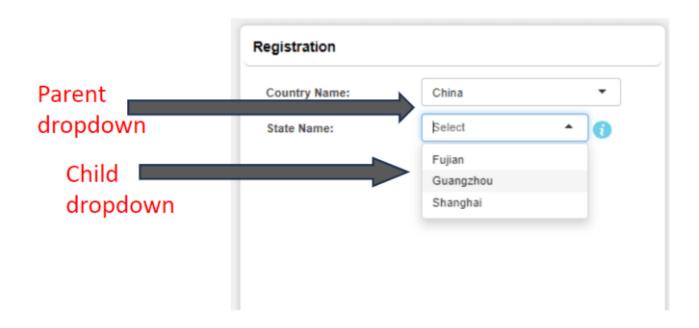
Relational Dropdown List
Relational Dropdown List
A relational dropdown list consists of two dropdown list.  We will call the two dropdown list parent and child dropdown lists.
When the user select a different option for the parent dropdown list, the child dropdown list wil show a different set of options based on the selected choice of the parent dropdown list.
In this example, the parent dropdown list contains a list of country names and the child dropdown which display the state of the country will display the choices based on the country you have selected.

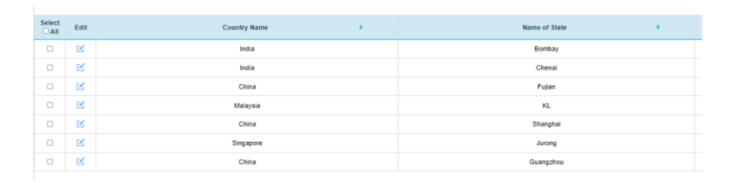


Step 1: We first create a form to contain all the country and state information

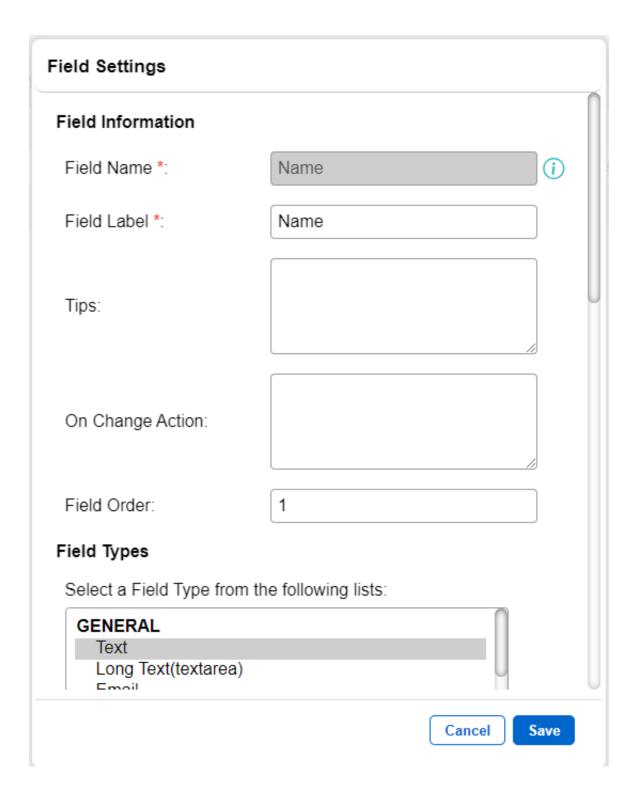
- Create a form named "Master"
- Add two fields, Country and State
- · Add the data into this form

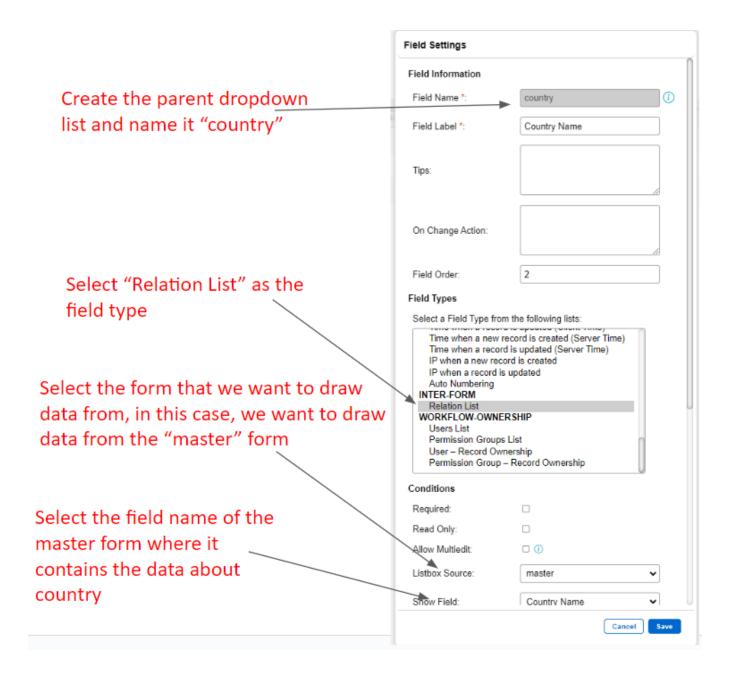
## master > Fields Mapping Form ID: 474

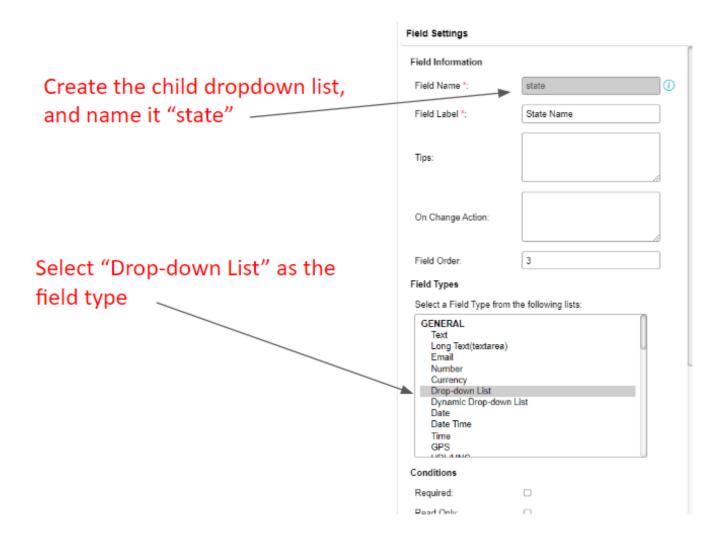
Select	Edit	Field Order	Field Name	Field Label	Required	Allow Multiedit	Read Only	Unique Value	Alphanumeric Sort	Same value	Listbox ID	Listbox Show Field	Field Type
	ß	1	countryname	Country Name	0	0	0	0	0	0	0		Dynamic Dropdown
	ß	2	statename	Name of State	0	0	0	0	0	0	0		text



**Step 2.** Next, we create the user form, in this case, we are creating a registration form. For this demo, we are only creating three fields.







The field mapping for the registration form should now look like this.



again.

Step 3: There are some additional commands required to make the relational dropdown list work properly.



```
loadDropdown({
parentDropdown: "country",
```

```
childDropdown: "state",
  dataSource: { formID: 474, parentField:"countryname", childField:"s
tatename"}
})
```

This command indicates the parent and child dropdown fieldname of the registration form, with the data coming from the relative parentField and childField of the Master form.

