

Universal Lookup Field Component

Developer & Admin Guide

Case

↶

Search Cases

Chatter Expert ×

Pyramid Construction Inc. ×

Case

00001001 ×

Lead

Tom James ×

Contract

00000100 ×

Order

00000100 ×

Campaign

User Conference - Jun 17-19, 2002 ×

Custom test_1

Test ×

Custom test_2

Test 2 ×

Custom test_3

Test 3 ×

Product

↶

Ins

Q "Ins" in Products

Installation: Industrial - High

IN7080

Installation: Portable

IN7020

Installation: Industrial - Low

IN7040

Installation: Industrial - Medium

IN7060

Content

Introduction	4
Installation	5
Aura Lookup	6
Component Attributes	6
Required attributes	7
objectType	7
selectedRecordId	8
Attributes to customize search	10
queryCondition	10
If you are using LULC in the Flow (use INPUT params):	10
If you are using LULC in the Custom Lightning component:	11
primarySearchField	12
additionalField	13
fuzzySearch	14
orderBy	15
withSharing	16
queryLimit	17
withAdvancedSearch	17
fieldSetName	18
Attributes to customize component UI/UX	20
label / pluralLabel	20
objectIconPath	21
isRequired	22
showFiveRecent	23
readOnly	23
dropDownQuery	24
dropDownLimit	25
helpText	26
placeholder	26
styleClass	27
width	28
errorMessage	28
isDropdownOverflowAvaliable	29
Component Events	30

onBlur	30
onFocus	31
onChange	31
Component Global Methods	32
fireChanging	32
Component Exception Handler	33
Translation support	36
LULC Dependent component	38
Lookup Advanced Search	40



Introduction

Boost the development of your custom components with our powerful lookup component. It looks like a standard one, pretty easy to customize and works everywhere – even in Salesforce1.

Lightning allows us to create convenient, feature-rich components, however, the problem is that there is not all the functionality from classic Salesforce that is currently implemented. This leads to a more time-consuming and expensive component development process. In Synebo we are always trying to find the best and the most optimized way for our customers in order to fully satisfy their needs and it goes without saying that we are adhering to save their budget.

We have created a Lightning Lookup component that will save your time when you want to add a lookup field in your custom component. Now you do not need to lose time on creating and testing your own component for lookup. Just use it.

We use the Lightning Design System, so the component looks and works like a standard Lightning Lookup field. Our component is easy to customize and implement. You can use it for custom and standard objects. Also it works in Salesforce1.



Installation

Notice that for correct Custom Lightning component's work you should deploy it for all Users custom Domain.

 **Install Universal Lookup Field Component**
By Synebo


☐ Install for Admins Only


☒ Install for All Users


☐ Install for Specific Profiles...

Install

Cancel

App Name	Publisher	Version Name	Version Number
Universal Lookup Field Component	Synebo		

[Additional Details](#) [View Components](#)

 **Install Universal Lookup Field Component**
By Synebo

 Installation Complete!

Done

App Name	Publisher	Version Name	Version Number
Universal Lookup Field Component	Synebo		

Aura Lookup

Component Attributes

Attribute	Type	Default
additionalField	String	-
dropDownQuery	String	-
dropDownLimit	Integer	5
withAdvancedSearch	Boolean	false
fieldSetName	String	-
fuzzySearch	Boolean	false
isRequired	Boolean	false
label	String	standard
limit	Integer	200
objectType*	String	-
orderBy	String	-
pluralLabel	String	standard
objectIconPath	String	-
primarySearchField	String	standard
queryCondition	String	-
readOnly	Boolean	false
selectedRecordId*	Id	-
showFiveRecent	Boolean	true
styleClass	String	-

width	String	100%
errorMessage	String	Complete this field
withSharing	Boolean	false

Required attributes

Attributes that are required for correct component work

objectType

Type

String

Description

Required attribute that defines a type of Object that will be used in Lookup.
It can be Standard or Custom Salesforce Object.
In this attribute a full API Name of Salesforce Object has to be defined.

E.g.:

```
<aura:attribute name="contactId" type="String" />
```

```
<|_lookup:Lookup objectType="Contact"
    selectedRecordId="{!v.contactId}" />
```

```
<aura:attribute name="customerId" type="String" />
```

```
<|_lookup:Lookup objectType="Customer__c"
    selectedRecordId="{!v.customerId}" />
```

selectedRecordId

Type

String

Description

Required string attribute that returns the Id of the record that was selected.

E.g.:

Component:

```
<aura:attribute name="accountId"
    type="String" />

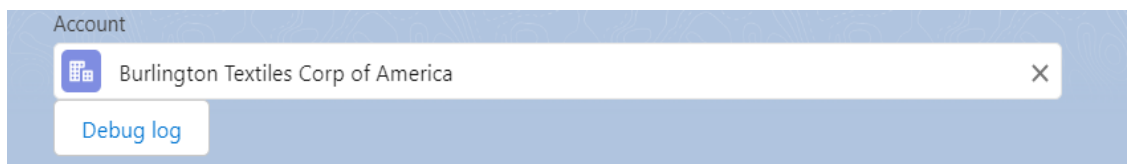
<!_lookup:Lookup objectType="Account"
    selectedRecordId="{!v.accountId}" />

<lightning:button label="Debug log"
    onclick="{!c.debugSelectedId}" />
```

Controller:

```
{
  debugSelectedId: function(cmp, event, helper) {
    console.log("recordId = ", cmp.get("v.accountId"));
  }
}
```

UI:



Console:

```
recordId = 0012x000003PTagAAG      AuraWrapperAndLWCTestCmp.js:52
```

Component:

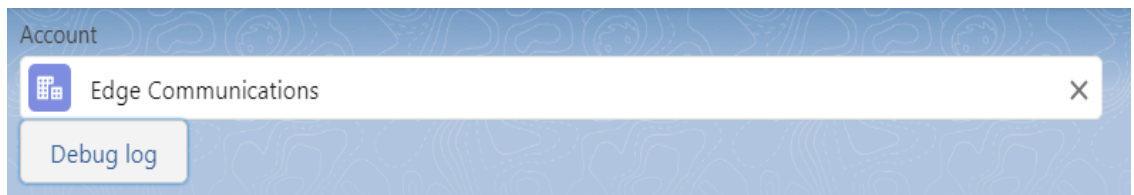
```
<|_lookup:Lookup aura:id="accountLookup"
    objectType="Account" />

<lightning:button label="Debug log"
    onclick="{!c.debugSelectedId}" />
```

Controller:

```
{
  debugSelectedId: function(cmp, event, helper) {
    console.log("recordId = ", cmp.find("accountLookup").get("v.selectedRecordId"));
  }
}
```

UI:



Console:

```
recordId = 0012x000003PTafAAG AuraWrapperAndLWCTestCmp.js:52
```

NOTE:

In console displayed record Id according ObjectType="Account" starts with Object Key Prefixes "001"

How to set Default record id

1. Set default value in the attribute.
(Not recommended because ID hardcode should not be used)
2. Set default value in doInit of main component

Attributes to customize search

queryCondition

Type

String

Description

Optional string attribute that allows adding additional condition to query. This attribute gives an opportunity to add a filter to lookup. Be careful with this attribute - if the query is invalid, the component will not work. You can use hardcoded values or values from attributes for conditions.

If the *queryCondition* is changed, the selected value will be removed. Use the attribute *removeValueOnQueryConditionChange* to change this behaviour.

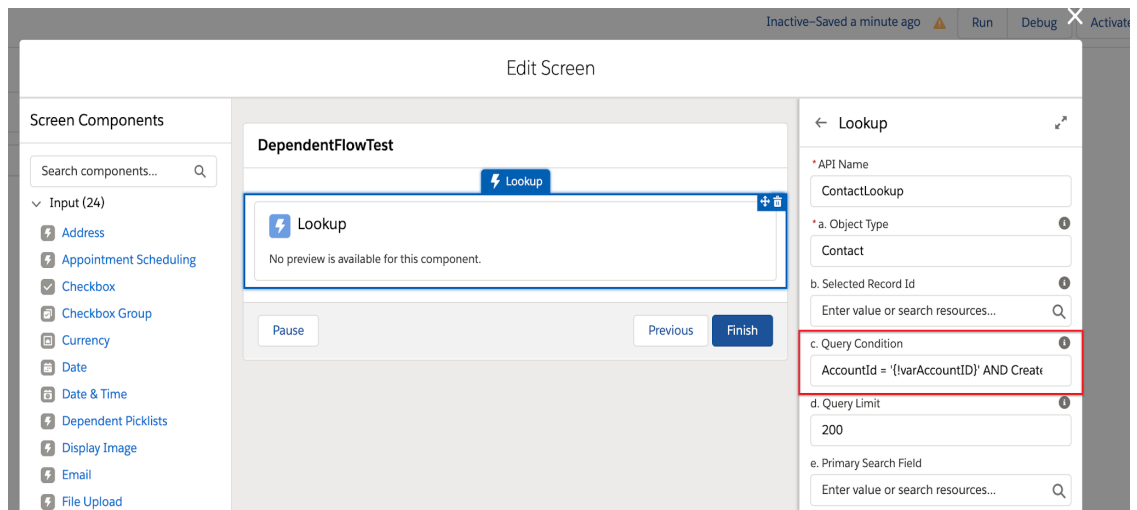
Salesforce documentation: [WHERE conditionExpression](#).

E.g.:

If you are using LULC in the Flow (use INPUT params):

```
AccountId = '{!varAccountID}' AND CreatedDate=Today
```

AccountId - API Name of the field that you want to use in condition
{!varAccountID} - variable from a Flow



If you are using LULC in the Custom Lightning component:

```
// Users with Profile "System Administrator" OR "Account Manager"
<|_lookup:Lookup aura:id="uniqueLookupId"
    objectType="User"
    selectedRecordId="{!v.selectedUserId}"
    queryCondition="Profile.Name = 'System Administrator' OR
    Profile.Name = 'Account Manager'" />
```

```
// Contacts with Last Name starts from "S"
<|_lookup:Lookup aura:id="uniqueLookupId"
    objectType="Contact"
    selectedRecordId="{!v.selectedUserId}"
    queryCondition="LastName LIKE 'S%'" />
```

```
// Accounts that are created this week
<|_lookup:Lookup aura:id="uniqueLookupId"
    objectType="Account"
    selectedRecordId="{!v.selectedUserId}"
    queryCondition="CreatedDate = THIS_WEEK" />
```

// Example with value from attribute in condition:

```
<aura:attribute name="selectedRecordId" type="String"/>
<aura:attribute name="accountId" type="Id" default="0010a00001LtTmiAAF" />

<!_lookup:Lookup aura:id="uniqueLookupId"
    objectType="Contact"
    queryCondition="{! 'AccountId=\' + v.accountId + '\''}"
    selectedRecordId="{!v.selectedRecordId}" />
```

primarySearchField

Type

String

Description

An API Name of the object field that will be primarily used in searching records (instead of standard fields).

E.g.:

```
<aura:attribute name="accountId" type="String" default="" />

<!_lookup:Lookup aura:id="myLookup"
    objectType="Account"
    selectedRecordId="{!v.accountId}"
    primarySearchField="Site"
    additionalField="Name" />
```

Demo:

Account

Search Accounts...

Q

additionalField

Type

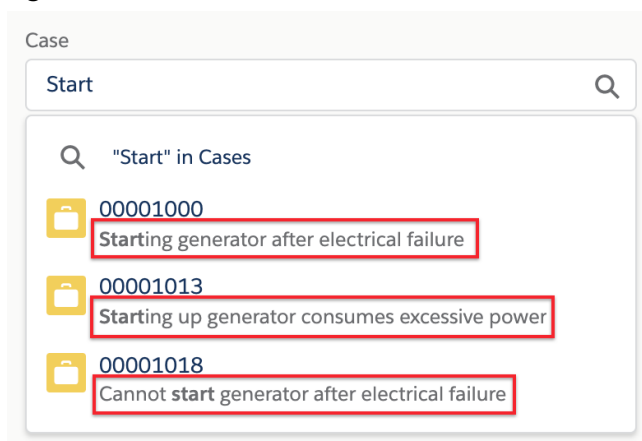
String

Description

Field that is displayed below name or number and used as an additional field for search.
An attribute has to contain the correct API Name of a field that is related to the object that is defined in “objectType” attribute.

Works for standard and custom objects.

E.g.:



If user does not define special value, it will contain standard default value in next cases:

Object	addutionalField	Display
Case	Subject	Subject
Contact	AccountId	Account.Name
Contract	AccountId	Account.Name
Order	AccountId	Account.Name
Lead	Title	Title
Product2	ProductCode	ProductCode

Field types that are allowed to use in the attribute:

string, email, fax, phone, picklist, reference (will be displayed the Name of record), *url*

fuzzySearch

Type

Boolean

Description

The optional attribute, that enables/disables a search within primary and additional object fields' values. Allow searching in the middle of a word or number starting with two symbols.

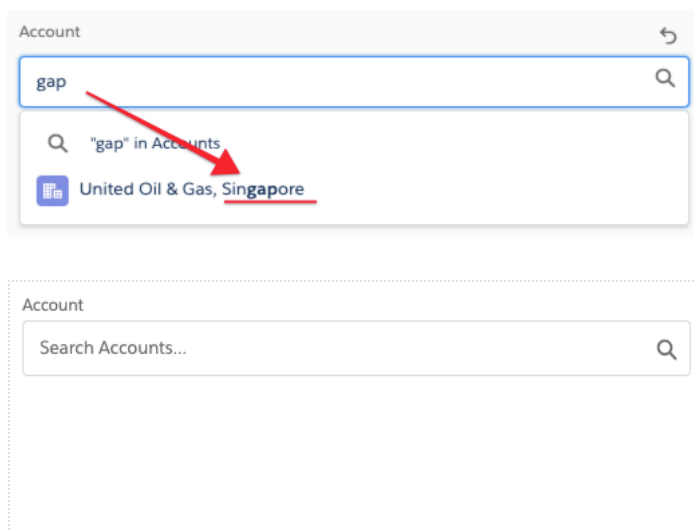
The default value is false.

E.g.

```
<aura:attribute name="selectedRecordId" type="String" default="" />

<!_lookup:Lookup aura:id="myLookup"
    objectType="Contact"
    selectedRecordId="{!v.selectedRecordId}"
    fuzzySearch="true" />
```

Demo:



The image shows two examples of a Salesforce 'Account' lookup component. The top example shows the component with the text 'gap' entered in the search field. A red arrow points to the search results, which include 'United Oil & Gas, Singapore', where 'Singapore' is underlined, indicating a fuzzy match. The bottom example shows the same component with the placeholder text 'Search Accounts...' in the search field.

orderBy

Type

String

Description

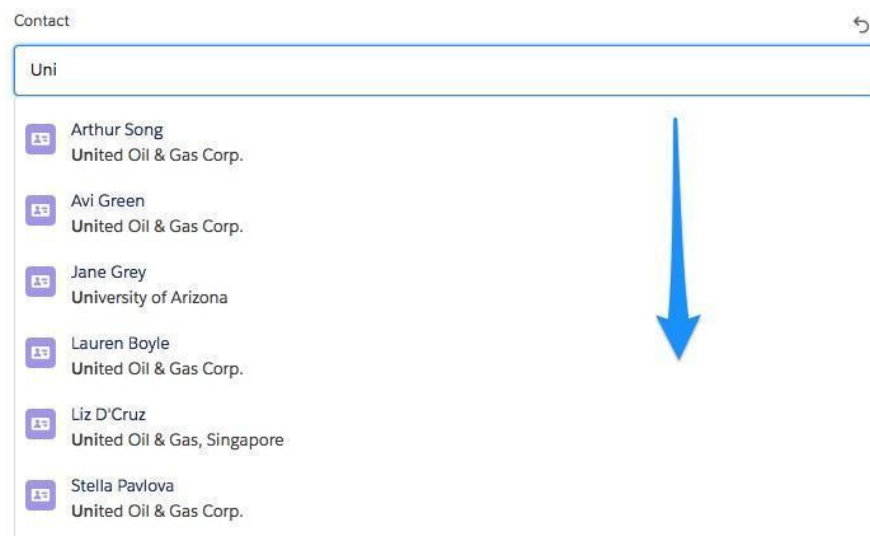
Optional attribute that allows sorting records.

Salesforce documentation: [ORDER BY in SOQL](#)

E.g.:

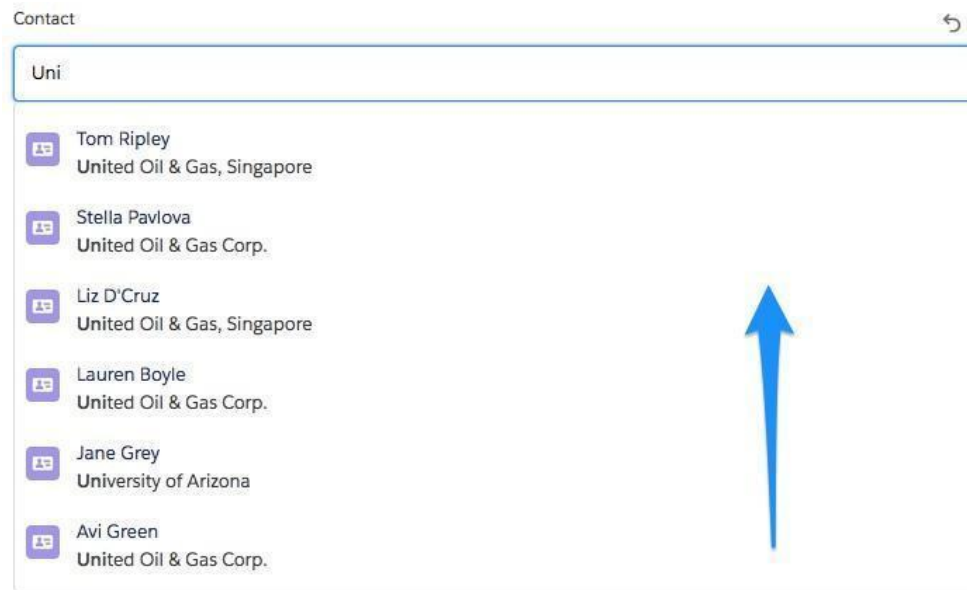
```
<aura:attribute name="selectedId" type="Id" />

// Order by Contact Name A-Z
<!_lookup:Lookup aura:id="myLookup"
  objectType="Contact"
  selectedRecordId="{!v.selectedId}"
  orderBy="Name" />
```



```
// Order by Contact Name Z-A
```

```
<|lookup:Lookup aura:id="myLookup"  
  objectType="Contact"  
  selectedRecordId="{!v.selectedId}"  
  orderBy="Name DESC" />
```



Also you can define several fields by comma separator:

```
// Order by Name and Created date  
<|lookup:Lookup aura:id="myLookup"  
  objectType="Contact"  
  selectedRecordId="{!v.selectedId}"  
  orderBy="Name DESC, CreatedDate" />
```

withSharing

Type

Boolean

Description

Optional attribute that controls retrieving records according to user's access level. The default value is **true**.

E.g.:

```
<aura:attribute name="selectedRecordId" type="String" default="" />

<!_lookup:Lookup aura:id="myLookup"
    objectType="Contact"
    selectedRecordId="{!v.selectedRecordId}"
    withSharing="true" />
```

queryLimit

Type

Integer

Description

Optional attribute that controls retrieving records according to user's access level.
The default value is **200**.

withAdvancedSearch

Type

Boolean

Description

Optional attribute that allows to open advanced search modal by clicking on the row with search icon.
The default value is **false**.

E.g.:

```
<aura:attribute name="withAdvancedSearch" type="Boolean" access="global"
    default="false" />

<!_lookup:Lookup aura:id="myLookup"
    objectType="Contact"
    fieldSetName="{!v.fieldSetName}"
    withAdvancedSearch="{!v.withAdvancedSearch}" />
```

Account

co

Q "co" in Accounts

- Burlington Textiles Corp of America
- Edge Communications
- Pyramid Construction Inc.

Accounts

Account

co

Accounts

6 Results

Account Name	Account Site	Account Fax	Billing City	Account Number
Sample Account for Entitlements				
Edge Communications	Edge Site	(512) 757-9000	Austin	CD451796
Burlington Textiles Corp of America		(336) 222-8000	Burlington	CD656092
Pyramid Construction Inc.		(014) 427-4428	Paris	CC213425
United Oil & Gas Corp.		(212) 842-5501	New York	CD355118

Cancel

fieldSetName

Type

String

Description

Attribute that defines a Field Set API Name which is used for Advanced Search. The fieldSetName is required when **withAdvancedSearch** is **true**. The fields from Field Set will be used for search and displayed in the Advanced Search table. If primarySearchField is not in the field set, it will be added and displayed in the advanced search table. If an additionalField is populated and not in the field set, it will be added and displayed in the advanced search table.

The fields maintained in Fields Set for Advanced Search:

- Like / Equals:
Auto Number, Lookup Relationship, Email, Phone, Picklist, Text, URL.
- Only Equals:
Number.
- Only View:
Currency, Roll-Up Summary, Checkbox, Date, Date/Time, Geolocation, Percent, Picklist (Multi-Select), Text (Encrypted), Time.

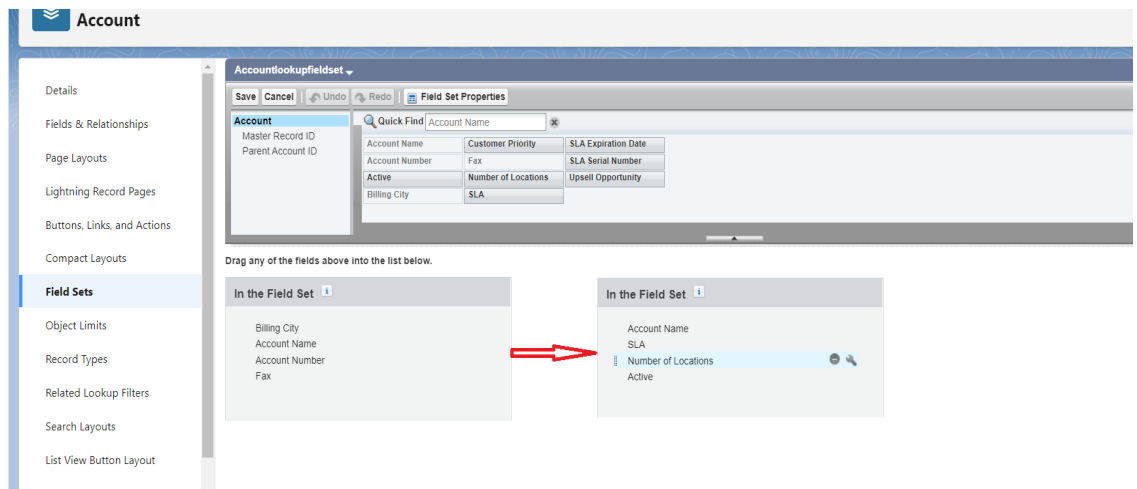
E.g.:

Component

```
<aura:attribute name="fieldSetName" type="String" access="global"
                default="!_lookup__accountlookupfieldset"
/>

<!_lookup:Lookup aura:id="myLookup"
                objectType="Account"
                fieldSetName="{!v.fieldSetName}"
                withAdvancedSearch="{!v.withAdvancedSearch}" />
```

Customer has possibility to change the Field Set suite:



Attributes to customize component UI/UX

label / pluralLabel

Type

String

Description

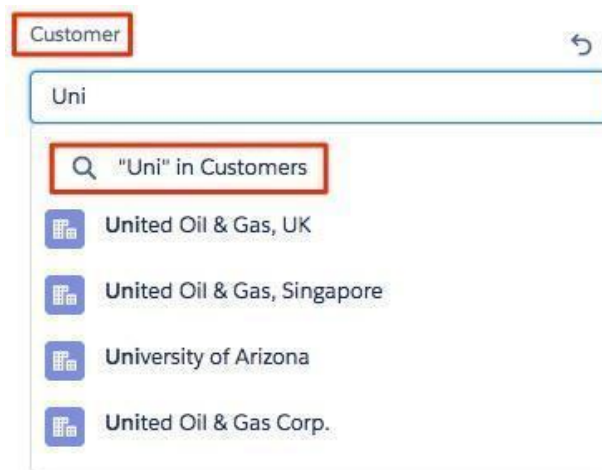
String attribute for Custom Label.

It could be useful if you use component for custom field and want to display not standard label.

E.g.:

```
<l_lookup:Lookup aura:id="myLookup"
  objectType="Account"
  label="Customer"
  pluralLabel="Customers"
  selectedRecordId="{!v.customerId}" />
```

Demo:



objectIconPath

Type

String

Description

Optional attribute that specifies the standard SLDS icon. It should be displayed for the records during the search.

Supported icons - <https://www.lightningdesignsystem.com/icons>

E.g.:

```
<l_lookup:Lookup aura:id="myLookup"
  objectType="Contact"
  selectedRecordId="{!v.contactId}"
  objectIconPath="standard:bot"/>
```

Demo:

Lookup

No preview is available for this component.

Pause Previous Finish

Lookup

i. Label Student

j. Plural Label Students

k. Icon Name standard:bot

* Student

Tom Barak

isRequired

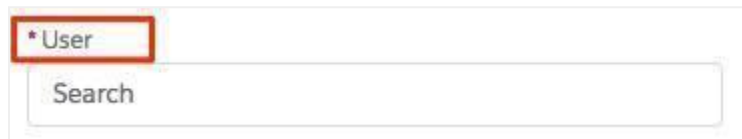
Type

Boolean

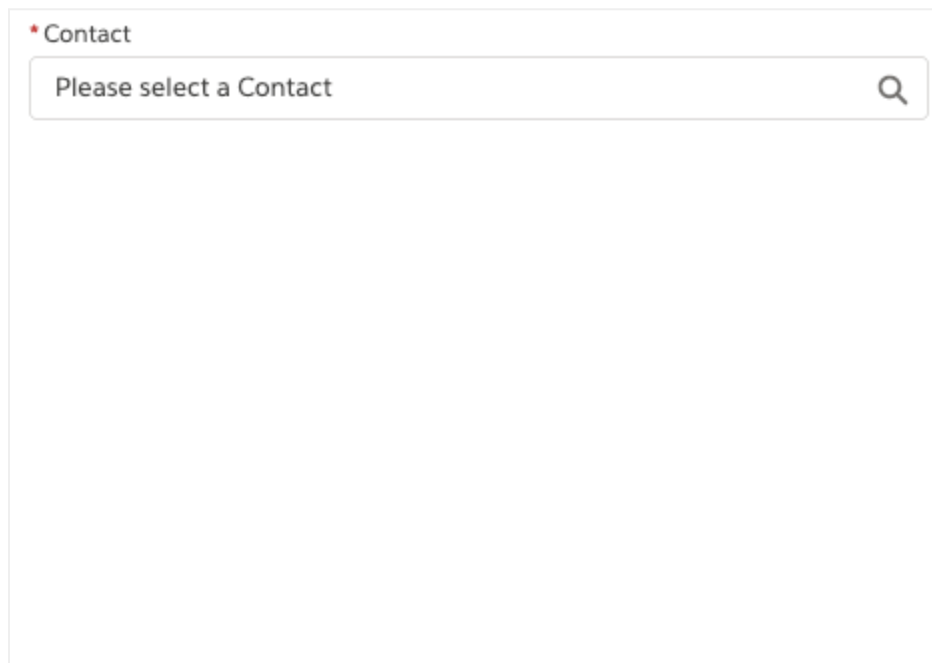
Description

The optional boolean attribute, which is false by default.

If it is set to true, a red asterisk will be displayed next to the Universal Lookup label.



A screenshot of a user selection dropdown. The label '* User' is highlighted with a red rectangular box. Below the label is a search input field with the placeholder text 'Search'.



A screenshot of a contact selection dropdown. The label '* Contact' is at the top. Below it is a search input field with the placeholder text 'Please select a Contact' and a magnifying glass icon on the right. The dropdown menu is currently empty.

showFiveRecent

Type

Boolean

Description

Optional boolean attribute that switches on or switches off Lightning salesforce feature. Standard lookups in Lightning by default show recent viewed records on focus. Maximum that will be displayed is 5 records. It is true by default.



readOnly

Type

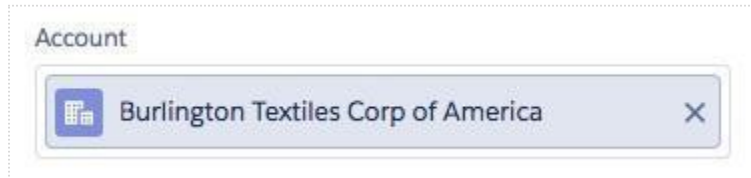
Boolean

Description

Boolean attribute to define if component is readOnly. If you define the lookup component as readOnly, you will not allow the user to change it.

E.g.:

```
<l_lookup:Lookup aura:id="uniqueLookupId"
  objectType="Account"
  readOnly="true"
  selectedRecordId="{!v.defaultValue}" />
```



dropDownQuery

Type

String

Description

Optional attribute that can contain query condition that will populate a drop-down with a list of suggestions instead of standard last viewed records.

It will allow for example to show a top list of records that are available for selection and the user will not have to type anything.

If the attribute is empty, a standard process will work that will show 5 recently viewed records if *showFiveRecent* is not false.

If you also use the *queryCondition* attribute, it will be added to *dropDownQuery* criteria.

E.g.:

```
<aura:attribute name="selectedRecordId"
  type="String"
  default="" />

<l_lookup:Lookup aura:id="myLookup"
  objectType="Contact"
  selectedRecordId="{!v.selectedRecordId}"
  dropDownQuery="IsActive__c = true" />
```

dropDownLimit

Type

Integer

Description

The optional attribute, which can be used in combination with the dropDownQuery attribute. It defines a number of records that will be displayed in drop-down by dropDownQuery attribute.

The value is 5 by default.

E.g.:

```
<aura:attribute name="selectedRecordId" type="String" default="" />

<!_lookup:Lookup aura:id="myLookup"
    objectType="Contact"
    selectedRecordId="{!v.selectedRecordId}"
    dropDownQuery="IsActive__c = true"
    dropDownLimit="3" />
```

Demo:

Contact

Search Contacts...

helpText

Type

String

Description

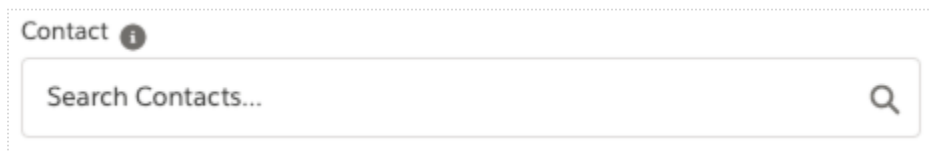
Optional string attribute that adds a help icon with defined text.

E.g.:

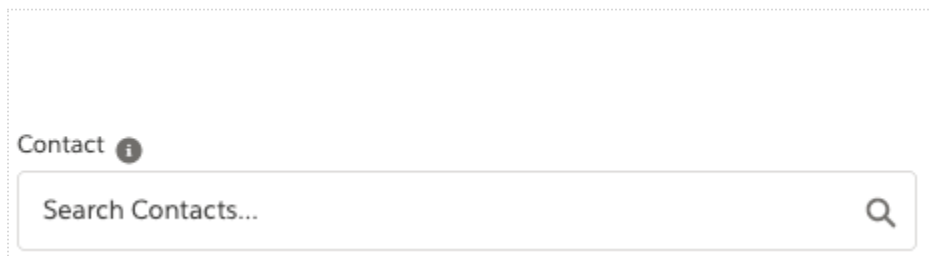
```
<aura:attribute name="contactId" type="String" default="" />

<!_lookup:Lookup aura:id="myLookup"
  objectType="Contact"
  selectedRecordId="{!v.contactId}"
  helpText="This contact will be used as primary" />
```

Demo:



A screenshot of a user interface element. At the top, the word "Contact" is followed by a small circular help icon (an 'i' inside a circle). Below this is a search bar with the placeholder text "Search Contacts..." and a magnifying glass icon on the right side.



A second screenshot of a user interface element, identical to the one above. It shows the word "Contact" with a help icon, followed by a search bar containing the text "Search Contacts..." and a search icon.

placeholder

Type

String

Description

Optional string attribute that allows to replace standard input placeholder.

E.g.:

```
<aura:attribute name="contactId" type="String" default="" />

<!_lookup:Lookup aura:id="myLookup"
    objectType="Contact"
    selectedRecordId="{!v.contactId}"
    placeholder="Please select a Contact" />
```

Demo:

Contact



styleClass

Type

String

Description

Optional string attribute that adds your own style classes to Lookup.

E.g.:

Style:

```
.THIS .red-border {
    border: 2px solid #c23934;
}
```

Component:

```
<aura:attribute name="myId"
    type="String" />

<!_lookup:Lookup aura:id="lookupCmpld"
    objectType="User"
    isRequired="true"
    styleClass="red-border"
    selectedRecordId="{!v.selectedId}"/>
```



width

Type

String

Description

Optional string attribute that controls the width of the component. By default it's equal to **"100%"**.

You can define it in percentage or pixels.

errorMessage

Type

String

Description

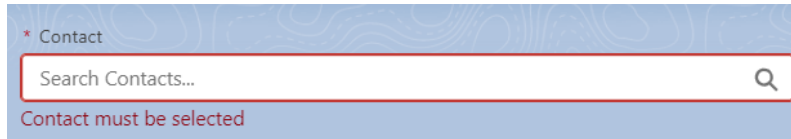
The optional string attribute, used for validation errors. Default value: 'Complete this field'.

E.g.:

```
<aura:attribute name="contactId" type="String" default="" />

<!lookup:Lookup aura:id="myLookup"
  objectType="Contact"
  selectedRecordId="{!v.contactId}"
  isRequired="true"
  errorMessage="Contact must be selected" />
```

Demo:



* Contact

Search Contacts...

Contact must be selected

isDropdownOverflowAvaliable

Type

Boolean

Description

The optional attribute that adds styles to the dropdown so that it is not cut off by the modal footer.

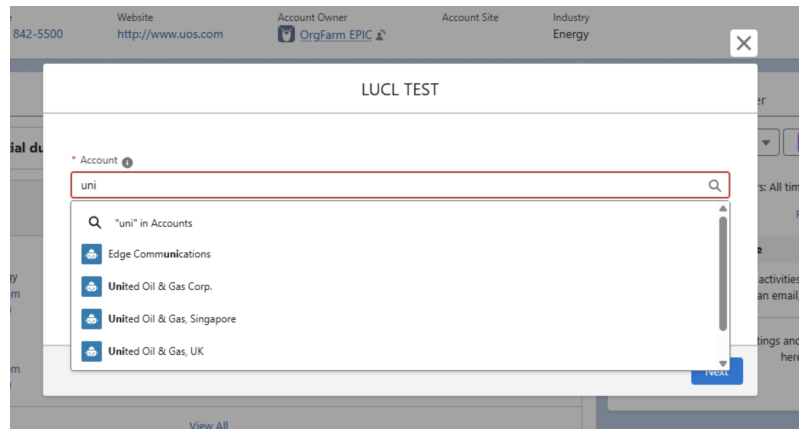
The default value is false.

E.g.

```
<aura:attribute name="selectedRecordId" type="String" default="" />

<!_lookup:Lookup aura:id="myLookup"
    objectType="Contact"
    selectedRecordId="{!v.selectedRecordId}"
    isDropdownOverflowAvailable="true" />
```

Demo:



Component Events

onBlur

Description

The event is fired when the user moves off from the component.

Code to handle the event:

```
<aura:handler name="onblur"
  event="!_lookup:OnBlur"
  action="{!c.handleBlur}"/>
```

In action you have to define your code that will be executed at onblur event.

If you have several lookups, you have to define **aura:id** in lookup components and you will get this as an event parameter called **cmpId**.

E.g.:

```
handleBlur : function(cmp, event, helper) {
  console.log('Lookup component Id: ' + event.getParam("cmpId"));
}
```

onFocus

Description

The event is fired when the user focuses on the component.

Code to handle the event:

```
<aura:handler name="onfocus"
  event="_lookup:OnFocus"
  action="{!c.handleFocus}"/>
```

In action you have to define your code that will be executed at onfocus event.

If you have several lookups, you have to define **aura:id** in lookup components and you will get this as an event parameter called **cmpId**.

E.g.:

```
handleFocus : function(cmp, event, helper) {
  console.log('Lookup component Id: ' + event.getParam("cmpId"));
}
```

onChange

Description

The event is fired when the user changes the content of the input.

Code to handle the event:

```
<aura:handler name="onChange"
  event="_lookup:OnChange"
  action="{!c.handleChange}"/>
```

In action you have to define your code that will be executed at onchange event.

If you have several lookups, you have to define **aura:id** in each of the lookup components



and you will get this as an event parameter called **cmpId**.

Also you could obtain previous value from **oldValue** event parameter.

E.g.:

```
handleChange : function(cmp, event, helper) {  
    console.log('Lookup component Id: ' + event.getParam("cmpId"));  
    console.log('Old Id: ' + event.getParam("oldValue"));  
}
```

Component Global Methods

fireChanging

- * To clear the selection you have to pass *null* as a parameter.

E.g.:

Component:

```
<aura:attribute name="selectedId"  
    type="String"/>  
  
<aura:attribute name="newValue"  
    type="String" />  
  
<l_lookup:Lookup aura:id="lookupId"  
    objectType="User"  
    selectedRecordId="{!v.selectedId}" />  
  
<lightning:button label="Change" onclick="{!c.changeIt}" />
```

Action in controller to change record Id:



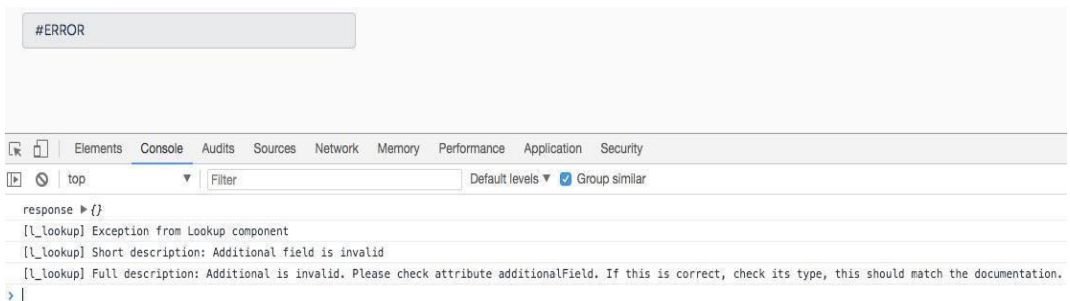
```
changelist : function(cmp, event, helper) {  
    var newVal = cmp.get("v.newValue");  
    cmp.find("lookupId").fireChanging(newVal);  
}
```

Notice that:

- value will not be set if **readOnly** is true
- value will not be set if new value is invalid
- value will not be set if you use **queryCondition** attribute and new value does not match criteria

Invalid value in **additionalField** attribute:

```
<l_lookup-lookup-lwc class="uniqueLookupClass"  
    object-type="Contact"  
    additional-field="Email">  
</l_lookup-lookup-lwc>
```



Component Exception Handler

For an easier development process we have added an Exception handler.

If you specified an incorrect value in the attribute or missed the required one, you could go to Browser developer tools and read the error description.

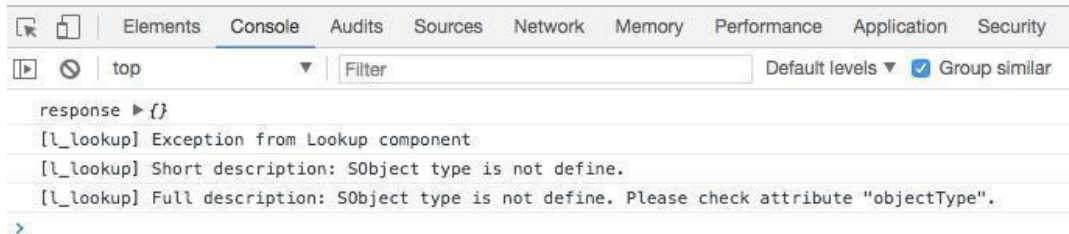
E.g.:

Missed required objectType attribute:

```
<aura:attribute name="selectedId"
    type="String"/>

<!_lookup:Lookup aura:id="lookupId"
    selectedRecordId="{!v.selectedId}" />
```

#ERROR



Invalid value in additionalField attribute:

```
<aura:attribute name="selectedId"
    type="String"/>

<!_lookup:Lookup aura:id="lookupId"
    objectType="Contact"
    additionalField="Email"
    selectedRecordId="{!v.selectedId}" />
```

#ERROR



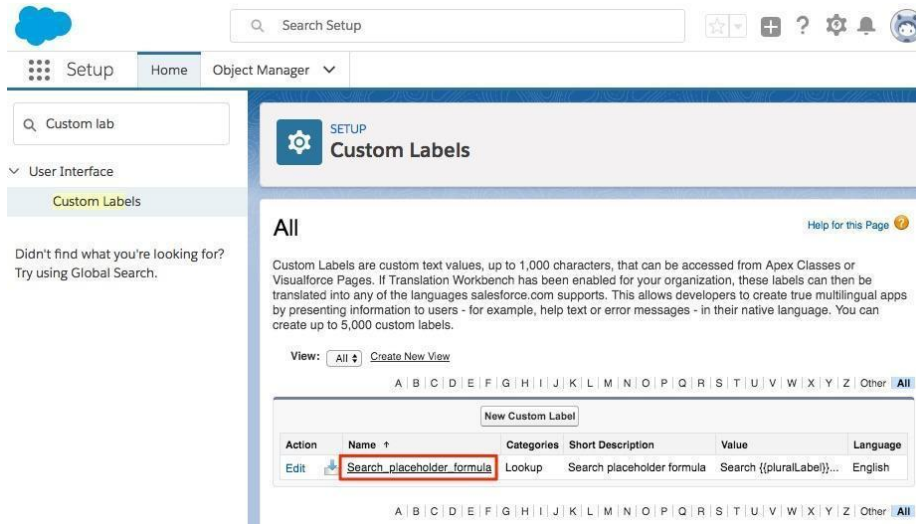
Translation support

Adding translation for Lookup placeholder:

1. Go to Setup -> Custom Labels

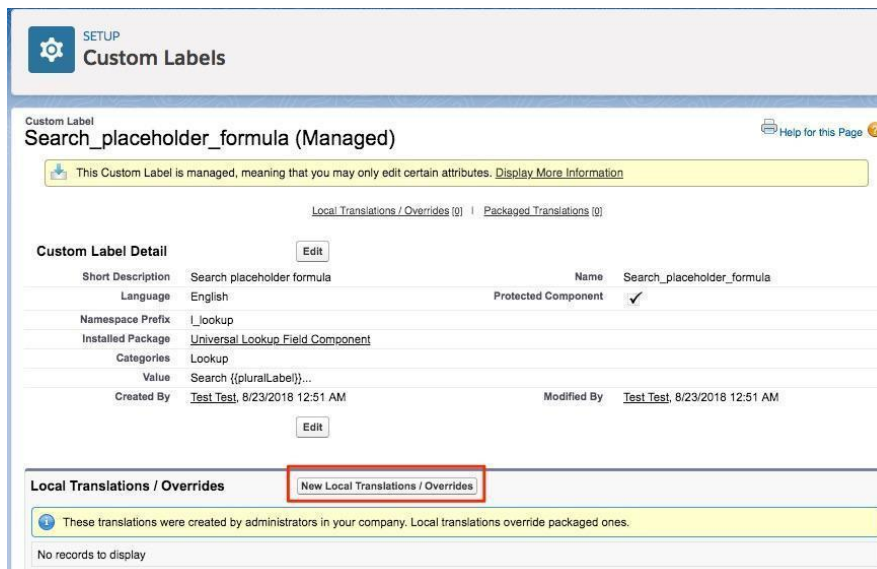
Find Label Search_placeholder_formula.

Click on Label.



2. Review the next screen.

Click "New Local Translations / Overrides" to add translation.



3. Choose language and add translation.

{{pluralLabel}} will be automatically replaced with the plural label of Object.

New Translation [Help for this Page](#)

Translation Edit

Master Label Information | = Required Information

Master Label	Search_placeholder_formula	Master Label Language	English
Master Label Text	Search {{pluralLabel}}...		

Translation Information

Language	German v i
Translation Text	{{pluralLabel}} durchsuchen...

E.g.:

Lookup on Contact in English.

Contact

Lookup on Contact in Deutsch.

Kontakt

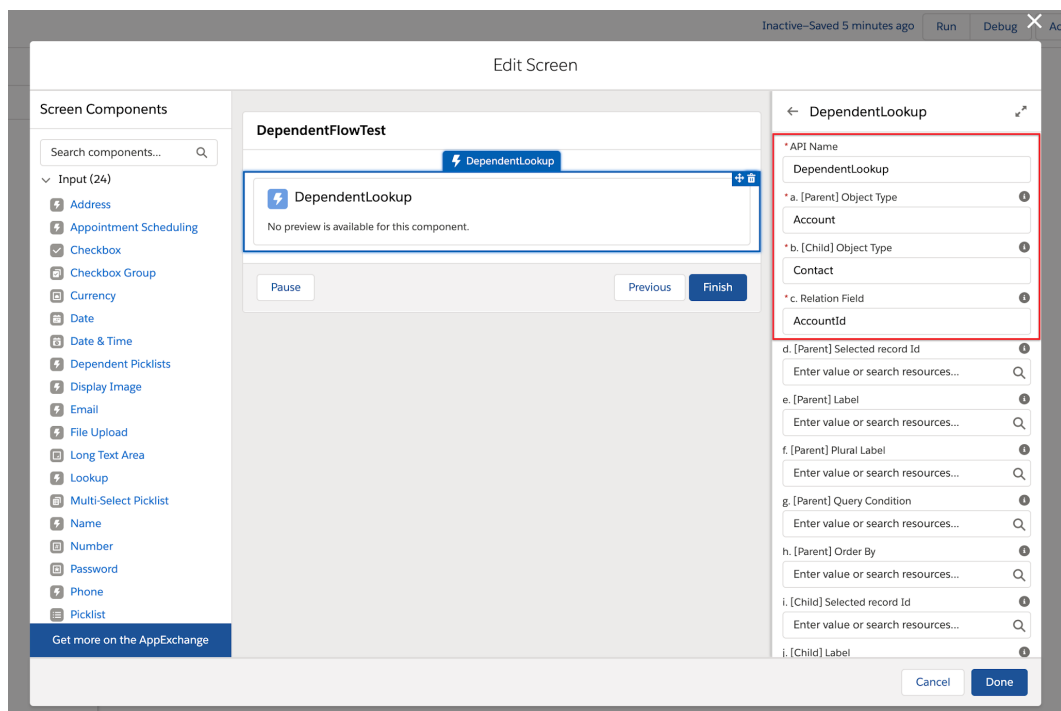
For this feature you have to add a Label: **{{pluralLabel}} durchsuchen...**

LULC Dependent component

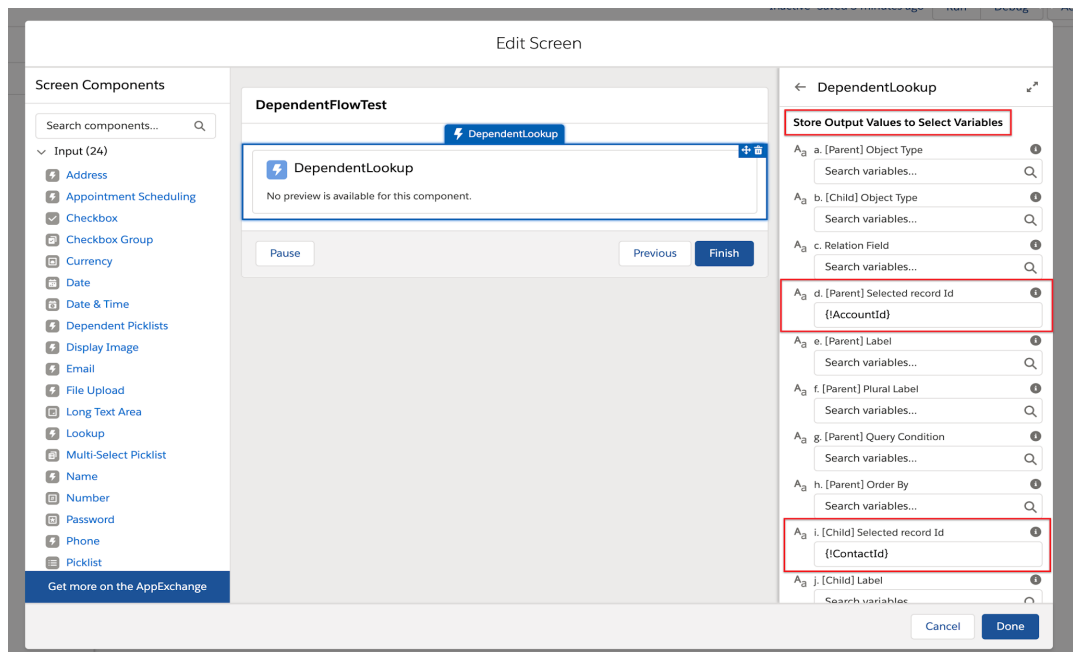
To help Salesforce admins create dependent lookups we have created a new component: DependentLookup.

Steps to use it in the Flow:

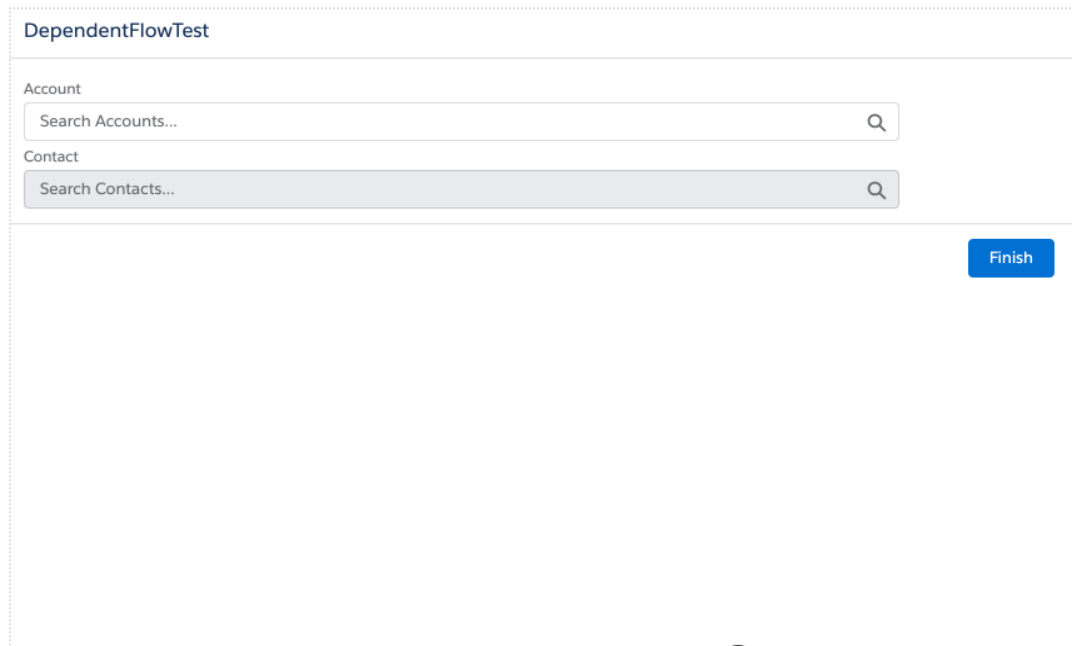
1. Add component to your screen
2. Specify required input params:
 - a. API Name of the parent Object
 - b. API Name of the child Object
 - c. API Name of the field that connect objects



3. Specify Output params:
 - a. Parent selected record Id
 - b. Child selected record Id



Demo result:



Lookup Advanced Search

Introduction

Lookup Advanced Search allows searching by the list of fields with the same additional condition like were provided in the lookup.

It appears by clicking on the row with the search icon in the search results drop down on the lookup component.

The Advanced Search result will be displayed on the modal window in a table. The table has a lazy loading functionality to load by 50 records.

The Lookup Advanced Search displays all fields specified in the Fields Set. Specified fields columns are sortable.

If the user clicks on the record - the result will be displayed in the input field of the page where the lookup component is located.

The Advanced Search functionality has the possibility to do searching using the input field above the table. It allows searching starting with two symbols.

For Advanced Search were added two new parameters: `fieldSetName` and `withAdvancedSearch`. When `withAdvancedSearch` is 'true' the `fieldSetName` is required.

The Advanced Search result is displayed on the modal window in a table:

Accounts

Account

ti

Q

Accounts
4 Results

Account Name	Billing City	Account Number	Account Fax
Edge Communications	Austin	CD451796	(512) 757-9000
Burlington Textiles Corp of America	Burlington	CD656092	(336) 222-8000
Pyramid Construction Inc.	Paris	CC213425	(014) 427-4428
Express Logistics and Transport	Portland	CC947211	(503) 421-7801

Cancel

Clicking on the record - the result will be displayed in the input field of the page where the lookup component is located :

Accounts

Account



Accounts
4 Results

Account Name	Billing City	Account Number	Account Fax
Edge Communications	Austin	CD451796	(512) 757-9000
Burlington Textiles Corp of America	Burlington	CD656092	(336) 222-8000
Pyramid Construction Inc.	Paris	CC213425	(014) 427-4428
Express Logistics and Transport	Portland	CC947211	(503) 421-7801

Cancel

Result displayed in the input field:

 Opportunity
Burlington Textiles Weaving Plant Generator

Account Name

Close Date

Amount

Opportunity Owner

[Burlington Textiles Corp of America](#)

3/11/2018

USD 235,000.00

 [Maria Kropinova](#)

✓

✓

✓

✓

✓

✓

✓

✓

Account

 Edge Communications 