

FORM CREATION

INTRODUCTION

- Form is a window or screen that contains numerous fields to enter data.
- Forms are created to deliver the application in a structured manner and also it is a particular way in which application exists or appears.

FORMS CREATION

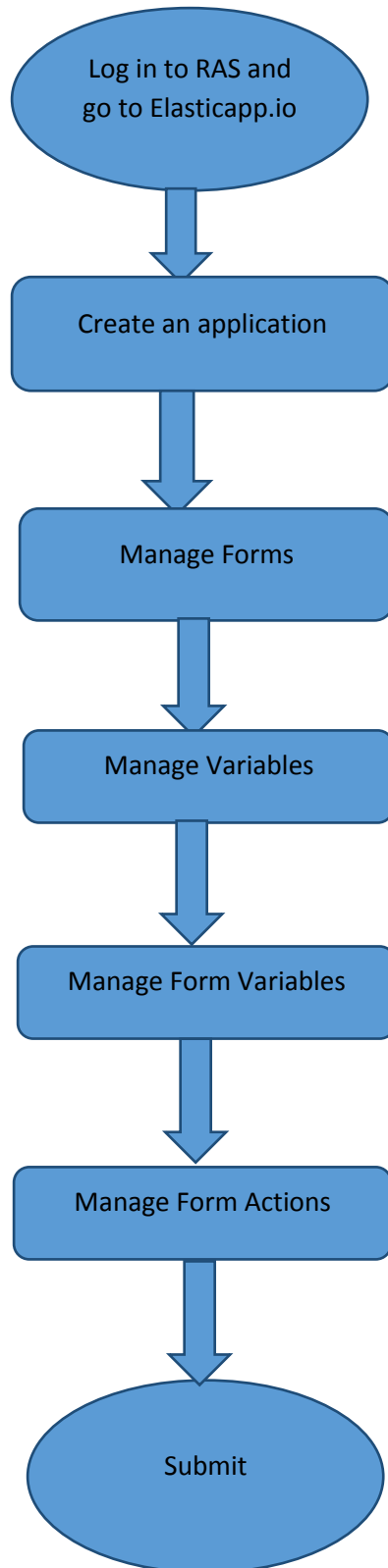
We have five steps to create a form. They are:

1. Manage Forms.
2. Mange Variables.
3. Manage Form Variables.
4. Manage Form Actions.
5. Manage UX Layout.

PREREQUISITIES

- Create account in RNC (<http://www.relgo.com>)
- Verify and create identity cast.
- Subscribe in RAS (<http://www.relgo.com/ras>) with generated identity cast.
- Complete Bootstrap installations.

RELGO APPS FORM CREATION FLOW



MANAGE FORMS

- Manage Forms are used to create forms in a specific application, Here we can create any number of forms in an application
- Manage form is a way to get into the functionality and succeed in handling the application

MANAGE VARIABLES

- These manage variables are used for variable creation and there is no possibility to exist a form without variables. By using some set of predefined relations, we can create variables.

MANAGE FORM VARIABLES

- Mandatory to select form in manage form.
- These manage form variables are used to bind the variables to the form

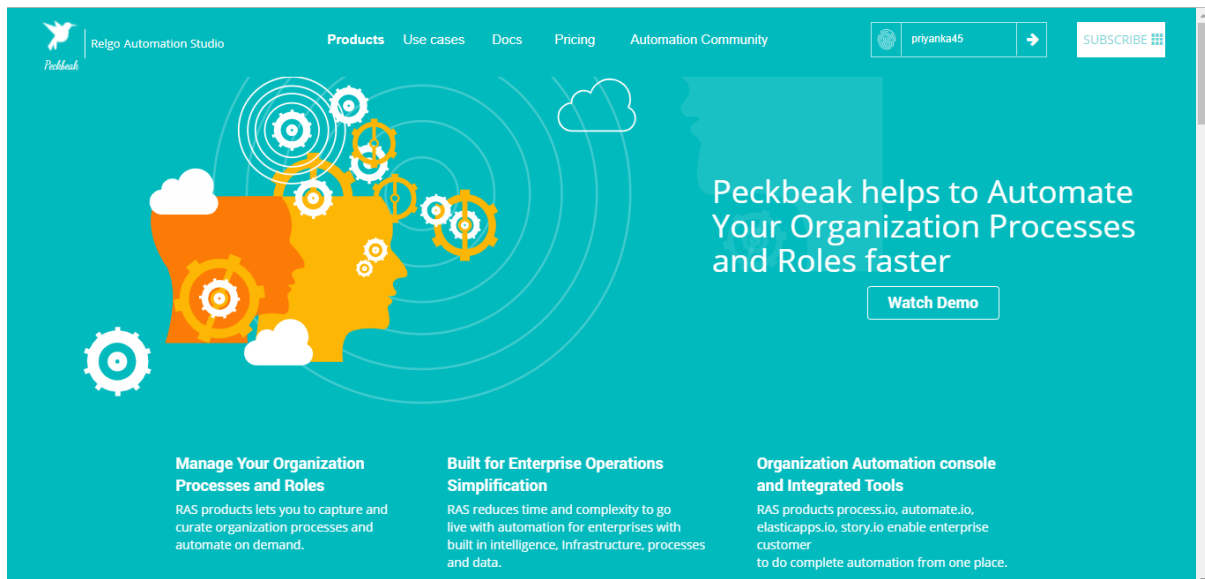
MANAGE FORM ACTIONS

- This section is used to create actions to a form
- A form without actions is useless and to make that form useful, we use this manage form actions.

MANAGE UX LAYOUT

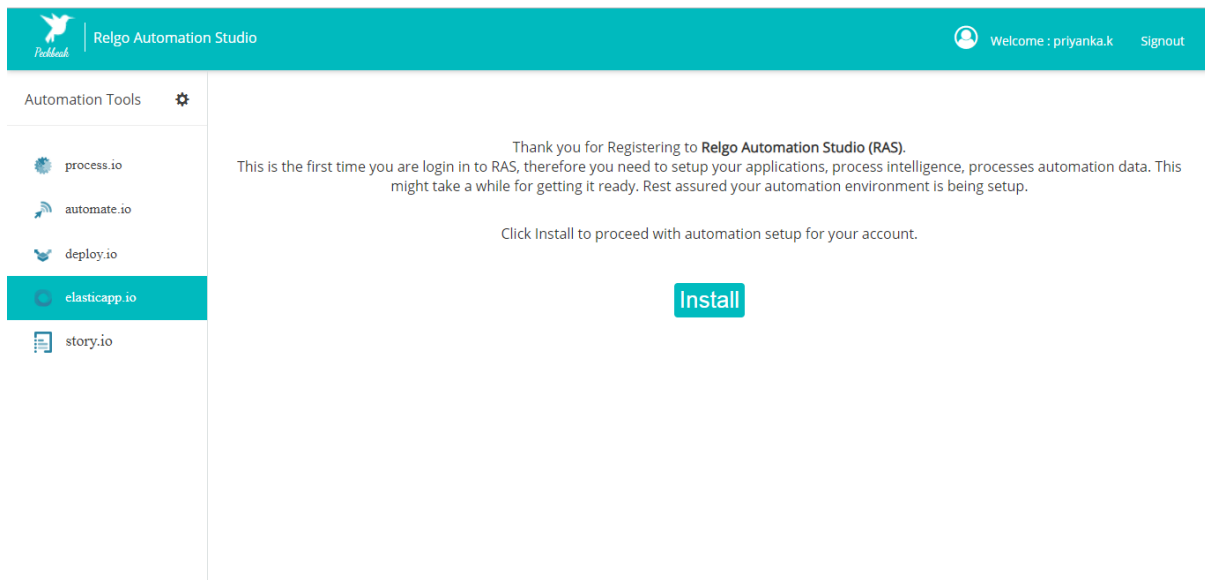
- This session is used to give richness to the form
- Layout is an empty template, based on requirement you can add variables to the layout.

Login to the account with your particular credentials



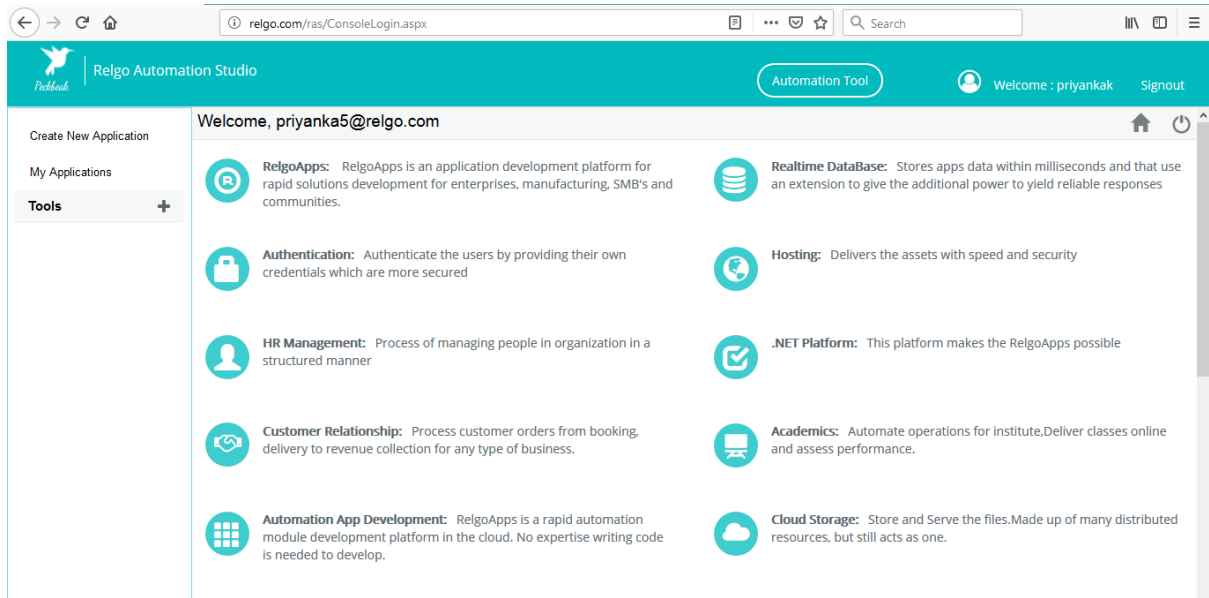
The image shows the landing page of the Peckbeak website. The header includes the Peckbeak logo, 'Relgo Automation Studio', and navigation links for 'Products', 'Use cases', 'Docs', 'Pricing', and 'Automation Community'. A user profile 'priyanka45' and a 'SUBSCRIBE' button are also visible. The main content area features a large graphic of a human head with gears and a cloud, symbolizing automation and intelligence. The headline reads 'Peckbeak helps to Automate Your Organization Processes and Roles faster', with a 'Watch Demo' button. Below this, three columns describe the platform's capabilities: 'Manage Your Organization Processes and Roles', 'Built for Enterprise Operations Simplification', and 'Organization Automation console and Integrated Tools'.

After logging in to the app with your credentials, you can see the below screen. Here you can see many tools, based on the user requirements they use their tools. To develop your own application, go to RAS Account → Elasticapp.io



The image shows the dashboard of the Relgo Automation Studio. The header includes the Peckbeak logo, 'Relgo Automation Studio', and a user profile 'Welcome : priyanka.k' with a 'Signout' link. The main content area is divided into two sections. On the left, there is a sidebar titled 'Automation Tools' with a gear icon, listing several tools: 'process.io', 'automate.io', 'deploy.io', 'elasticapp.io' (highlighted in teal), and 'story.io'. On the right, a message reads: 'Thank you for Registering to Relgo Automation Studio (RAS). This is the first time you are login in to RAS, therefore you need to setup your applications, process intelligence, process automation data. This might take a while for getting it ready. Rest assured your automation environment is being setup.' Below this message is a button labeled 'Install'.

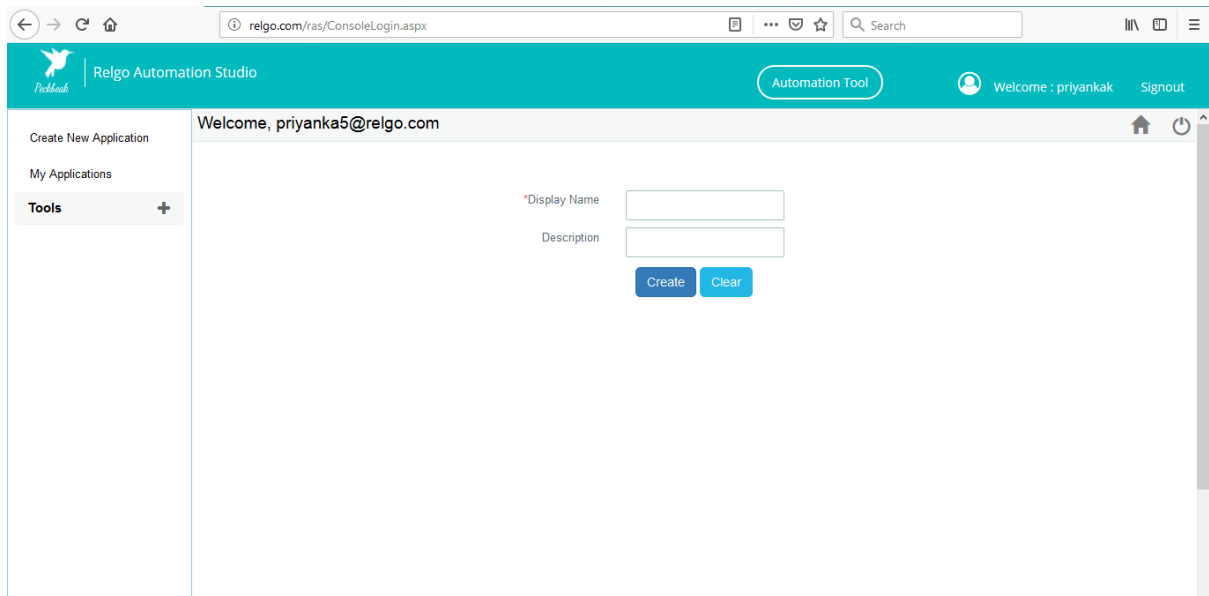
The below screen shows the home page of elasticapp.io



Application Creation

To create your own application, click on “Create New Application” which is shown below

In one application we can create number of related forms



- **Display Name:** It is nothing but application name.
- **Description:** we can add description if needed.

After entering the details click on “create” option then application will be created.

To find the created application, click on “My Applications”.

The screenshot shows a web interface with a sidebar on the left containing 'Create New Application', 'My Applications', and 'Tools +'. The main content area displays a table of applications. The table has columns for 'Expand', 'Application Name', 'Import Url', 'Description', 'App CreatedTime', and 'Editor'. There are five rows of data, each with an 'edit' button in the 'Editor' column.

Expand	Application Name	Import Url	Description	App CreatedTime	Editor
+	baseapps	Url		7/25/2018 11:03:57 AM	edit
+	hrmanagement	Url		7/25/2018 11:10:03 AM	edit
+	organizationmanagement	Url		7/25/2018 11:37:56 AM	edit
+	registrationapplication	Url	sample register form	7/25/2018 2:48:46 AM	edit
+	relgobusinessservices	Url		7/25/2018 11:07:40 AM	edit

To create your own form, select the created application which is shown in above screen.

To design the application, click on “Designer” which is shown in below screen.

The screenshot shows the 'Designer' tool interface. At the top, there is a navigation bar with icons for Dashboard, Designer, Source Control, Analytics, Provisioning, Automization, and Import&Export. Below the navigation bar is a form with the following fields:

- Application Name: Registration application
- Description: sample register form
- Application Owner: sampleappstore@relgo.con
- App Creation Time: 7/25/2018 2:48:46 AM

At the bottom of the form, there are two buttons: 'Update' (green) and 'Cancel' (blue).

Click on Designer then below screen appears.

The screenshot shows the application designer interface for the 'Registration application'. The top navigation bar includes 'FORMS', 'EXPRESSION BUILDER', 'WIDGETS', 'SEARCHSCOPE', 'MODULES', 'THEMES', and 'MODULEDEFINITION'. A dropdown menu is open under 'WIDGETS', showing the following options:

- ManageReports
- ManageForms
- Forms List
- New Widget

The left sidebar shows a navigation menu with 'Home', 'Application Dashboard', and 'Administrator' (expanded) with sub-items like 'Application Expressions - 0', 'Dashboard Widgets - 0', 'Email - 1', 'Gateway - 0', 'Modules - 0', 'Notification - 1', 'Search Scope - 6', 'SMS - 1', and 'Themes - 0'.

Select widgets → Manage Forms

In manage forms we have four steps to create a form. They are:

1. Manage Forms.
2. Manage variables.
3. Manage Form Variable.
4. Mange Form Action.

The flow to create forms, variables and buttons in an application.

Step 1-Manage Forms

This is used to create a form.

Go to RAS Account → Elastiapp.io → My Applications → Designer → Widgets → Manage Forms.

The screenshot shows the 'Manage Forms' interface. On the left is a sidebar with a tree view containing 'Application Dashboard', 'Administrator', 'Application Expressions - 0', 'Dashboard Widgets - 0', 'Email - 1', 'Gateway - 0', 'Modules - 0', 'Notification - 1', 'Search Scope - 6', 'SMS - 1', and 'Themes - 0'. The top navigation bar includes 'FORMS', 'EXPRESSION BUILDER', 'WIDGETS', 'SEARCHSCOPE', 'MODULES', 'THEMES', and 'MODULEDEFINITION'. Below this is a sub-navigation bar with 'Manage Forms', 'Manage Variables', 'Manage Form Variables', 'Manage Form Actions', and 'Manage UX Layout'. The 'Manage Forms' section contains the following fields and options:

- Form Name:
- Form Description:
- Existing Forms:
- Layouts:
- Form Type:
- Table Border: (optional)
- Module:
- Feature:
- Group Index:
- Form Index:
- Variables per Row:
- Text Align:
- Control Align:
- Validator Align:
- Data Collection:
- Mark Form As: Task Form Customizable

At the bottom are two buttons: 'Add Form' (purple) and 'Cancel' (blue).

In this step enter the details of the form location and alignment of the fields.

- **Form Name:** Desired form name will be given here
- **Form Description:** To give any additional information about the form.
- **Layouts:** Layouts is an optional field, if user wants any layout to the application then user can fill this field.
- **Form type:** There are two types of form types
 - One type.
 - Recursive.
 1. **One type:** If you wants to show the created form to end user for only one time then select as one type
 2. **Recursive:** If you wants to show the created form to end user for many times then select as Recursive.
- **Table Border:** This is used for Css File.
- **Alignments:** This will allow the user to select the alignment for the variables (like left, right or centre).
- **Feature:** There will be features on the dashboard. Select the feature in which the form should be displayed.
- **Group Index:**This is used to show the form in Dashboard. In order to maintain the sequence of groups we provide index to them.
- **Form Index:** This is used to show the form to a particular group.To maintain them in order we provide sequence number.
- Variables per row, Text Alignment, Control Alignment, Validator Alignment all these are useful for alignment purposes.
- **Data Collection:**There are two types of data collection
 - One type.
 - Recursive.
 1. **One type:** If you want to collect data for one time select type as one type

2. **Recursive:** If you want to collect data for many times select type as Recursive type

- **Task Form:** To display form on the dashboard, mark it as task form. Otherwise form will not appear on the dashboard.
- **Customizable:** To hide form on the dashboard, mark it as customizable.
- **Existing Forms:** Created forms will be displayed in existing forms.

Finally click on add form button. A pop-up will be displayed-
"FormAdded Successfully".

Step-2 Manage Variables

These manage variables are used for variable creation

Here we have two sections:

1. Manage Variables.
2. Manage Validators.

1. Manage Variables

In this some additional fields will be displayed based on selecting variable type. Variable name must be given and the remaining part of manage variables can be entered based on the requirement.

2. Manage Validators

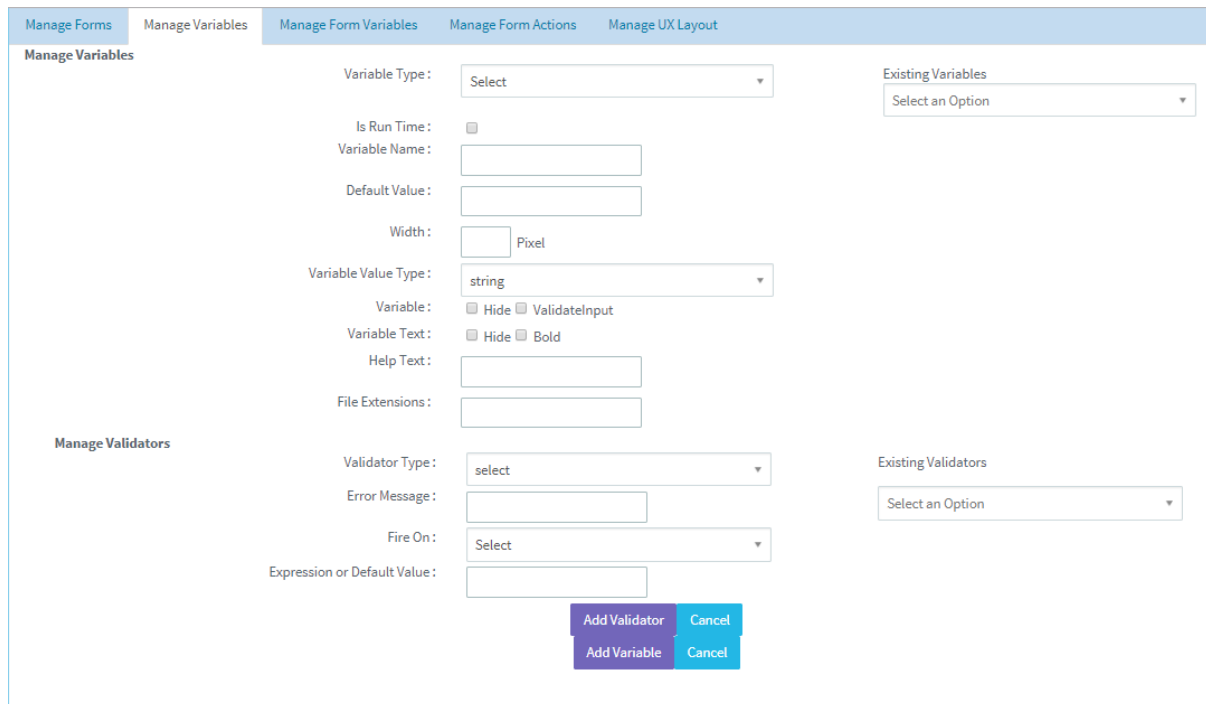
For one variable, you can create number of validators.

If we require Validators to the current variable select the Validator type and enter the Error message.

Variable Creation

This is used to create variables for the created form

Go to RAS Account → Elastiapp.io → My Applications → Designer → Widgets → Manage Variables.



Manage Variables

- **Variable Type:** This Variable Type has some predefined variables like Select line text box, single select, Multi line text box, Multi select, Multiple Text Box, Sequence number and so on...
These are the different types for binding the data to your variable.

When you select single line text box there you will get one parameter and in that there are some fields like...

1. **Result from a method:** If you don't know the exact data for the field.
2. **Custom Item:** If you know the values like male, female, others.
3. If you want to give provision to user to enter the data then don't select anything.

In Manage method bindings, there are two fields like select bind name and select binding value. For every method there is a return type, from that return type you can bind the values to variables.

- **Variable Name:** Enter Desired “Variable name” (Name).
- **Default value:** If you want to show default value to user then give the field of default value.
- **Variable Value Type:** This Variable value type has some predefined data types like int, string, double.
- If you want to hide that variable to user then select as “Hide” otherwise select as “Bold”.
- **Help Text:** If you want to give any instructions to the user to enter the data.

Manage Validator

- **Validator type:** This validator has some predefined validators like required field validator, regular express validator, range validator, custom validator, compare validator.
 - a. **Required field validator:** If you want to select this validator then you must enter the data in that field.
 - b. **Regular Express Validator:** If you want to validate data like email, phone number... for that purpose use this field.
 - c. **Range Validator:** This validator is applicable for range
E.g. Age cannot exceed 100
 - d. **Custom Validator:** If you want any requirement (like only upper case letters, lower case letters and no numbers) other than the above options then use this validator.
- **Error message:** If you want to show any message then fill this field.
- **Fire on:** If you want to show error message at the time of actions (submit) then select button submit which is shown in drop down.

For better understanding, there are some examples which are shown below.

Example-1: If we want to create a variable input text

- **Variable Type:** Select “Single Line Text Box”.
- **Variable Name:** Enter Desired “Variable name” (Name).
- **Variable Value type:** select “String”.
- **Variable:** select the “valid input” checkbox.

Provide “Help Text”.

Finally click on add variable button, a pop-up will be displayed as “Variable added successfully”. After adding the variable it will display in existing variables.

Example-2: If we want to create a variable type date

- **Variable Type:** Select AnsDateTime
- **Variable Name:** Enter Desired Variable name.
- **Variable Value type:** select Date.
- **Variable:** select the valid input checkbox.

Provide Help Text and Select “Date Format” mm/dd/yy, dd/mm/yy.

Condition: Specifies whether calendar date should be greater (or) lesser than the current date.

Finally click on “add variable” button, a pop-up will be displayed as ‘Variable added successfully’. After adding the variable it will be displayed in existing variables.

Example-3: If we want to create a variable type number

- **Variable Type:** Select “phone number”.
- **Variable Name:** Enter Desired “Variable name” (phone no).
- **Variable Value type:** select “int”.
- **Variable:** select the “valid input” checkbox.

Provide “Help Text”.

Finally click on add variable button, a pop-up will be displayed as “Variable added successfully”. After adding the variable it will be displayed in existing variables.

Example-4: If we want to create a variable type as sequence number, there is a procedure which is shown below.

Sequence number

Sequence number is a special case where it should be automatically generated and assigned to any item in the application. Sequence Number has some unique properties to generate.

PROCEDURE

For this we need to go for following procedure:

- Create New Series Format.
- Create New Series.
- Create Scope.
- Create Variable.

The Following forms which are appearing in the dashboard will be imported to your account by administrator.

NEW SERIES FORMAT

Details of how sequence should appear will be mentioned here.

Fill the fields of that form and click on “Add Format”

Go to dashboard → Series → New Series Format.

The screenshot shows a web application interface for creating a new series format. On the left, there is a sidebar menu with items: conditionalreport, City, Country, Manage Scope, Manage Scope, Country report view, State, Register, and Series. The main content area is titled 'Series > New Series Format'. The form contains the following fields and controls:

- *Name:** A text input field with a blue border and an information icon.
- Include Branch Code:** A dropdown menu currently set to 'Yes'.
- *Prefix:** A text input field with an information icon.
- *Length:** A text input field.
- Regular Expression:** A text input field with an information icon.
- *Description:** A text input field with an information icon.

At the bottom of the form, there are two buttons: a blue 'Cancel' button and a purple 'Add Format' button.

We are trying to generate alphanumeric sequence for “Order No.”. If we require branch code select “yes” and the value of branch code will be given in “**prefix**”. The length of the Order Number will be declared here.

FIRST PREVIOUS 1 NEXT LAST

Export Delete Relation

Name	Include Branch Code	Prefix	Length	Regular Expression	Description	ACTION
Leave Request Format	Yes	LR	6		Sequence of leave request	   

CREATE NEW SERIES

“**New Series**” form is used to mention the Sequence range.

Go to → Dashboard → Series → New series.

conditionalreport

City

Country

Manage Scope

Manage Scope

Country report view

State

Register

Series

Series > New Series

Select Form

*Format

*Starting Number




*Ending Number

Status

Select the form to which you want to map this “Series Format” and then select previously created “Series Format”. Enter the starting and ending number of the series. Finally click on “**Add**” button, a pop-up will be displayed like “**Series added successfully**”.

FIRST PREVIOUS 1 NEXT LAST

Export Delete Relation

Select Form	Format	Starting Number	Ending Number	Status	Issued Numbers	Hold Numbers	ACTION
Leave Request	Leave Request Format	0001	9999	Running			   

CREATE SCOPE

To get the “Series Format” into “Employee Id.” variable we write scope.

To create scope go to Dashboard→Manage Scope→Scope.

*Scope Name	<input type="text" value="LeaveRequestseq"/>
Scope Type	<input type="text" value="RelationsSearch"/>
Scope Owner	<input type="text" value="Requested User"/>
Requested User	<input type="text" value="itadministrator@relgo.com"/>
Scope Criteria	<input type="text" value="Item"/>
Deleted Relations	<input type="text" value="Select an Option"/>
Select Query	<input type="text" value="Select an Option"/>
Criteria Type	<input type="text" value="Select an Option"/>
Select Form	<input type="text" value="Select an Option"/>
Variable	<input type="text" value="Select an Option"/>
Operator	<input type="text" value="Select an Option"/>
Value	<input type="text"/>

Scope Name:Give a unique scope Name.

Scope Type:Select scope type which decides from where we need to get the data into the variable.

Scope Owner:select the Application owner.

Requested User:Give the user address based on selection type of scope owner.

Scope Criteria:If your “Scope Type” is Group, enter the group name in “Scope Criteria”

To fetch the data with some conditions, specify the conditions in scope. After filling the required fields click on “Add” button, a pop-up will be displayed like “**Scope added successfully**”.

VARIABLE CREATION

This is used to create variables for the created form

Go to RAS Account → Elastiapp.io → My Applications → Designer → Widgets → Manage Variables.

The screenshot displays the 'Manage Variables' configuration page. It features a navigation bar with tabs for 'Manage Forms', 'Manage Variables', 'Manage Form Variables', 'Manage Form Actions', and 'Manage UX Layout'. The main content area is divided into two sections: 'Manage Variables' and 'Manage Validators'. The 'Manage Variables' section includes fields for 'Variable Type' (SequenceNumber), 'Select Source Variable' (Name - d22bc65d-d61e-4753-983e-74ce07a36a99), 'Select Form' (Series), 'Select Scope' (employeiddetails), 'Is Run Time' (checkbox), 'Variable Name' (Employeeid), 'Default Value', 'Width' (Pixel), 'Variable Value Type' (string), 'Variable' (Hide, Validatelnput), 'Variable Text' (Hide, Bold), and 'Help Text'. The 'Manage Validators' section includes 'Validator Type' (select), 'Error Message', 'Fire On' (Select), and 'Expression or Default Value'. On the right side, there are dropdown menus for 'Existing Variables' (Employeeid - 3d2a2819-08a3-48ef-9cd6-a449ade3372e) and 'Existing Validators' (Select an Option). At the bottom, there are buttons for 'Add', 'Cancel', 'Update Variable', 'Delete Variable', and 'Cancel'.

Variable Type:Sequence Number

Select Source:Based on series format name we fetch the series format into current variable.That’s why we select source as “Name”.

Select Form:This “Name” variable is declared in “Series” form, that’s why we select series form.

Select Scope:Select the scope here.

Variable:Enter desired name for variable.

Variable Value Type:Select as String and select “valid input” checkbox.

Finally click on “add variable” button, a pop-up will be displayed as “**Variable added successfully**” After adding the variable, it will displayed in existing forms.

Similarly create the remaining variables.

Step 3-Manage Form Variables

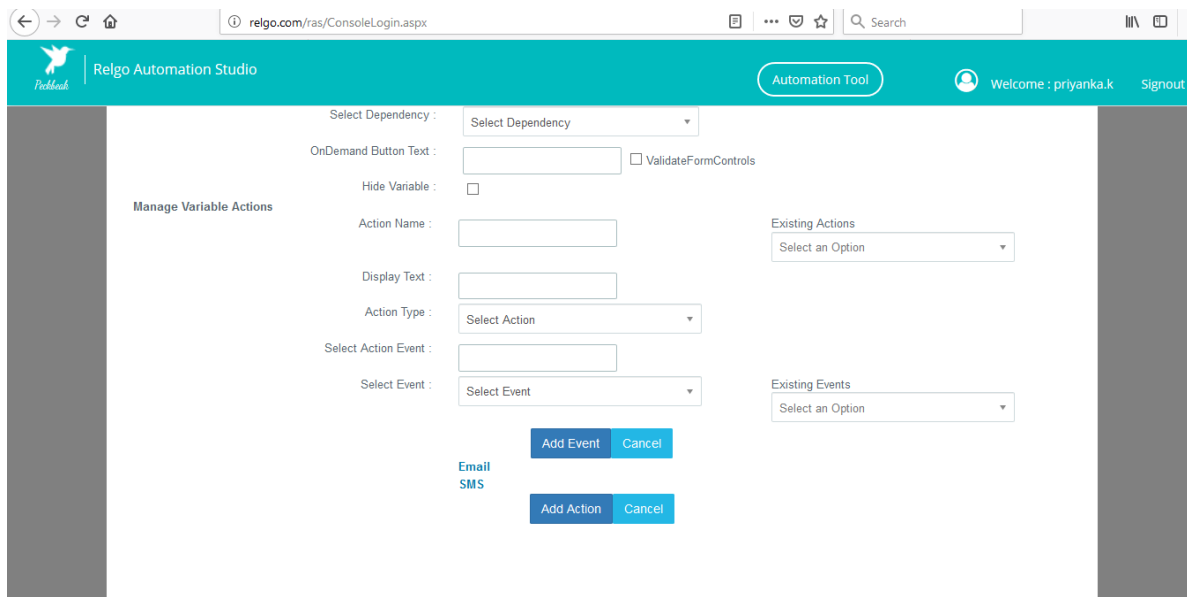
- Mandatory to select form in manage form.
- These manage form variables are used to bind the variables to the form

The variables present in Manage form variables are specific to one form.

Go to RAS Account → Elastiapp.io → My Applications → Designer → Widgets → Manage Form Variables.

- **Select variable:** The variables defined in step-2 will be displayed in this drop-down.
- **Tab index:** In one form we have number of variables (fields). To set the position of each variable, we use tab index.
- **Display text:** Variable name will be changed using this field.
- **Variable Insert Position:** To arrange the multiple variables in order.
- **Alignment:** It is used to set the variables in the same format.
(If you want to change the alignment which you have given in manage form then these text align, control align and validator align are used.)

When you click on “show variable properties” check box, a popup will be displayed which is shown below.



The screenshot shows the Relgo Automation Studio interface. The browser address bar displays 'relgo.com/ras/ConsoleLogin.aspx'. The page header includes the Relgo logo, 'Relgo Automation Studio', an 'Automation Tool' button, and a user profile for 'Welcome : priyanka.k' with a 'Signout' link. The main content area is titled 'Manage Variable Actions' and contains the following form elements:

- Select Dependency :** A dropdown menu with 'Select Dependency' selected.
- OnDemand Button Text :** A text input field.
- ValidateFormControls**
- Hide Variable :**
- Action Name :** A text input field.
- Existing Actions :** A dropdown menu with 'Select an Option' selected.
- Display Text :** A text input field.
- Action Type :** A dropdown menu with 'Select Action' selected.
- Select Action Event :** A text input field.
- Select Event :** A dropdown menu with 'Select Event' selected.
- Existing Events :** A dropdown menu with 'Select an Option' selected.
- Add Event** and **Cancel** buttons.
- Email** and **SMS** options.
- Add Action** and **Cancel** buttons.

DEPENDENCY RELATION

Dependency Relation means one variable value depends on another variable value.

E.g.: Consider a form which is called register and in that form there are some fields like state & city. Based on the chosen state, city list will vary i.e. for the state.

➤ A.P cites are Guntur, Vijayawada etc.,

This is called Dependency Relation i.e. cities are dependent on the state variable.

When you select dependency, there you will get a drop down

On Load: At the time of page rendering, if you want to fetch the data

On Selected: If you want to bind the data to a variable based on the selection of another variable.

On Demand: If you want to fetch the data by clicking on some button.

Manage Variable Actions

If you want to perform any actions then use manage variable actions.

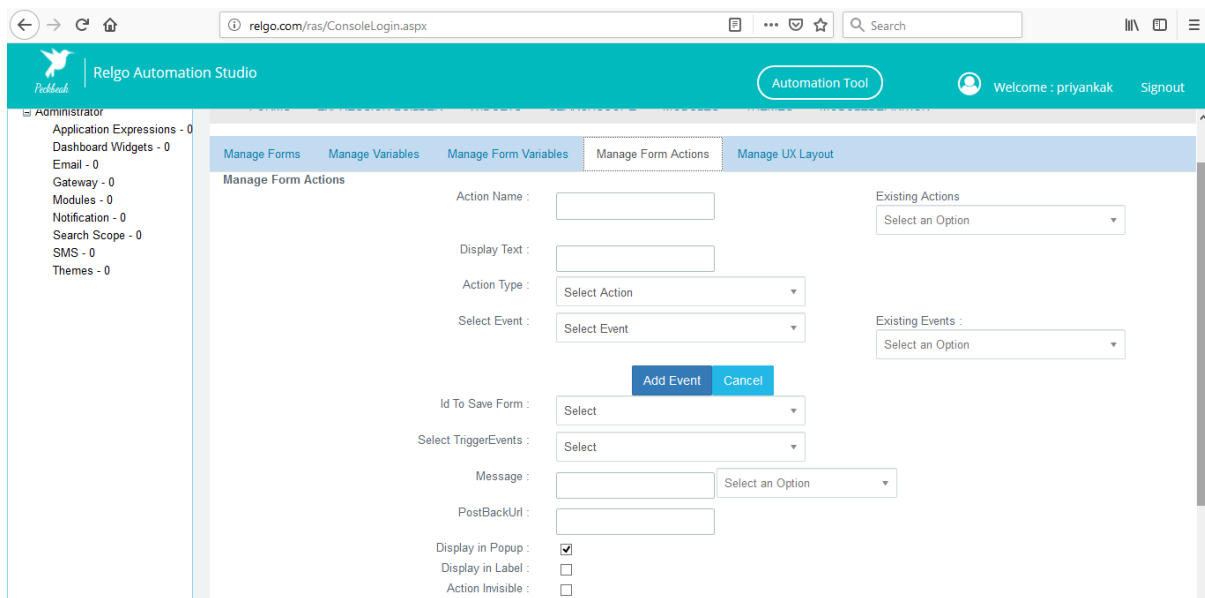
Action Type: If you want to store the data in another relation then provide action type.

Event: It is nothing but the function, by using this function you can fetch the data from data base (or) insert data in to data base.

Step 4- Manage Actions

- To perform any operation related to the filled form for that we require actions.

Go to RAS Account → Elastiapp.io → My Applications → Designer → Widgets → Manage Form Actions.



The screenshot shows the 'Manage Form Actions' interface in the Relgo Automation Studio. The browser address bar shows 'relgo.com/ras/ConsoleLogin.aspx'. The application header includes the 'Relgo Automation Studio' logo, an 'Automation Tool' button, and a user profile for 'Welcome : priyankak' with a 'Signout' link. The left sidebar lists various application components like 'Administrator', 'Application Expressions', 'Dashboard Widgets', 'Email', 'Gateway', 'Modules', 'Notification', 'Search Scope', 'SMS', and 'Themes'. The main content area has tabs for 'Manage Forms', 'Manage Variables', 'Manage Form Variables', 'Manage Form Actions' (selected), and 'Manage UX Layout'. The 'Manage Form Actions' section contains the following fields and controls:

- Action Name :** Text input field.
- Display Text :** Text input field.
- Action Type :** Dropdown menu with 'Select Action'.
- Select Event :** Dropdown menu with 'Select Event'.
- Existing Actions :** Dropdown menu with 'Select an Option'.
- Existing Events :** Dropdown menu with 'Select an Option'.
- Buttons:** 'Add Event' (blue) and 'Cancel' (grey).
- Id To Save Form :** Dropdown menu with 'Select'.
- Select TriggerEvents :** Dropdown menu with 'Select'.
- Message :** Text input field and a dropdown menu with 'Select an Option'.
- PostBackUrl :** Text input field.
- Display in Popup :**
- Display in Label :**
- Action Invisible :**

The screenshot displays the Relgo Automation Studio interface. The top header includes the Relgo logo, the text 'Relgo Automation Studio', and a navigation bar with 'Automation Tool', 'Welcome : priyankak', and 'Signout'. The main workspace is divided into a left sidebar with 'SMS - 0' and 'Themes - 0', and a central configuration area. The configuration area contains several fields and controls:

- Display text :** A text input field.
- Action Type :** A dropdown menu with 'Select Action'.
- Select Event :** A dropdown menu with 'Select Event'.
- Existing Events :** A dropdown menu with 'Select an Option'.
- Add Event** and **Cancel** buttons.
- Id To Save Form :** A dropdown menu with 'Select'.
- Select TriggerEvents :** A dropdown menu with 'Select'.
- Message :** A text input field and a dropdown menu with 'Select an Option'.
- PostBackUrl :** A text input field.
- Display in Popup :**
- Display in Label :**
- Action Invisible :**
- Print Form :**
- Is Integration Point :**

At the bottom, there are tabs for 'Email', 'SMS', 'IntegrationTemplate', and 'Notification'. Below these tabs are 'Add Action', 'Cancel', and 'Finish' buttons.

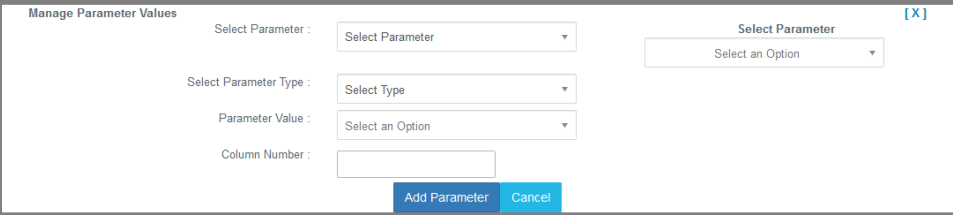
- **Action name:** Give action name as submit button, cancel button, update button.
- **Display text:** It is used to show the form name in dashboard (submit).
- **Action type:** In this type, there is a drop down list like form submit, form update, form delete etc.
 - i. **Form Submit:** It is used to get the data in to the form and save that data to a database.
 - ii. **Form Update:** It is used for modifying any data.
 - iii. **Form Delete:** It is used to delete the form all over record.
 - iv. **Form Cancel:** It is used to reset the form.
 - v. **View Form:** It is used to view the data.
 - vi. **Details View:** This is other view for form details.
 - vii. **Print Form:** It is used to print the form.
 - viii. **Save to File:** This is used for saving the form to other format.

After the selection of action type, two “multi - select text boxes” will appear. In one box you can select hidden variables and in another one you can select non-editable variables.

In order to save the form data we have to create an event

- **Select Event:** Here number of methods has defined to save/ hold the date.

When you select “Add Relation”, a popup will be displayed which is shown below.



The screenshot shows a modal window titled "Manage Parameter Values" with a close button "[X]" in the top right corner. The window contains the following fields:

- Select Parameter :** A dropdown menu with "Select Parameter" as the selected option.
- Select Parameter Type :** A dropdown menu with "Select Type" as the selected option.
- Parameter Value :** A dropdown menu with "Select an Option" as the selected option.
- Column Number :** An empty text input field.

At the bottom of the window, there are two buttons: "Add Parameter" (highlighted in blue) and "Cancel".

Select parameter: In this type, there is a drop down list like

1. **Additional Data:** This is used to add any additional information.
2. **Group Id:** This is used to store the data in a group which you want to add.
3. **Source address:** Source address is nothing but your own address.
4. **Relation Type:** In which relation you want to save this data.

Select parameter Type: In this type, there is a drop down list like

1. **Form Variable:** What are the variables present in the form
2. **Form:** Form present in your application.
3. **Group:** Groups present in both application and admin account.
4. **Networks:** Networks present in both application and admin account.
5. **Relgo Variables:** Variables present in your application.
6. **Method Output:** This is used to pass method output data.

Based on parameter, you can add parameter value then click on “Add parameter” which is shown in above screen.

This is about Add Relation.

Id to Save Form: In a form there are multiple relations, to which relation you want to save the data.

Trigger events: If you want to continue this flow in server set.

Post back URL: If you want the user to redirect to other website after clicking on “Submit” button.

After completion of filling the fields click on “Add Action” button which is shown in above screen.

Let us see how to add this relation using Relgo Apps

Procedure

- Create three forms in an application.
- Consider the forms as Add Country, Add State, and Account form.

Form 1: Add country

I. Manage Form

- Name the form as Add Country and Choose the Task Form and click on “Add form” which has explained in manage form.

II. Manage Variables

- Select variable type as single line textbox and variable name as Add Country which has explained in manage variables.

Manage Variables

Variable Type:

Is Run Time :

Variable Name :

Default Value :

III. Manage Form Variables

- Fill the fields and click on “add” and add the tab index as 1 which has explained in manage form variables.

IV. Manage Form Actions

- Enter the fields as follows

Action Name :

Display Text :

Action Type :

In select event tab choose the event as **Add Relation**

Select Event :

- AddMultipleUserLicense
- AddBasicSubscription
- ChangePassword
- UpdateCSS
- SendNotificationToAdministrator
- InvokeInternalMethod
- UpdateInvoiceStatus
- AddRelation**
- AddRelationsAndTags
- AddModule
- AddModuleGroup
- AddRelationsToGroupMap
- DeleteRelation
- DeleteRelationFromGroup
- DeleteRelationEntries
- DeletePosition
- DeleteRole

A dialog window appears as follows:

Manage Parameter Values [X]

Select Parameter :

Select Parameter Type :

Parameter Value :

Column Number :

Select Parameter

Select an Option

String : targetname

String : targetaddress

String : sourceaddress

String : groupid

String : relationtype

Cancel

Enter the values as:

Parameter	Type	Value
Target Name	Form Variable	Add Country
Target Address	Value	Null
Source Address	Value	sampleappstore@relgo.com
Group id	Group/value	Choose group name/ Enter Group Name
Relation Type	Value	Item
Form id	Form	Select the form

After completion of four steps click on “Add Event & Add Action” buttons. Choose the Application Dashboard and run the created form. Enter the country names.



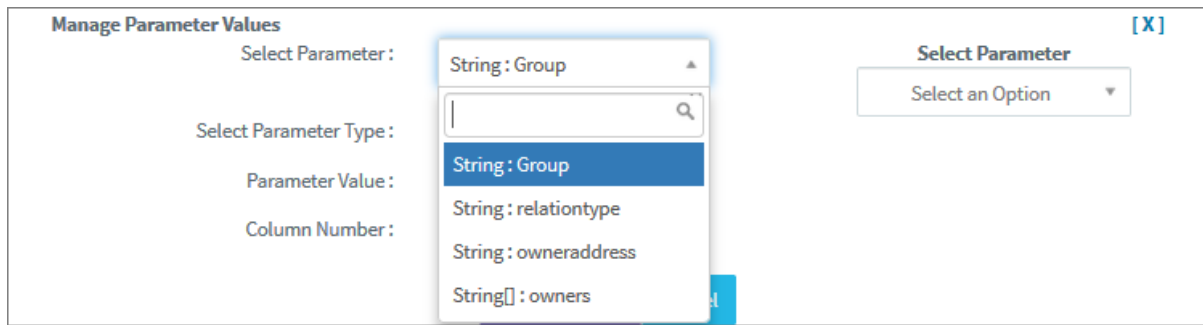
Form 2: Add States

I. Manage Form:

- Name the form as Add States and choose the task form and click on “Add form”.

II. Manage Variables:

- Create a variable name as “select country” which is of “Single Select”.
- Select the values as “Result from a method” and choose the method as “Get Group Members” then a dialog box will appear which is shown below.



Enter values as:

- ✓ Group-group-select a group.
- ✓ Relation type-value-enter relation of add country.
- ✓ Owner [Address-value-sampleappstore@relgo.com](mailto:sampleappstore@relgo.com)
- ✓ Bind name – target name
- ✓ Bind value – target address

III. Create another variable

- Variable name as “states” and select it as “Single Line TextBox”.

IV. Manage Form Variables

- Select variable:
 - ✓ Select Country tab index as 1.
 - ✓ Select State tab index as 2.

V. Manage Form Actions:

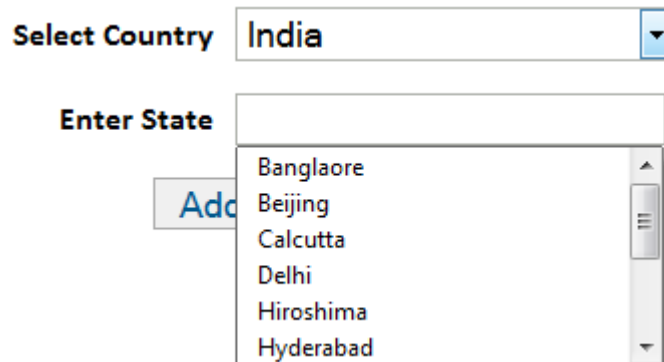
- Enter the action name, display text & action type as usual.
- Select event as “Add Relation”and enter values as:

Parameter	Type	Value
Target Name	Form Variable	Enter State
Target Address	Value	Null
Source Address	Form Variable	Select Country
Group id	Group	Choose group name
Relation Type	Value	Item
Form id	Form	Add States

Note:

For source address we had given the value as “Select Country” because country is the source for the variable state. But in the country form the value given as user mail-id (as user is the owner for country).

Application Dashboard-> Add states.



The image shows a web form interface. At the top, there is a dropdown menu labeled "Select Country" with "India" selected. Below it, there is an input field labeled "Enter State" which is open, showing a list of states: Bangalore, Beijing, Calcutta, Delhi, Hiroshima, and Hyderabad. To the left of the "Enter State" field, there is a button labeled "Add".

Form 3:Account

I. Manage Form:

- Name the form as “Add States” and choose the task form and click on “Add Form”.

II. Manage variables:

- Enter variables like name, e-mail id, address, phone number etc.
- Now we need to import the country and state variables into this account form.

a) **Country variable:**

- ✓ Create variable

- ✓ Select Country as type Single Select, import values as **Result from a method.**
- ✓ Enter the parameter values as:
 - Group-group-select a group
 - Relation type-value-add country relation type
 - Owner Address-value-sampleappstore@relgo.com
 - Binding name-target name
 - Binding address-target address

b) **State Variable:**

- ✓ create variable
- ✓ Select state as type single select.
- ✓ Enter the item name as First choose Country.

III. Manage Form Variables:

- Select variables and add tab index for each variable in required order.
- But for state variable add the “dependency” by choosing the variable from existing variables and set the values as
 - ✓ Group-group-choose a group
 - ✓ Relation type-value-item
 - ✓ Owner address-form variable-select country*

Properties [X]

Select Dependency : OnSelected

OnDemand Button Text : ValidateFormControls

Select Parent Variable : select country

Select Method : GetGroupMembers

Manage Parameter Values

Select Parameter : String : owneraddress String : owneraddress

Select Parameter Type : FormVariable

Parameter Value : select country

Column Number :

Select Bind Name : TargetName

Select Binding Value : TargetAddress

Hide Variable :

Manage Variable Actions

Action Name :

Existing Actions :

Display Text :

Action Type : Select Action

Select Action Event :

IV. Manage Form Actions:

- Now enter Submit for action & click for display text and choose add action which is shown in below screen.

Name

E-mail id

Address

Select Country* India

select state Calcutta

Employed Yes

Note:

Here based on country, states appear thus the dependency relation works.