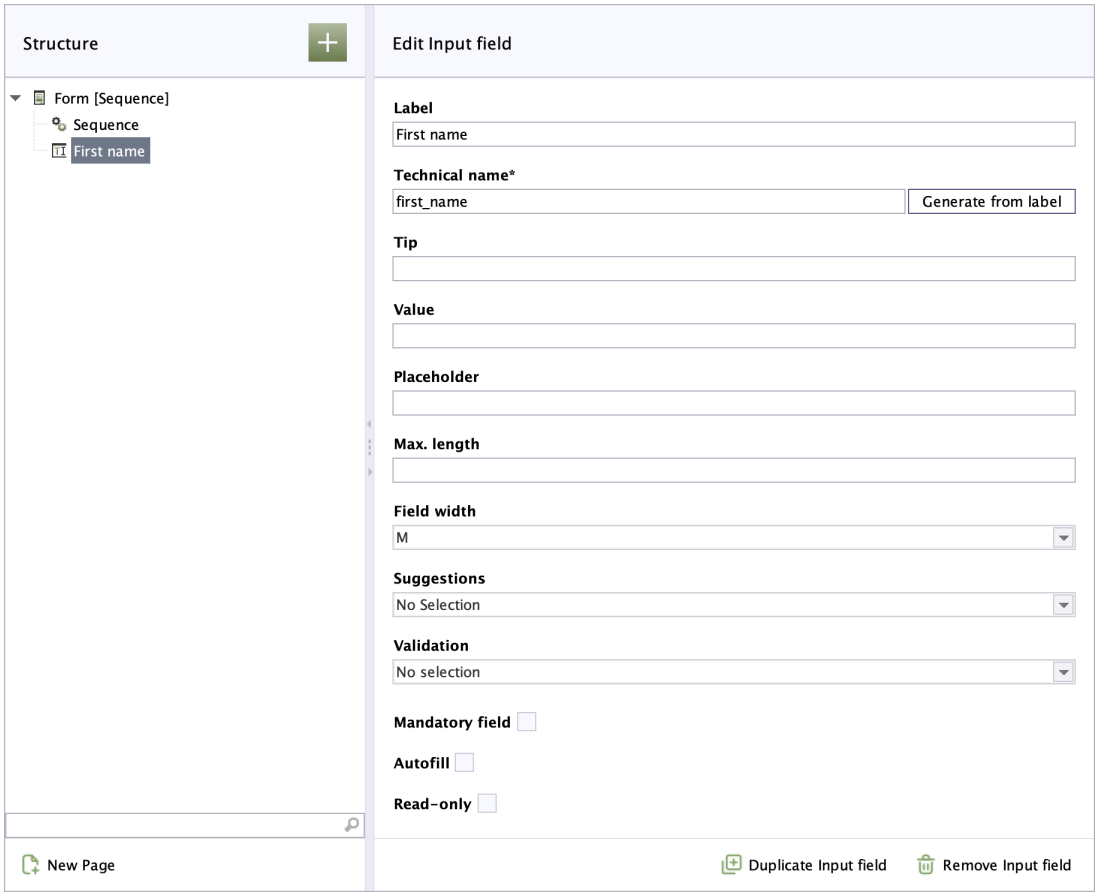


Figure 2.8. Input field

Label: Enter the text of the label that is displayed next to the form field. By default, the label is placed above the input 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 default technical name is formed from the type name of 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 *Generate from label*.

Please 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. By default, the note is placed underneath the form field.

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 2.3, “Variables”).

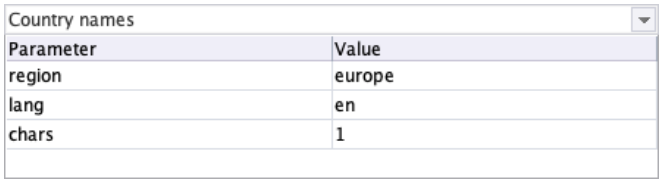
Placeholder: In this field, you can enter a piece of placeholder text that is displayed in the input field. This placeholder text disappears as soon as the user starts typing into the form field.

Max. length: You can use this field to specify a maximum number of characters that the user may 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 or more 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.

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 data sources, including country names, months, year numbers and week-days.



The table gives you the option of entering additional parameters to pass to the data source (see also Section 2.5, “Data sources”).

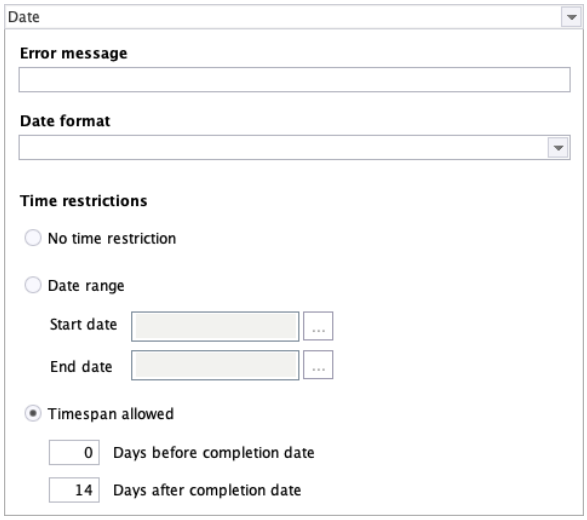
Validation: Pick a validator from this picklist to specify the format to be used for the text that the user enters into the input field. For example, the *email* validator checks to confirm that a valid email address has been entered into the field.

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

Email	The *email* 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

The *date* 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.

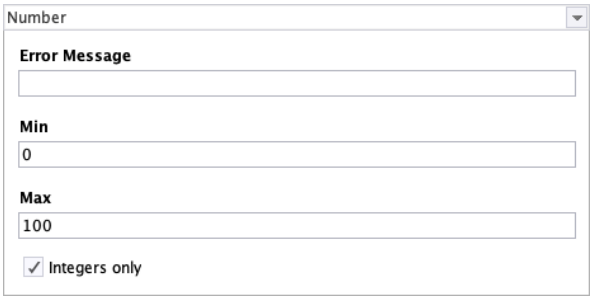
The following placeholders can be used in the format string:

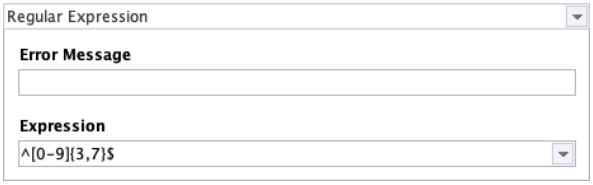
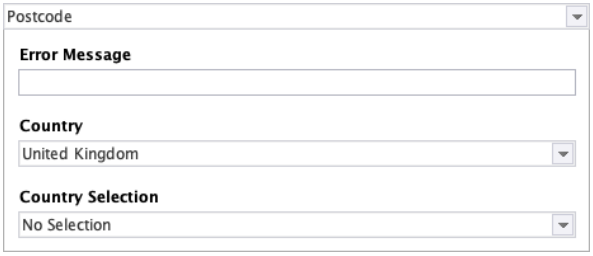
y	Year
MM	Month of the year with leading zero
d	Day of the month
H	Hour of the day (0–23)
m	Minute of the hour
s	Second of the minute

A list of suggestions already contains some commonly-used data formats. You can either use these directly or modify them to suit your personal requirements.

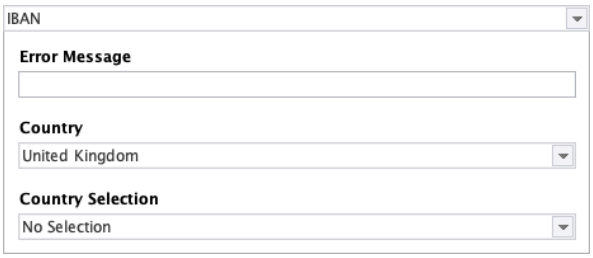
By default, the format *dd/MM/yyyy* is applied.

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.

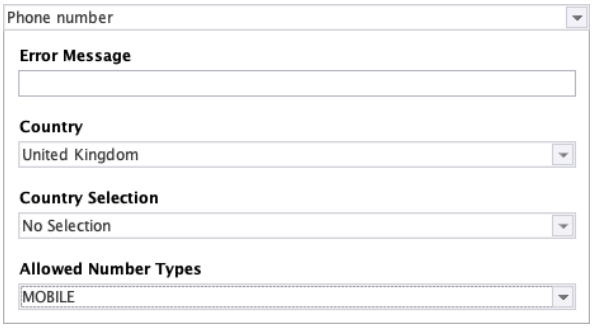
	No time restrictions: If you select *No time restrictions*, then the user will be able to enter any date. **Date range:** Specify a date range here if the user needs to enter a date that lies within a specified period of time. **Date from:** Select the start date for this period of time. **Date to:** 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).
Number	The *number* validator checks to see if the value input is a 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	The *regular expression* 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. **Expression:** In this field, you can enter or select a regular expression that defines the input format you require. A list of suggestions 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	The *postcode* validator checks to see if the value 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 2.2.5, “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	The *number of characters* 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	The *IBAN* 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 2.2.5, “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.
European Union VAT identification number	The *European Union VAT ID number* validator checks to confirm that user input matches the structure of an EU VAT ID 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 2.2.5, “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	The *BIC* 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	The *equal value* validator compares two input fields and checks to confirm that the input in these fields 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	The *phone number* 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.

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 2.2.5, “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.

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.
Credit card	The *Credit card* 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. **Error message:** Enter the text for the error message to be displayed instead of the default error message. This message will be shown to the user if the form field is not filled out correctly. **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. The following card types are supported: Visa Card (VISA), Master-Card (MASTERCARD), American Express (AMEX), Diners Club (DINERS), JCB Card (JCB), and Discover Card (DISCOVER). 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.

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 the form field, which also helps the user to fill out 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*”.

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.

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 2.3, “Variables”).

Duplicating an input field: Click *Duplicate input field* to add a copy of this form field to your form.

Deleting an input field: Click *Delete input field* to delete this form field.

2.2.3. 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.

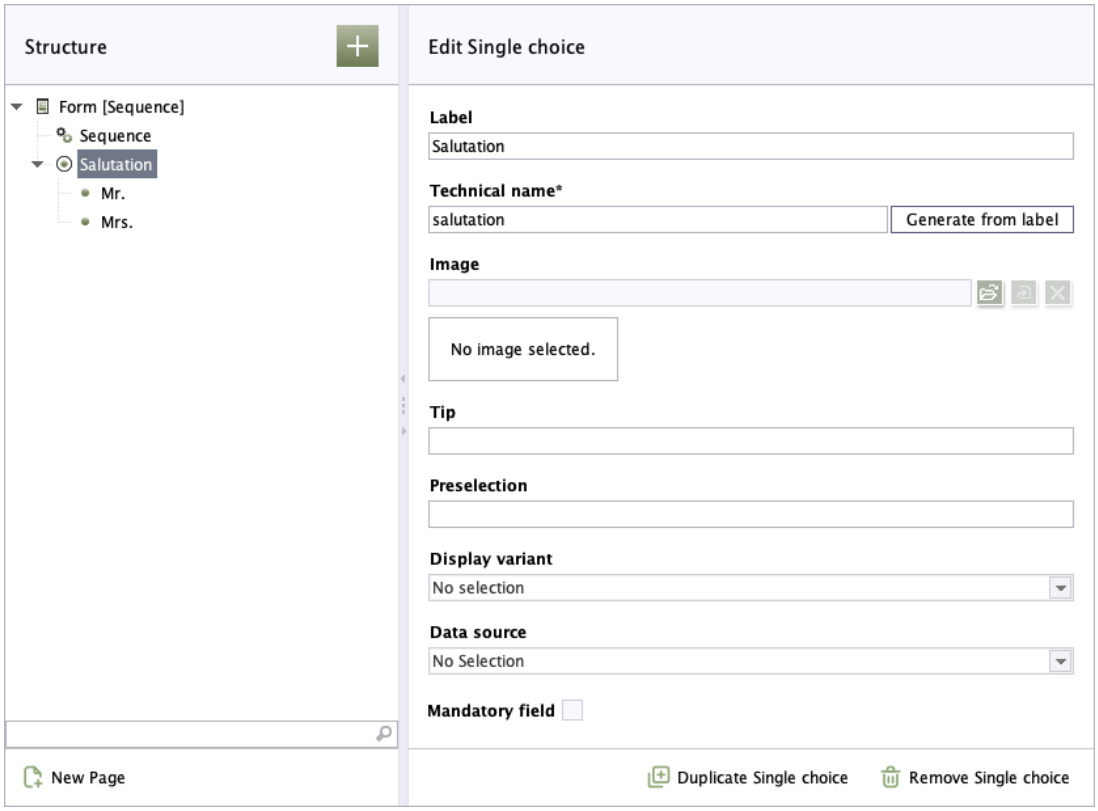


Figure 2.9. Single choice

Label: Enter the text of the label that is displayed next to the form field. By default, the label is placed above the single choice 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 default technical name is formed from the type name of 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 *Generate from label*.

Image: You can add an image here from the FirstSpirit MediaStore that will be shown together with the select field.

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. By default, the note is placed directly underneath the label.

Preselection: If you have linked your single choice field to a data source (see below), then you can specify that one option from the data source is preselected when the form is first accessed. Enter the *value* for the corresponding option here.

Let"s assume that you have added the *Country names* data source and you would like "Germany" to be preselected. In the *Country names* data source, the values (not the labels) for the options match the country codes according to ISO 3166. This means that you need to enter *DE* here for Germany.

Field width: Specify how wide the single choice field should be.

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

Data source: Select a data source that will be used to fill the single choice field with external data at runtime.

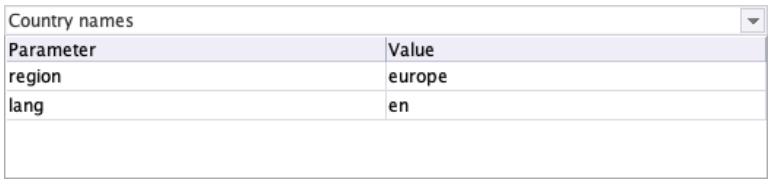


Figure 2.10. Data source

The table gives you the option of entering additional parameters to pass to the data source (see also Section 2.5, "Data sources").

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.

Adding options: Once you have added a single choice field to your form, you then need to specify the individual options that are displayed for this form element. To do this, click the green plus sign in the *Structure* area to add options.

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, then this option will be ignored in later processing steps, even if selected by the user. This function can be used if you want to add a "Please select" option, for example.

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

Preselected: If you check this check box, then the option will be preselected.

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

Deleting a single choice field: Click *Delete single choice field* to delete this form field.

2.2.4. Multiple choice

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

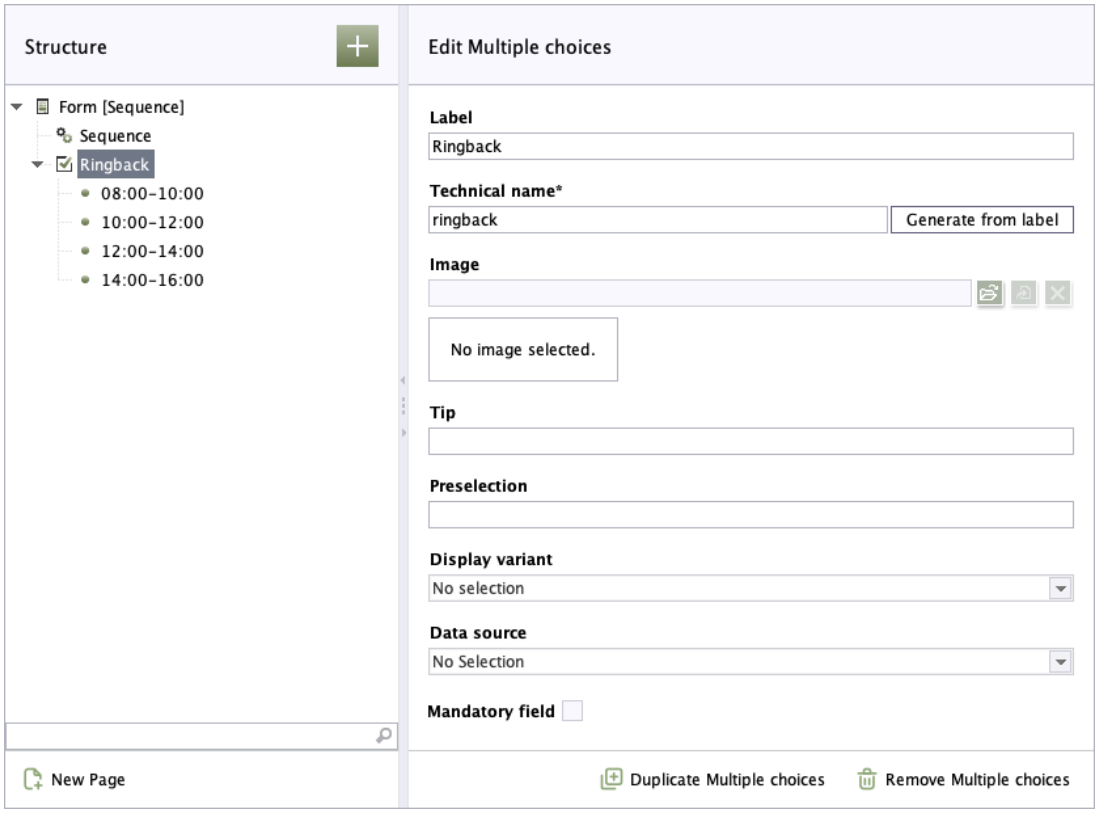


Figure 2.11. Multiple choice

Label: Enter the text of the label that is displayed next to the form field. By default, the label is placed above the multiple choice 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 default technical name is formed from the type name of 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 *Generate from label*.

Image: You can add an image here from the FirstSpirit MediaStore that will be shown together with the select field.

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. By default, the note is placed directly underneath the label.

Preselection: If you have linked your multiple choice field to a data source (see below), then you can specify that one option from the data source is preselected when the form is first accessed. Enter the *value* for the corresponding option here.

Let"s assume that you have added the *Country names* data source and you would like "Germany" to be preselected. In the *Country names* data source, the values for the options match the country codes according to ISO 3166. This means that you need to enter *DE* here for Germany.

Field width: Specify how wide the multiple choice field should be.

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

Data source: Select a data source that will be used to fill the multiple choice field with external data at runtime.

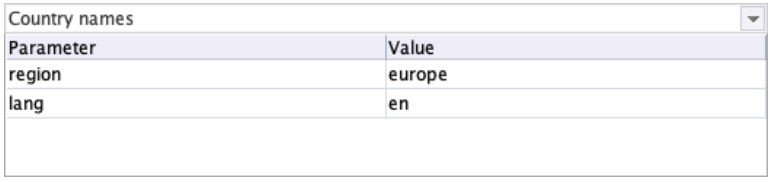


Figure 2.12. Data source

The table gives you the option of entering additional parameters to pass to the data source (see also Section 2.5, "Data sources").

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.

Adding options: Once you have added a multiple choice field to your form, you then need to specify the individual options that are displayed for this form element. To do this, click the green plus sign in the *Structure* area to add options.

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, then this option will be ignored in later processing steps, even if selected by the user. This function can be used if you want to add a “Please select” option, for example.

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

Preselected: If you check this check box, then the option will be preselected.

Duplicating a multiple choice field: Click *Duplicate multiple choice* to add a copy of this form field to your form.

Deleting a multiple choice field: Click *Delete Multiple choice* to delete this form field.

2.2.5. 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.

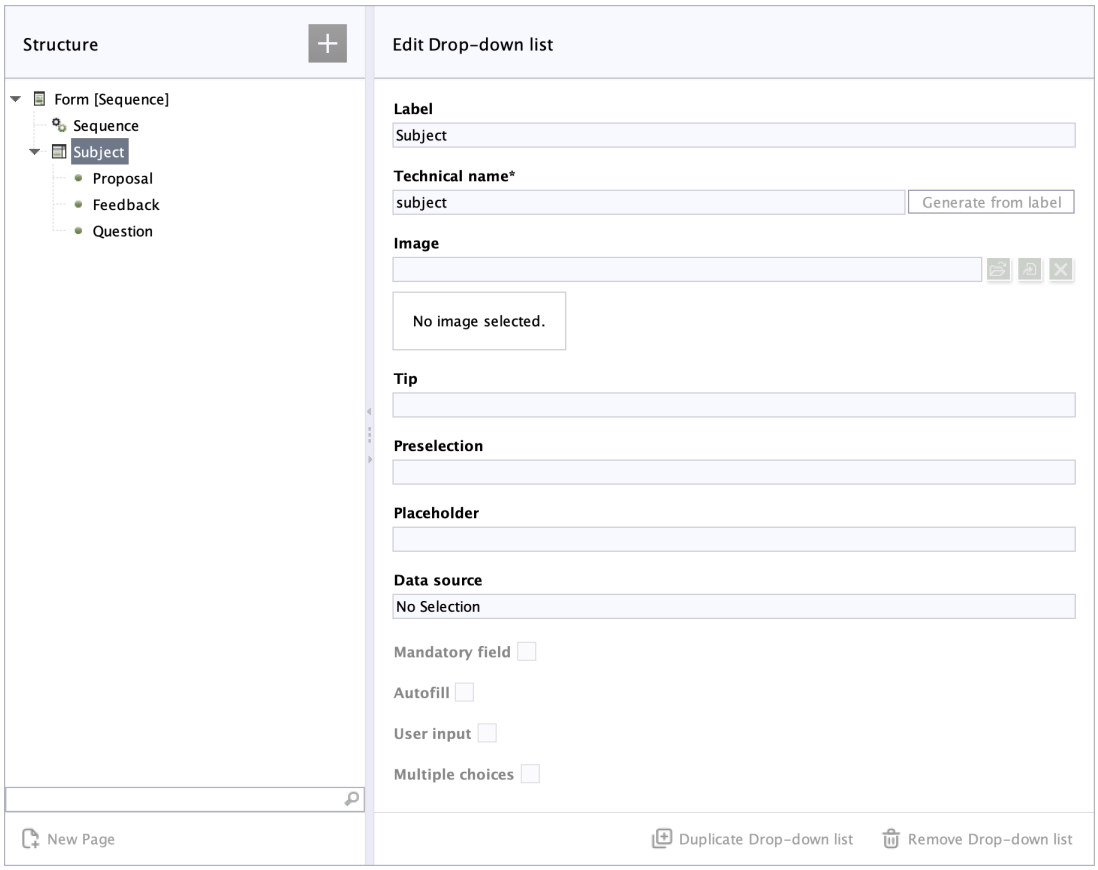


Figure 2.13. Drop-down list

Label: Enter the text of the label that is displayed next to the form field. By default, the label is placed above the drop-down list.

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 default technical name is formed from the type name of 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 *Generate from label*.

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

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.

By default, the note is placed underneath the form field.

Preselection: If you have linked your drop-down list to a data source (see below), then you can specify that one option from the data source is preselected when the form is first accessed. Enter the *value* for the corresponding option here.

Let"s assume that you have added the *Country names* data source and you would like "Germany" to be preselected. In the *Country names* data source, the values for the options match the country codes according to ISO 3166. This means that you need to enter *DE* here for Germany.

Field width: Specify how wide the drop-down list should be.

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

Data source: Select a data source that will be used to fill the drop-down list with external data at runtime.

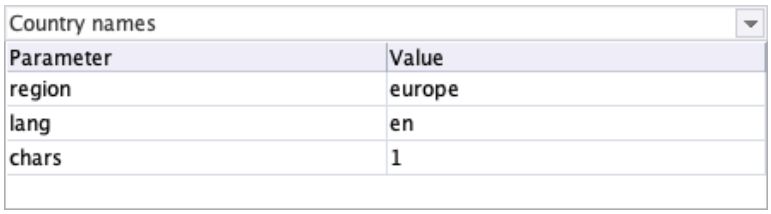


Figure 2.14. Data source

The table gives you the option of entering additional parameters to pass to the data source (see also Section 2.5, "Data sources").

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 the field, which also helps the user to fill out 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*".

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.

User input: Check *User input* to let users enter their own option(s).

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

Adding options: Once you have added a drop-down list to your form, you then need to specify the individual options that are displayed for this form element. To do this, click the green plus sign in the *Structure* area to add options.

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, then this option will be ignored in later processing steps, even if selected by the user. This function can be used if you want to add a "Please select" option, for example.

Image: You can add an image here from the FirstSpirit MediaStore that will be shown together with the select field.

Preselected : If you check this check box, then the option will be preselected.

Duplicating a drop-down list: Click *Duplicate drop-down list* to add a copy of this form field to your form.

Deleting a drop-down list: Click *Delete drop-down list* to delete this form field.

2.2.6. Short text

You use the predefined *short text* form field 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.

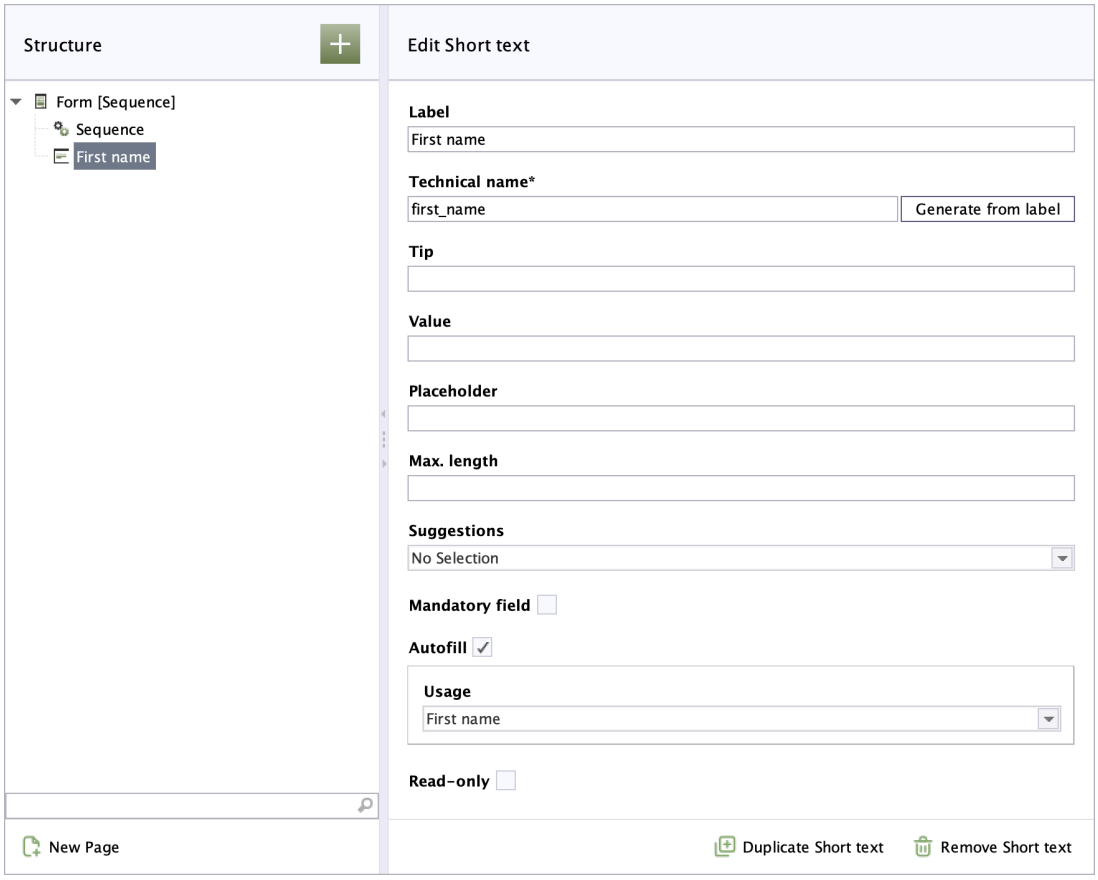


Figure 2.15. Short text

Label: Enter the text of the label that is displayed next to the form field. By default, the label is placed above 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 default technical name is formed from the type name of 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 *Generate from label*.

Please 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.

By default, the note is displayed directly underneath the form 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 make use of a variable here. This could insert the date automatically into the form field, for example (see Section 2.3, “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: You can use this field to specify a maximum number of characters that the user may enter into the form field.

Field width: Specify how wide the form element should be.

Display variant: Select one or more of the available display variants here, so as to specify how the form element is displayed in the form.

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 a default data source containing country names.

Country names

Parameter	Value
region	europe
lang	en
chars	1

The table gives you the option of entering additional parameters to pass to the data source (see also Section 2.5, “Data sources”).

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 the field, which also helps the user to fill out 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*".

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.

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 2.3, "Variables").

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

Deleting a short text: Click *Delete short text* to delete this form field.

2.2.7. Long text

You use the predefined *long text* form field (previously known as a "textarea" field) 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.

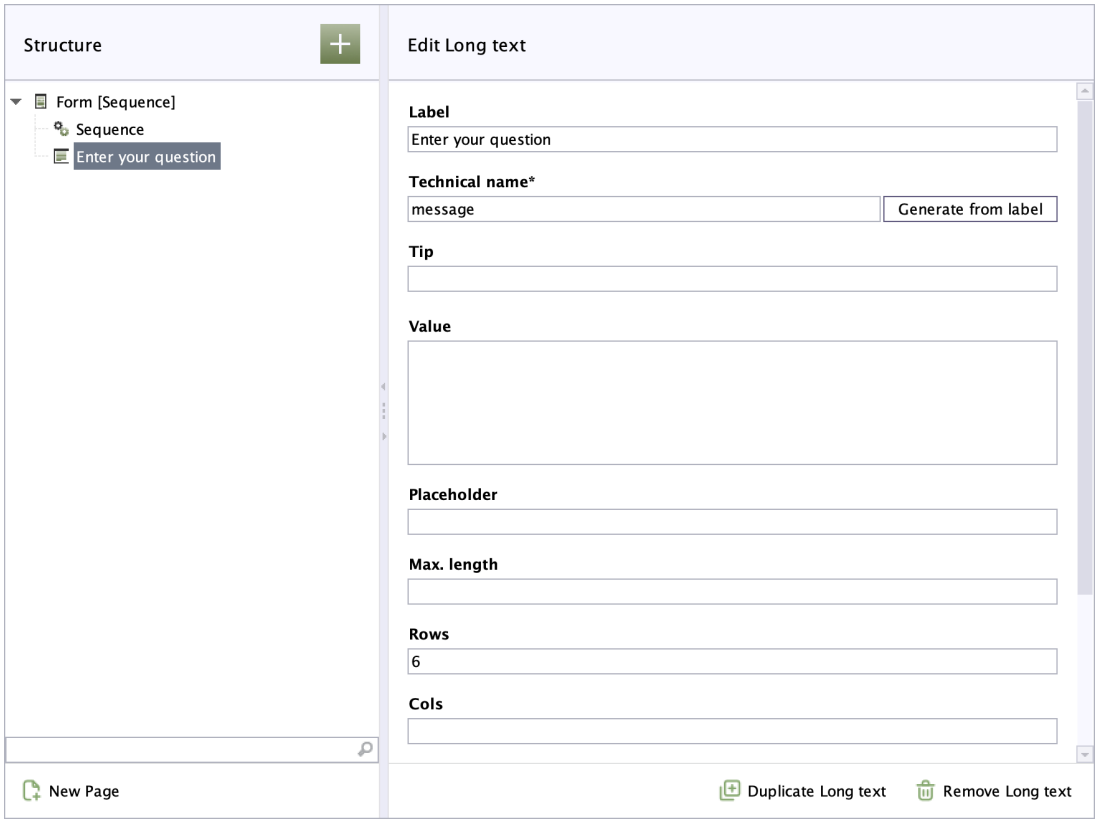


Figure 2.16. Long text

Label: Enter the text of the label that is displayed next to the form field. By default, the label is placed above 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 default technical name is formed from the type name of 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 *Generate from label*.

Please 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.

By default, the note is placed underneath the form 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. This could insert the date automatically into the form field, for example (see Section 2.3, “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: You can use this field to specify a maximum number of characters that the user may enter into the form field.

Rows: You use this field to set the height of the form field. If you enter the value “5”, for example, then the form field will be displayed so that five lines of text will be visible at any one time.

Columns: You use this field to set the width of the form field. If you enter the value “30”, for example, then the form field will be shown word-wrapped to about 30 characters per line.

Field width: Specify how wide the form element should be.

Display variant: Select one or more of the available display variants here, so as to specify how the form element is displayed in the form.

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 2.3, “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 the field, which also helps the user to fill out forms faster.

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

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*”.

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.

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

Deleting a long text: Click *Delete long text* to delete this form field.

2.2.8. Email address

You use the predefined *email address* form field 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.

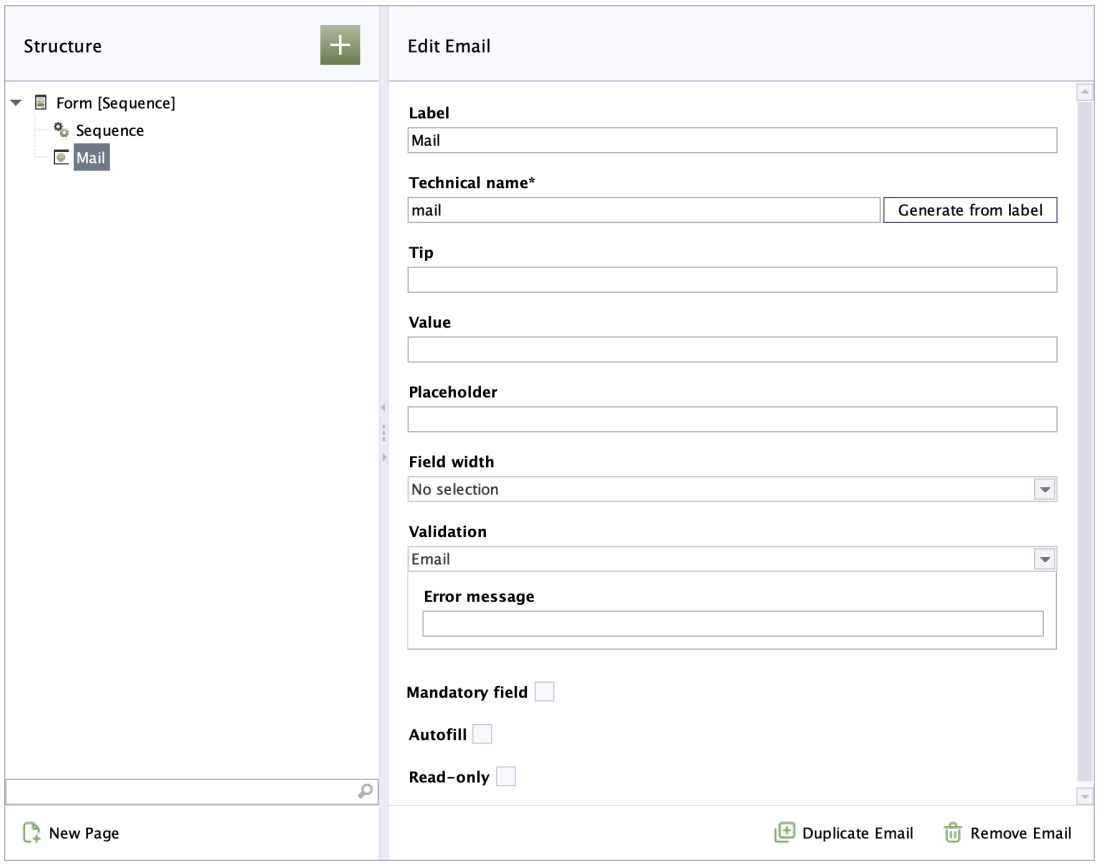


Figure 2.17. Email address

Label: Enter the text of the label that is displayed next to the form field. By default, the label is placed above 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 default technical name is formed from the type name of 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 *Generate from label*.

Please 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.

By default, the note is placed underneath the form 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 2.3, “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.

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 2.3, “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.

Field width: Specify how wide the form element should be.

Display variant: Select one or more of the available display variants here, so as to specify how the form element is displayed in the form.

Validation: You can make the following setting, as 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 their input does not match the format that is required for a valid email address.

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 the field, which also helps the user to fill out 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*”.

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.

Duplicating an email address: Click *Duplicate email address* to add a copy of this form field to your form.

Deleting an email address: Click *Delete email address* to delete this form field.

2.2.9. Number

You use the predefined *number* form field 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.

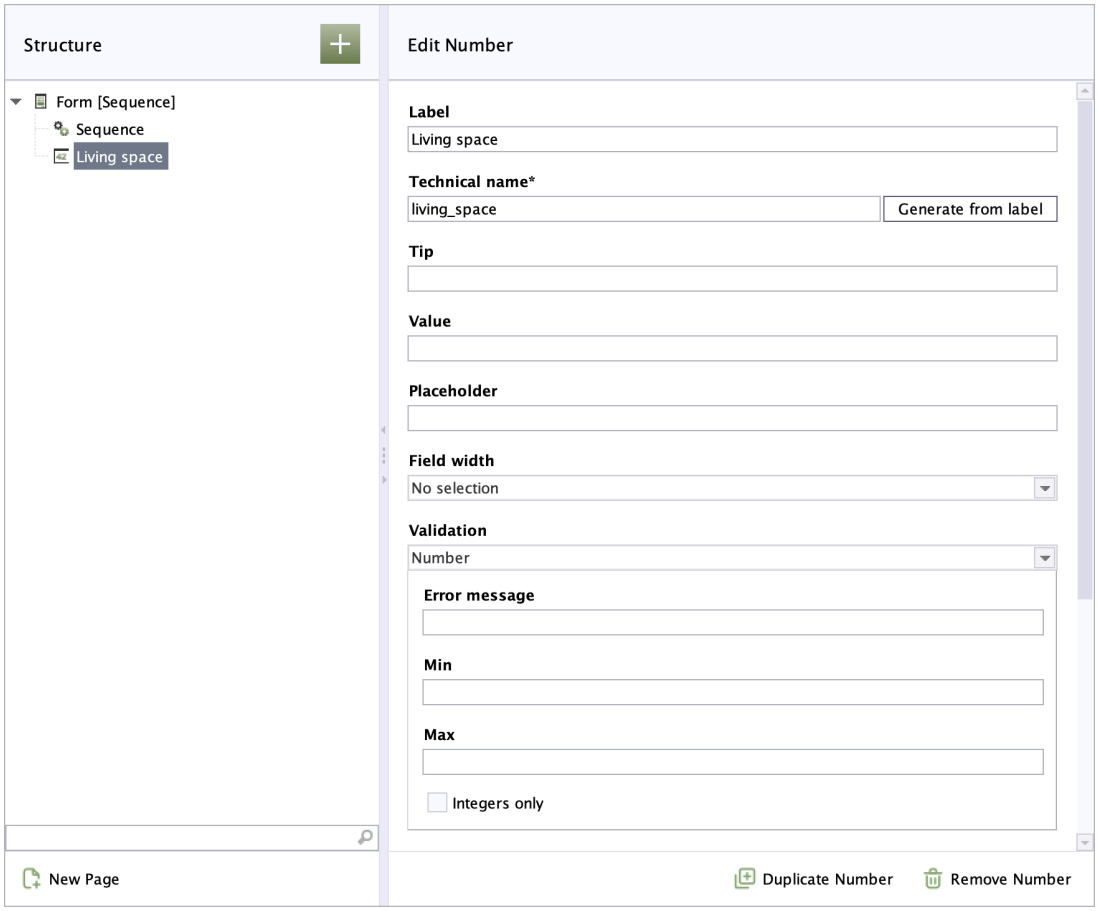


Figure 2.18. Number

Label: Enter the text of the label that is displayed next to the form field. By default, the label is placed above 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 default technical name is formed from the type name of 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 *Generate from label*.

Please 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.

By default, the note is placed underneath the form 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 2.3, “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.

Field width: Specify how wide the form element should be.

Display variant: Select one or more of the available display variants here, so as to specify how the form element is displayed in the form.

Validation: You can make the following settings, as 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.

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.

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 2.3, “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 the field, which also helps the user to fill out 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*”.

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.

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

Deleting a number: Click *Delete number* to delete this form field.

2.2.10. Phone number

You use the predefined *phone number* form field 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.

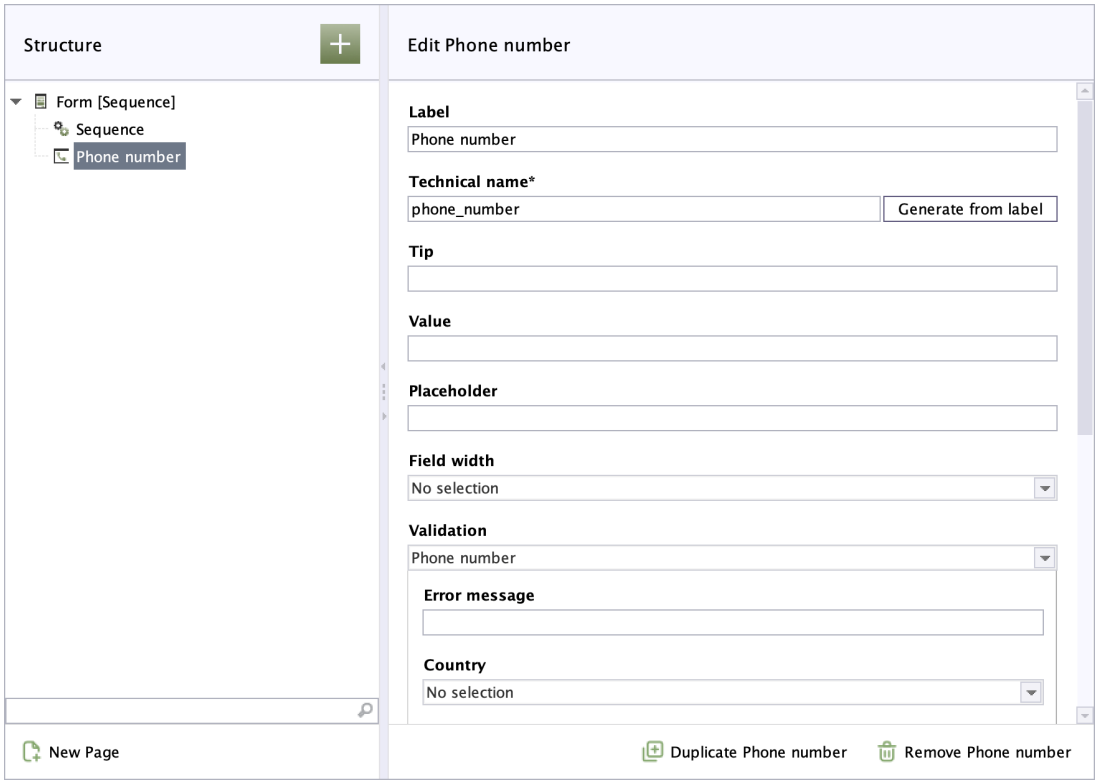


Figure 2.19. Phone number

Label: Enter the text of the label that is displayed next to the form field. By default, the label is placed above 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 default technical name is formed from the type name of 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 *Generate 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.

By default, the note is displayed directly underneath the form 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 2.3, "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.

Field width: Specify how wide the form element should be.

Display variant: Select one or more of the available display variants here, so as to specify how the form element is displayed in the form.

Validation: You can make the following settings, as 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 2.2.5, "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.

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 the field, which also helps the user to fill out 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*".

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.

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 2.3, "Variables").

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

Deleting a phone number: Click *Delete phone number* to delete this form field.

2.2.11. Date

You use the predefined *date* form field 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.

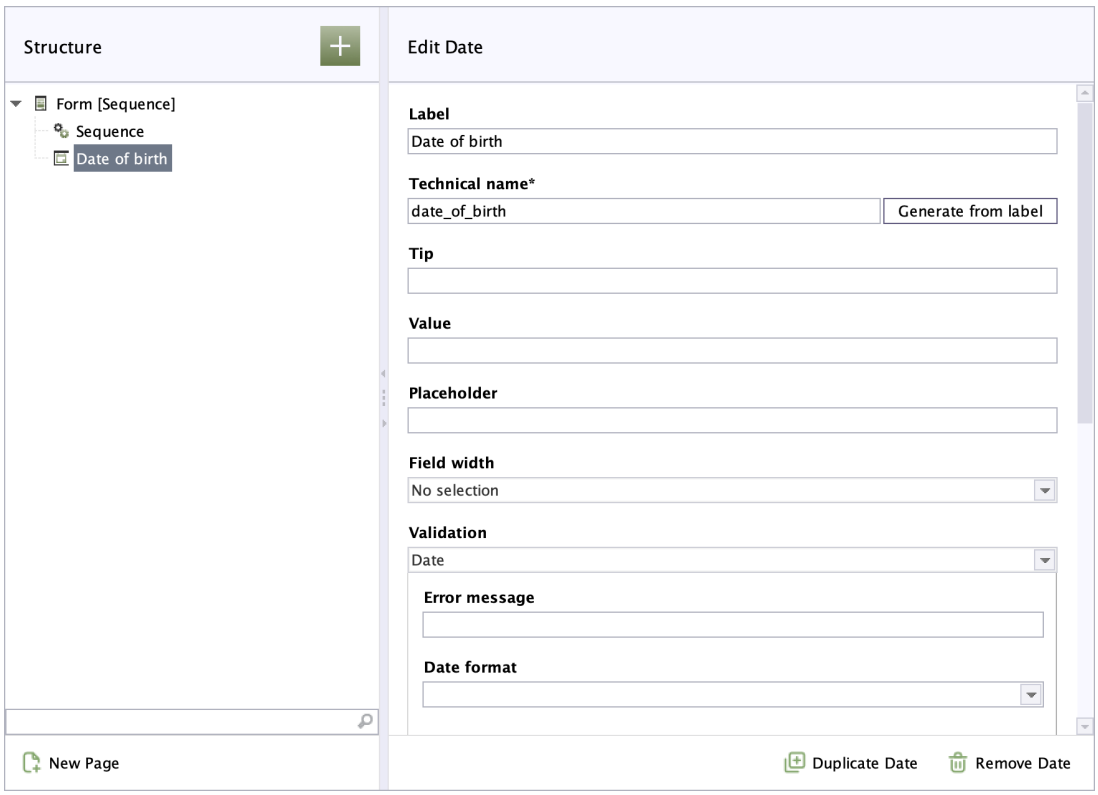


Figure 2.20. Date

Label: Enter the text of the label that is displayed next to the form field. By default, the label is placed above 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 default technical name is formed from the type name of 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 *Generate from label*.

Please 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.

By default, the note is placed directly underneath the form 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 2.3, “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.

Field width: Specify how wide the form element should be.

Display variant: Select one or more of the available display variants here, so as to specify how the form element is displayed in the form.

Validation: You can make the following settings, as 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.

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

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

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

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 2.3, “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 the field, which also helps the user to fill out 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*”.

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.

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

Deleting a date: Click *Delete date* to delete this form field.

2.2.12. Password

You use the predefined *password* form field 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.

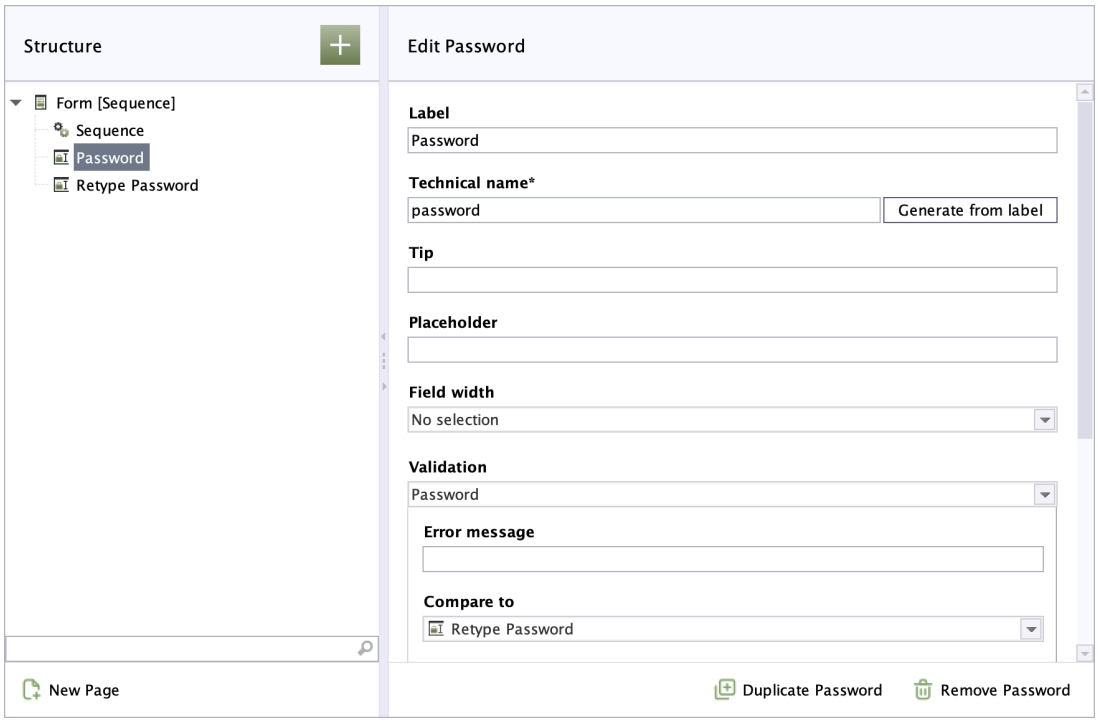


Figure 2.21. Password

Label: Enter the text of the label that is displayed next to the form field. By default, the label is placed above 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 default technical name is formed from the type name of 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 *Generate from label*.

Please 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.

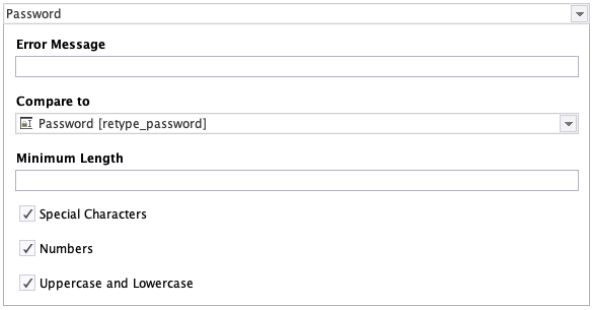
By default, the note is placed underneath the form field.

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: Specify how wide the password field should be.

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

Validation: You can configure various settings here.

Password	This validator checks whether the password entered complies with the security requirements.  **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.

	Compare with: Select a second form field to compare with the input from this first form field. **Minimum length:** Specify the minimum number of characters that must be used for the password. **The password must include:** Check the relevant checkbox if the password must include special characters, numbers and/or lower-case/uppercase letters.

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 the field, which also helps the user to fill out 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.

2.2.13. 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 an email attachment or stored in the web server's file system.

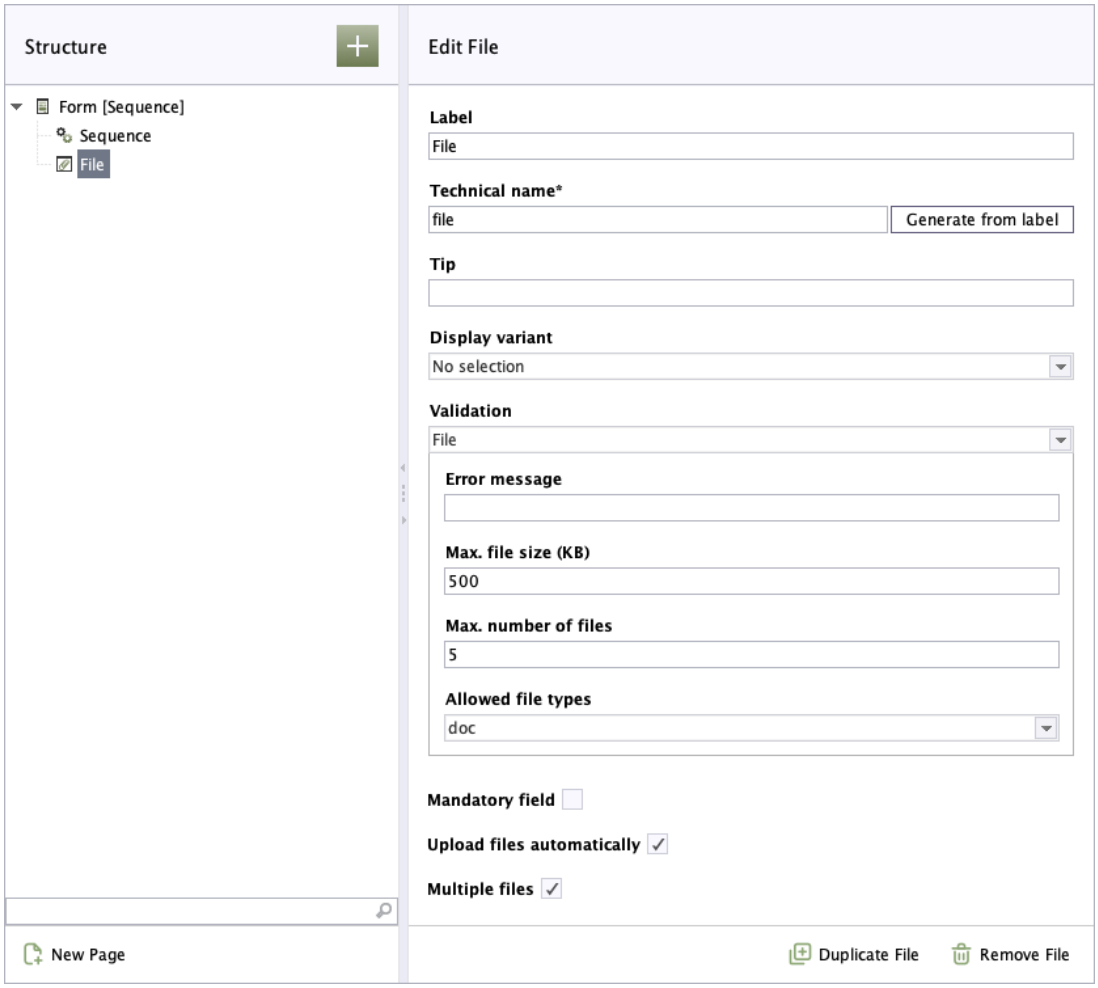


Figure 2.22. Upload file

Label: Enter the text of the label that is displayed next to the form field. By default, the label is placed above 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 default technical name is formed from the type name of 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 *Generate from label*.

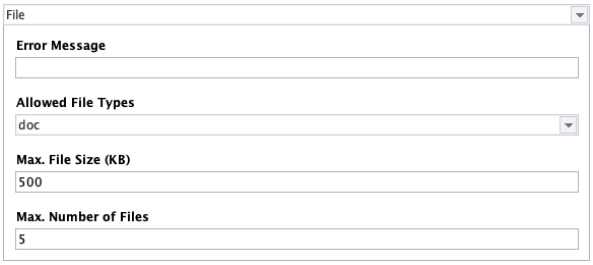
Note: Here you have the option of adding some text to your upload area that gives the user additional information. This could be information about the maximum file size allowed, for example.

By default, the note is displayed directly underneath the upload area.

Field width: Specify how wide the form field should be.

Display variant: Select one or more of the available display variants here, so as to specify how the form element is displayed in the form.

Validation: You can configure various settings here.

File	This validator checks the size and the type of an uploaded file.  **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 of files that users can upload. Otherwise, you may run into problems with data traffic handling if several users attempt to upload large files at the same time. **Maximum no. of files:** Use this to specify how many files the user can upload simultaneously. **Permitted 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.

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.

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.

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

2.2.14. 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.

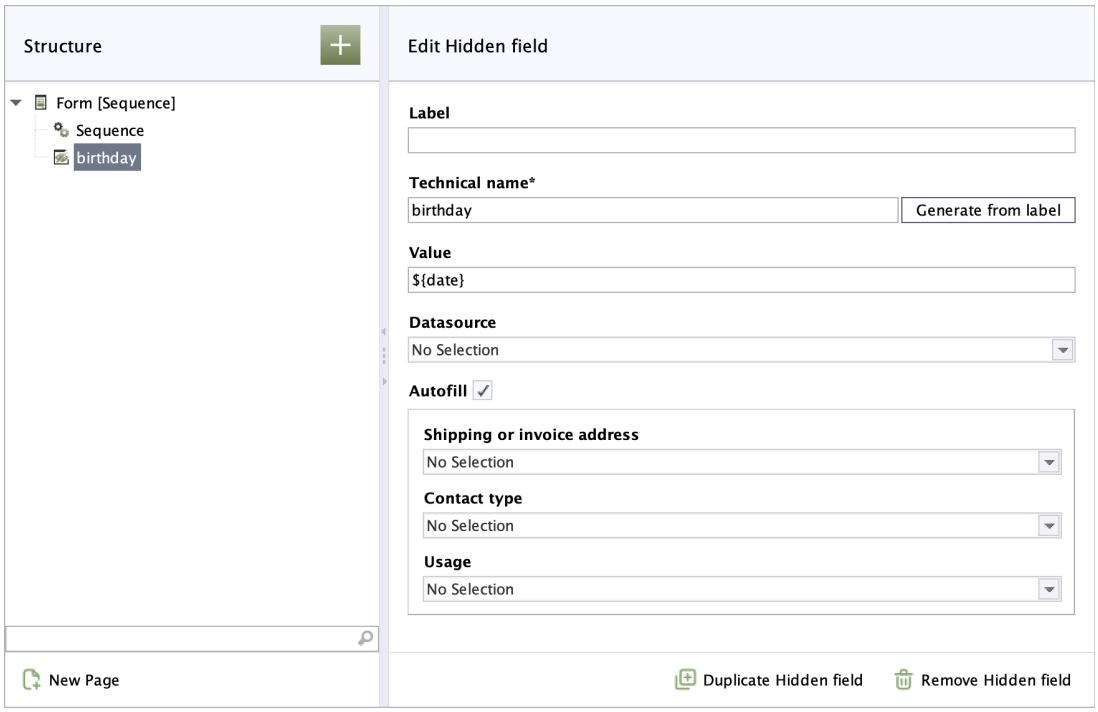


Figure 2.23. Hidden field

Label: Enter a piece of descriptive text that will be sent together with the value from the hidden field. This helps you to distinguish one data item from another 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 default technical name is formed from the type name of 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 *Generate 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 2.3, “Variables”).

Data source: Select a data source here that determines the value of the hidden field dynamically at runtime (see also Section 2.5, “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 the field, which also helps the user to fill out 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*”.

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.

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

Deleting a hidden field: Click *Delete hidden field* to delete this form element.

2.2.15. Page break

You use the *page break* element to construct multi-page forms. Insert this element at a position in the form where you want to start a new page. There are no restrictions on placement. All of the form elements following a page break will be shown on the next page. With multi-page forms, the user is shown “Next” and “Back” buttons. The user can then use these buttons to page back and forth between the individual form pages.

Forms can consist of as many pages as necessary, and you can even create empty pages.

If a page should only be displayed if the user has made specific kinds of input on the previous pages of the form, then you can define a condition to provide this functionality.

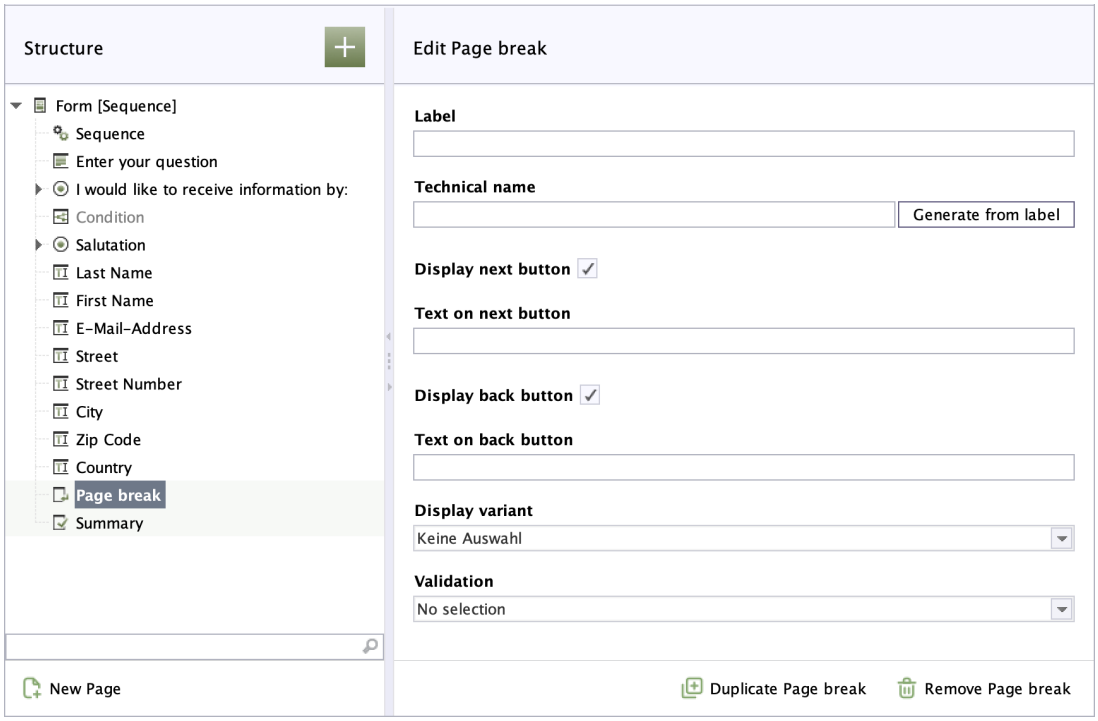


Figure 2.24. Page break

Label: This text is output as a heading shown to the user on the form page.

Technical name: Enter the technical name of the form page here.

Text on next button: This text is displayed on the “Next” button that is shown on the first page of a multi-page form. This text is not used for forms that have only one page.

Text on back button: This text is displayed on the “Back” button.

Display variant: Use this to pick an alternative presentation style for the form page.

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

The same validation options as are explained in the the section called “Page properties tab” section under *Validation* are also available to you here.

2.2.16. Page break condition

Placing the condition element under a page break makes it possible for you to dynamically show or hide the form page depending on user input.

If there are multiple conditions for a form page, their evaluation occurs in the order in which they are assigned below the Page Break element. Note that individual condi-

tion elements are linked using *OR*. This means that evaluation stops as soon as one condition is met and the following conditions are then no longer evaluated.

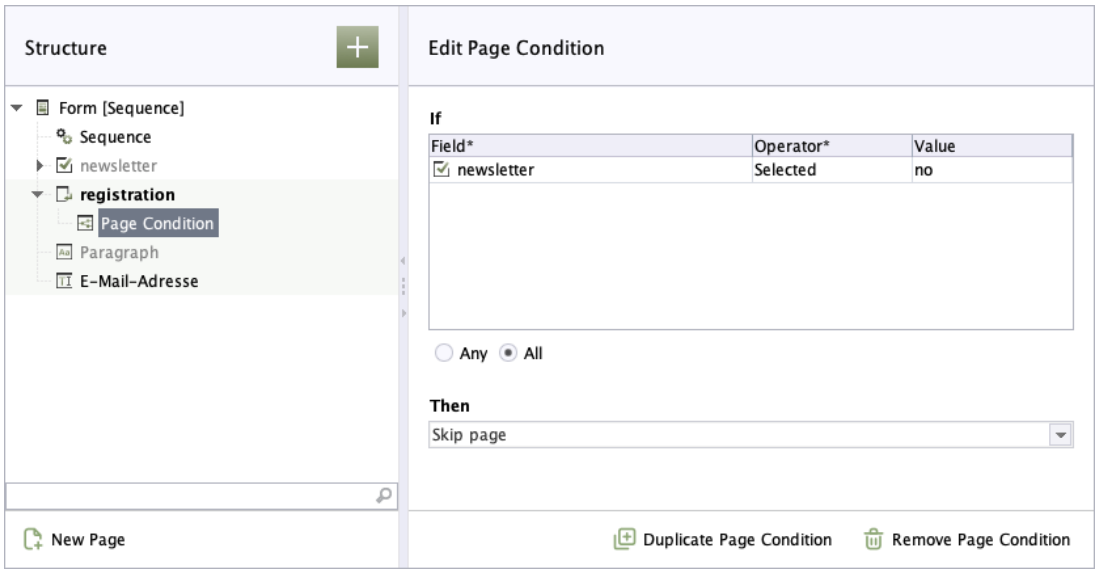


Figure 2.25. Page break condition

If: If this condition is satisfied, this page is not 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.

The condition can be created as described in Section 2.2.19, “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, for example, then you must ensure that the fields concerned are mandatory fields.

Then: In this drop-down list, you can 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.

This list only displays form pages where the *Technical name* field is filled out.

2.2.17. 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.

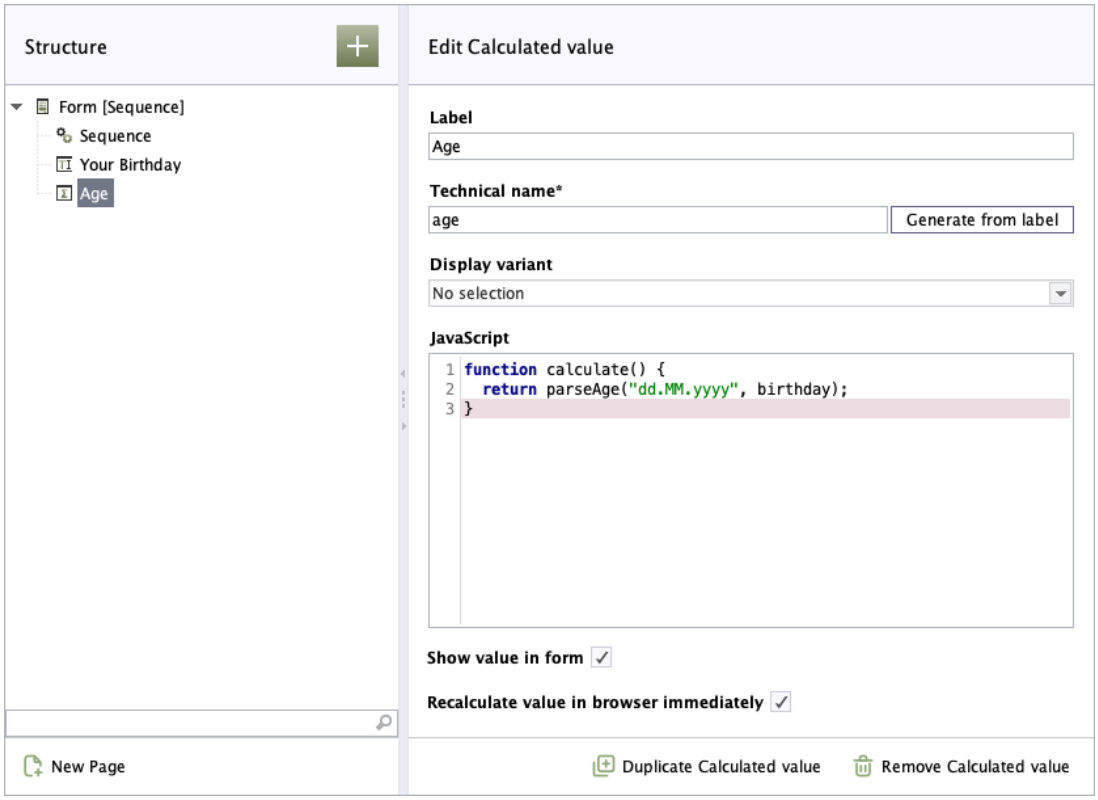


Figure 2.26. Calculated value

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 default technical name is formed from the type name of 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 *Generate 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); }
```

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.

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.

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

Deleting a calculated value: Click *Delete calculated value* to delete this form element.

2.2.18. 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.

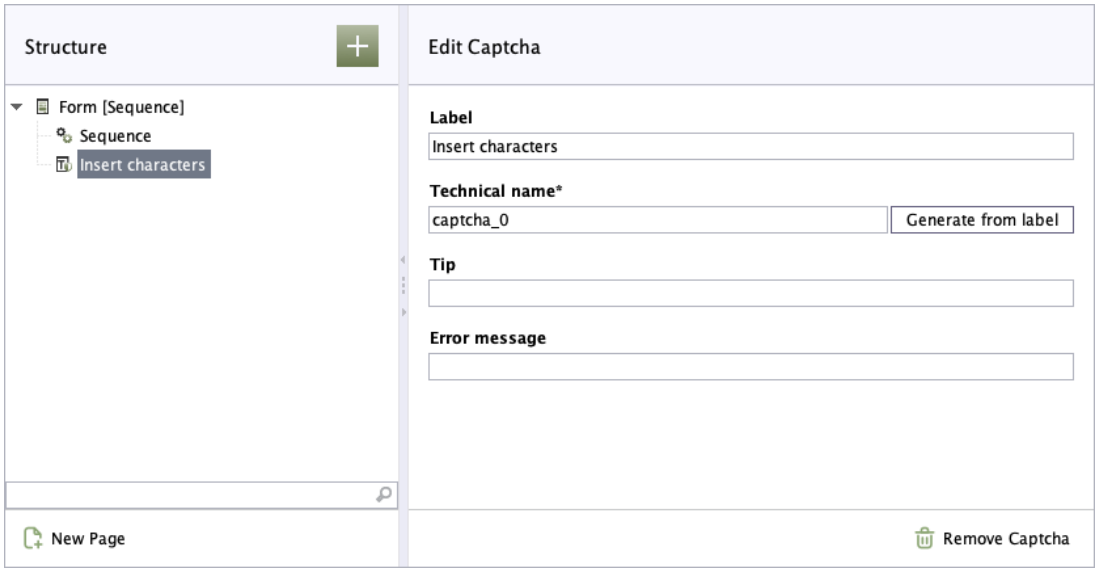


Figure 2.27. Captcha

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 default technical name is formed from the type name of 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 *Generate from label*.

Please 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.

Error message: Enter some text here for an error message that should be displayed instead of the default error message.

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

Deleting a captcha: Click *Delete captcha* to delete this form element.

2.2.19. 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.

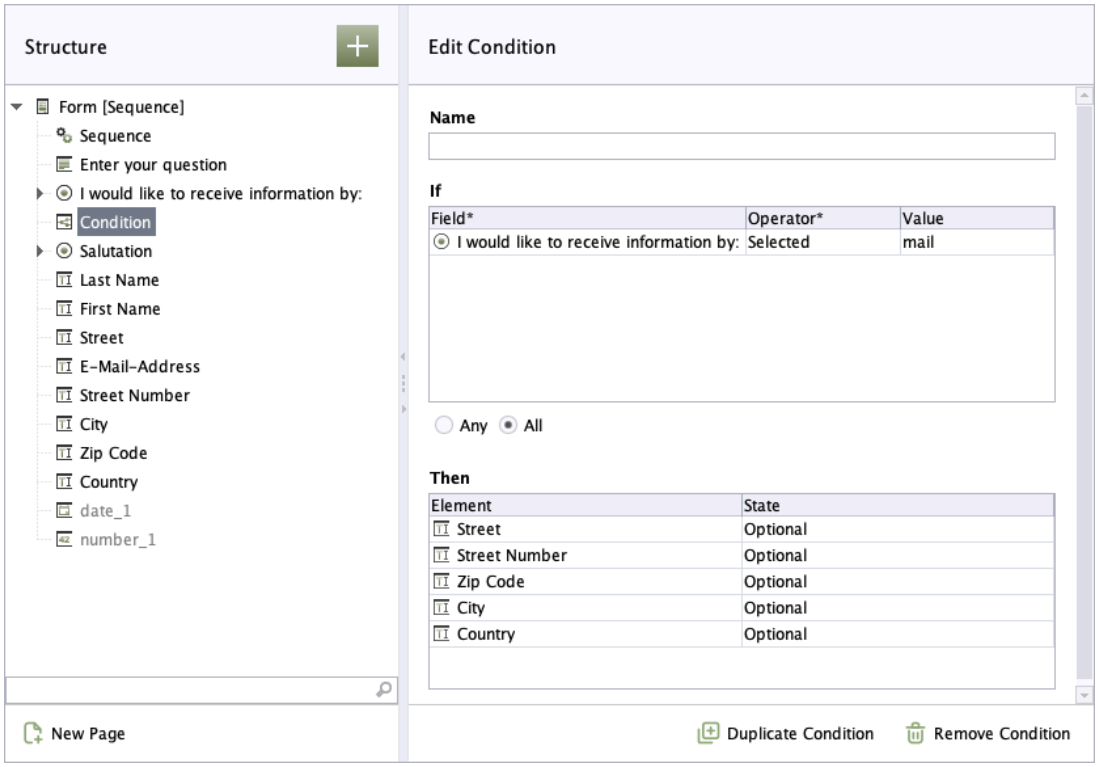
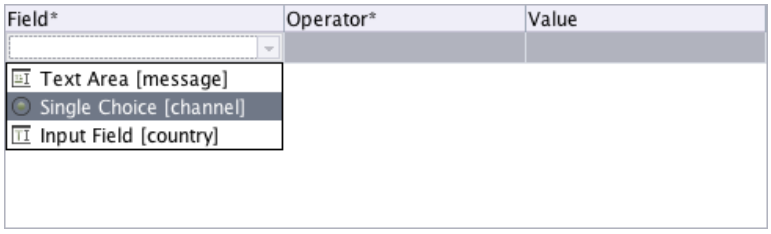


Figure 2.28. Condition

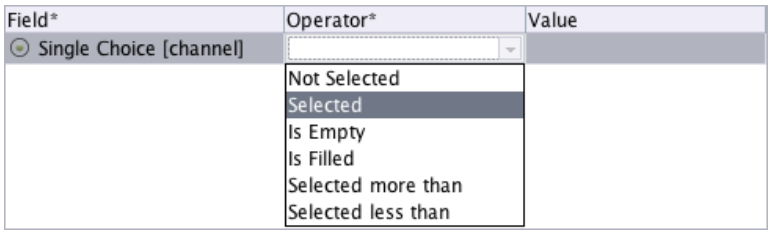
Name: Enter a name for the condition to help you identify it later. The name is shown only in the form tree.

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

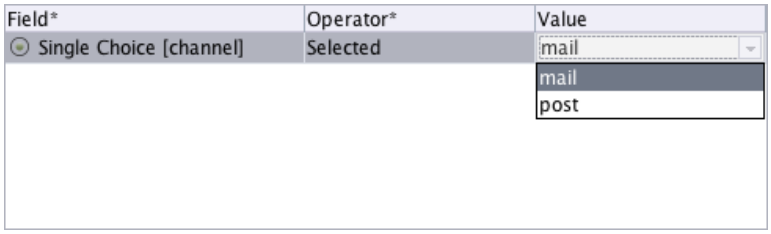
From *Field*, select the form element that you want to include in the condition that you have just created.



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. The operators offered to you here will depend on the field content and whether the field is a selection field or a text input field.



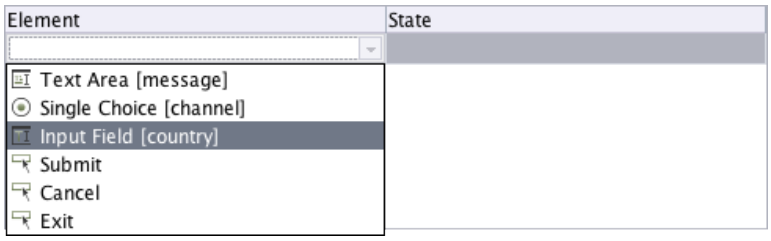
In the *Value* column, enter the comparison value or – in the case of a drop-down list – select the corresponding option.



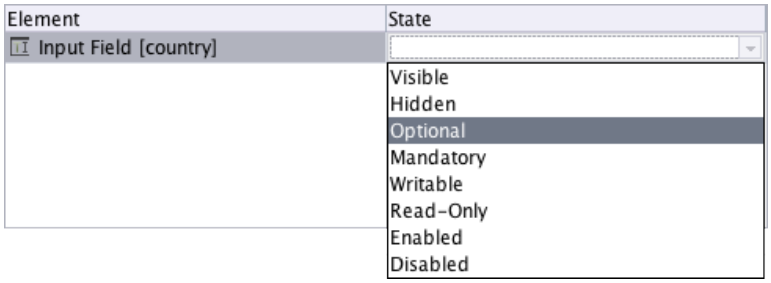
One condition – all conditions: If you define multiple triggers, then you need to specify whether one or all triggers must be activated for the outcome to happen specified in *Then*.

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

In *Field*, select the form element that will be referenced by the trigger that you have specified in the *If* step.



In *State*, select the outcome that should happen when the criteria for the trigger are fulfilled. You can choose from several states here, depending on the form element selected.



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.

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

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

2.2.20. 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.

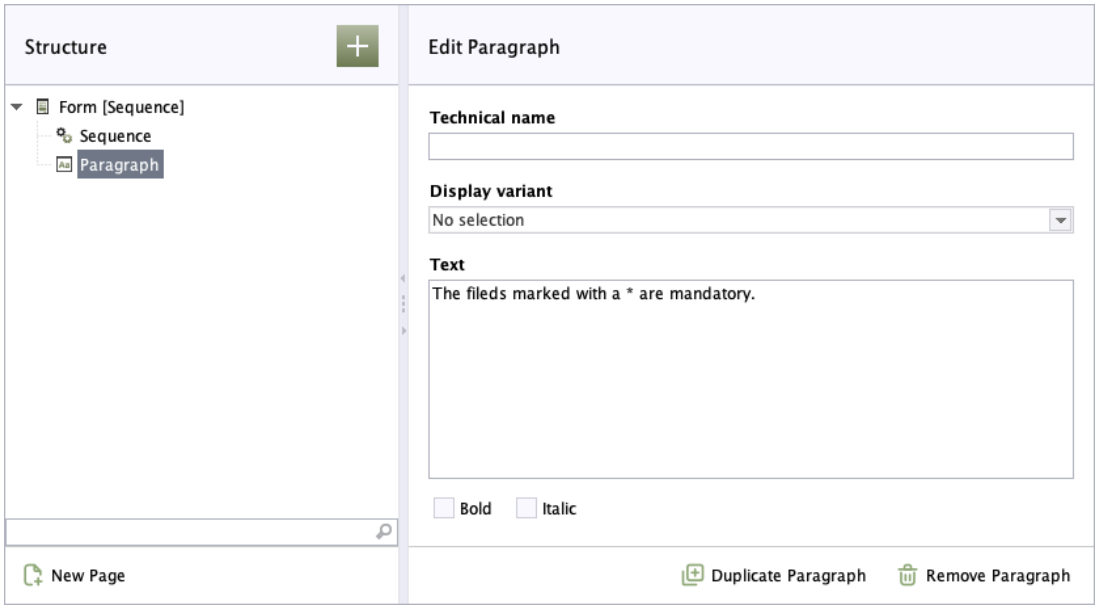


Figure 2.29. Paragraph

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 default technical name is formed from the type name of 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.

Display variant: Select one or more of the available display variants here, so as to specify how the form element is displayed in the form.

Text: Enter a piece of text here to be displayed in your form. You can use Section 2.6, “Markdown” to format the text.

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

Deleting a paragraph: Click *Delete paragraph* to delete this form element.

2.2.21. 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.

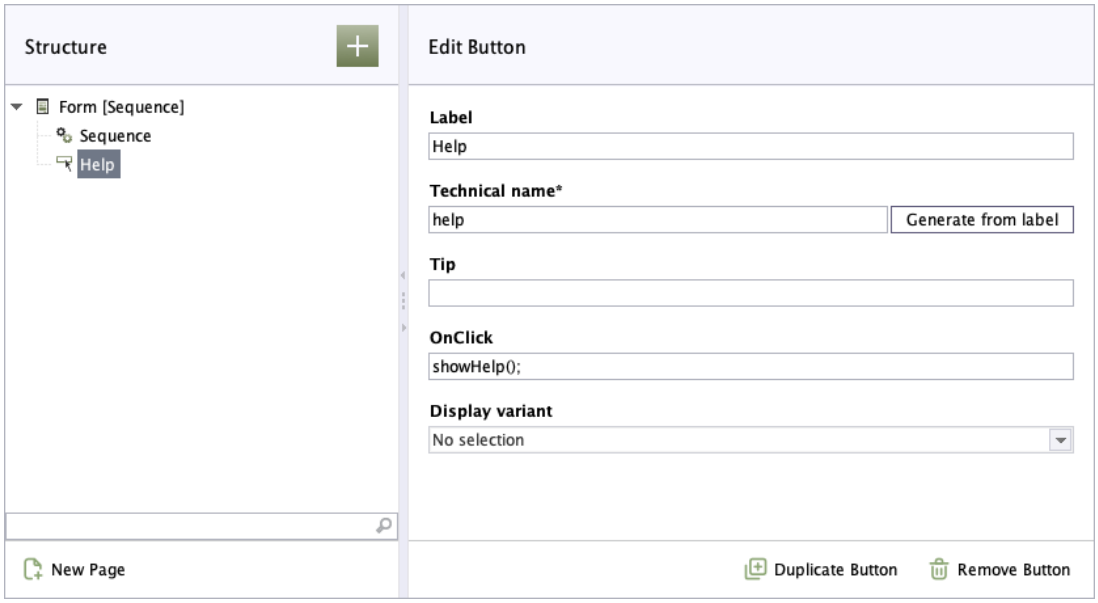


Figure 2.30. Button

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 default technical name is formed from the type name of 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 *Generate 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 or more of the available display variants here, so as to specify how the form element is displayed in the form.

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

Deleting a button: Click *Delete button* to delete this form element.

2.2.22. Fieldset

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

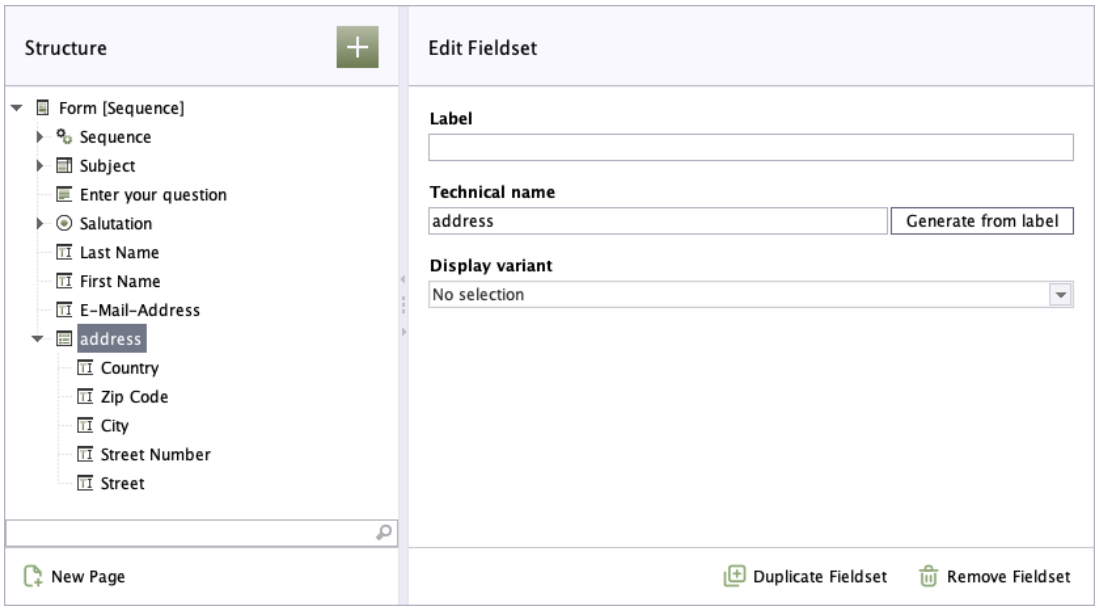


Figure 2.31. Fieldset

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 default technical name is formed from the type name of 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 *Generate from label*.

Display variant: Select one or more 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.

2.2.23. Image

The image form element offers you countless ways to make your form more individual. Adding your company logo or product photos are just two examples of how to do this. And you can also add imagery to give your form a certain style.

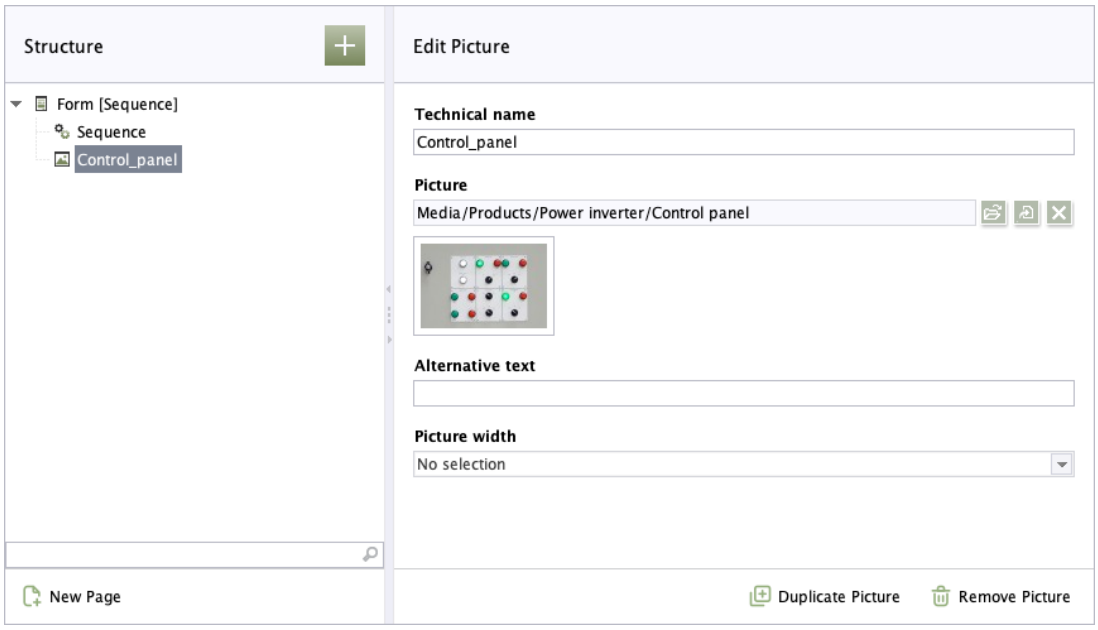


Figure 2.32. Image

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.

Image: Select the image that you want to add to your form. You can select an image from the FirstSpirit Media Store.

Alternative text: Enter a piece of alternative text that describes the image. This text is shown if the image itself cannot be displayed and will be read out if a screen reader is being used.

Image width: Specify how wide the image should be.

Display variant: Select one or more of the available display variants here, so as to specify how the image is displayed in the form.

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

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

2.2.24. 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.

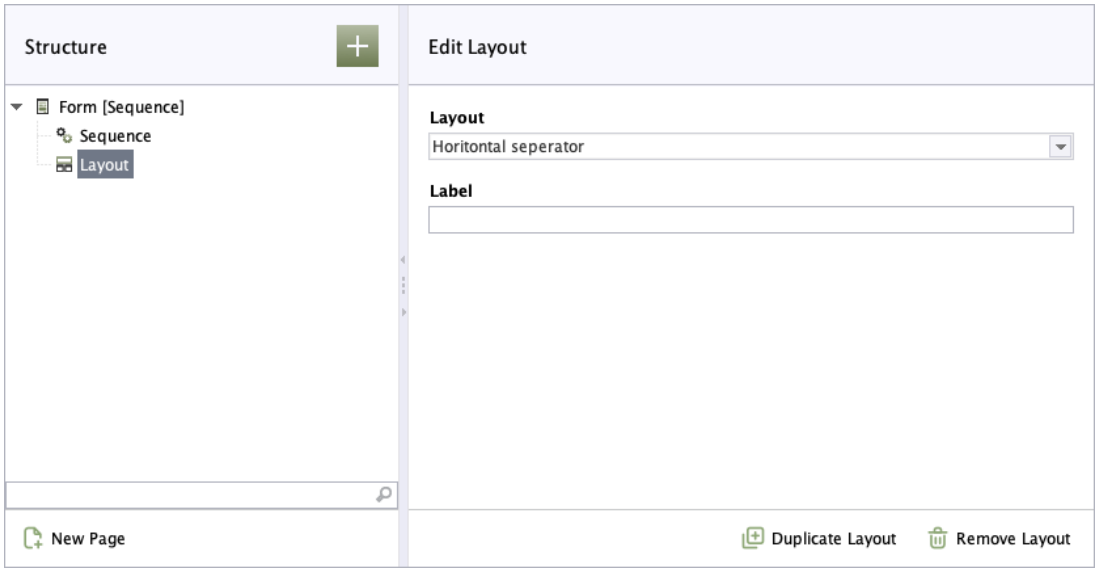


Figure 2.33. Layout

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.

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

Deleting a layout: Click *Delete layout* to delete this form element.

2.2.25. 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 to create a “Check your input” page as the last page in your form, for example, which lists all of the input that the user has entered. The user can then check the accuracy of the data here and correct their input as required. The user can use the “Back” button to go to the relevant form page and make any necessary changes.

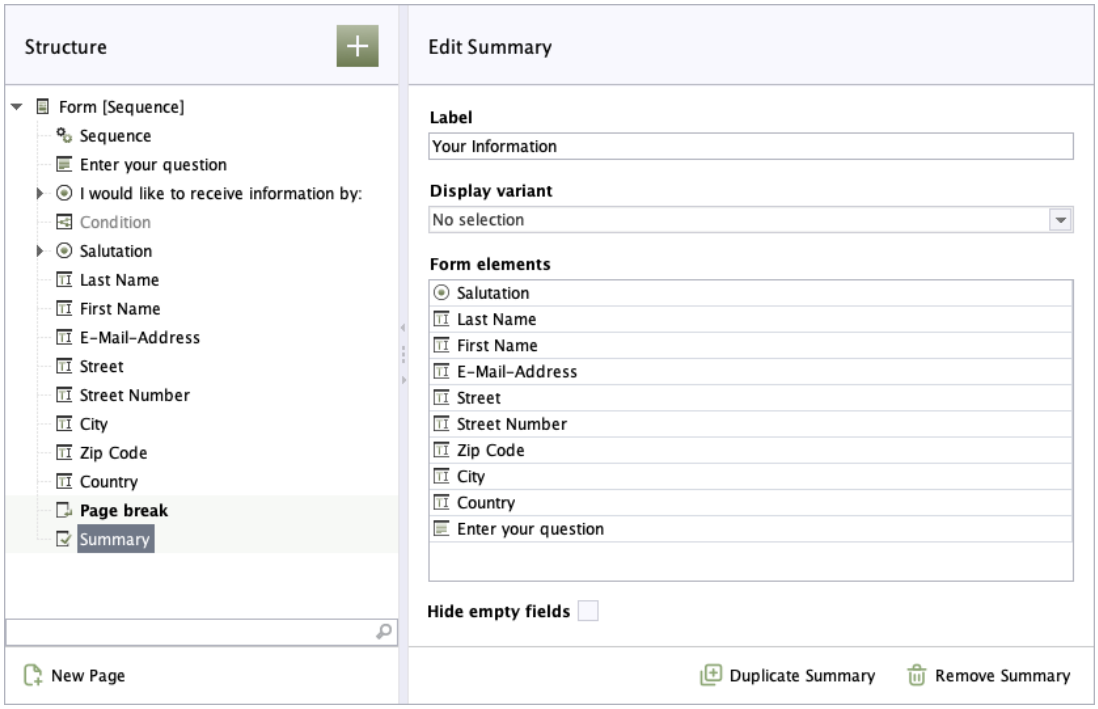


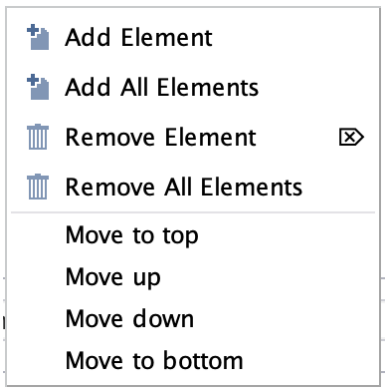
Figure 2.34. Summary

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

Display variant: Select one or more of the available display variants here, so as to specify how the summary is displayed in the form.

Form elements: In this list, you specify the form values that will be displayed in the summary. The values are output in the order in which they are arranged in the list.

To add a new form element to the summary, right-click the field. In the menu that is then displayed, select the *Add element* menu option.



If you double-click the newly-added row, a drop-down menu is shown. From this list, select your chosen form element.

To add all existing form elements to the summary, right-click the field. In the menu that is then displayed, select the *Add all elements* menu option.

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.

If you make no selections in this list, then the user is simply shown the values for all form elements that would normally be included in the summary. If you want to have form elements of the *password* or *hidden field* type displayed, then you will need to select these explicitly.

Paragraph elements can be included in the summary if you assign a name to the element beforehand.

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.

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

Deleting a summary: Click *Delete summary* to delete this form element.

2.3. 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 = 30/05/2013).
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 = 17:33).
serverDate	The date, in the server'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).
serverTime	The time, in the server'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).

Variable	Description
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. 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). 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: *30/05/2013 18:36*

2.4. Actions

You use actions to specify how the data entered by the user should be processed. Various actions are available for you to use. You can add these to your form either individually or together with a sequence action (see Section 2.4.8, "Sequence").

The action is executed when the user clicks the submit button. For multi-page forms, this is located on the last page of the form. Accordingly, the position at which an action is inserted below the form element has no effect on the point in time at which it is executed.

For all actions (with the exception of the sequence action), you have the option of specifying the user input where the action is executed or not executed, as appropriate.

As one example, this functionality could be used to send the user a copy of their form input via email if they have agreed to this as part of filling out the form.

To use this functionality, switch to the *Condition* tab in the detail view for the respective action.

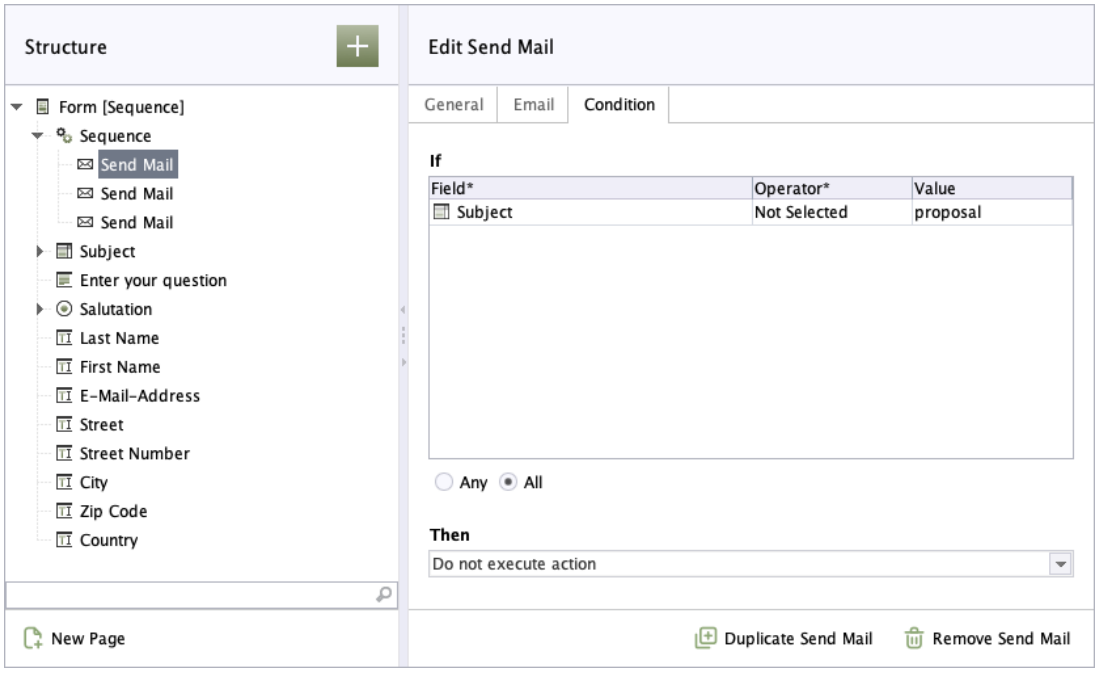


Figure 2.35. Condition tab

If: Use this field to specify the user input for which this action should not be executed/ not executed.

The condition can be created as described in Section 2.2.19, “Condition”.

Then: In this drop-down list, you can specify whether the action is executed or not executed if the abovementioned condition is fulfilled.

All of the other configuration options for form actions are described in detail in the following sections.

2.4.1. Send as Email

The send as email action collects all of the data entered into the form and sends it as an email attachment to any number of recipients.

General tab

Figure 2.36. Send as Email – “General” tab

Confirmation: Enter the confirmation message that is shown to the user on the web page after the form has been submitted. The confirmation text can also be formatted with the help of markdown (see Section 2.6, “Markdown”).

Email tab

Enter the recipient, a subject line and the message body – just as you would when composing an email normally. In addition, you also have the option here of specifying which of the data items entered by the user should actually be sent in the email.

When configuring the send as email action, you can also utilise the data entered by the user directly – so as to send a copy of the message to the user, for example. To do so, simply enter a placeholder using the format `${technicalName}` into the corresponding property field in the send as email action.

Example: On a contact form, you have defined an input field *email*. The user is asked to enter their email address into this field. If you now give the field *CC* the variable `${email}` the user will receive a copy of the email created.

You can re-use the labelling for an option by extending the field name with the suffix *#label*, like this: `${technicalName#label}`

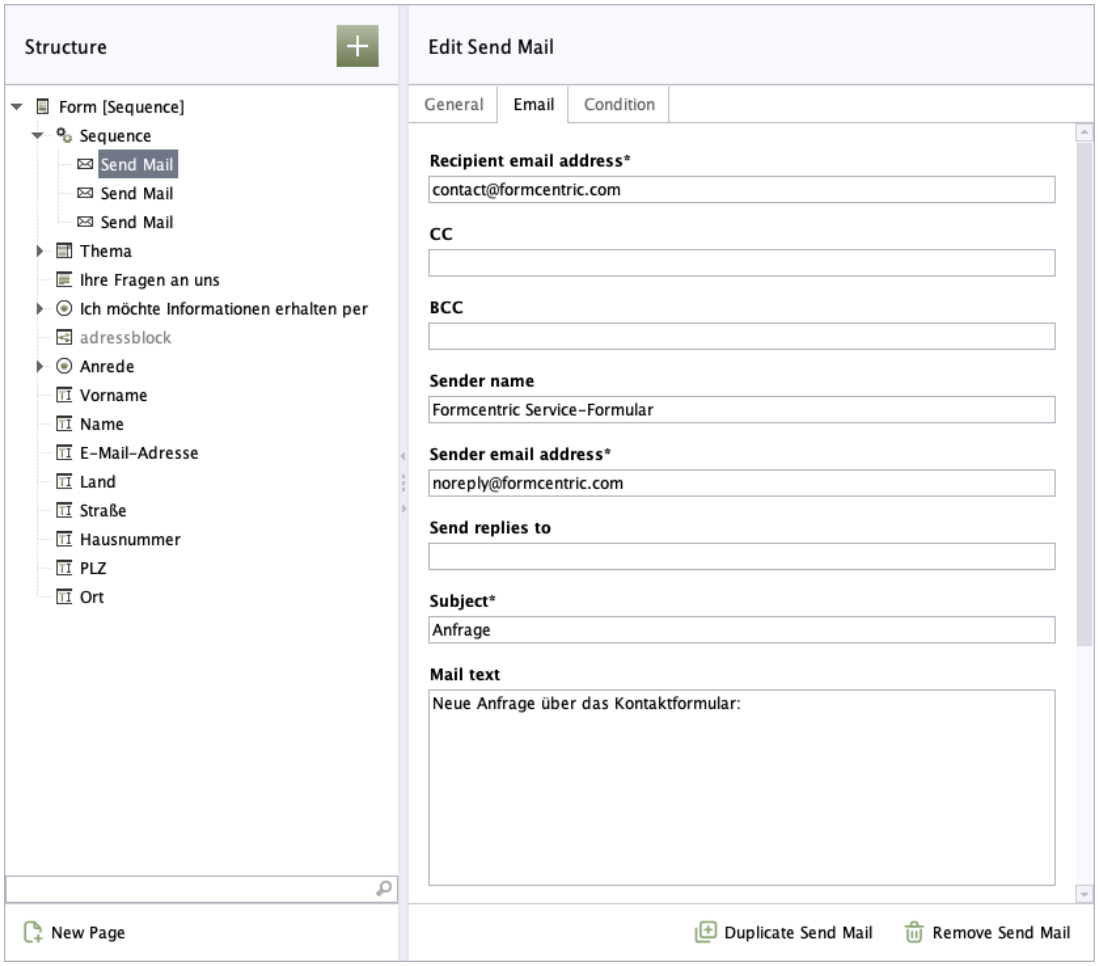


Figure 2.37. Send as Email – “Email” tab

You can send the email created to multiple recipients at the same time by entering multiple addresses (recipient email address, CC, BCC) into one address field. Use a comma to separate the individual addresses.

Recipient email address: The email addresses to which the data entered by the user should be sent.

Cc: The email addresses to which a copy of the data entered by the user should be sent.

Bcc: The email addresses to which a blind carbon copy of the data entered by the user should be sent.

Sender name: The name to be used as the sender.

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

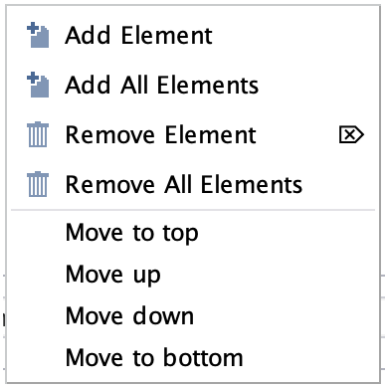
Send replies to: One or more comma-separated email addresses, to which any replies to the email should be sent (if not the same as the sender email address).

Subject: The subject line of the email to be sent.

Message: The text that is inserted into the body of the email, in addition to the form data.

Elements: In this list, you specify the form values that will be displayed in the email. The values are output in the order in which they are arranged in the list.

To include a new form element in the email, first right-click the field with the mouse. In the menu that is then displayed, select the menu option *Add element*.



If you double-click the newly-added row, a drop-down menu is shown. From this list, select your chosen form element.

To add all existing form elements to the email, right-click the field. In the menu that is then displayed, select the *Add all elements* menu option.

If you do not select anything here, then the email sent contains only the message text specified.

Fields that have been hidden by conditions are never included in the email.

Paragraph elements can be included in the email if you assign a name to the element beforehand.

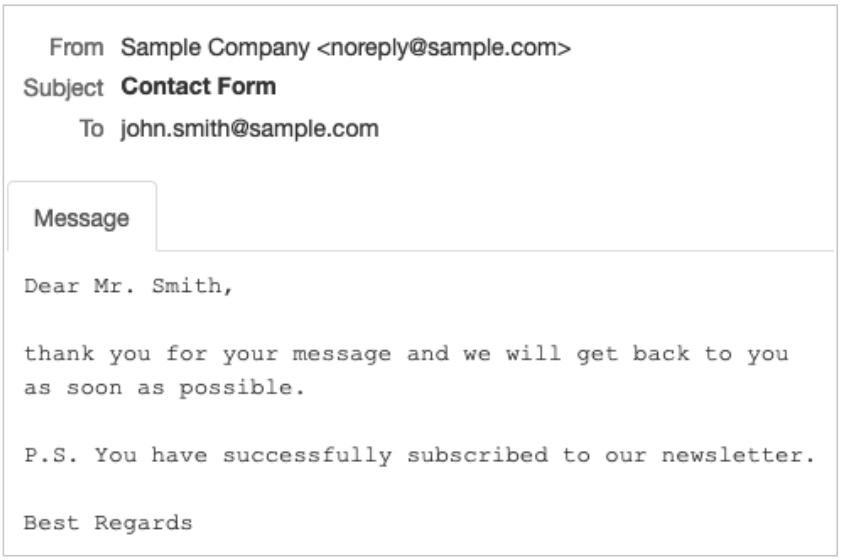
Exclude empty fields: If you activate this check box, then only fields that have been filled out will be included in the email that is sent. Empty fields will be filtered out.

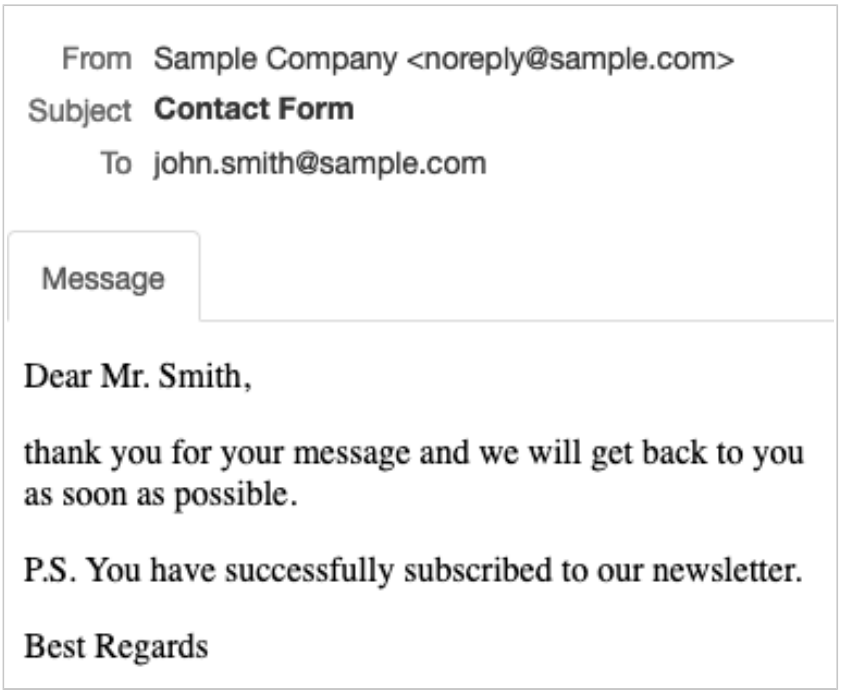
Email format: This field lets you specify whether the email should be sent in HTML format or as a plain text message.

Format	Description
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.
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.

Format	Description

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.
	`Dear ${salutation#label} ${lastname},`
	
	`thank you for your message and we will get back to you as soon as possible.`
	`<#if newsletter == 'true'>`
	
	`P.S. You have successfully subscribed to our newsletter.`
	`</#if>`
	
	`Best Regards`

Format	Description
	
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. The following example shows how you can modify the message text with the help of `#if` statements. `<p>Dear ${salutation#label} ${lastname},</p>` `<p>thank you for your message and we will get back to you as soon as possible.</p>` `<#if newsletter == 'true'>` `<p>P.S. You have successfully subscribed to our newsletter.</p>` `</#if>`

Format	Description
	

Formcentric offers you the option of sending multiple emails with different content. To do so, simply create another send as email action. If you want to send the exact same content again, simply enter multiple recipients for the mail.

Duplicating a send as email: Click *Duplicate send as email* to add a copy of this action to your form.

Deleting a send as email: Click *Delete send as email* to delete this action.

2.4.2. 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 means the user can then download a PDF file containing their data.

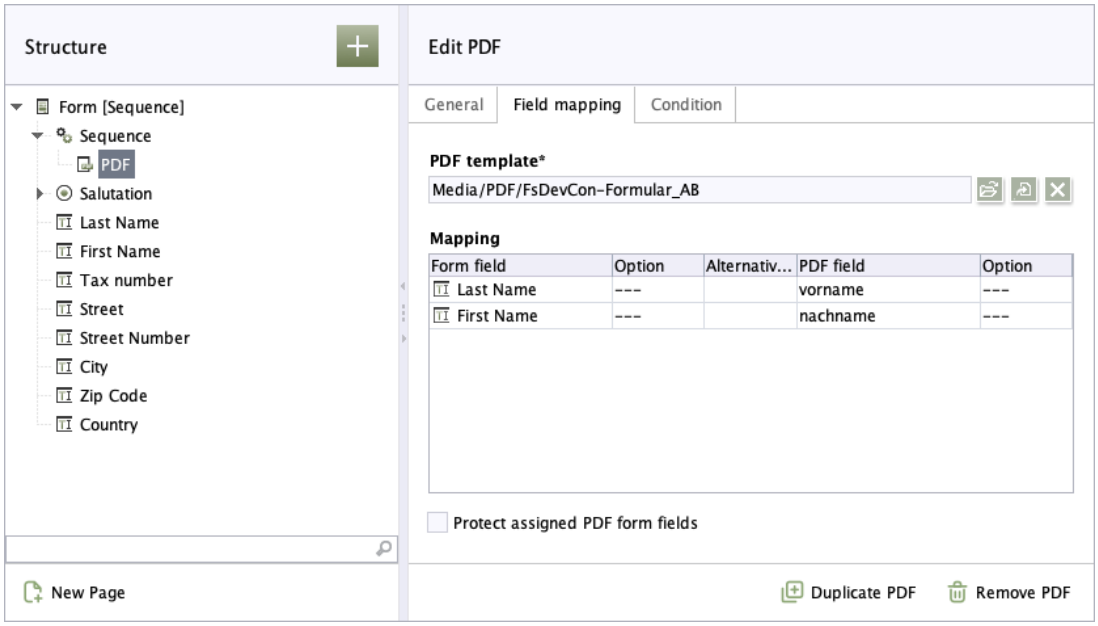
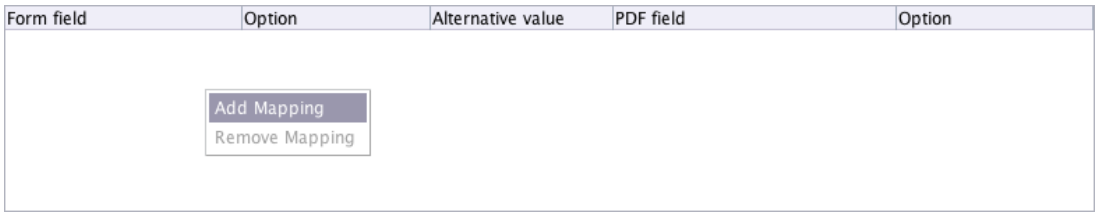


Figure 2.38. PDF

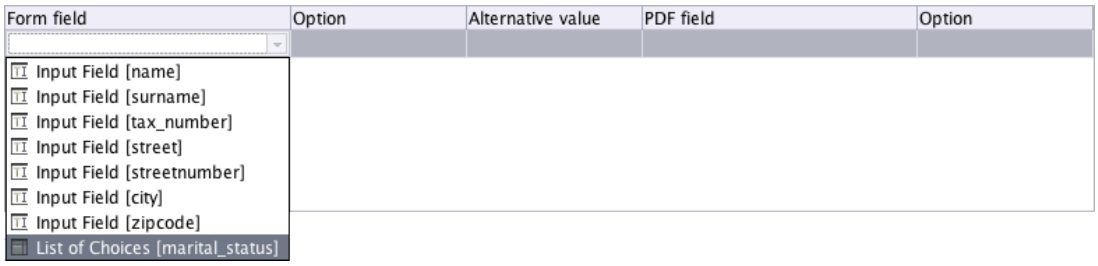
Template: In this field, you can select a PDF document (PDF template) from the FirstSpirit Media Management.

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.

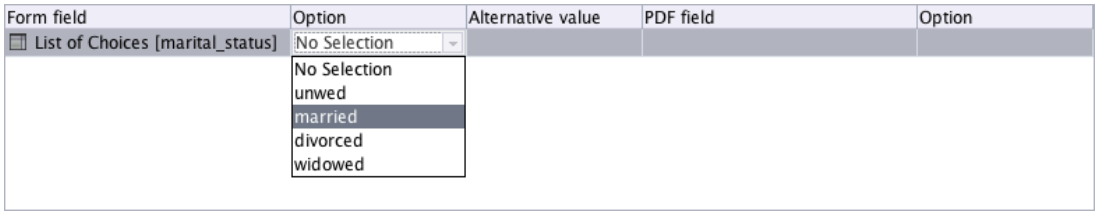
To add a new assignment to the list, right-click with the mouse on the assignment field.



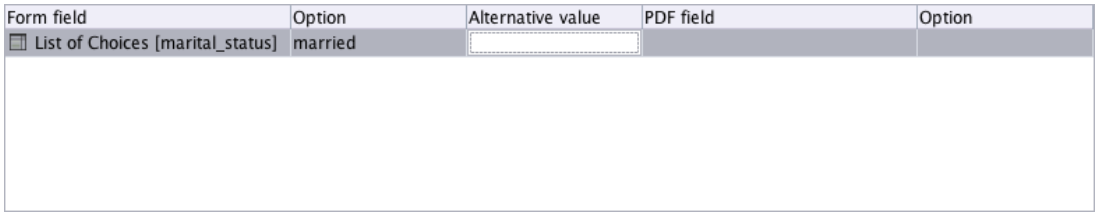
If you double-click the row added, a drop-down menu is shown. From this list, select the form element that you wish to include in the assignment.



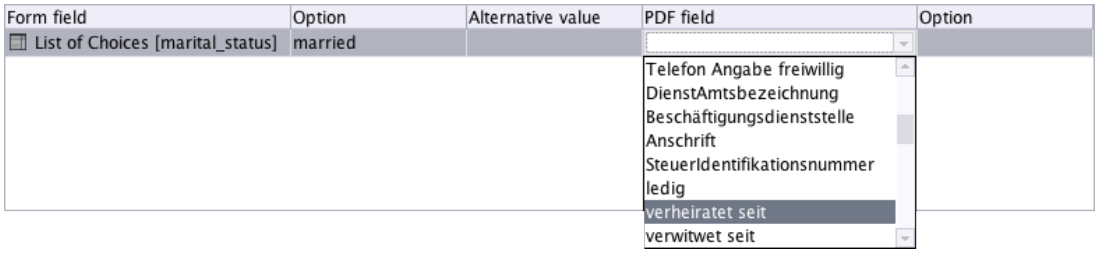
If the form field selected is itself a selection field (single choice, multiple choice or drop-down list), then you can use the *Option* column to specify an individual option from the selection field.



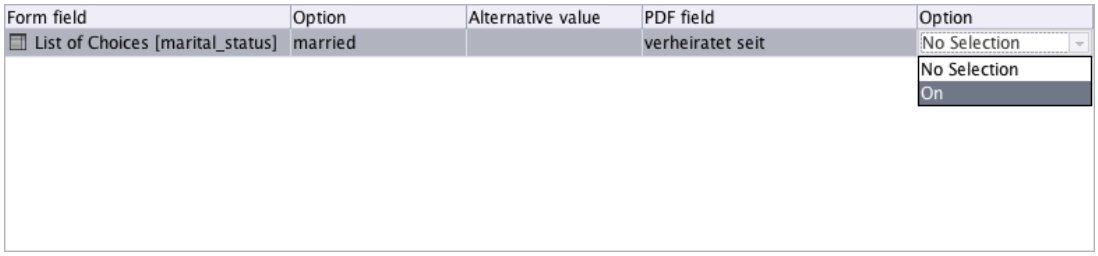
You can enter an alternative value in the *Alternative value* column. This will be entered into the PDF form instead of the value entered or selected by the end user.



You can select a form field from the PDF template in the *Field in the template* column.



If the field selected is itself a selection field you can use the last column *Option* to select an option.



Protect assigned PDF form fields: 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. The confirmation text can also be formatted with the help of markdown (see Section 2.6, “Markdown”).

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.

Duplicating a PDF: Click *Duplicate PDF* to add a copy of this action to your form.

Deleting a PDF: Click *Delete PDF* to delete this action.

Before you can execute the PDF action, you must publish the selected PDF template: this ensures that the template is available both in the Preview and the Live website.

To publish PDF templates, a special publication schedule must be both created and configured (see section 4.4 of the Developer Manual). Available publication schedules can be accessed via the menu option "*Projects > Execute schedule*".

2.4.3. Data source

This action saves the form data in a FirstSpirit data source.

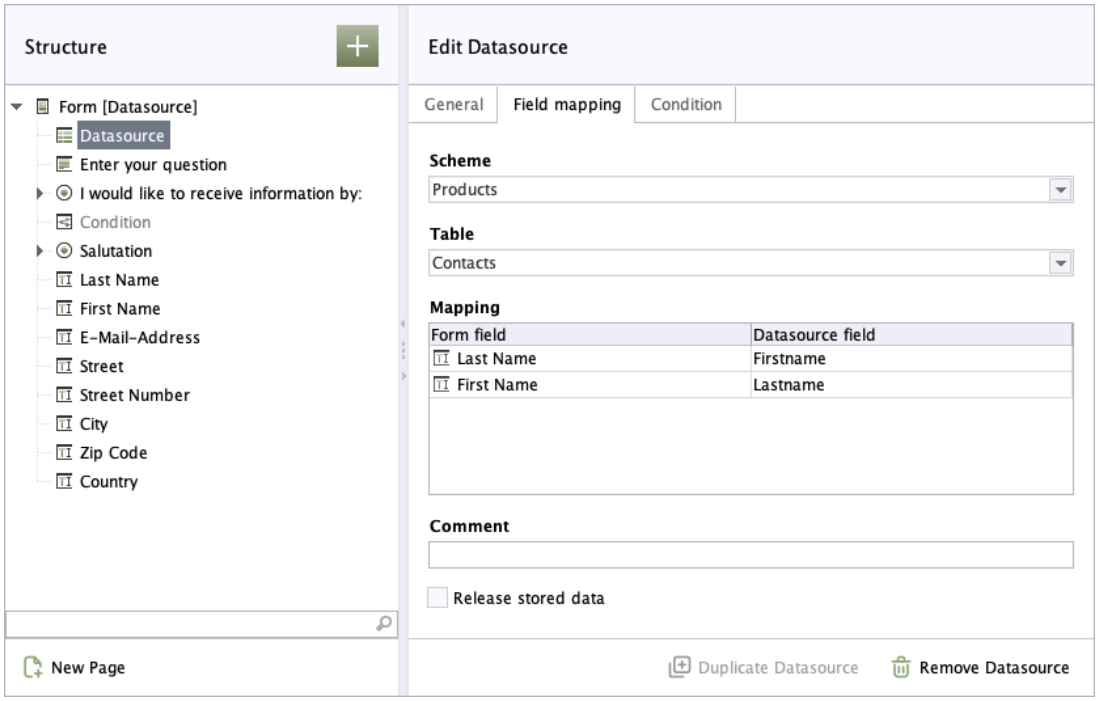


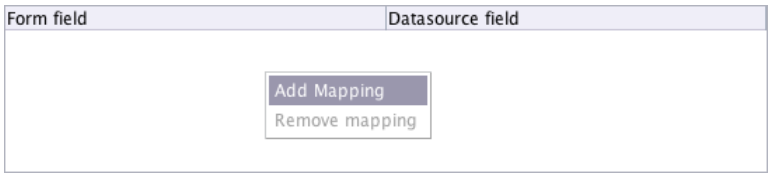
Figure 2.39. Data source

Schema: The schema in which the data source table is located.

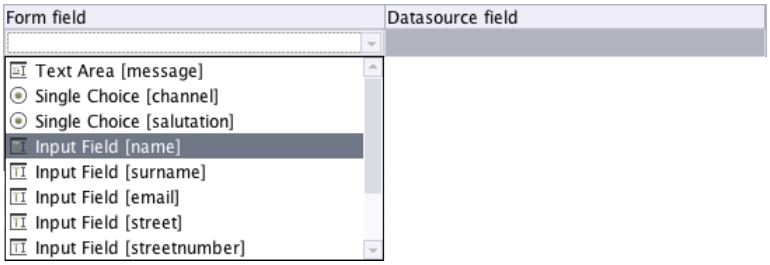
Table: The data source table.

Assignment: The list of assignments between the form field and the data source.

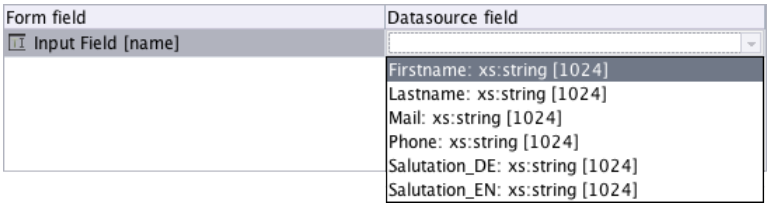
To add a new assignment to the list, right-click with the mouse on the assignment field.



If you double-click the row added, a drop-down menu is shown. You can select the input element here that you wish to include in the list.



In the *Data source field* column, you can select a field from the data source into which the value from the form should be saved. The select field displays only the kinds of data source fields that are compatible with the type of the form field selected in the first column.



Comment: The comment that should be added to the data record when it is saved. This comment is then visible in the record's version history.

Confirmation: The confirmation message that is shown to the user on the web page after the form has been submitted. The confirmation text can also be formatted with the help of markdown (see Section 2.6, “Markdown”).

Release stored data: If you check this box, the stored data record is automatically released in the FirstSpirit server.

Duplicating a data source: Click *Duplicate data source* to add a copy of this action to your form.

Deleting a data source: Click *Delete data source* to delete this action.

2.4.4. Media Management

This action stores uploaded files in the FirstSpirit Media Management.

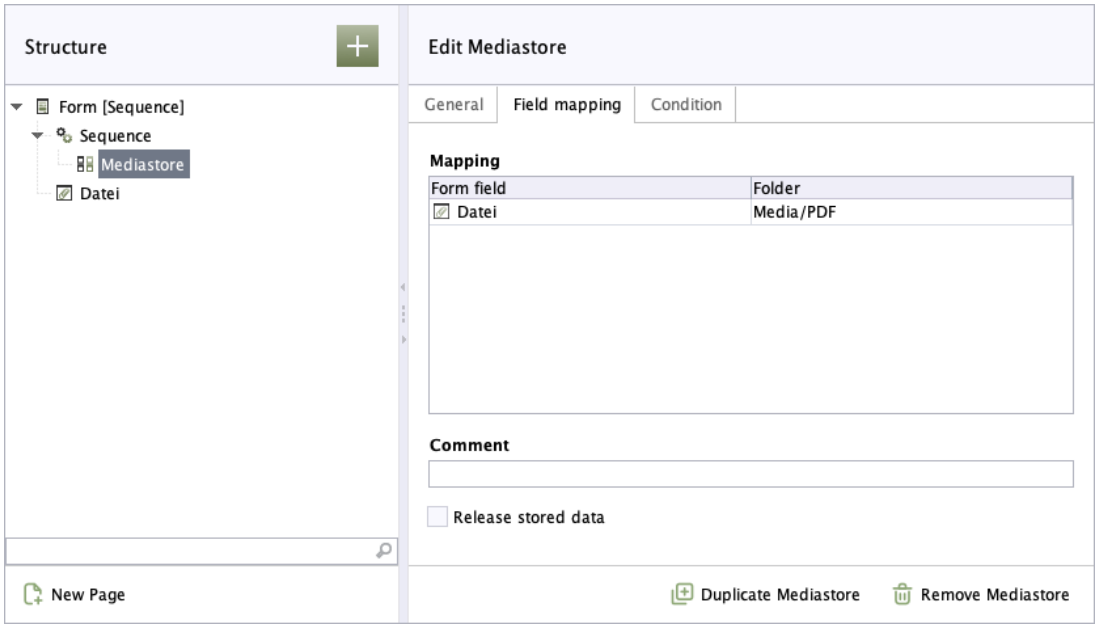
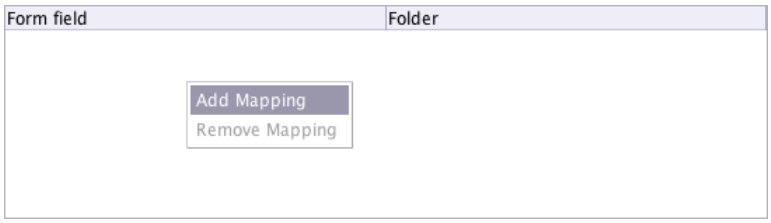


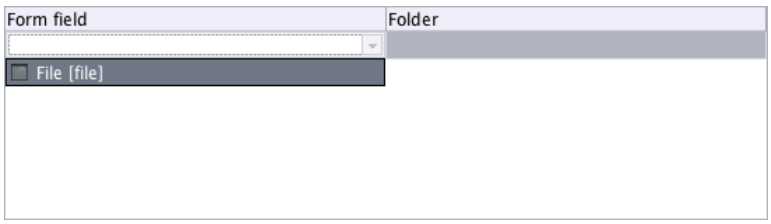
Figure 2.40. Media Management

Assignments: The list of assignments between the file form field and a directory within Media Management.

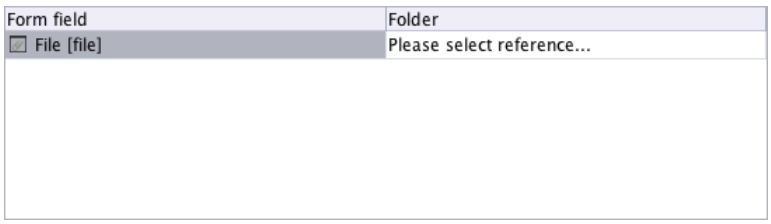
To add a new assignment to the list, right-click with the mouse on the assignment field.



If you double-click the row added, a drop-down menu is shown. You can select the file input element here that you wish to include in the list.



In the *Directory* column, you can select a directory from Media Management into which the uploaded file from the form should be saved. Double-click the column to open up a selection box.



Comment: The comment that should be added to the file when it is saved. This comment is then visible in the version history and the media record description.

Confirmation: The confirmation message that is shown to the user on the web page after the form has been submitted. The confirmation text can also be formatted with the help of markdown (see Section 2.6, “Markdown”).

Release stored data: If you check this box, the stored media record is automatically released in the FirstSpirit server.

Duplicating Media Management: Click *Duplicate Media Management* to add a copy of this action to your form.

Deleting Media Management: Click *Delete Media Management* to delete this action.

2.4.5. Formcentric Analytics

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

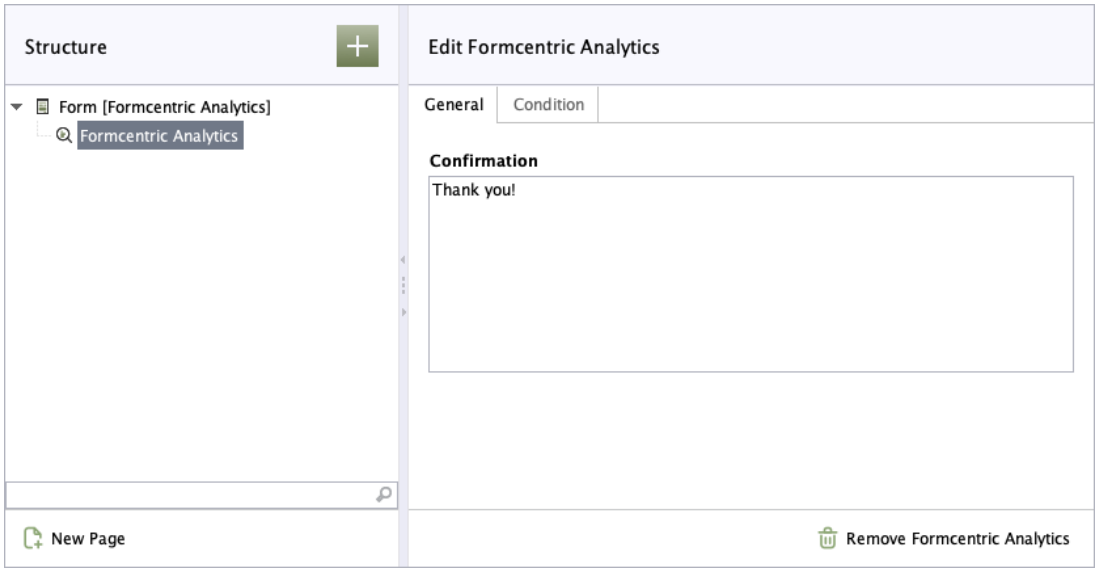


Figure 2.41. Formcentric Analytics

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 2.6, “Markdown” to format the text.

Duplicating Formcentric Analytics: Click *Duplicate Formcentric Analytics* to add a copy of this action to your form.

Deleting Formcentric Analytics: Click *Delete Formcentric Analytics* to delete this action.

2.4.6. 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 FirstSpirit page.

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

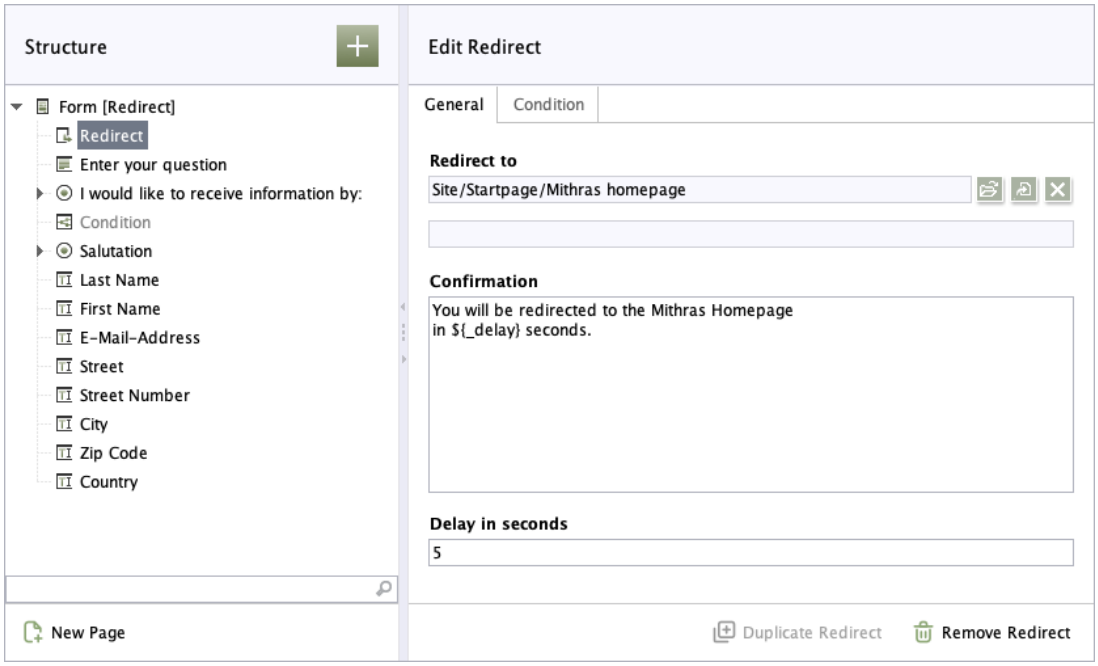


Figure 2.42. Forward to

Forward to: If you want to redirect the user to a FirstSpirit page after submitting the form, then you can select the page here from the FirstSpirit content management system. Alternatively, you can specify an external URL.

As illustrated by the example shown below, you can also use Formcentric variables in the external URL (see Section 2.3, “Variables”) as well as form values.

```
https://www.mydomain.com/site/${country}
```

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.

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

Duplicating a forward to: Click *Duplicate forward to* to add a copy of this action to your form.

Deleting a forward to: Click *Delete forward to* to delete this action.

2.4.7. 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.

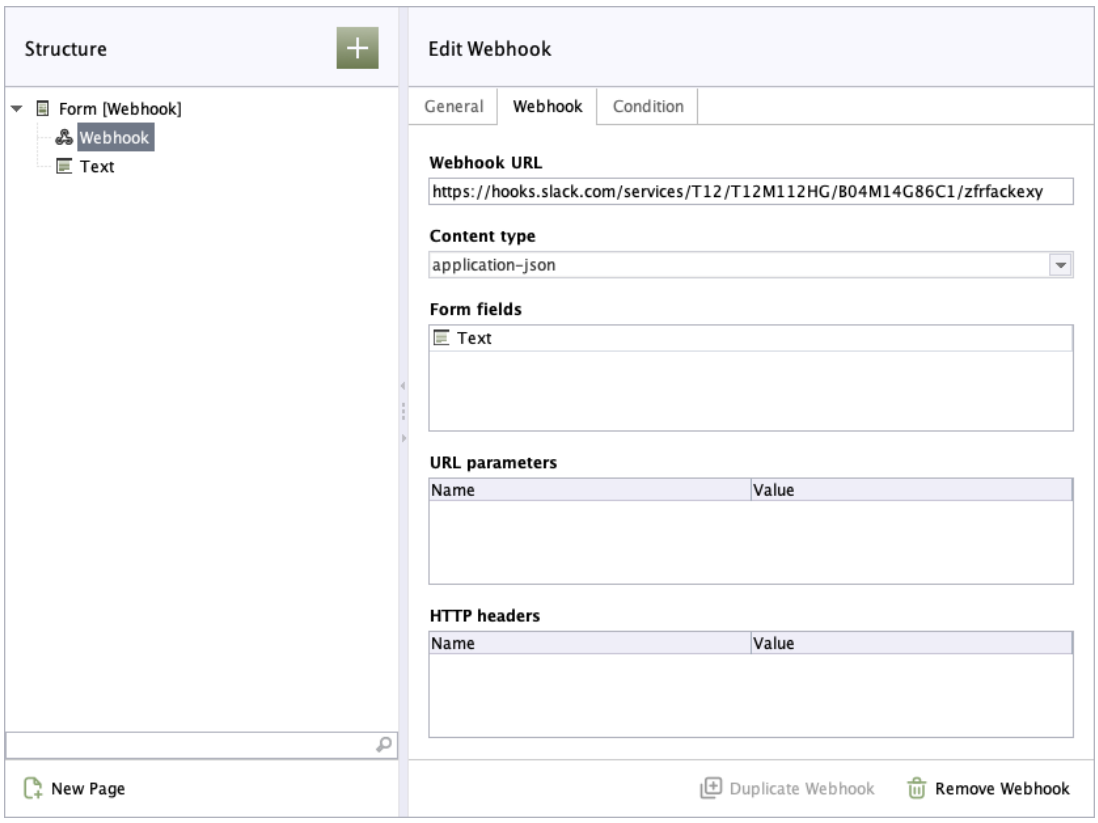


Figure 2.43. Webhook – “Webhook” tab

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. This restriction can be lifted on the server side by your system administrator for individual target addresses.

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.

Format	Description
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.

Form fields: 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 `*${fieldName}*` or `*${variableName}*`.

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.

Duplicating a webhook: Click *Duplicate webhook* to add a copy of this action to your form.

Deleting a webhook: Click *Delete webhook* to delete this action.

2.4.8. Sequence

You use the sequence action to consolidate multiple actions into a single sequence. The individual actions are then executed in the order in which they are assigned below the sequence action in the tree view. If an error occurs while the sequence is being processed, then the sequence stops at the action in which the error occurred.

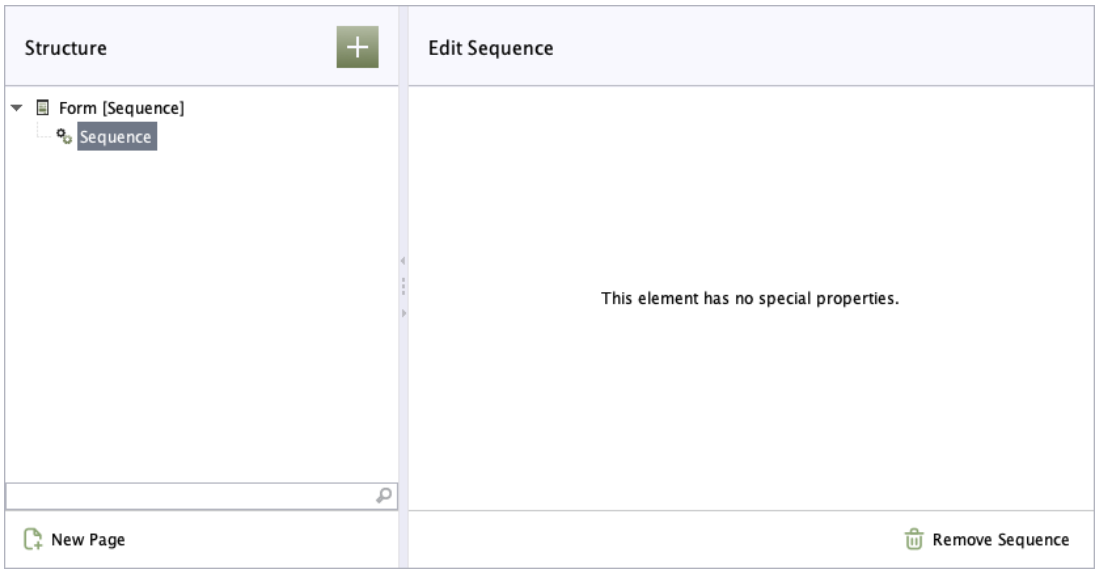


Figure 2.44. Sequence

Deleting a sequence: Click *Delete sequence* to delete this action and the actions arranged below this action.

2.5. 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.

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.

Please note: these data sources are not FirstSpirit data sources but web services.

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

2.5.1. 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.

Parameter name	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: • *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

Parameter name	Description
	• *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

2.5.2. Weekdays

This data source creates a list of the days of the week. You can use the *first* parameter to specify which weekday appears first in the list.

Parameter name	Description
first	• *mon* – Monday • *tue* – Tuesday • *wed* – Wednesday • *thu* – Thursday • *fri* – Friday • *sat* – Saturday • *sun* – Sunday

2.5.3. 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).

2.5.4. Year numbers

This data source creates a list of year numbers. Years must not be more than 100 years apart. The keys *from* and *to* can be used to limit the period for the list of results.

2.6. 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/.

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, hasownproperty, 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, formcentric_redirect, while, with, yield