



**BUILDING BLOCKS / UI Designer / UI component types / Form elements**

# Contents

- BUILDING BLOCKS / UI Designer / UI component types / Form elements / Input
  - Configuring the input element
    - Input settings
    - Input styling
- BUILDING BLOCKS / UI Designer / UI component types / Form elements / Text area
  - Configuring the text area element
    - Text area settings
    - Text area styling
- BUILDING BLOCKS / UI Designer / UI component types / Form elements / Select
  - Configuring the Select element
    - Select Settings
    - Select styling
  - Example - Dynamic dropdowns
    - Creating the process
    - Configuring the nodes
    - Configuring the UI
- BUILDING BLOCKS / UI Designer / UI component types / Form elements / Checkbox
  - Configuring the checkbox element
    - Checkbox settings
    - Checkbox styling
- BUILDING BLOCKS / UI Designer / UI component types / Form elements / Radio
  - Configuring the radio field element

- Radio settings
- Radio styling
- BUILDING BLOCKS / UI Designer / UI component types / Form elements / Switch
  - Configuring the switch element
    - Switch settings
- BUILDING BLOCKS / UI Designer / UI component types / Form elements / Datepicker
  - Configuring the datepicker element
    - Datepicker settings
    - Datepicker styling
- BUILDING BLOCKS / UI Designer / UI component types / Form elements / Slider
  - Configuring the slider element
    - Slider settings
    - Multiple sliders
    - Slider styling
- BUILDING BLOCKS / UI Designer / UI component types / Form elements / Segmented button
  - Configuring the segmented button
    - Segmented button settings
    - Segmented button styling

# BUILDING BLOCKS / UI Designer / UI component types / Form elements / Input

The screenshot shows the FLOWX.AI UI Designer interface. On the left is a sidebar with categories like Layout, Collection, and Basic, each containing various UI component icons. The main area displays a form titled "First Form Title" with a subtitle "First Form subtitle". The form contains several input fields: "Input 1" (a simple text input), "Input 3" (a numeric input with a placeholder ".000"), and a "Gender" dropdown. Below these are two radio buttons for "Contact via Email" and "Contact via SMS". There is also a date input field labeled "Birthday" and a toggle switch labeled "Does the client have a house?". To the right of the form is a detailed configuration panel for an "Input" component. It shows the process data key as "application.input2", properties for "Label" (set to "Income") and "Placeholder" (set to "Income"), and options for "Type" (set to "Select a Type"), "Prefix" (\$), and "Suffix" (eg. suffix). A "Helpertext" field contains "Additional information" and a checked "Hide inside infopoint" checkbox.

An input field is a form element that enables users to input data with validations and can be hidden or disabled.

## Configuring the input element

### Input settings

The Input Field offers the following configuration options:

- General
- Properties
- Datasource
- Validators
- Expressions
- UI actions
- Input styling

## General

- **Process data key** - creates the binding between form element and process data, so it can be later used in **decisions**, **business rules** or **integrations**

## Properties

- **Label** - the label that appears on the input field
- **Placeholder** - the placeholder text that appears in the input field when it is empty
- **Type** - the type of data that the input field can accept, such as text, number, email, or password
- **Prefix** - a label that appears as a prefix to the input field
- **Suffix** - a label that appears as a suffix to the input field
- **Helpertext** - additional information about the input field (can be hidden inside an infopoint)

The screenshot shows the FLOWX.AI UI Designer interface. On the left, there's a sidebar with icons for Image, TEXT, LINK, Forms (Form, Input), Textarea, Select, Checkbox, Radio, Switch, and Datepicker. The main area displays a form titled "First Form Title" with a subtitle "First Form subtitle". The form contains several input fields: "Input 1" (Text), "input" (Text with placeholder "\$ Income" and helpertext "This is a helper text"), "Input 3" (Text with suffix ".000"), "Gender" (Select), "Contact via Email" and "Contact via SMS" (Radio buttons), "Birthday" (Datepicker), and a "Does the client have a house?" section with a Switch button and a checkbox. To the right, there's a panel for "Input" with "Process data key" set to "application.input2", and another panel for "Properties" where "Label" is "Income", "Placeholder" is "Income", "Type" is "number", "Prefix" is "\$", "Suffix" is "eg. suffix", "Helpertext" is "This is a helper text", and a checkbox for "Hide inside infopoint" is checked.

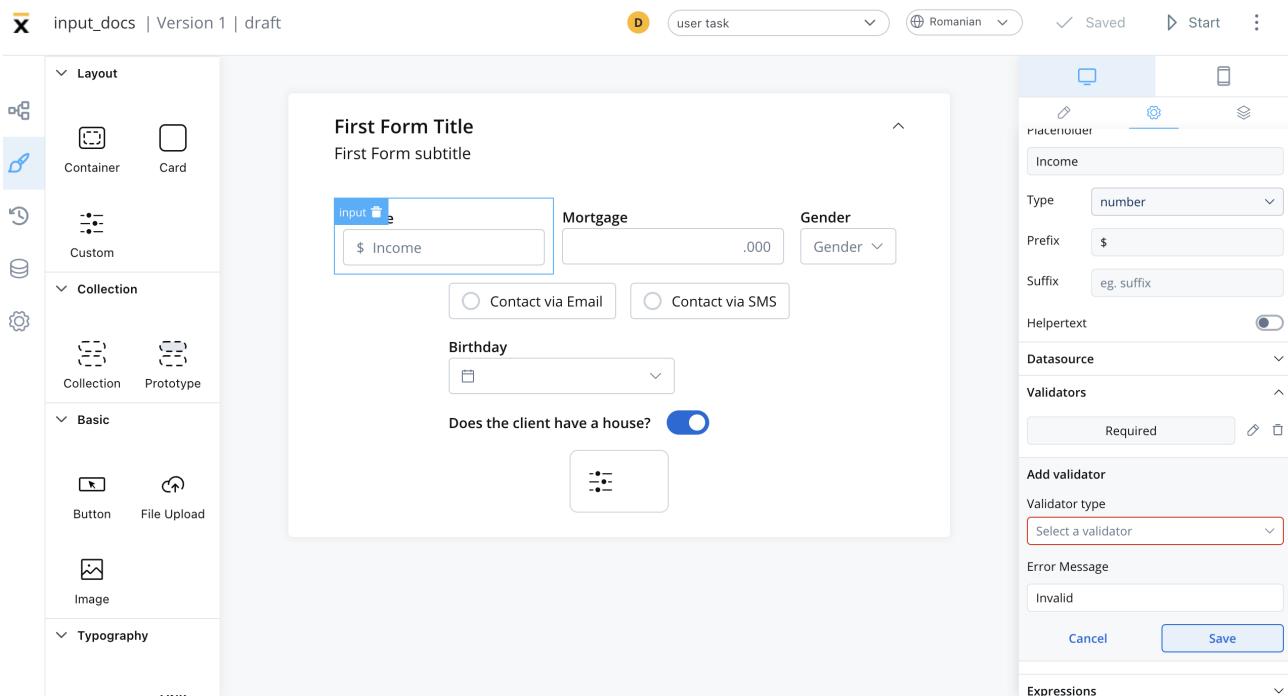
## Datasource

The default value for the element can be configured here, this will autofill the input field when you will run the process.

The screenshot shows the FLOWX.AI UI Designer interface. On the left, there is a sidebar with various components: Image, Typography (Text, Link), Forms (Form, Input), Textarea, Select, Checkbox, and Radio. The main area displays a form titled "First Form Title" with a subtitle "First Form subtitle". The form contains several input fields: "Input 1" (empty), "Income" (with a value of "\$ 555" and a helper text "This is a helper text"), "Mortgage" (empty), "Gender" (dropdown menu), "Contact via Email" (radio button), "Contact via SMS" (radio button), "Birthday" (date picker), and a toggle switch labeled "Does the client have a house?". To the right of the form, there is a properties panel for the "Input" component, showing settings like "Process data key: application.input2", "Default value: 555", and "UI Action". Below the designer, a preview window shows the final form with the same layout and data. The preview window has a header with "First Form Title" and "First Form subtitle". It includes the "Income" field with the value "\$ 555" and the helper text "This is a helper text", the "Contact via Email" and "Contact via SMS" radio buttons, the "Birthday" date picker, and the toggle switch for "Does the client have a house?" which is currently turned on.

## Validators

There are multiple validators can be added to an input (more details [here](#)).



## Expressions

The input field's behavior can be defined using JavaScript expressions for hiding or disabling the element. The following properties can be configured for expressions:

- **Hide** - JavaScript expression used to hide the Input Field when it returns a truthy value
- **Disabled** - JavaScript expression used to disable the Input Field when it returns a truthy value

### ! INFO

It's important to make sure that disabled fields have the same expression configured under the path expressions → hide.

The screenshot shows the FLOWX.AI UI Designer interface. On the left, there's a sidebar with categories like Layout, Collection, and Basic, each containing icons for Container, Card, Custom, Collection, Prototype, Button, File Upload, and Image. The main area displays a form titled "First Form Title" with a subtitle "First Form subtitle". The form contains several input fields: an "Income" input field with a placeholder "\$ Income", a "Mortgage" input field with a placeholder ".000", a "Gender" dropdown menu, and two radio buttons for "Contact via Email" and "Contact via SMS". Below these are a date input field labeled "Birthday" and a toggle switch labeled "Does the client have a house?". To the right of the form is a detailed configuration panel for the "Income" field. It includes sections for Placeholder (Income), Type (number), Prefix (\$), Suffix (eg. suffix), Helpertext, Datasource, Validators, and Expressions. The "Expressions" section is highlighted with a red box and contains the expression \${application.key}!=='TEST'. There are also sections for Hide and Disabled.

## UI actions

UI actions can be added to the Input Field to define its behavior and interactions.

- **Event** - possible value: CHANGE
- **Action Type** - select the action type

The screenshot shows the FLOWX.AI UI Designer interface. On the left, there's a sidebar with categories like Layout, Collection, and Basic, each containing icons for Container, Card, Custom, Collection, Prototype, Button, File Upload, and Image. The main area displays a form titled "First Form Title" with a subtitle "First Form subtitle". The form contains several input fields: a text input with a dollar sign prefix and a file icon, a text input for "Mortgage", a dropdown for "Gender", two radio buttons for "Contact via Email" and "Contact via SMS", a date input for "Birthday", and a toggle switch for "Does the client have a house?". To the right of the form is a detailed configuration panel for the first input field. It includes tabs for desktop and mobile views, settings for placeholder text ("Income"), type ("number"), prefix ("\$"), suffix ("eg. suffix"), help text, data source, validators, expressions, hide logic (\${application.key} != 'TEST'), disabled status, and UI actions. There are also icons for edit, settings, and copy.

## (!) INFO

For more details on how to configure a UI action, click [here](#).

## Input styling

### Icons

- **Icon Key** - the key associated in the Media library, select the icon from the **Media Library**
- **Icon Color** - select the color of the icon using the color picker

## (!) INFO

When setting the color, the entire icon is filled with that color, the SVG's fill. Avoid changing colors for multicolor icons.

You have the option to enhance the Input element by incorporating two types of icons:

- **Left Icon:** You can include an icon on the left side of the Input element. This icon can serve as a visual cue or symbol associated with the input field's purpose or content.
- **Right Icon:** Same as left icon.

By utilizing these two types of icons, you can provide users with a more intuitive and visually appealing experience when interacting with the Input element.

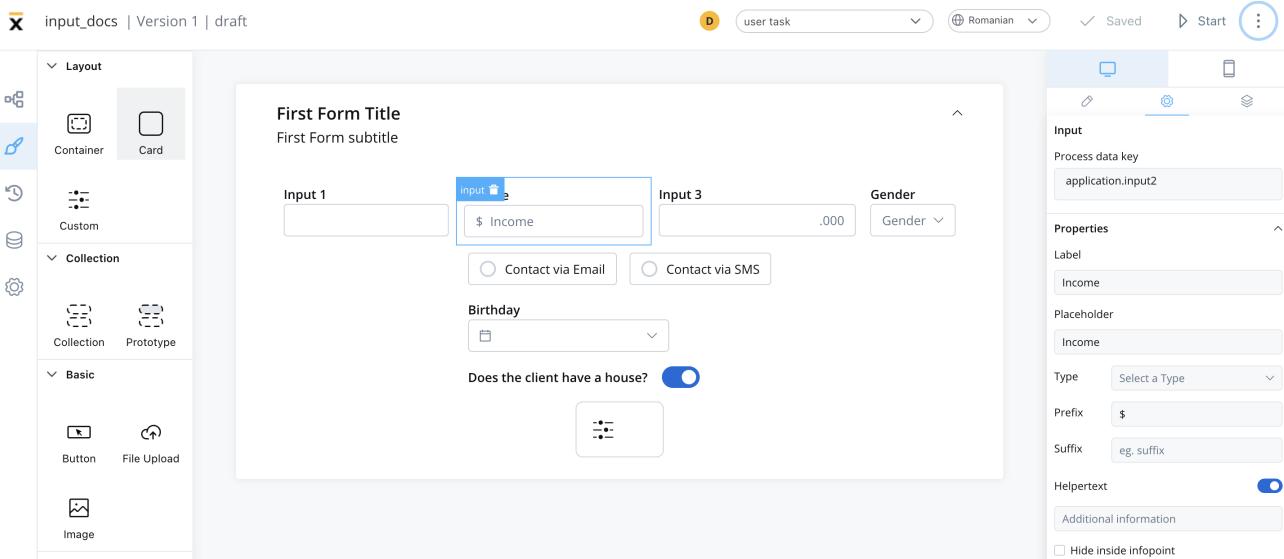
The screenshot shows a user interface for 'Enter Personal Information'. On the left, there's a form with fields for First Name, Last Name, Date of birth, Employment type (radio buttons for Employed and Pensioner), a Save personal information toggle, and a Loan amount slider set at 255000 \$. On the right, a sidebar provides detailed component settings for the 'Input' element, including options for Left Icon and Right Icon, both of which are currently enabled. The sidebar also includes sections for Properties, Sizing, Spacing, and Typography.

- The Input Field can be styled using valid CSS properties (more details [here](#))

The screenshot shows the FLOWX.AI UI Designer interface. On the left, there's a sidebar with categories: Layout (Container, Card, Custom), Collection (Collection, Prototype), and Basic (Button, File Upload, Image). The main area displays a form titled "First Form Title" with a subtitle "First Form subtitle". The form contains several input fields: a text input for "Income" with a dollar sign prefix, a text input for "Mortgage" with a ".000" suffix, a dropdown for "Gender", two radio buttons for "Contact via Email" and "Contact via SMS", a date input for "Birthday", and a toggle switch for "Does the client have a house?". To the right of the form is a panel with tabs for desktop and mobile. The "Input" tab is selected, showing settings for "Process data key" (application.input2), "Label" (Income), "Placeholder" (Income), "Type" (number), "Prefix" (\$), "Suffix" (eg. suffix), "Helpertext" (with a toggle switch), "Datasource" (dropdown), and "Validators" (dropdown). There are also tabs for "Properties" and "Behaviors".

Was this page helpful?

# BUILDING BLOCKS / UI Designer / UI component types / Form elements / Text area



A text area is a form element used to capture multi-line input from users in a conversational interface. The text area component is typically used for longer inputs such as descriptions, comments, or feedback, providing users with more space to type their responses.

It is an important tool for creating intuitive and effective conversational interfaces that can collect and process large amounts of user input.

## Configuring the text area element

### Text area settings

The text area offers the following configuration options:

- General
- Properties
- Datasource
- Validators

- **Expressions**
- **UI actions**
- **Text area styling**

## General

- **Process data key** - creates the binding between form element and process data, so it can be later used in [decisions](#), [business rules](#) or [integrations](#)

## Properties

- **Label** - the label of the text area
- **Placeholder** - the placeholder text that appears in the text area
- **Helpertext** - additional information about the text area field (can be hidden inside an infopoint)

## Datasource

The default value for the element can be configured here, this will autofill the text field when you will run the process.

## Validators

There are multiple validators can be added to a text area element (more details [here](#)).

The screenshot shows the FLOWX.AI UI Designer interface. On the left, there's a sidebar with various component categories like TEXT, LINK, FORMS, INDICATORS, and a MESSAGE component. The FORMS section is expanded, showing options for Form, Input, Textarea, Select, Checkbox, Radio, Switch, and Datepicker. The INDICATORS section is also expanded, showing a Message component. In the center, there's a preview of a form with fields for Customer Name, Income, Gender, Contact via Email/SMS, Select a date (Date of birth dropdown), a newsletter subscription toggle, and a large Textarea field containing placeholder text 'Some text here'. A 'Submit' button is at the bottom. On the right, there's a detailed configuration panel for the Textarea component. It includes sections for 'Process data key' (set to 'textKey'), 'Properties' (Label: 'Input label', Placeholder: 'Some text here', Helpertext: checked), 'Datasource' (Default value: 'eg. Name'), and 'Validators' (a button to 'Add a validator').

## Expressions

The text area's behavior can be defined using JavaScript expressions for hiding or disabling the element. The following properties can be configured for expressions:

- **Hide** - JavaScript expression used to hide the text area when it returns a truthy value
- **Disabled** - JavaScript expression used to disable the text area when it returns a truthy value

### INFO

It's important to make sure that disabled fields have the same expression configured under the path expressions → hide.

The screenshot shows a user interface for building forms. On the left, a sidebar lists components: Layout (Container, Card, Custom), Collection (Collection, Prototype), and Basic (Button, File Upload). The main area displays a form titled "First Form Title" with a subtitle "First Form subtitle". The form includes fields for "Income" (number type, value: \$ Income), "Mortgage" (number type, value: .000), "Gender" (dropdown), and two radio buttons for "Contact via Email" and "Contact via SMS". There is also a date input field labeled "Birthday". A toggle switch is labeled "Does the client have a house?". On the right, there are several configuration panels: Placeholder (Income), Type (number), Prefix (\$), Suffix (eg. suffix), Helpertext (checkbox), Datasource (dropdown), Validators (dropdown), and Expressions. The Expressions panel contains the code \${application.key}!='TEST'. The Disabled section is empty.

## UI actions

UI actions can be added to the text area field to define its behavior and interactions.

- **Event** - possible value: CHANGE
- **Action Type** - select the action type

The screenshot shows the FLOWX.AI UI Designer interface. On the left, there's a sidebar with categories like TEXT, LINK, FORMS, INDICATORS, and a MESSAGE icon. The FORMS section is expanded, showing options for Form, Input, Textarea, Select, Checkbox, Radio, Switch, and Datepicker. The INDICATORS section is also expanded, showing a Message icon. The main area displays a form titled "Form title". It includes fields for Customer Name, Income, Gender, and contact preferences (Email or SMS). Below these are sections for selecting a date (Date of birth dropdown) and subscribing to a newsletter (checkbox). A large text area labeled "Some text here" is also present. At the bottom is a "Submit" button. To the right of the form, there are several configuration panels: Helpertext (with a toggle switch), Datasource, Default value (set to "eg. Name"), Validators (with a "Add a validator" button), Expressions (with a "Hide" condition: `\${textKey} !== 'TEST'`), Disabled (set to "-"), and UI Action (with a "Add UI action" button). The top right of the interface shows language settings (Romanian), a save status (Saved), and navigation links (Start, More).

## INFO

For more details on how to configure a UI action, click [here](#).

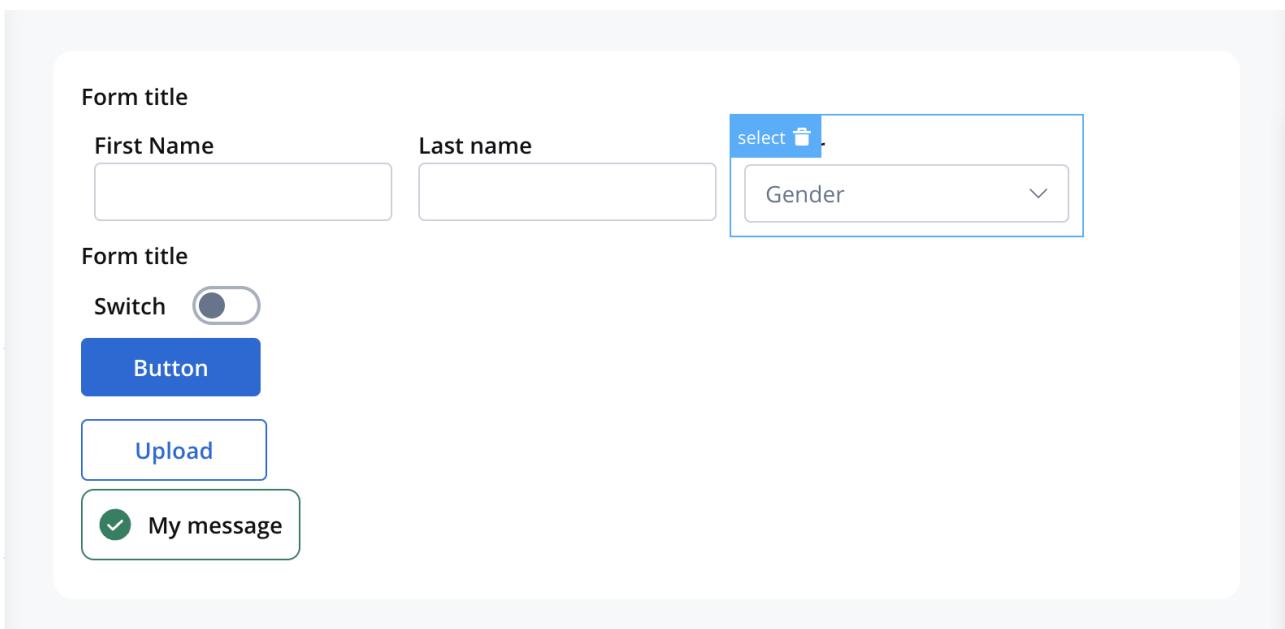
## Text area styling

The ability to style the text area element using CSS properties is relevant because it allows you to customize the appearance of the text area to match the overall design of the website or application.

» [UI Designer styling](#)

Was this page helpful?

# BUILDING BLOCKS / UI Designer / UI component types / Form elements / Select



The Select form field is an element that enables users to make a choice from a list of predefined options. It consists of multiple values, each of which is defined by a label that is displayed in the dropdown menu, and a code that is saved.

## (!) INFO

For instance, you could have a label of "Female" with the value "F" and "Male" with the value "M". This means that when a user selects "Female" in the process instance, the value "F" will be stored for the "Select" key.

## Configuring the Select element

## Select Settings

These allow you to customize the settings for the Select Field:

- **General**
- **Properties**
- **Datasource**
- **Validators**
- **Expressions**
- **UI actions**
- **Select styling**

### General

- **Process data key** - creates the binding between form element and process data so it can be later used in **decisions**, **business rules** or **integrations**

### Properties

- **Label** - the label of the select
- **Placeholder** - placeholder when the field has no value
- **Empty message** - text displayed for custom type when no results are found
- **Search for options** - displays a search to filter options
- **Helpertext** - additional information about the select field (can be hidden inside an infopoint)

### Datasource

- **Default value** - autofill the select with this value. Going back to the example with Woman label with F value and Man with M to have a default value of Woman we need to configure here F
- **Source Type** - it can be Static, Enumeration, or Process Data
- **Add option** - label - value pairs can be defined here

## Validators

There are multiple validators can be added to a select (more details [here](#)).

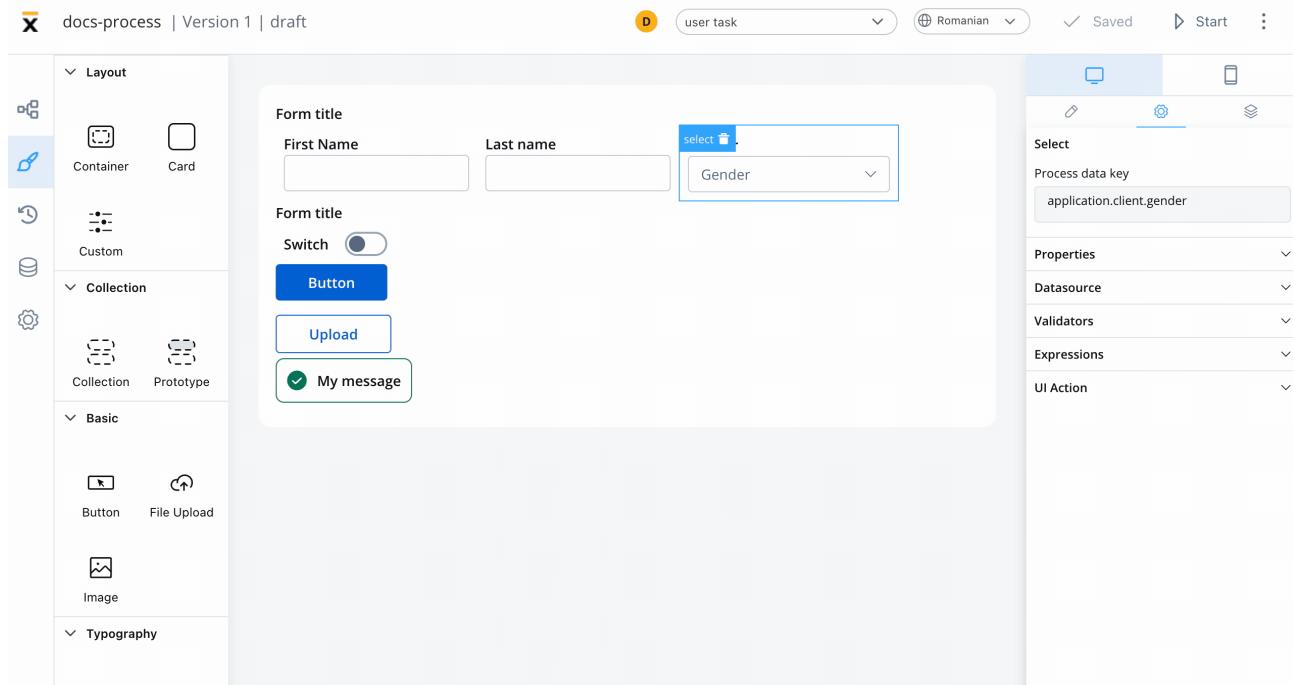
## Expressions

The select field's behavior can be defined using JavaScript expressions for hiding or disabling the element. The following properties can be configured for expressions:

- **Hide** - JavaScript expression used to hide the Select Field when it returns a truthy value
- **Disabled** - JavaScript expression used to disable the Select Field when it returns a truthy value

### (!) INFO

It's important to make sure that disabled fields have the same expression configured under the path expressions → hide.



## UI actions

UI actions can be added to the select element to define its behavior and interactions.

- **Event** - possible value: `CHANGE`
- **Action Type** - select the action type

### !(INFO)

For more details on how to configure a UI action, click [here](#).

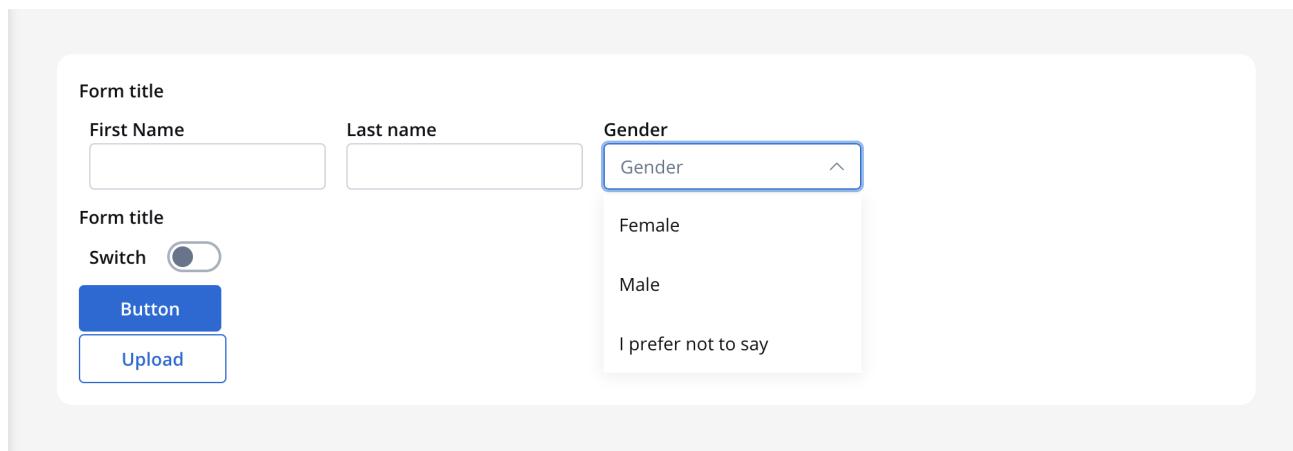
## Select styling

Styling the Select field using CSS properties allows you to customize the appearance of the dropdown list and make it more visually appealing and

consistent with the overall design of the website or application.

### » UI Designer styling

For example, a FORM element with a **layout** configuration including direction of Horizontal and some inputs, and a select element will look like this:



## Icons

When customizing the appearance of a Select UI element that includes an icon, you can utilize the following properties:

- **Icon Key** - the key associated in the Media library, select the icon from the **Media Library**
- **Icon Color** - select the color of the icon using the color picker

### (!) INFO

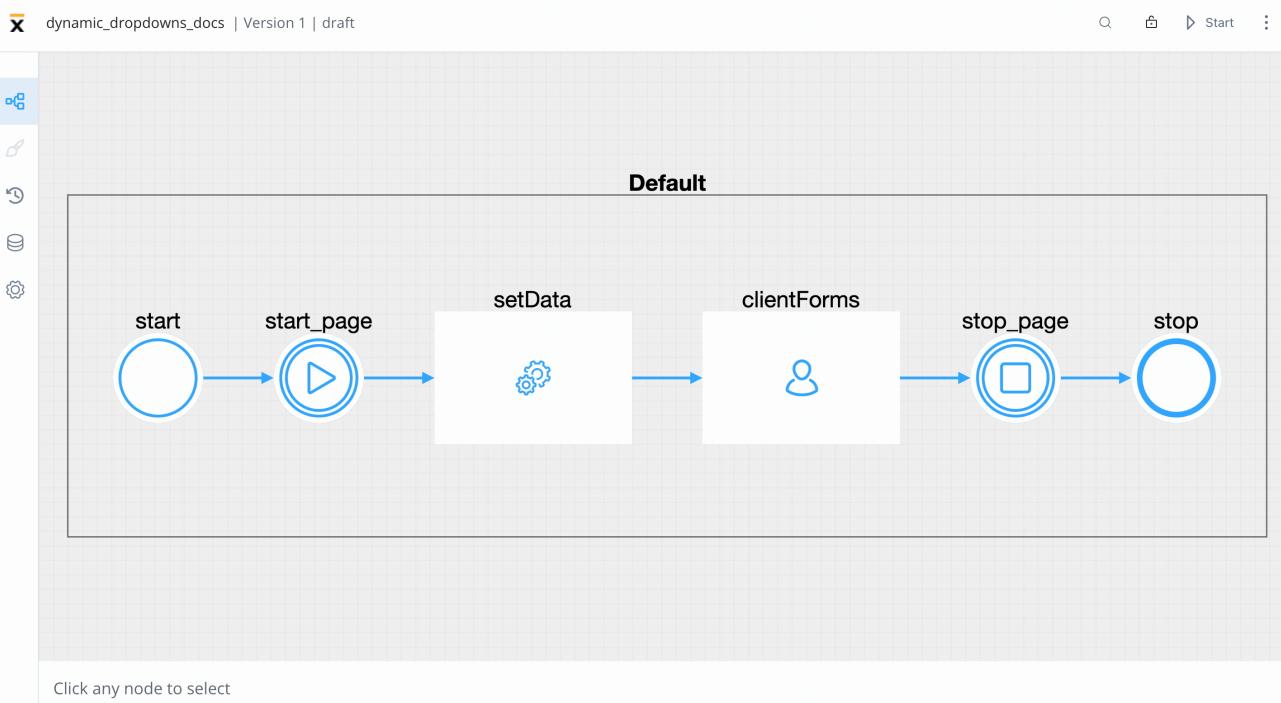
When setting the color, the entire icon is filled with that color, the SVG's fill. Avoid changing colors for multicolor icons.

## Example - Dynamic dropdowns

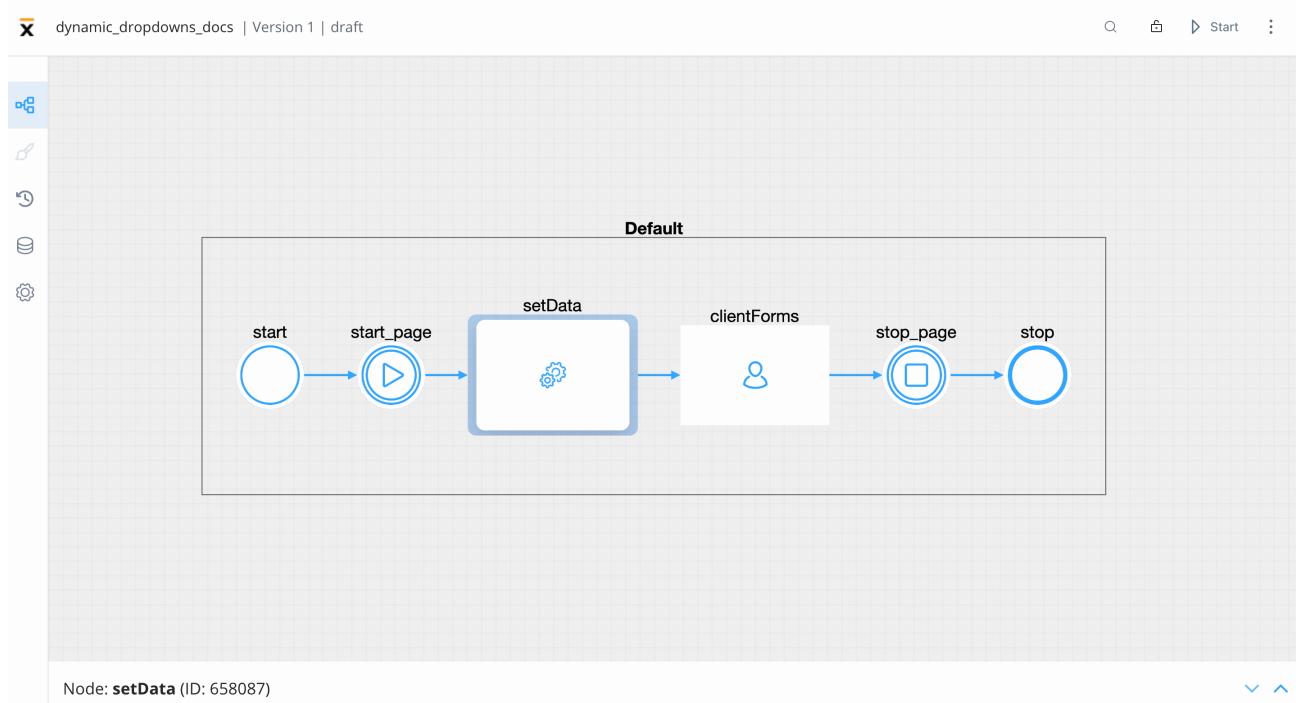
As mentioned previously, you can create dropdowns including static data, enumerations, or **process data**. Let's create an example using **process data** to create a process that contains **dynamic dropdowns**.

To create this kind of process, we need the following elements:

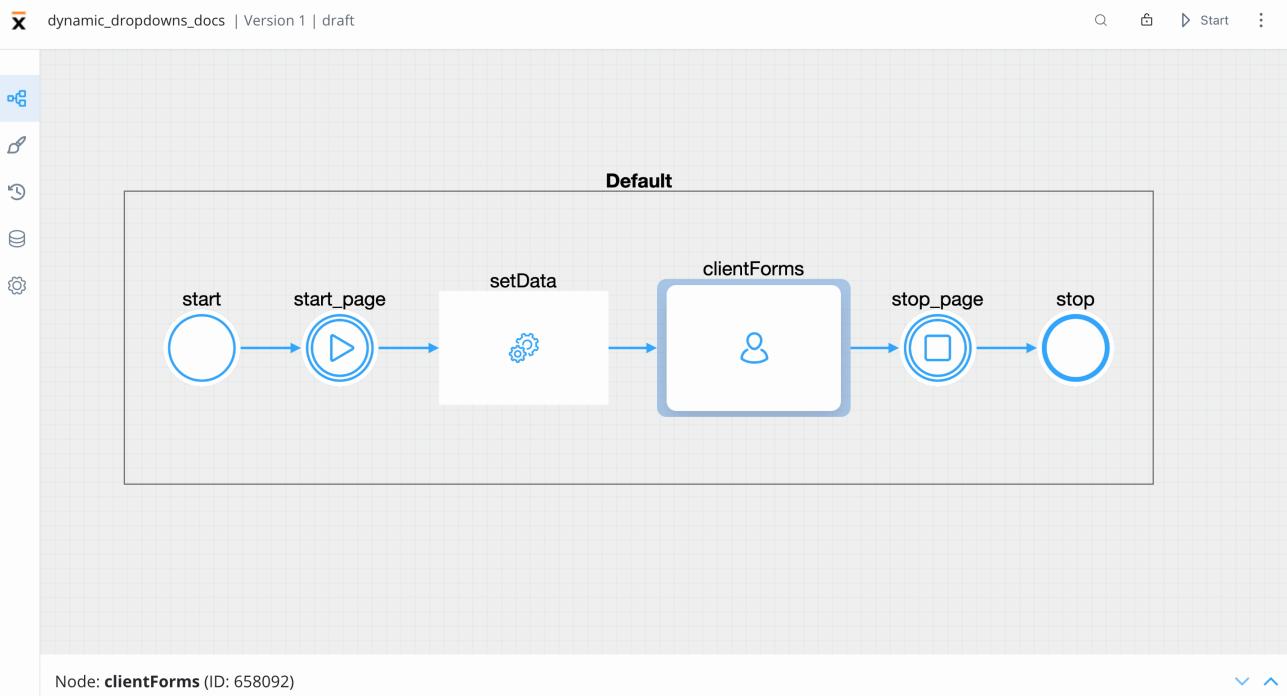
- a **start** node and an **end** node
- a **start milestone** UI element to it and an **end milestone** node



- a **task node** (this will be used to set which data will be displayed on the dropdowns)



- a **user task node** (here we have the client forms and here we add the SELECT elements)



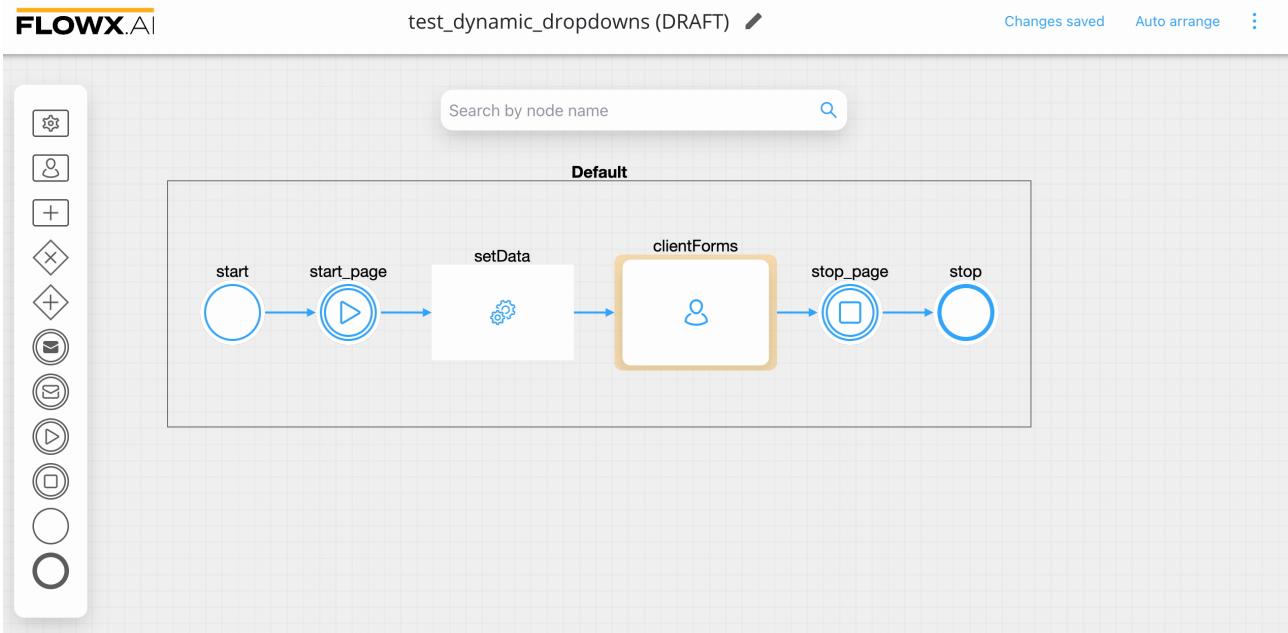
## Creating the process

Follow the next steps to create the process from scratch:

1. Open **FLOWX Designer** and from the **Processes** tab select **Definitions**.
2. Click on the breadcrumbs (top-right corner) then click **New process** (the Process Designer will now open).
3. Now add all the **necessary nodes** (as mentioned above).

## Configuring the nodes

1. On the **start milestone** node, add a **page** UI element.
2. On the **task node**, add a new **Action** (this will set the data for the dropdowns) with the following properties:
  - Action type - **Business Rule**
  - **Automatic**
  - **Mandatory**
  - **Language** (we used an **MVEL** script to create a list of objects)
3. On the **user task node**, add a new **Action** (submit action, this will validate the forms and save the date) with the following properties:
  - **Action type** - Save Data
  - **Manual**
  - **Mandatory**
  - **Data to send** (the key where the data will be sent) - **application**



Below you can find the MVEL script used in the above example:

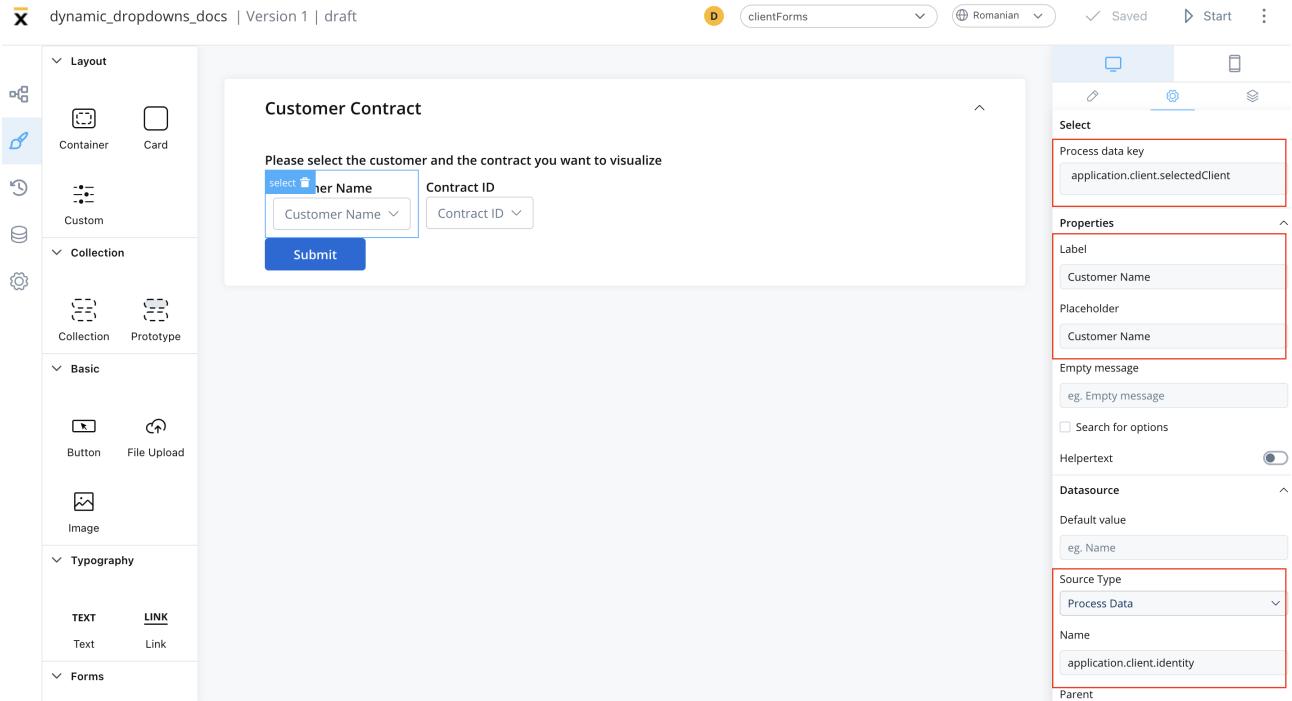
```
output.put("application",
{
    "client": {
        "identity": [
            {
                "value": "001",
                "label": "Eddard Stark"
            },
            {
                "value": "002",
                "label": "Sansa Stark"
            },
            {
                "value": "003",
                "label": "Catelyn Stark"
            }
        ]
    },
}];
```

```
"contracts": {
    "001": [
        {
            "value": "c001",
            "label": "Eddard Contract 1"
        },
        {
            "value": "c007",
            "label": "Eddard Contract 2"
        }
    ],
    "003": [
        {
            "value": "c002",
            "label": "Catelyn Contract 1",
        },
        {
            "value": "c003",
            "label": "Catelyn Contract 2",
        },
        {
            "value": "c004",
            "label": "Catelyn Contract 3"
        }
    ],
    "002": [
        {
            "value": "c005",
            "label": "Sansa Contract 1",
        }
    ]
});
```

## Configuring the UI

Follow the next steps to configure the UI needed:

1. Select the **user task node** and click the **brush icon** to open **UI Designer**
2. Add a **card** element as a **root component** (this will group the other elements inside it) with the following properties:
  - **Message** - `{"application": ${application}}`
  - **Title** - *Customer Contract*
3. Inside the **card**, add a **form element**.
4. Inside the **form** add two **select elements**, first will represent, for example, the *Customer Name* and the second the *Contract ID*.
5. For first select element (Customer Name) set the following properties:
  - **Process data key** - `application.client.selectedClient`
  - **Label** - Customer Name
  - **Placeholder** - Customer Name
  - **Source type** - Process Data (to extract the data added in the **task node**)
  - **Name** - `application.client.identity`



6. For the second select element (Contract ID) set the following properties:

- o **Process data key** - application.client.selectedContract
- o **Label** - Contract ID
- o **Placeholder** - Contract ID
- o **Source Type** - Process Data
- o **Name** - application.contracts
- o **Parent** - SELECT (choose from the dropdown list)

The screenshot shows the FLOWX.AI UI Designer interface. On the left, there's a sidebar with categories like Layout, Collection, Basic, and Typography, each containing various UI components such as Container, Card, Custom, Collection, Prototype, Button, File Upload, and Image. In the center, there's a form titled "Customer Contract" with the following fields:

- A text input field labeled "Customer Name".
- A dropdown menu labeled "select Contract ID" with a placeholder "Contract ID".
- A blue "Submit" button.

On the right, there's a detailed configuration panel for the "Contract ID" dropdown. It's defined as a "Select" type component. The "Source Type" is set to "Process Data", the "Name" is "application.contracts", and the "Parent" is set to "SELECT". The "Select" configuration panel is highlighted with a red box, showing fields for "Process data key" (set to "application.client.selectedContract") and "Properties".

7. Add a button under the form that contains the select elements with the following properties:

- **Label** - Submit
- **Add UI action** - add the submit action attached earlier to the user task node

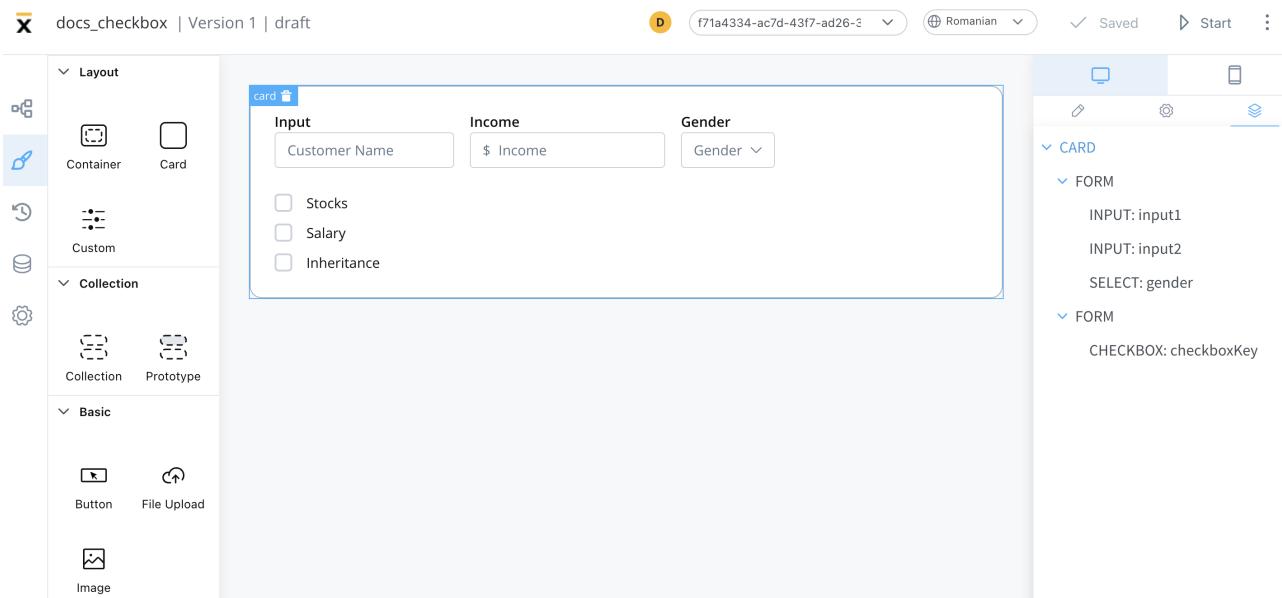
The screenshot shows the FLOWX.AI UI Designer interface. On the left, there's a sidebar with categories like Layout, Collection, Basic, and Typography, each containing various UI components. The main workspace displays a form titled "Customer Contract" with instructions: "Please select the customer and the contract you want to visualize". It contains two dropdown menus labeled "Customer Name" and "Contract ID", and a blue button labeled "mit". To the right, the "Properties" panel is open for a "Button" component. The "Label" field is set to "Submit". Below it, the "UI Action" section is expanded, showing an "Add action" dialog for a "CLICK" event. The "Action Type" is set to "ACTION" and the "Node Action Name" is "submit". There are also checkboxes for "Use a different name for UI action", "Add custom body", "Add form to validate", "Dismiss process?", and "Show loader?".

## 8. Test and run the process by clicking **Start process**.

The screenshot shows the FLOWX.AI process editor. It displays a workflow titled "Default". The process starts at a "start" node, transitions to a "start\_page" node (represented by a play button icon), then to a "setData" node (with a gear icon). From "setData", the flow goes to a "clientForms" node (with a person icon). Finally, it transitions to a "stop\_page" node (represented by a square icon) and ends at a "stop" node. At the bottom of the workspace, there's a note: "Click any node to select".

Was this page helpful?

# BUILDING BLOCKS / UI Designer / UI component types / Form elements / Checkbox



A checkbox form field is an interactive element in a web form that provides users with multiple selectable options. It allows users to choose one or more options from a pre-determined set by simply checking the corresponding checkboxes.

This type of form field can be used to gather information such as interests, preferences, or approvals, and it provides a simple and intuitive way for users to interact with a form.

## Configuring the checkbox element

## Checkbox settings

The available configuration options for this form element are:

- [General](#)
- [Properties](#)
- [Datasource](#)
- [Validators](#)
- [Expressions](#)
- [UI actions](#)
- [Checkbox styling](#)

### General

- **Process data key** - creates the binding between form element and process data, so it can be later used in [decisions](#), [business rules](#) or [integrations](#)

### Properties

- **Label** - the label that appears on the checkbox
- **Helpertext** - additional information about the checkbox (can be hidden inside an infopoint)

The screenshot shows the FLOWX.AI UI Designer interface. On the left is a sidebar with categories like Layout, Collection, Basic, and others, each containing icons for different components. The main area displays a form with three input fields: 'Customer Name' (Input), '\$ Income' (Input), and 'Gender' (Select). Below these is a section titled 'checkbox' containing three checkboxes labeled 'Stocks', 'Salary', and 'Inheritance'. To the right of the form is a panel for configuring the 'Checkbox' component. It includes fields for 'Process data key' (checkboxKey), 'Label' (eg. Name), 'Helpertext' (Select an income source), and a checked checkbox for 'Hide inside infopoint'. There are also dropdowns for 'Datasource', 'Validators', 'Expressions', and 'UI Action'.

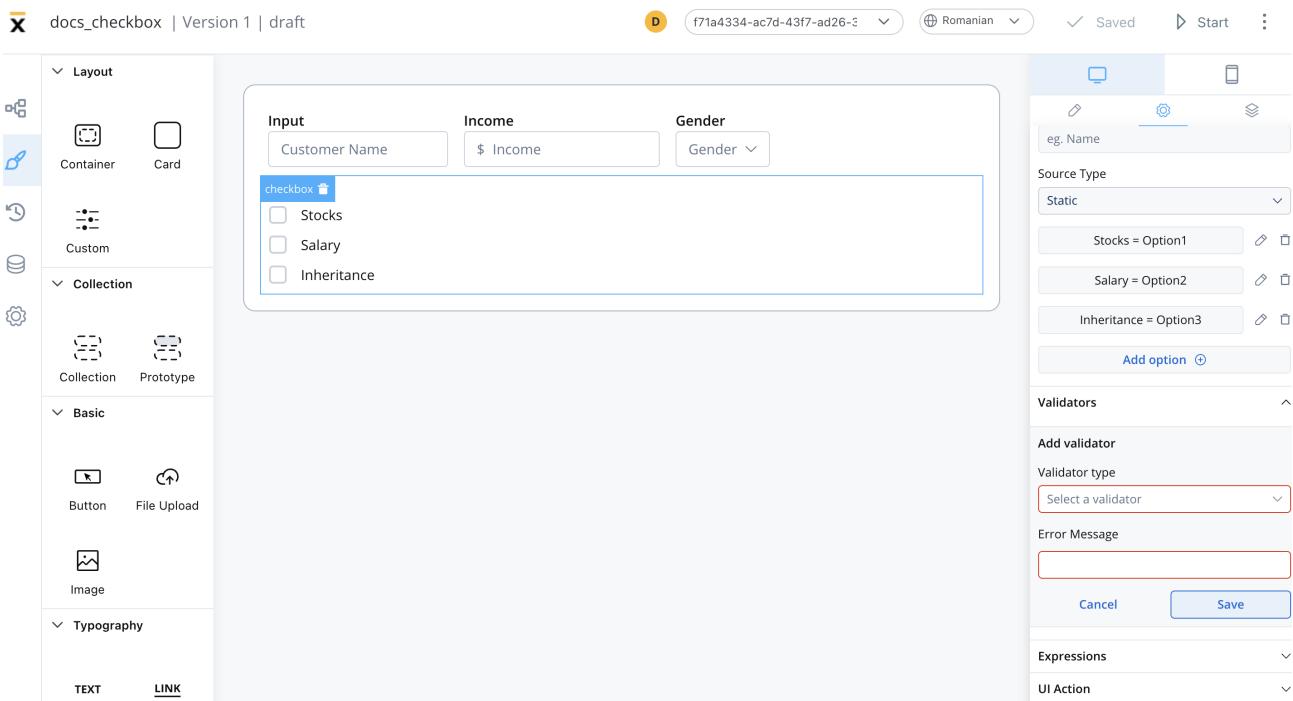
## Datasource

- **Default Value** - the default value of the checkbox
- **Source Type** - it can be Static, Enumeration, or Process Data
- **Add option** - label - value pairs can be defined here

The screenshot shows the FLOWX.AI UI Designer interface. On the left is a sidebar with categories like Layout, Collection, and Basic, each containing icons for Container, Card, Custom, Collection, Prototype, Button, File Upload, and Image. The main area displays a form with three input fields: 'Input' (Customer Name), 'Income' (\$ Income), and 'Gender'. Below these is a 'checkbox' field with three options: Stocks, Salary, and Inheritance. To the right is a preview of the mobile UI and a 'Properties' panel. The 'Properties' panel has sections for Datasource, Default value, Source Type (Static), Validators, Expressions, and UI Action. The 'Datasource' section is highlighted with a red box and contains three entries: 'Stocks = Option1', 'Salary = Option2', and 'Inheritance = Option3', each with edit and delete icons. At the bottom of the 'Datasource' section is a button labeled 'Add option'.

## Validators

The following validators can be added to a checkbox: `required` and `custom` (more details [here](#)).



## Expressions

The checkbox behavior can be defined using JavaScript expressions for hiding or disabling the element. The following properties can be configured for expressions:

- **Hide** - JavaScript expression used to hide the checkbox when it returns a truthy value
- **Disabled** - JavaScript expression used to disable the checkbox when it returns a truthy value

### INFO

It's important to make sure that disabled fields have the same expression configured under the path expressions → hide.

## UI actions

UI actions can be added to the checkbox element to define its behavior and interactions.

- **Event** - possible value: `CHANGE`
- **Action Type** - select the action type

**!** **INFO**

For more details on how to configure a UI action, click [here](#).

## Checkbox styling

The type of the checkbox can be selected by using the **styling** tab in **UI Designer**, possible values:

- clear
- bordered

**!** **INFO**

For more valid CSS properties, click [here](#).

A clear checkbox element with three options added, and a column layout will look like as it follows:

**✓ First Form Title**

First Form subtitle

Input 1

\$ Income

Input 3

.000

Gender

▼

- Stocks
- Salary
- Inheritance

**Was this page helpful?**

# BUILDING BLOCKS / UI Designer / UI component types / Form elements / Radio

The screenshot shows the FLOWX.AI platform's UI Designer. On the left, there's a sidebar with navigation links for Processes (Definitions, Active process, Process Instances, Failed process start) and Content Management (Enumerations, Substitution tags, Content models, Languages, Source systems, Media Library). The main area displays a form titled "Form title". The form contains three input fields: "Customer" (with "Customer Name" placeholder), "Income" (with "\$ Income" placeholder), and "Gender" (with a dropdown menu). Below these are two radio buttons: "Contact via Email" and "Contact via SMS", where "Contact via SMS" is selected. A blue "Submit" button is at the bottom of the form.

Radio buttons are normally presented in radio groups (a collection of radio buttons describing a set of related options). Only one radio button in a group can be selected at the same time.

## Configuring the radio field element

### Radio settings

The available configuration options for this form element are:

- **General**
- **Properties**
- **Datasource**
- **Validators**
- **Expressions**
- **UI actions**
- **Radio styling**

#### General

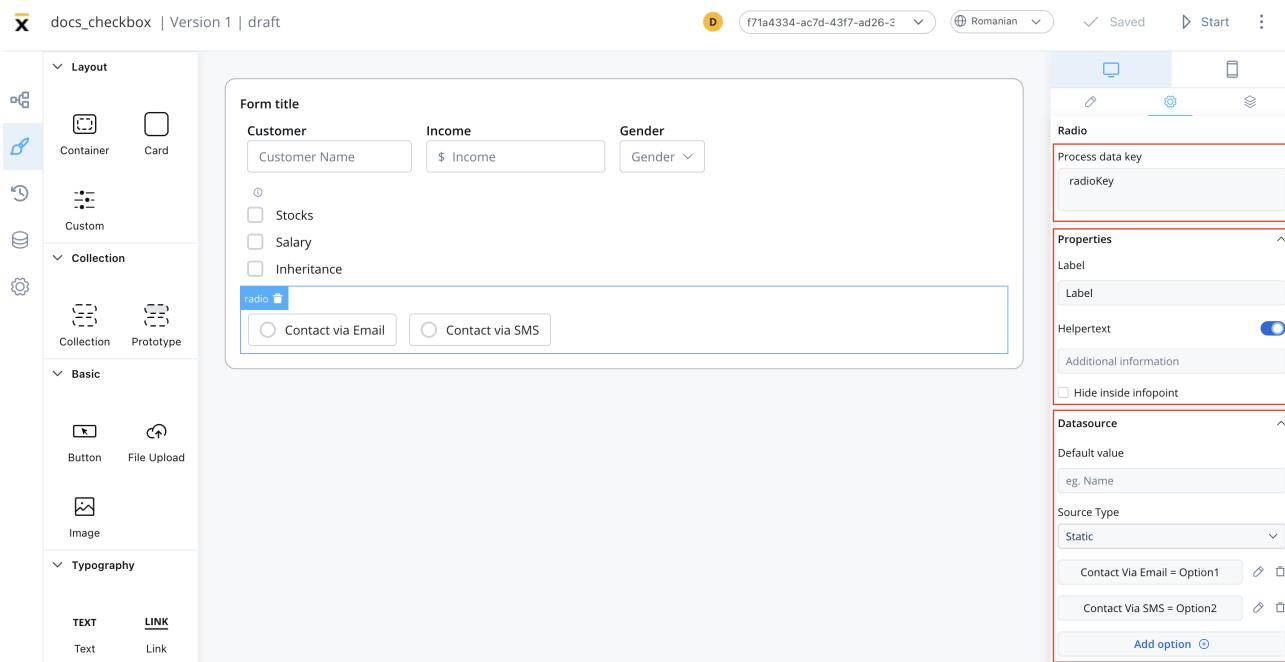
- **Process data key** - creates the binding between form element and process data so it can be later used in **decisions**, **business rules** or **integrations**

#### Properties

- **Label** - the label that appears on the radio
- **Helpertext** - additional information about the radio (can be hidden inside an infopoint)

## Datasource

- **Default Value** - the default values of the radio element
- **Source Type** - it can be Static, Enumeration, or Process Data
- **Add option** - label - value pairs can be defined here



## Validators

The following validators can be added to a radio: `required` and `custom` (more details [here](#))

## Expressions

The radio's element behavior can be defined using JavaScript expressions for hiding or disabling the element. The following properties can be configured for expressions:

- **Hide** - JavaScript expression used to hide the Radio element when it returns a truthy value
- **Disabled** - JavaScript expression used to disable the Radio element when it returns a truthy value

### ! INFO

It's important to make sure that disabled fields have the same expression configured under the path expressions → hide.

The screenshot shows the FLOWX.AI UI Designer interface. On the left, there is a sidebar with categories like Layout, Collection, and Basic, each containing icons for Container, Card, Custom, Collection, Prototype, Button, File Upload, and Image. The main canvas displays a form titled "Form title". The form contains three input fields: "Customer" (Customer Name), "Income" (\$ Income), and "Gender" (Gender dropdown). Below these are three checkboxes labeled "Stocks", "Salary", and "Inheritance". At the bottom of the form is a section labeled "radio" with two radio buttons: "Contact via Email" and "Contact via SMS". To the right of the form, there is a panel with tabs for "Radio", "Properties", "Datasource", and "Validators". The "Validators" tab is currently selected and has a red border around it. It contains a "Required" field and a "Add a validator" button. Below the validators is an "Expressions" section with "Hide" and "Disabled" fields, also both with red borders around them. The top of the screen shows the project name "docs\_checkbox | Version 1 | draft", a save status indicator, and language settings for "Romanian".

## UI actions

UI actions can be added to the radio element to define its behavior and interactions.

- **Event** - possible value: CHANGE
- **Action Type** - select the action type

**!** INFO

For more details on how to configure a UI action, click [here](#).

## Radio styling

The type of the radio can be selected by using the **styling** tab in **UI Designer**, possible values:

- clear
- bordered

**!** INFO

For more valid CSS properties, click [here](#).

A Radio element with two options added, and with a layout configuration set to horizontal will look like as it follows:

**✓ First Form Title**

First Form subtitle

The form consists of several input fields and a radio group. At the top, there are three input fields: 'Input 1' (text), 'Income' (with a dollar sign icon and a dropdown arrow), and 'Input 3' (text). To the right of 'Input 3' is a field labeled 'Gender' with a dropdown arrow. Below these are three radio buttons: 'Stocks' (unchecked), 'Salary' (checked), and 'Inheritance' (unchecked). At the bottom, there are two radio buttons for contact preferences: 'Contact via Email' (checked) and 'Contact via SMS'.

**Was this page helpful?**

# BUILDING BLOCKS / UI Designer / UI component types / Form elements / Switch

The form interface includes:

- Form title:** Customer, Income, Gender.
- Customer:** Customer Customer.
- Income:** \$ 999999999.
- Gender:** Gender dropdown.
- Contact methods:** Radio buttons for "Contact via Email" (selected) and "Contact via SMS".
- Date selection:** A date picker showing 09.02.1972.
- Newsletter subscription:** A toggle switch labeled "Do you want to subscribe to our newsletter?" which is turned on.
- Submit button:** A blue "Submit" button at the bottom left.

A switch, a toggle switch, is another form element that can be utilized to create an intuitive user interface. The switch allows users to select a response by toggling it between two states. Based on the selection made by the user, the corresponding Boolean value of either true or false will be recorded and stored in the process instance values for future reference.

## Configuring the switch element

### Switch settings

The available configuration options for this form element are:

- General
- Properties
- Datasource
- Validators
- Expressions
- UI actions
- Switch styling

## General

- **Process data key** - creates the binding between form element and process data so it can be later used in decisions, business rules or integrations

## Properties

- **Label** - the label of the switch
- **Helpertext** - additional information about the switch element (can be hidden inside an infopoint)

## Datasource

- **Default Value** - the default value of the switch form field (it can be switched on or switched off)

## Validators

The following validators can be added to a switch element: `requiredTrue` and `custom` (more details [here](#)).

The screenshot shows the FLOWX.AI UI Designer interface. On the left is a sidebar with categories like Layout, Collection, Basic, and Typography, each containing icons for various components. The main area displays a form titled "Customer" with fields for "Customer Name", "Income", and "Gender". Below these are two radio buttons for "Contact via Email" and "Contact via SMS". A "Select a date" dropdown is set to "Date of birth". A "switch" element is present with the label "want to subscribe to our newsletter?" and a toggle switch. At the bottom is a "Submit" button. To the right of the form is a panel titled "Properties" which includes sections for "Label" (set to "Do you want to subscribe to ou"), "Helpertext" (disabled), "Datasource" (with a checked "Default value" checkbox), and "Validators" (set to "requiredTrue" and "custom"). There are also sections for "Expressions" and "UI Action". The top of the screen shows the project name "docs\_checkbox | Version 1 | draft", a session ID "f71a4334-ac7d-43f7-ad26-3", language "Romanian", and status indicators.

## Expressions

- **Hide** - JavaScript expression used to hide the Switch element when it returns a truthy value
- **Disabled** - JavaScript expression used to disable the Switch element when it returns a truthy value

## UI actions

UI actions can be added to the switch element to define its behavior and interactions.

- Event - possible value: CHANGE
- Action Type - select the action type



For more details on how to configure a UI action, click [here](#).

## Switch styling

The label of the switch can be positioned either as `start` or `end`.

The screenshot shows the FLOWX.AI UI Designer interface. On the left is a sidebar with categories like Layout, Custom, Collection, Basic, and Media. In the center is a form titled "Form title". The form contains fields for "Customer Name", "Income", and "Gender". Below these are two radio buttons for "Contact via Email" and "Contact via SMS". A date picker for "Date of birth" is also present. At the bottom is a switch component with the label "Do you want to subscribe to our newsletter?". To the right of the form is a panel titled "Switch" with properties for "Label position" set to "end", and sections for Sizing, Spacing, Background, Typography, and Advanced.

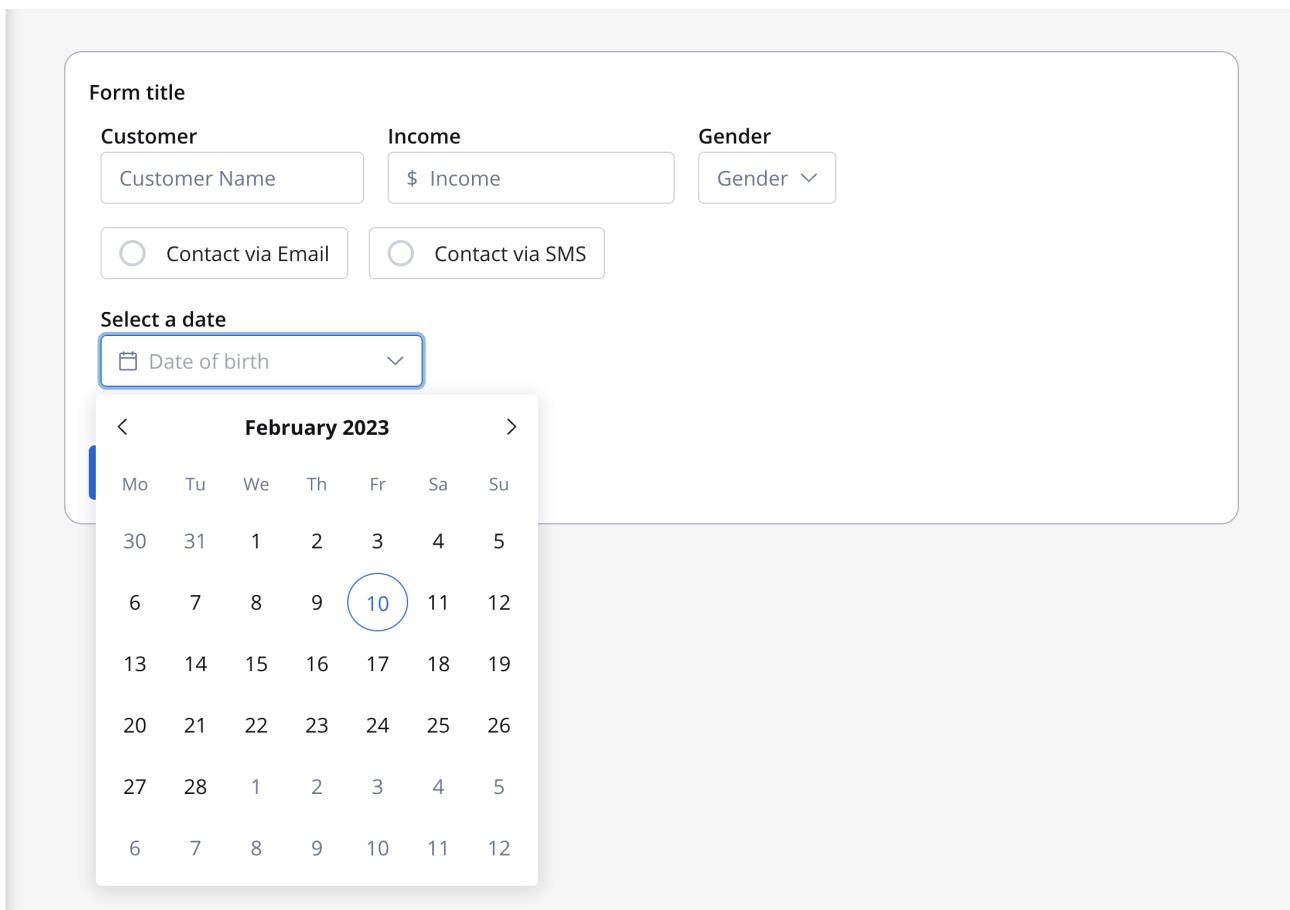
### INFO

For more valid CSS properties, click [here](#).

Was this page helpful?

**BUILDING BLOCKS / UI Designer / UI component types / Form elements /**

# Datepicker



The datepicker (Calendar Picker) is a lightweight component that allows end users to enter or select a date value.

## INFO

The default datepicker value is `DD.MM.YYYY`.

## Configuring the datepicker element

## Datepicker settings

The available configuration options for this form element are:

- **General**
- **Properties**
- **Datasource**
- **Validators**
- **Expressions**
- **UI actions**
- **Datepicker styling**

### General

- **Process data key** - creates the binding between form element and process data so it can be later used in **decisions**, **business rules** or **integrations**

### Properties

- **Label** - the label of the datepicker
- **Placeholder** - placeholder when the field has no value
- **Min Date** - set the minimum valid date selectable in the datepicker
- **Max Date** - set the maximum valid date selectable in the datepicker
- **Min Date, Max Date error** - when a date is introduced by typing, define the error message to be displayed
- **Helpertext** - additional information about the input field (can be hidden inside an infopoint)

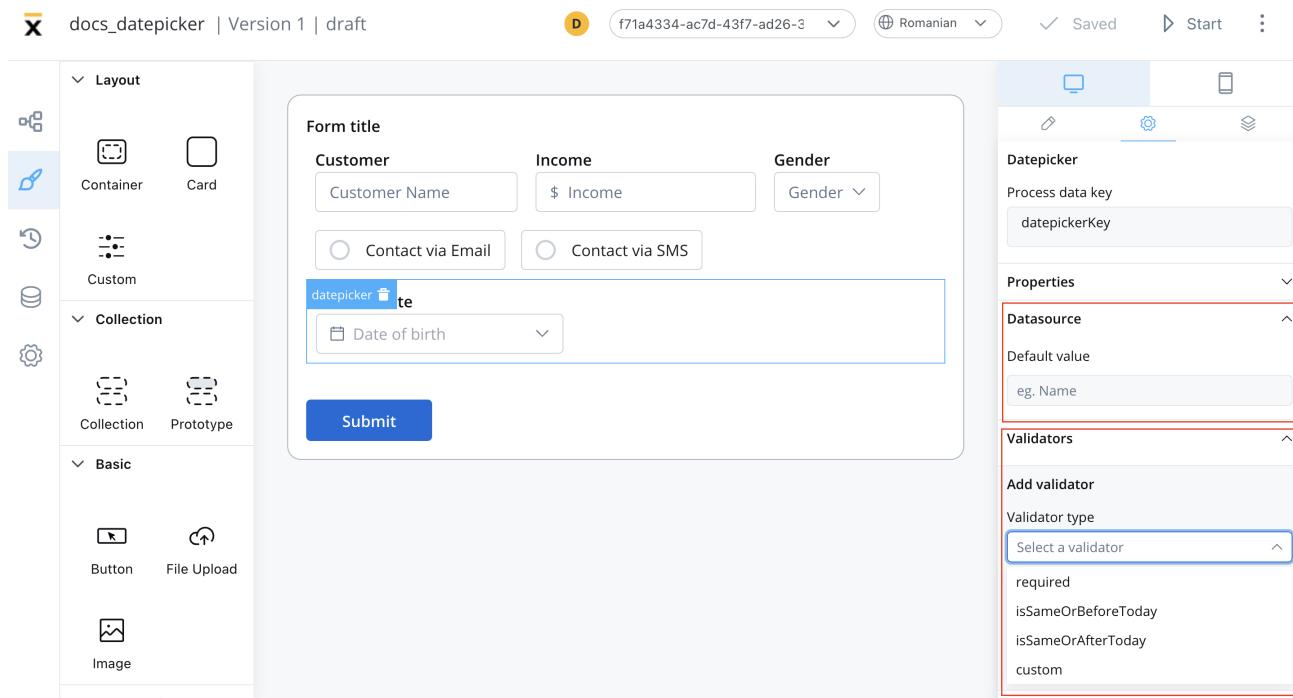
The screenshot shows the FLOWX.AI UI Designer interface. On the left is a sidebar with various components: Layout (Container, Card, Custom), Collection (Collection, Prototype), and Basic (Button, File Upload, Image). The main area displays a form titled "Form title" with sections for "Customer" (Customer Name, Income, Gender), "Contact via Email/SMS", and a "datepicker" input field labeled "Date of birth". A "Submit" button is at the bottom. To the right of the form is a properties panel for the "datepicker" element, which includes fields for "Process data key" (datepickerKey), "Label" (Select a date), "Placeholder" (Date of birth), "Min date" and "Max date" (both set to dd.mm.yyyy), "Min date error" and "Max date error" (empty), and a "Helpertext" toggle switch.

## Datasource

- **Default Value** - the default values of the datepicker element, this will autofocus the datepicker when you will run the process

## Validators

The following validators can be added to a datepicker: `required`, `custom`, `isSameOrBeforeToday` or `isSameOrAfterToday` (more details [here](#)).



## Expressions

The datepicker behavior can be defined using JavaScript expressions for hiding or disabling the element. The following properties can be configured for expressions:

- **Hide** - JavaScript expression used to hide the datepicker when it returns a truthy value
- **Disabled** - JavaScript expression used to disable the datepicker when it returns a truthy value

### INFO

It's important to make sure that disabled fields have the same expression configured under the path expressions → hide.

## UI actions

UI actions can be added to the datepicker element to define its behavior and interactions.

- **Event** - possible value: `CHANGE`
- **Action Type** - select the action type

### INFO

For more details on how to configure a UI action, click [here](#).

The screenshot shows the FLOWX.AI UI Designer interface. On the left, there's a sidebar with categories like Layout, Collection, and Basic, each containing icons for Container, Card, Custom, Collection, Prototype, Button, File Upload, and Image. The main area displays a form titled "Form title". The form contains fields for "Customer Name", "Income", "Gender", and two radio buttons for "Contact via Email" and "Contact via SMS". Below these is a datepicker field labeled "Date of birth". At the bottom is a blue "Submit" button. To the right of the form, there's a panel titled "Expressions" which includes sections for "Hide" and "Disabled", both currently empty. Below this is a "UI Action" section. Under "Event", the value "CHANGE" is selected. Under "Action Type", a dropdown menu is open with the placeholder "Select an action type". At the bottom of the UI action panel are "Cancel" and "Save" buttons. The overall interface has a clean, modern design with a light blue header and sidebar.

## Datepicker styling

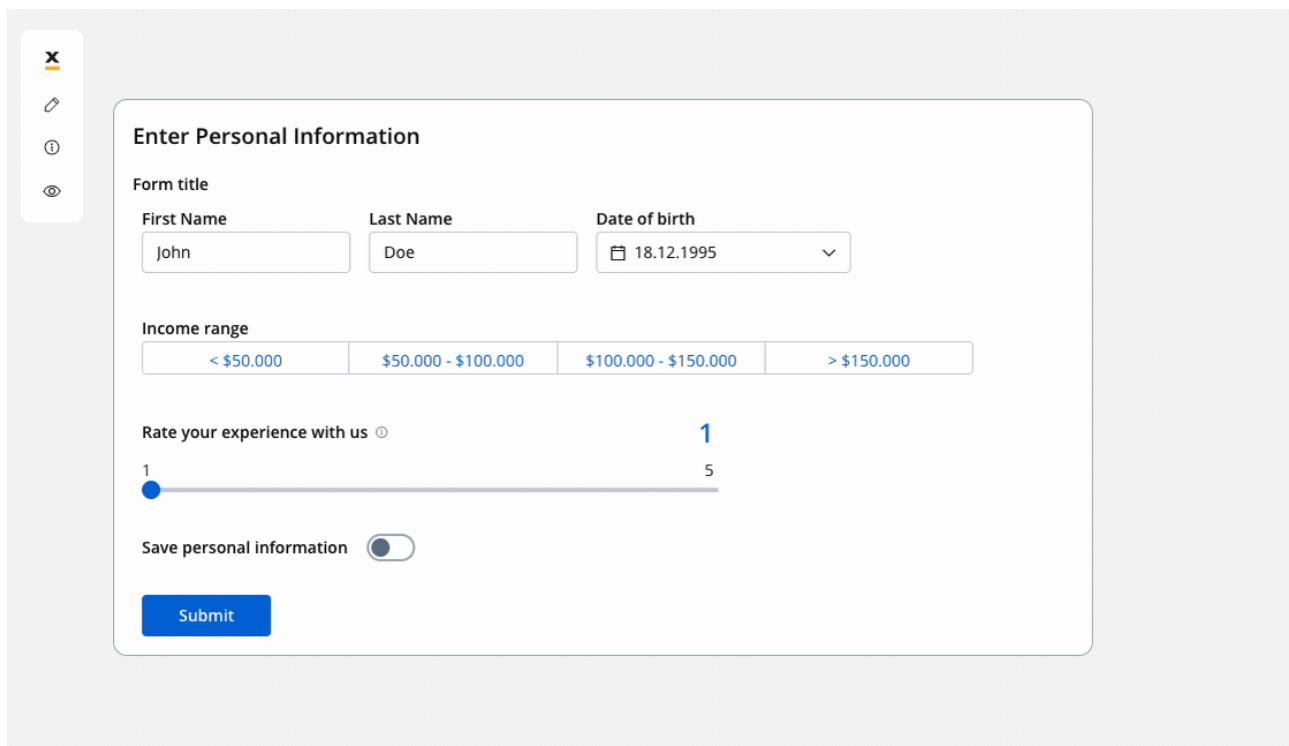
The styling of a datepicker element can be customized in various ways using CSS properties like typography color, border-radius/width, or advanced CSS params. This allows you to create a datepicker that fits seamlessly with the overall design of the application you are developing.

**!** INFO

For more valid CSS properties, click [here](#).

Was this page helpful?

# BUILDING BLOCKS / UI Designer / UI component types / Form elements / Slider



The screenshot shows a user interface for "Enter Personal Information". On the left is a toolbar with icons for close, edit, info, and preview. The main area has a title "Enter Personal Information" and a "Form title" section. It includes fields for "First Name" (John), "Last Name" (Doe), and "Date of birth" (18.12.1995). Below this is a "Income range" section with four options: < \$50.000, \$50.000 - \$100.000, \$100.000 - \$150.000, and > \$150.000. A "Rate your experience with us" section features a horizontal slider with a midpoint at 1, ranging from 1 to 5. A "Save personal information" toggle switch is turned on. At the bottom is a blue "Submit" button.

It allows users to pick only one option from a group of options, and you can choose to have between 2 and 5 options in the group. The segmented button is

easy to use, and can help make your application easier for people to use.

## Configuring the slider element

### Slider settings

The available configuration options for this form element are:

- **General**
- **Properties**
- **Datasource**
- **Validators**
- **Expressions**
- **UI actions**
- **Slider styling**

#### General

- **Process data key** - creates the binding between form element and process data so it can be later used in **decisions**, **business rules** or **integrations**

#### Properties

- **Label** - the label of the slider
- **Show value label** - a toggle option that determines whether the current selected value of the slider is displayed as a label alongside the slider handle
- **Helptext** - an optional field that provides additional information or guidance related to the usage or function of the slider, it can be hidden within an

infopoint, which users can expand or access for further detail

- **Min Value** - the minimum value or starting point of the slider's range, it defines the lowest selectable value on the slider
- **Max Value** - the maximum value or end point of the slider's range, it sets the highest selectable value on the slider
- **Suffix** - an optional text or symbol that can be added after the displayed value on the slider handle or value label, it is commonly used to provide context or units of measurement
- **Step size** - the increment or granularity by which the slider value changes when moved, it defines the specific intervals or steps at which the slider can be adjusted, allowing users to make more precise or discrete value selections

## Datasource

- **Default Value** - the default value of the slider (static value - integer) the initial value set on the slider when it is first displayed or loaded, it represents a static value (integer), that serves as the starting point or pre-selected value for the slider, users can choose to keep the default value or adjust it as desired

The screenshot shows the FLOWX.AI UI Designer interface. On the left, there's a sidebar with various UI component icons categorized under 'Layout' (Container, Card, Custom), 'Forms' (Form, Input, Textarea, Select, Checkbox, Radio, Switch, Datepicker, Segmented Button, Slider), and 'Collection'. The main area displays a form titled 'Enter Personal Information'. This form includes fields for 'First Name', 'Last Name', and 'Date of birth'. Below these are sections for 'Employment type' (radio buttons for 'Employed' and 'Pensioner'), 'Income range' (a segmented button with options: < \$50.000, \$50.000 - \$100.000, \$100.000 - \$150.000, > \$150.000), and a 'Loan amount' section featuring a slider. The slider has a value of '55000 €' and a range from '10000 €' to '100000 €'. There's also a 'Save personal information' toggle switch and a 'Submit' button. To the right of the form, a detailed configuration panel is open for the 'Slider' component. It shows the 'Process data key' as 'sliderKey' and contains several sections: 'General properties' (Slider label: 'Loan amount', Show value label checked, Helpertext: 'Min Value 10000, Max Value 100000, Suffix €, Step size 5000'), 'Datasource' (Default value: 20000), and a 'Slider' section with icons for edit, copy, and delete.

## Validators

The following validators can be added to a slider: `required` and `custom` (more details [here](#)).

## UI actions

UI actions can be added to the slider element to define its behavior and interactions.

- **Event** - possible value: CHANGE
- **Action Type** - select the action type, ! for more details on how to configure a UI action, click [here](#)

## Multiple sliders

You can also use multiple sliders UI elements that are interdependent, as you can see in the following example:

The screenshot shows a form titled "Enter Personal Information". It includes fields for First Name, Last Name, and Date of birth. Under "Employment type", there are two radio buttons: "Employed" and "Pensioner". A toggle switch labeled "Save personal information" is turned on. Below these, there are three sliders with numerical inputs:

- Loan amount:** Slides from 10000 \$ to 500000 \$. The current value is 255000 \$.
- Down payment:** Slides from 38250 \$ to 89250 \$. The current value is 38250 \$.
- Loan type:** A dropdown menu currently set to "Conventional".

A blue "Submit" button is at the bottom.

### INFO

You can improve the configuration of the slider using computed values as in the example above. These values provide a more flexible and powerful approach for handling complex use cases. You can find an example by referring to the following documentation:

## Dynamic & computed values

# Slider styling

To create a slider with specific styling, sizing, typography, and color settings, you can use the following configuration:

- **Sizing**
- **Typography**
- **Background**

## Sizing

- set the width of the button - fill/fixed/auto

## Typography

Choose an appropriate font family, size, and weight for the button label text.

- **Label Color** - set the color of the button label text
- **Min/Max values** - set the color of
- **Result** - set the color of the

## Background

- **Selected BG** - set the background color of the button.
- **ComponentBg** - set the background color of the button.

The screenshot shows the FLOWX.AI UI Designer interface. On the left, there's a sidebar with categories like Layout, Forms, and Collections, each containing various UI component icons. The main area displays a form titled "Enter Personal Information". This form includes fields for First Name, Last Name, Date of birth, Employment type (radio buttons for Employed and Pensioner), Income range (a segmented button with options <\$50.000, \$50.000 - \$100.000, \$100.000 - \$150.000, >\$150.000), and a Slider for Loan amount ranging from 10000 € to 100000 €. A "Save personal information" toggle switch is also present. At the bottom is a "Submit" button. To the right of the form is a detailed configuration panel for the "Slider" component, divided into sections for Sizing, Spacing, Typography, Background, and Advanced settings.

### INFO

For more valid CSS properties, click [here](#).

Was this page helpful?

# BUILDING BLOCKS / UI Designer / UI component types / Form elements / Segmented button

The screenshot shows a 'Form title' section with three input fields: 'First Name' (John), 'Last Name' (Doe), and 'Date of birth' (18.12.1995). Below this is a 'Income range' section with four options: '< \$50.000', '\$50.000 - \$100.000', '\$100.000 - \$150.000', and '> \$150.000'. A 'Save personal information' toggle switch is turned on. At the bottom is a blue 'Submit' button.

It allows users to pick only one option from a group of options, and you can choose to have between 2 and 5 options in the group. The segmented button is easy to use, and can help make your application easier for people to use.

## Configuring the segmented button

### Segmented button settings

The available configuration options for this form element are:

- General
- Properties
- Datasource
- Validators
- Expressions

- **UI actions**
- **Segmented button styling**

## General

- **Process data key** - creates the binding between form element and process data so it can be later used in **decisions**, **business rules** or **integrations**

## Properties

- **Label** - the label of the segmented button
- **Helpertext** - additional information about the segmented button (can be hidden inside an infopoint)

## Datasource

- **Default Value** - the default value of the segmented button (it can be selected from one of the static source values)
- **Source Type** - it is by default Static
- **Add option** - value/label pairs can be defined here

## Validators

The following validators can be added to a segmented button: **required** and **custom** (more details [here](#)).

The screenshot shows the FLOWX.AI UI Designer interface. On the left, there's a sidebar with categories like Layout, Forms, Collection, etc., each containing various UI component icons. The main area displays a form titled "Enter Personal Information". Inside the form, there's a "Segmented Button" element labeled "Income range" with four options: "< \$50.000", "\$50.000 - \$100.000", "\$100.000 - \$150.000", and "> \$150.000". Below the button is a "Save personal information" toggle switch and a "Submit" button. To the right of the form is a panel with three sections: "Properties", "Datasource", and "Validators", all of which are currently expanded and highlighted with red boxes.

## UI actions

UI actions can be added to the segmented button element to define its behavior and interactions.

- **Event** - possible value: CHANGE
- **Action Type** - select the action type

### INFO

For more details on how to configure a UI action, click [here](#).

## Segmented button styling

To create a segmented button with specific styling, sizing, typography, and color settings, you can use the following configuration:

- **Sizing**
- **Typography**
- **Background**

## Sizing

- set the width of the button - fill/fixed/auto

## Typography

Choose an appropriate font family, size, and weight for the button label text.

- **Label Color** - set the color of the button label text
- **Selected State** - set the color of the label text when the button is selected
- **Unselected State** - set the color of the label text when the button is not selected

## Background

- **Selected state** - set the background color of the button
- **Unselected state** - set the background color of the button

The screenshot shows the FLOWX.AI UI Designer interface. On the left is a sidebar with categories like Layout, Forms, and Collections, each containing various UI component icons. The main area displays a form titled "Enter Personal Information". The form includes fields for First Name, Last Name, Date of birth, and an income range segmented button. A "Save personal information" toggle and a "Submit" button are also present. To the right of the form is a detailed styling panel with sections for Sizing, Spacing, Typography, Background, Border, and Advanced settings. The "Sizing" section has a red box around it, showing "Fit W: auto". The "Background" section also has a red box around it, showing "Selected" and "Unselected" color swatches. The "Border" section shows radius and width settings.

## INFO

For more valid CSS properties, click [here](#).

Was this page helpful?