

TRANZEVO

**ADVANCED
MULTISELECT
LOOKUP**
LIGHTNING COMPONENT



Developer & Admin Guide



Contents

1. Introduction	3
2. Installation	4
3. Custom Metadata Details	6
4. How to use the Lookup Component?	10
5. Key Features	13
6. Dependent Lookup Component.....	14
7. Examples of Component Usage in Code	16



1. Introduction

Salesforce comes up with many out of box UI components and Lightning Base components to make developers life easy. The Look up filter is one such essential feature, widely used in classic but wasn't there in lightning.

We have created a lightning component which is one step ahead of the normal lookup and gives you advanced version of the lookup component which makes development process much easier and faster. It has all the basic features of any lightning lookup plus it allows selection of multiple records. Multiple record selection will give us more flexibility when we need to create junction object records from a custom component. Some of the examples include adding invitees to event, link multiple contacts to an Account, etc.

The component is metadata driven. So, no more code changes are required when we need to update the lookup configurations like, updating the search criteria, updating fields to be displayed on UI, etc. The package also contains a dependent lookup component which will help to configure the results of one lookup to be based on results of another lookup (Parent – Child Relationship).

This component supports standard as well as custom objects. It is also supported in Salesforce1.



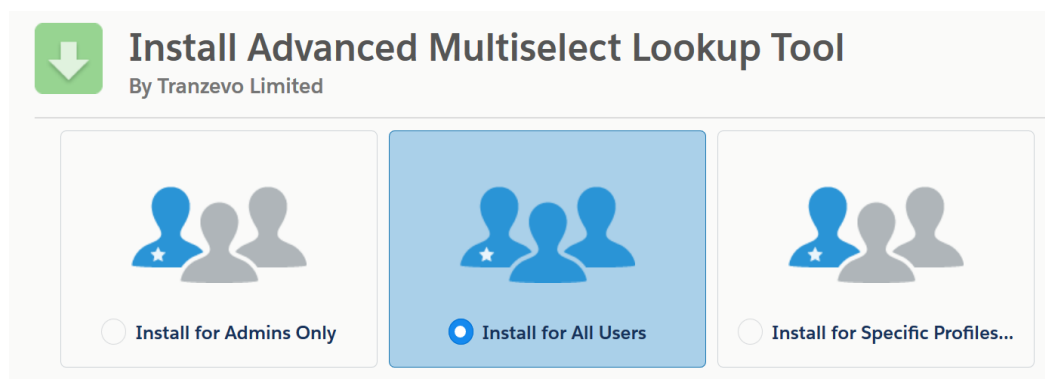
2. Installation

1. Install the Advanced Multiselect Lookup Tool managed package from link here, <https://login.salesforce.com/package/installPackage.apexp?p0=04t2w000007EMWR>

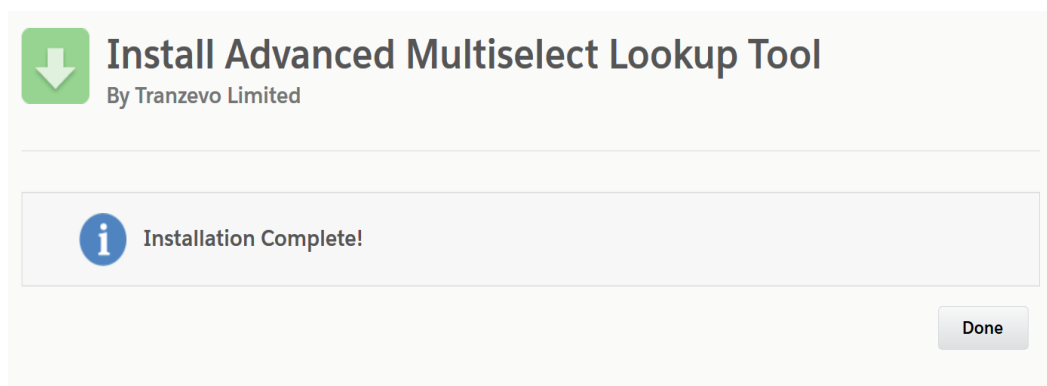
Or install the package from app exchange <<AppExchange listing URL>>

We recommend installing this component package from AppExchange so that you will get the latest version of the component.

2. Select **Install for All Users**



3. Click on **Done** once the installation is completed. That is all you need to do. The component is ready to be used.





4. To verify if the Installation is completed successfully, check if the packages components are available in your org.

Navigate to Setup → Custom Metadata → Verify if Custom Lookup Setting is present in the org.

New Custom Metadata Type						
Action	Label	Namespace Prefix	Visibility	Api Name	Record Size	Description
Del Manage Records	Custom Lookup Setting	TnzMultilookup	Public	TnzMultilookup__Custom_Lookup_Setting__mdt	685	

5. Create a permission set to enable access to the component's controller. This permission set should be assigned to the users who will be using the Advanced Multiselect Lookup Tool.
 - a. Navigate to Setup → Permission Sets → New
 - b. Set Permission Set Label as **"Advanced Lookup Access"**
 - c. Set Permission Set API Name as **"Advanced_Lookup_Access"**
 - d. Set Permission Set Description (Optional)
 - e. Select **None** in the License section and Save.

Save Cancel

Enter permission set information

Label

Advanced Lookup Access

API Name

Advanced_Lookup_Access

Description

Provides access to advanced multi-select lookup.

Session Activation Required

☐

Select the type of users who will use this permission set

Who will use this permission set?

-Choose '--None--' if you plan to assign this permission set to multiple users with different user and permission set licenses.
 -Choose a specific user license if you want users with only one license type to use this permission set.
 -Choose a specific permission set license if you want this permission set license auto-assigned with the permission set.

Not sure what a permission set license is? [Learn more here.](#)

License

--None--

Save Cancel

- f. Navigate to **Apex Class Access** → Add **TNZMultilookup.CustomLookUpComponent__c** to the Enabled Apex class list.



Settings that apply to Salesforce apps, such as Sales, and custom apps built on the Lightning Platform
[Learn More](#)

Assigned Apps Settings that specify which apps are visible in the app menu
Assigned Connected Apps Settings that specify which connected apps are visible in the app menu
Object Settings Permissions to access objects and fields, and settings such as tab availability
App Permissions Permissions to perform app-specific actions, such as "Manage Call Centers"
Apex Class Access Permissions to execute Apex classes
Visualforce Page Access Permissions to execute Visualforce pages
External Data Source Access Permissions to authenticate against external data sources
Flow Access Permissions to execute Flows
Named Credential Access Permissions to authenticate against named credentials
Custom Permissions Permissions to access custom processes and apps
Custom Metadata Types Permissions to access custom metadata types

Apex Class Access Save Close

Available Apex Classes	Enabled Apex Classes
LinkContactToMultipleAccounts_CC TnzMultilookup.CustomLookupComponent_CC_ UpdateAccountOnContact_CC	TnzMultilookup.CustomLookUpComponent_CC

Add
 Remove

3. Custom Metadata Details

The Custom Lookup Setting metadata plays an important role in configuring the lookup field to be used. All the configurations of the lookup field are defined in a single metadata record.

Below are the field details for Custom Lookup Setting metadata:

1. Label

This is the standard field. It refers to the metadata record label.

2. Custom Metadata Record Name (Developer Name)

This is a standard field and it refers to the Developer Name of the metadata record. This field is unique and it will be used while using the lookup component.



Custom Lookup Setting Detail [Edit] [Delete] [Clone]

Label: Account Lookup

Custom Lookup Setting Name: Account_Lookup

Maximum No of Records: 1

Allowed to Select: []

Object Name: Account

Field Text: Name

Placeholder: Type here to search for an Account

```
<aura:attribute name="accList" type="Object[]" />
<TnzMultilookup:CustomLookupComponent metadataName = "Account_Lookup" selectedRecordList="{!v.accList}" />
```

Lookup Icon: standard goals

Condition String: Rating=1to5

Lookup Field Label: Account

Relationship Field Name: (Parent)

3. Maximum No of Records Allowed to Select

Used to specify maximum number of records that can be selected from list. If this value is not set, then user can select as many records as he/she wants. (Maximum up to – 50)

Enter space after text to search for Accounts

Edge Communications X United Oil & Gas Corp. X

United Oil & Gas, Singapore X

Here we specified numberOfEntries as 3
So User can Only select 3 records

4. Object Name

API Name of the Object to be searched. This can be any custom or standard object.

5. Field Text

API Name of the field, to be used to show text on the search UI.

Account

Edge C

Edge Communications
http://edgecomm.com

Here the value for **Field Text** is Name.
Hence Account Name is displayed.

6. Placeholder Text

To specify the placeholder text for input field.



Account

Type here to Search for Accounts

7. Additional Field

Additional field to be displayed in search layout. Currently only one field can be used as additional field on the search layout.

Account

Edge Comm

Edge Communications
http://edgecomm.com

The value for Additional Field is set to Website. Hence the Account website is displayed on search layout.

8. Field Value

API Name of the field, to be returned from component. This value needs to be Id of the record but if user wants, he/she can use any of the other fields as well. This value can be used by the parent component for further processing. Note: This value will not be displayed on the UI

9. Maximum No of Records to be Displayed

Total number of records to be returned from the apex. It is defaulted to 5.

Account

United

United Oil & Gas Corp.
http://www.uos.com

United Oil & Gas, Singapore
http://www.uos.com

United Oil & Gas, UK
http://www.uos.com

Here the field value is set to 3. Hence three records will be display which contains the word **united**.

10. Record Type

To apply filter based on the record types, For. Ex if user wants to filter all the accounts having record type as "Individual" or "Business" put value as ('Business', 'Individual').

Note: The value used should be API Name of the Record Type.



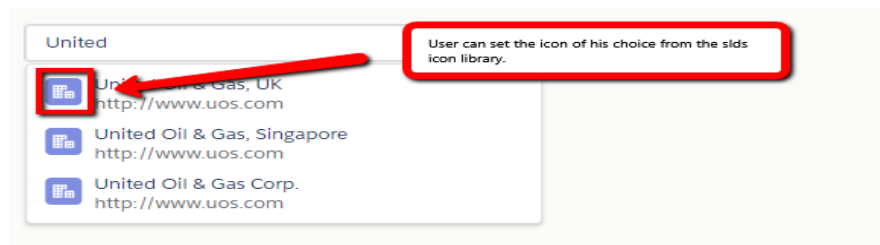
11. Search Field

API Name of field to be searched. This field can be any Text, Phone or Email field from the Object.

12. Lookup Icon

Used to hold icon name to display the specific Icon. Icons can be found at <https://www.lightningdesignsystem.com/icons/>

For Ex. To show Account standard icon, you can specify 'standard:account' or to display the goal icon you can specify 'standard:goals'



13. Condition String

To specify conditions for WHERE clause in SOQL query. User can specify condition string according to use case.

Here are some examples of condition string:

Sr No.	Criteria	Condition String
1	Filter Active account with Ratings Hot. Here Active__c is custom field on Account Object	<i>Active__c = 'Yes' AND Rating = 'Hot'</i>
2	Filter Active Products on a Product Lookup	<i>IsActive = true</i>
3	Filter contact records with either of Email or Phone populated.	<i>Email!=Null OR Phone!=Null</i>

14. Relationship Field Name (Parent)

This field is used when user wants to create dependency between 2 lookup components (Parent-Child relationship). Specify the API Name of the Parent Lookup field.

Use case: Fetch the contact records based on the selected account. Here we can create first lookup as lookup to Account and second lookup as lookup to Contact. Specify the relationship name as "AccountId"



Considerations:

- a. Only the objects having lookup or master detail relationship can be used to create dependencies.
- b. While creating dependency the Parent Lookup Component's "Maximum No of Records Allowed to Select" is already set to 1 from the backend. Hence only one parent record will be allowed to select. Refer [DependentLookupComponent](#) Section for further details.
- c. A custom metadata record with DeveloperName as **AccountLookup** is included in the package with default settings for Account object's lookup. Avoid using **AccountLookup** as a DeveloperName for the custom metadata as the default settings will override them.
- d. Mandatory fields while creating a metadata record
 - **Field Text**
 - **Field Value**
 - **Search Field**
 - **Label**
 - **Custom Metadata Name**
 - **Object Name**

Validations: Field Text and Field Value cannot be the same fields.

4. How to use the Lookup Component?

1. Create a Custom Metadata Record for Custom Lookup Setting

- a. Navigate to **Setup**.
- b. Search and **Select Custom Metadata Types**.
- c. Select **Custom Lookup Setting**.

New Custom Metadata Type							
Action	Label	Installed Package	Namespace Prefix	Visibility	Api Name	Record Size	Description
Manage Records	 Custom Lookup Setting	Advanced Multiselect Lookup Tool	TnzMultilookup	Public	TnzMultilookup__Custom_Lookup_Setting__mdt	685	

- d. Select **Manage Custom Lookup Setting**.



Custom Metadata Type

Custom Lookup Setting (Managed)

Help for this Page ?

This custom metadata type is managed. You can only edit certain attributes. [Display More Information](#)

Standard Fields [6] | Custom Fields [13] | Validation Rules [1] | Page Layouts [1]

Custom Metadata Type Detail

Edit **Manage Custom Lookup Settings**

Singular Label	Custom Lookup Setting	Description	
Plural Label	Custom Lookup Settings	Visibility	Public
Object Name	Custom_Lookup_Setting	Protection Level	
Namespace Prefix	TnzMultilookup	Record Size	685
API Name	TnzMultilookup__Custom_Lookup_Setting__mdt		

e. Select **New**.

Custom Lookup Settings

Help for this Page ?

View: All [Create New View](#)

New

Label ↑	Custom Lookup Setting Name	Namespace Prefix
No records to display.		

f. Fill the required details as per the requirement for the lookup field and Save the record. Refer below image for sample Account Lookup field.

Custom Lookup Setting Edit

Save Save & New Cancel

Information

Label

Custom Lookup Setting Name ⓘ

Maximum No of Records Allowed to Select

Object Name

Field Text

Placeholder

Additional Field

Field Value

Maximum No of Records to be Displayed

Record Type

Search Field

Lookup Icon

Condition String

Lookup Field Label

Relationship Field Name (Parent)



Custom Lookup Setting Detail Edit Delete Clone

Label	Account Lookup	Protected Component	☐
Custom Lookup Setting Name	Account_Lookup	Namespace Prefix	
Maximum No of Records Allowed to Select	1		
Object Name	Account		
Field Text	Name		
Placeholder	Type here to search for an Account		
Additional Field	Website		
Field Value	Id		
Maximum No of Records to be Displayed	5		
Record Type			
Search Field	Name		
Lookup Icon	standard goals		
Condition String	Rating='Hot'		
Lookup Field Label	Account		

2. Use the Metadata DeveloperName to configure the Lookup component

Use the metadata record's developerName and add the lookup component to your parent lightning component as below:

Custom Lookup Setting Detail Edit Delete Clone

Label	Account Lookup
Custom Lookup Setting Name	Account_Lookup
Maximum No of Records Allowed to Select	1
Object Name	Account
Field Text	Name
Placeholder	Type here to search for an Account

```
<aura:attribute name="accList" type="Object[]" />
<TnzMultilookup:CustomLookupComponent metadataName = "Account_Lookup" selectedRecordList="{!v.accList}" />
```

Lookup Icon	standard goals
Condition String	Rating='Hot'
Lookup Field Label	Account
Relationship Field Name	(Parent)

Use the “*selectedRecordList*” attribute of the component to get the values of the selected records. This attribute holds an Object Array (Object[]). Developer can access below properties of the object to get the associated values for selected records:

Wrapper Property	Description
text	- The value contained in the field of the record specified in “Field Text”. - For. Ex. If a lookup is configured for Account and the Field Text in the Custom Metadata is set to Name, then wrapper property text will hold Account Name of the selected record.



val	- The value contained in the field of the record specified in “Field Value”. - For. Ex. If a lookup is configured for Account and the Field Value in the Custom Metadata is set to Id, then wrapper property val will hold Account Id of the selected record.
additionalInfo	- The value contained in the field of the record specified in “Additional Info”. - For. Ex. If a lookup is configured for Account and the Additional Field in the Custom Metadata is set to Website, then wrapper property additionalInfo will hold Account Website of the selected record.

Refer section [Examples of Component Usage in Code](#) for detailed usage in code.

5. Key Features

a. Allows Single Record Selection for the Lookup Field

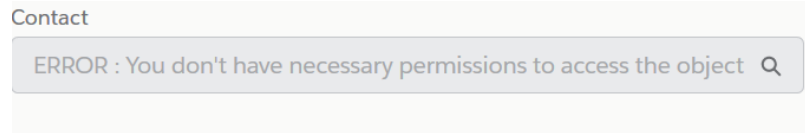
b. Allow Multiple Record Selection for the Lookup Field

c. If there are more records in the list, click the search icon and you can select the required records.

FULL NAME	EMAIL
☐ Rose Gonzalez	rose@edge.com
☐ Sean Forbes	sean@edge.com
☐ Jack Rogers	jrogers@burlington.com
☐ Pat Stumuller	pat@pyramid.net
☐ Tim Barr	barr_tim@grandhotels.com
☐ Lauren Boyle	lboyle@uog.com
☐ Babara Levy	b.levy@expressl&t.net



- d. On component initialization, the object and field level permissions are checked for the running user. If the user does not have any of the permissions then, the lookup field is disabled, and error message is displayed as below. A message will be displayed as below:



The message in the component can be configured by changing value of custom label **Lookup Alert message**.

6. Dependent Lookup Component

A dependent lookup is a relationship between a parent and child object with a lookup filter that references fields on the source object. For example, you can configure the Contact lookup field to only show contacts associated with the account selected in the parent lookup component.

How to create a dependent Lookup?

1. Create separate metadata record for child and parent component.
2. Specify the Relationship Field on the metadata setting of the child lookup component.
Consider creating dependency between Account Lookup and Contact Lookup, then specify the **Relationship field** on Contact Lookup Setting as **AccountId**.

Parent Lookup:

Custom Lookup Setting Detail		Edit	Delete	Clone
Label	Account Lookup	Protected Component ☐		
Custom Lookup Setting Name	Account_Lookup	Namespace Prefix		
Maximum No of Records Allowed to Select	1			
Object Name	Account			
Field Text	Name			
Placeholder	Type here to search for an Account			
Additional Field	Website			
Field Value	Id			
Maximum No of Records to be Displayed	5			
Record Type				
Search Field	Name			
Lookup Icon	standard:goals			
Condition String	Rating='Hot'			
Lookup Field Label	Account			



Child Lookup:

Custom Lookup Setting

Custom Lookup Setting Detail Edit Delete Clone

Label	ContactLookup	Protected Component	☐
Custom Lookup Setting Name	ContactLookup	Namespace Prefix	
Maximum No of Records Allowed to Select	10		
Object Name	Contact		
Field Text	Name		
Placeholder	Type here to Search for Accounts		
Additional Field	Email		
Field Value	Id		
Maximum No of Records to be Displayed	5		
Record Type			
Search Field	Name		
Lookup Icon	standard:contact		
Condition String			
Label	Search for related contacts		
Relationship Field Name (Parent)	AccountId		

3. In the component, use below code to declare the dependent lookup component.

```
<aura:attribute name="parentResults" type="Object[]"/>
<aura:attribute name="childRecords" type="Object[]"/>

<c:DependantLookup parentMetadataName="AccountLookup"
    childMetadataName="ContactLookup"
    parentRecordList="{!v.parentResults}"
    childRecordList="{!v.childRecords}"/>
```

Attribute details for DependentLookup:

Attribute Name	Description
parentMetadataName	Specify the custom metadata record's developer name configured for parent lookup.
childMetadataName	Specify the custom metadata record's developer name configured for child lookup.
parentRecordList	Holds the value of selected records for the parent lookup. It returns Object []. It is same as selectedRecordList mentioned in section 3.
childRecordList	Holds the value of selected records for the child lookup. It returns Object []. It is same as selectedRecordList mentioned in section 3.



7. Examples of Component Usage in Code

1. Single/Multiple Record Selection

Step 1: Create a Custom Metadata Record for Account Lookup under Custom Lookup Setting.

[Navigate to Setup → Custom Metadata Types → Click on Manage Record for Custom Lookup Setting → Click New]

Custom Lookup Setting Detail		Protected Component
Label	Account Lookup	☐
Custom Lookup Setting Name	Account_Lookup	Namespace Prefix
Maximum No of Records Allowed to Select	1	
Object Name	Account	
Field Text	Name	
Placeholder	Type here to search for an Account	
Additional Field	Website	
Field Value	Id	
Maximum No of Records to be Displayed	5	
Record Type		
Search Field	Name	
Lookup Icon	standard:goals	
Condition String	Rating='Hot'	
Lookup Field Label	Account	
Relationship Field Name (Parent)		

Step 2: Create new Lightning Application to use this lookup setting. Copy the code snippet below and paste it in .app file of the new component.

[Open Developer Console → Go to File → New → Lightning Application → Copy and paste below code snippet]

```
<aura:application extends="force:slds">
<!--Attribute to hold the selected records-->
<aura:attribute name="accList" type="Object[]"/>
<lightning:card footer="" title="Single Record Selection/Multiple Record Selection">
  <p class="slds-p-horizontal_small">
    <!--Here the metadataName attribute is the developerName of the metadata record created in Step 1-->
    <TnzMultilookup:CustomLookupComponent metadataName = "Account_Lookup" selectedRecordList="{!v.accList}"/>
    <aura:if isTrue="{!v.accList.length > 0}"> <br/>
      <!--This is how we can access the Account Name (The value for Field Text in metadata record is set to Name)-->
      Selected Account Name : <Strong>{!v.accList[0].text}</Strong> <br/>
      <!--This is how we can access the Account Website (The value for Additional Field in metadata record is set to Website)-->
      Account Website: <Strong>{!v.accList[0].additionalInfo}</Strong><br/><br/>
    </aura:if>
  </p>
</lightning:card>
</aura:application>
```



Step 3: Preview the Lightning Application. (Update the custom metadata to make changes in the Label, Placeholder, Search Criteria, etc.)

Single Record Selection/Multiple Record Selection

Account

Type here to search for an Account

Single Record Selection/Multiple Record Selection

Account

Edge Communications

Selected Account Name : **Edge Communications**
Account Website: **http://edgecomm.com**

2. Dependent Lookup

Step 1: Create a Custom Metadata Record for Account Lookup under Custom Lookup Setting. This will be the configuration for the parent lookup.

[Navigate to Setup → Custom Metadata Types → Click on Manage Record for Custom Lookup Setting → Click New]

Custom Lookup Setting Detail			Edit	Delete	Clone
Label	Account Lookup	Protected Component	☐		
Custom Lookup Setting Name	Account_Lookup	Namespace Prefix			
Maximum No of Records Allowed to Select	1				
Object Name	Account				
Field Text	Name				
Placeholder	Type here to search for an Account				
Additional Field	Website				
Field Value	Id				
Maximum No of Records to be Displayed	5				
Record Type					
Search Field	Name				
Lookup Icon	standard:goals				
Condition String	Rating="Hot"				
Lookup Field Label	Account				
Relationship Field Name (Parent)					

Step 2: Create a Custom Metadata Record for Child Lookup under Custom Lookup Setting. This will be configurations for the child lookup. [Do not forget to mention the Relationship Field Name (Parent) in the child lookup.]

[Navigate to Setup → Custom Metadata Types → Click on Manage Record for Custom Lookup Setting → Click New]



Custom Lookup Setting Detail Edit Delete Clone

Label	Contact Lookup	Protected Component	
Custom Lookup Setting Name	Contact_Lookup	Namespace Prefix	
Maximum No of Records Allowed to Select	2		
Object Name	Contact		
Field Text	Name		
Placeholder	Type here to search for related Contacts		
Additional Field	Email		
Field Value	Id		
Maximum No of Records to be Displayed	5		
Record Type			
Search Field	Name		
Lookup Icon	standard:contact		
Condition String			
Lookup Field Label	Related Contacts		
Relationship Field Name (Parent)	AccountId		

Step 3: Create a Lightning Application. Copy the code snippet below and paste it in .app file of the new application.

[Open Developer Console → Go to File → New → Lightning Application]

```
<aura:application extends="force:slds">
  <!-- Attribute to hold the value of selected parent record-->
  <aura:attribute name="accountList" type="Object[]"/>
  <!-- Attribute to hold the value of selected child records-->
  <aura:attribute name="contactList" type="Object[]"/>
  <div class="slds-m-around_large">
    <lightning:card footer="" title="Dependent Lookup">
      <p class="slds-p-horizontal_small">
        <!--Here the parentMetadataName and childMetadataName attribute is the developerName of the metadata records created in Step 1 and 2-->
        <TnzMultilookup:DependantLookup parentMetadataName="Account_Lookup"
          childMetadataName="Contact_Lookup"
          parentRecordList="{!v.accountList}"
          childRecordList="{!v.contactList}"/>
      </p>
    </lightning:card>
  </div>
</aura:application>
```

Step 4: Preview the Lightning Application (Update the custom metadata to make changes in the Label, Placeholder, Search Criteria, etc.)



Dependent Lookup

Account

Type here to search for an Account



Related Contacts

Type here to search for related Contacts



Dependent Lookup

Account

 Edge Communications



Related Contacts

Type here to search for related Contacts



 Rose Gonzalez