

# Multiple Dynamic Cascading Drop-Down List

Joget Form Builder allows you to **dynamically** change the **options/selections** of a **select box/radio button/radio** based on multiple form field values within the same form.

Tips

Dynamic Cascading Drop-Down List is also supported in List Filters.

- [How it works](#)
- [Configuring multiple dynamic cascading drop-down list](#)
- [Multiple Dynamic Form in App Run Time](#)
- [Demo App](#)

## How it works

Each option is a list of data from another form with an [option data store configuration](#). Below will show the steps of configuring each form.

## Configuring multiple dynamic cascading drop-down list

Create multiple simple forms with a 'Name' text field with no Advance Options configuration to create an option data list.

In this example, we will create 'Climate' and 'Continent' forms with the same configuration.

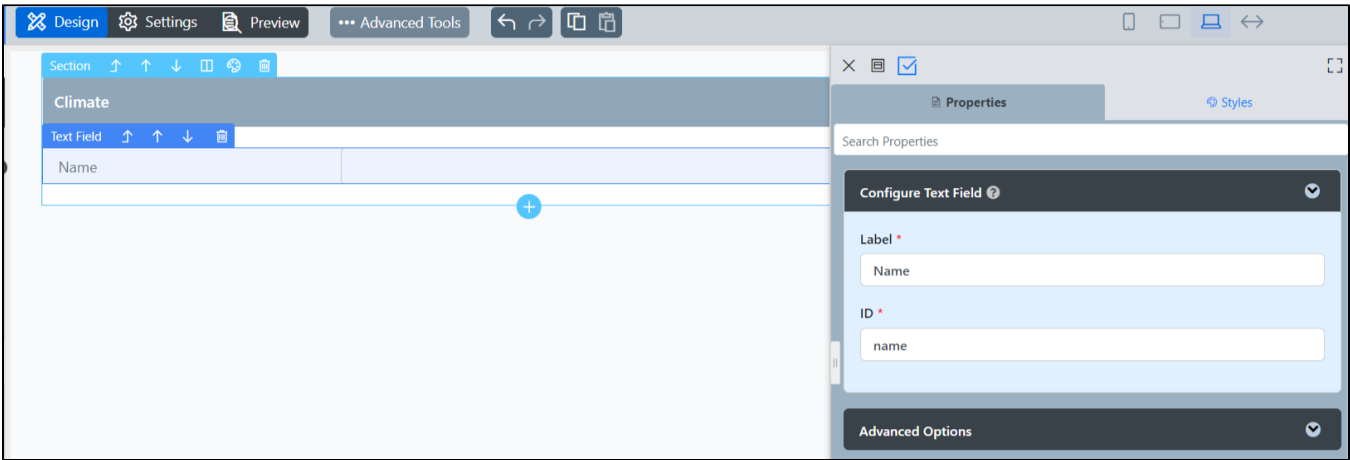
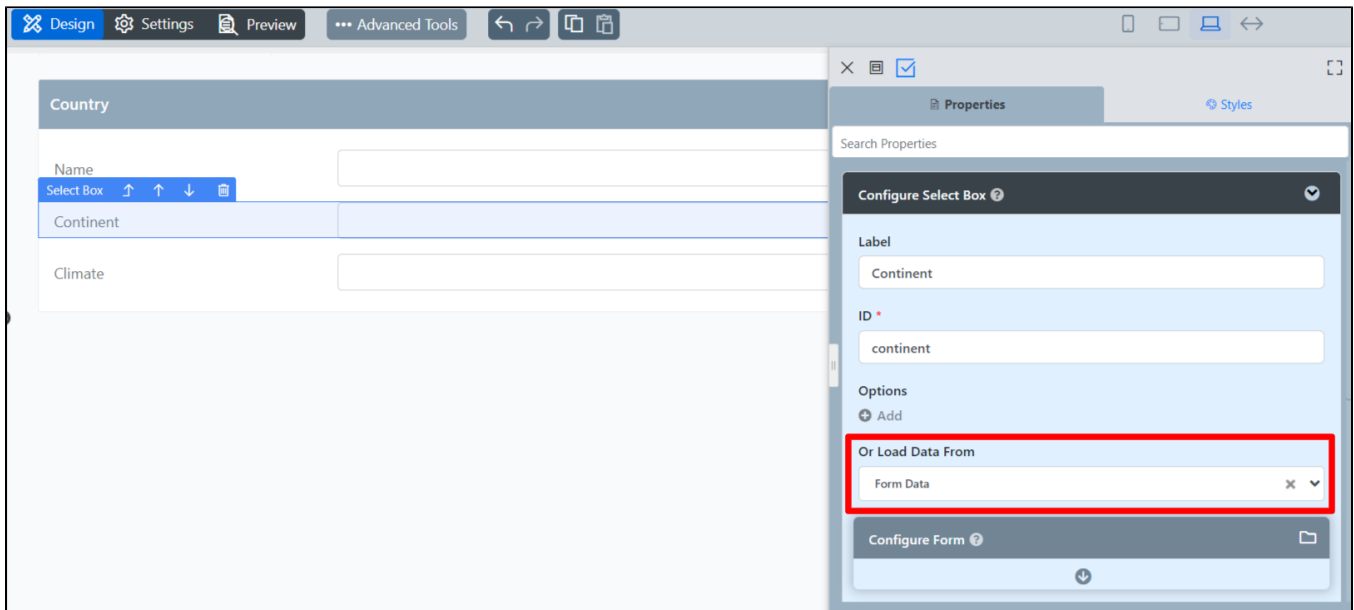


Figure 1: Configuring Form Data Options

Create a 'Country' Form with the same configuration as Figure 1 but with an addition of the 'Climate' and 'Continent' select boxes which will be using the Form Data Options Data Store on the 'Continent' select box field to load data from the 'Continent' Form.



Configure Form ?

Form \*

Continent

ID Column

Label Column \*

name

Grouping Column

Extra Filter Condition

☒ Add Empty Option

Empty Option Label

☐ Use AJAX for Cascade Options?

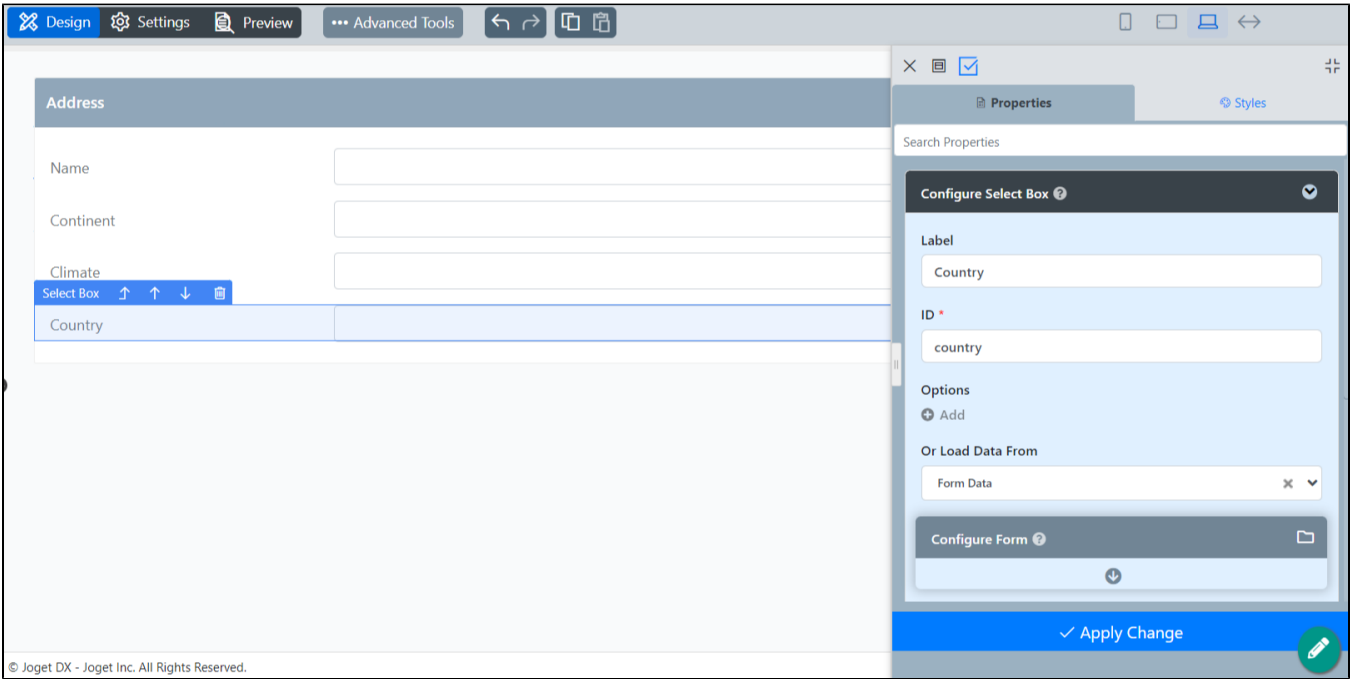
FORM OPTIONS CACHING

Sync Cache Interval ?

Figure 2: Configure Form Data Options Data Store

As shown in Figure 2, use the same configuration on the 'Climate' select box. Using Form Data Options Data Store, the 'Continent' field will view the values from the Continent List that contains only one column "Name", and the Climate field will view the values from the Climate List that contains only one column "Name".

Create 'Address' Form with all the previous forms as a select box with 'Continent', 'Climate', and 'Country' with the same configuration in Figure 2. But with additional configuration for the 'Country' select box.



## Configure Form ?



Form \*



Country



ID Column



Label Column \*

name



Grouping Column

climate



continent



Extra Filter Condition

☒ Add Empty Option

Empty Option Label

☒ Use AJAX for Cascade Options?

Figure 3: Configure Country as Data Option Form


The Country field will view the values from the Country List under the Name column based on the grouping by the other 2 columns "Continent ", and "Climate" with the help of the 'use AJAX for Cascade Options?' checkbox.

DEPENDENCY

Field Id to Control Available Options Based on Grouping

climate;continent

Figure 4: Shows the multiple dependency fields of Country Field



The **Field ID to Control Available Options Based on the Grouping** Field (figure 4) must be in the same order as the order in the **Grouping Column** (figure 3).

Make sure to click on the Field IDs rather than type them. You should only be typing the ';' symbol.

Multiple Dynamic Form in App Run Time

The figure below is the example data of which Country will appear based on the Climate and Continent combination.

	Name	Continent	Climate	
<input type="checkbox"/>	Bulgaria	Europe	Seasonal	<button>Edit</button>
<input type="checkbox"/>	Malaysia	Asia	Tropical	<button>Edit</button>
<input type="checkbox"/>	UK	Europe	Seasonal	<button>Edit</button>
<input type="checkbox"/>	Turkey	Europe	Tropical	<button>Edit</button>
<input type="checkbox"/>	Singapore	Asia	Tropical	<button>Edit</button>
<input type="checkbox"/>	Canada	North America	Seasonal	<button>Edit</button>
6 items found, displaying all items.				

Figure 5: Example Form Data option

In the app run time, we can see the 'Country' column is different based on the 'Climate' and 'Continent' columns as was set in Figure 5.

Home > Manage Address

### Address

Name

Continent

Climate

Country

Turkey

Home > Manage Address

### Address

Name

Continent

Climate

Country

Bulgaria  
UK

Figure 6: Multiple Form Data option testing

## Demo App

