

# ReusableForm Component Documentation

---

## Overview

The **ReusableForm** component is a flexible form builder that allows you to create and manage forms using a **json** configuration object. The form structure and fields are defined by a configuration object that can be declared in the script or as a **json** file. This document describes the configuration options available for each form element and their properties.

## Form Configuration

The **formConfig** object consists of several sections, each containing a set of fields. Below is the structure and options for the configuration.

Sure, here are the options for **formConfig**, **section**, and **field** as tables in markdown:

### FormConfig

Option	Type	Description
<b>endpoint</b>	string	The URL where the form data will be submitted.
<b>showProgressIndicator</b>	string	The type of progress indicator to show (" <b>vertical</b> " or " <b>horizontal</b> ").
<b>sectionHighlightColor</b>	string	The highlight color for the current section in the progress bar.
<b>sections</b>	array	An array of section objects.

### Section

Option	Type	Description
<b>title</b>	string	The title of the section.
<b>fields</b>	array	An array of field objects.

### Field

Option	Type	Description	Field Usage
<b>name</b>	string	The unique name of the field.	All fields
<b>label</b>	string	The label text for the field.	All fields
<b>type</b>	string	The type of the field (e.g., " <b>text</b> ", " <b>select</b> ", " <b>textarea</b> ", " <b>file</b> ", " <b>multitext</b> ", " <b>table</b> ", " <b>description</b> ").	All fields

Option	Type	Description	Field Usage
<code>required</code>	boolean	Whether the field is required.	All fields except <code>"description"</code>
<code>placeholder</code>	string	The placeholder text for the field (optional).	<code>"text"</code> , <code>"textarea"</code> , <code>"password"</code>
<code>helperText</code>	string	Additional helper text for the field (optional).	All fields
<code>validationPattern</code>	string	The regex pattern for validating the field input (optional).	<code>"text"</code> , <code>"textarea"</code> , <code>"password"</code>
<code>validationWarning</code>	string	The warning message to display if validation fails (optional).	<code>"text"</code> , <code>"textarea"</code> , <code>"password"</code>
<code>options</code>	array	An array of options for select fields (optional).	<code>"select"</code> , <code>"multiselect"</code>
<code>fetchOptions</code>	object	Configuration for dynamically fetching options (optional).	<code>"select"</code> , <code>"multiselect"</code>
<code>allowMultiple</code>	boolean	Whether multiple files can be selected (for file fields).	<code>"file"</code>
<code>maxSize</code>	number	The maximum file size allowed (for file fields).	<code>"file"</code>
<code>allowedTypes</code>	array	An array of allowed file types (for file fields).	<code>"file"</code>
<code>table</code>	string	The name of the table this field is part of (for table fields).	<code>"table"</code>
<code>labelName</code>	string	The label name for table headers.	<code>"table"</code>
<code>valueName</code>	string	The value name for table headers.	<code>"table"</code>
<code>units</code>	string	The units for table values.	<code>"table"</code>
<code>summary</code>	string	The summary text for table totals.	<code>"table"</code>
<code>rows</code>	number	The number of rows for textarea fields.	<code>"textarea"</code>
<code>condition</code>	object	Condition to determine if the field should be visible (optional).	All fields

## FetchOptions

Option	Type	Description
<code>url</code>	string	The URL to fetch options from.

Option	Type	Description
<code>method</code>	string	The HTTP method to use for fetching options ("GET" or "POST").
<code>target</code>	string	The key in the response data that contains the options.
<code>valueField</code>	string	The field in the response data to use as the option value.
<code>labelField</code>	array	An array of fields in the response data to use for the option label.
<code>labelSeparator</code>	string	The separator to use between label fields (optional).

## Condition

Option	Type	Description
<code>field</code>	string	The field name to check.
<code>value</code>	string	The value to compare.
<code>values</code>	array	An array of values to compare.
<code>notEmpty</code>	boolean	Whether the field should be visible if the condition field is not empty.

## Examples

### Example Configuration

Here is an example of a simple form configuration:

```
{
  "endpoint": "https://example.com/submit",
  "showProgressIndicator": "horizontal",
  "sectionHighlightColor": "#4caf50",
  "sections": [
    {
      "title": "Personal Information",
      "fields": [
        {
          "name": "firstName",
          "label": "First Name",
          "type": "text",
          "required": true,
          "placeholder": "Enter your first name"
        },
        {
          "name": "lastName",
          "label": "Last Name",
          "type": "text",
          "required": true,
          "placeholder": "Enter your last name"
        }
      ]
    }
  ]
}
```

```

    {
      "name": "gender",
      "label": "Gender",
      "type": "select",
      "required": true,
      "options": ["Male", "Female", "Other"]
    },
    {
      "name": "bio",
      "label": "Bio",
      "type": "textarea",
      "rows": 5,
      "placeholder": "Tell us about yourself"
    }
  ]
},
{
  "title": "Account Details",
  "fields": [
    {
      "name": "email",
      "label": "Email",
      "type": "text",
      "required": true,
      "placeholder": "Enter your email",
      "validationPattern": "^[\\S+@[\\S+\\.\\S+$$",
      "validationWarning": "Please enter a valid email address"
    },
    {
      "name": "password",
      "label": "Password",
      "type": "password",
      "required": true,
      "placeholder": "Enter your password"
    }
  ]
}
]
}

```

This configuration will create a form with two sections, "Personal Information" and "Account Details," each containing several fields.

## Custom Fetch Options

Fields can dynamically fetch options from an API using the `fetchOptions` property. Here is an example of a field with fetch options:

```

{
  "name": "country",

```

```
"label": "Country",
"type": "select",
"required": true,
"fetchOptions": {
  "url": "https://example.com/countries",
  "method": "GET",
  "target": "countries",
  "valueField": "id",
  "labelField": ["name"],
  "labelSeparator": ", "
}
}
```

In this example, the field will fetch options from the specified URL and use the `id` field as the value and the `name` field as the label for each option.