Lightning Search Enabled Drop-Down Component

Developer Guide
Version 1.0

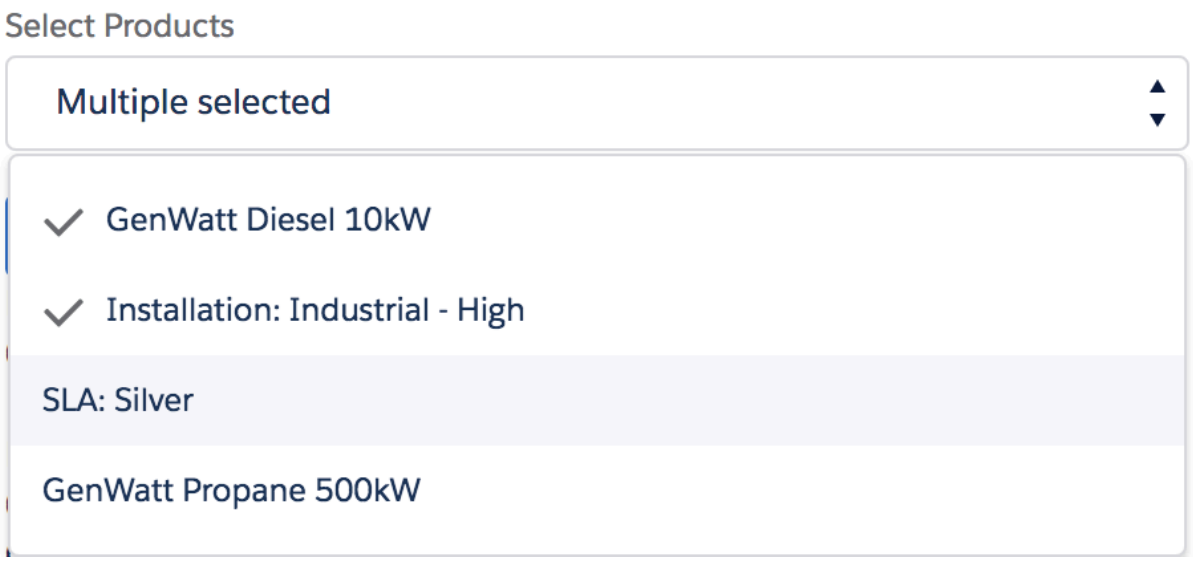
1. Introduction: Boost the development of your custom components with our custom lighting picklist Component. It uses a lightning design system so it looks like a standard field. The component provides an option to show a single picklist, multi-select, and an option to show a search box for searching.

2. Single Picklist Screen Image

Related Account



3. Multi-Select Picklist Screen Image




```
<div class="slds-form-element">
<label class="slds-form-element__label" for="accountInput">Related Account (Single Select Demo)</label>
<ASTPC:PicklistWidget options="{!v.accounts}" aura:id="accountInput"
                selectedOptions="{!v.selectedAccounts}" showSearchBox="true"/>
</div>

<div class="slds-form-element">
    <label class="slds-form-element__label" for="productsInput">Select Products(Multi-Select Demo)</label>
    <ASTPC:PicklistWidget options="{!v.products}" aura:id="productsInput" multiselect="true"
                        selectedOptions="{!v.selectedProducts}"/>
```

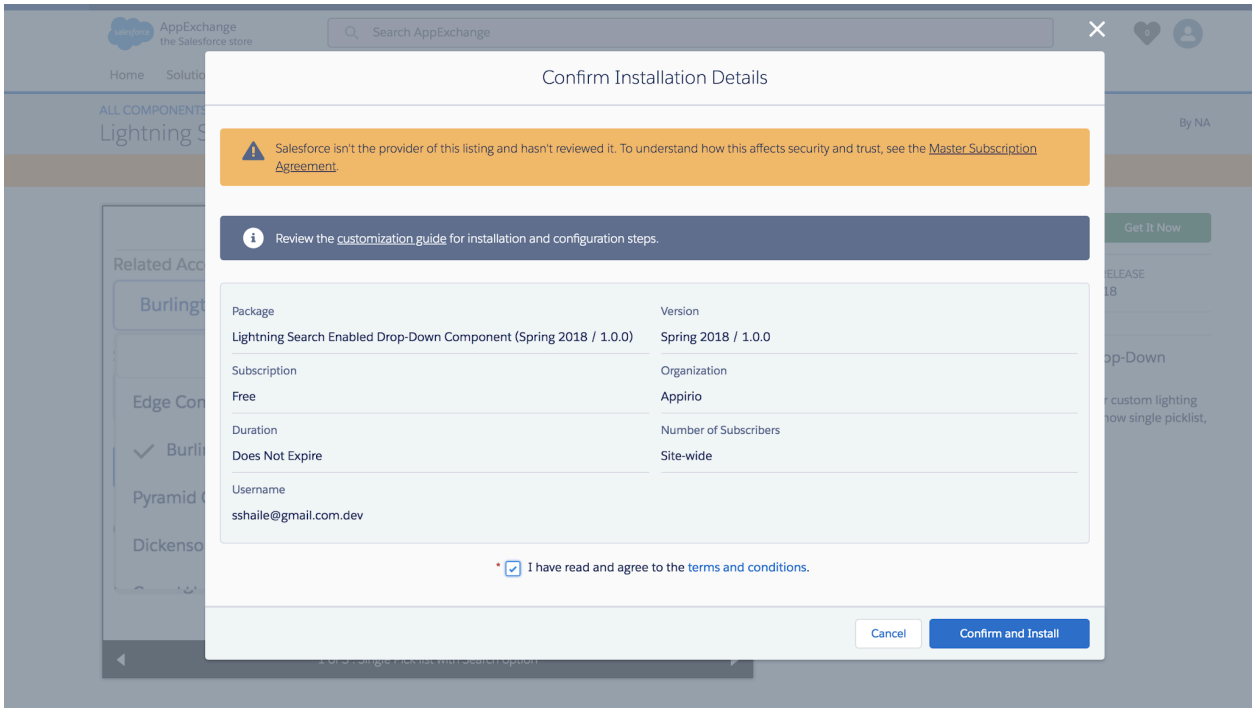
4. Component Attributes

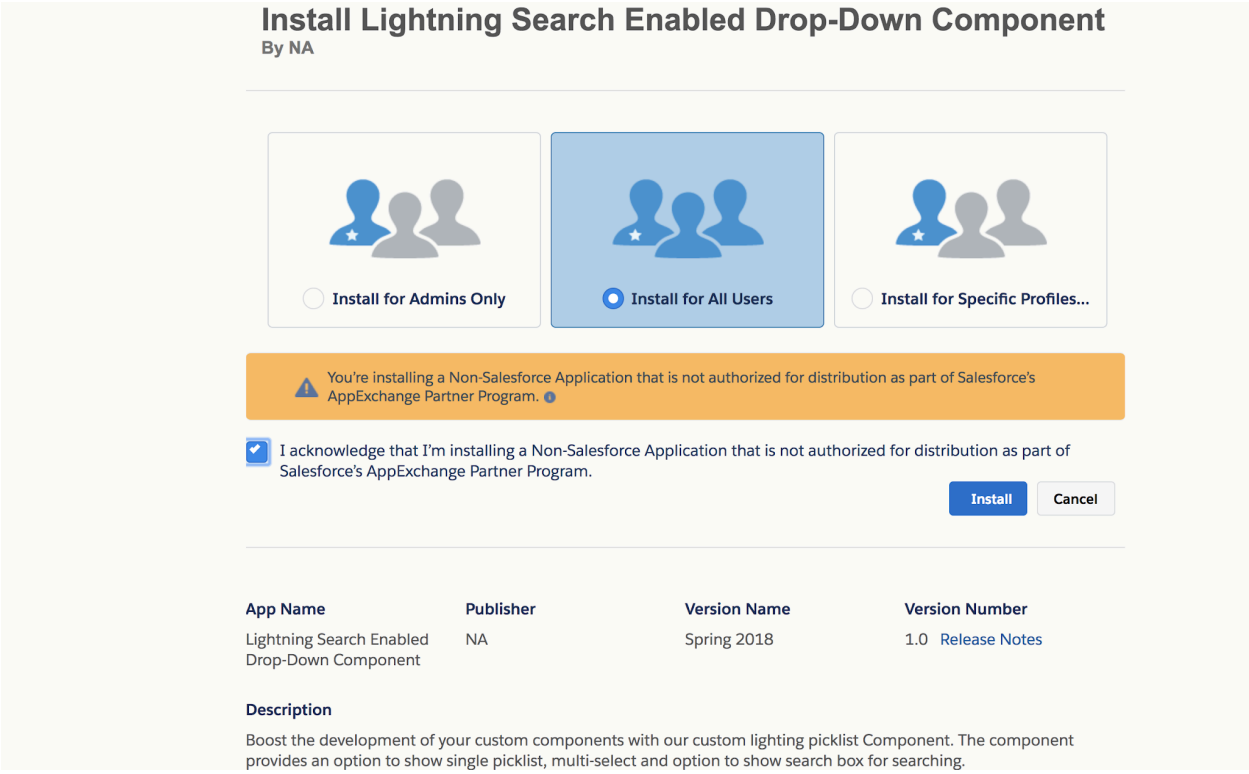
Attribute	Type	Details
options	List<SelectOption>	List of Select Options. As the aura method can't return List<SelectOption> so you can return List<Object> and then convert them in Javascript init method. **options.push({ label: 'red,' value: 1 });**

		(See example in sample code)
selectedOptions	String[]	List if selected Keys
showSearchBox	Boolean	Default = false, Set true if you want to show search box to filter options
multiselect	Boolean	Default = false, Set true for multiselect

5. Installation :

Install package from following appexchange link.
Package Link :- [Lightning Search Enabled Drop-Down Component - Astonous]





6. **How to Use it :-** After installing a package, a component is available to use in any other lighting component or application. Here is sample code to use. Refer Component attribute section for all available options.

```
<lightning:layout>
  <lightning:layoutItem padding="horizontal-small" size="6" smallDeviceSize="12"
mediumDeviceSize="6">
     <ASTPC:PicklistWidget  options="[{label: 'Red',value : 1},{label: 'Blue',value: 2}]"
                  selectedOptions="{!v.selectedColors}" showSearchBox="true"/>
  </lightning:layoutItem>
</lightning:layout>
```

7. **Support :-** We would love to provide you any support and hear from you. Please get in touch at info@astonous.com

8. Sample Code of Use

Aura Component

```
<aura:component controller="QuickOrderController"
implements="flexipage:availableForAllPageTypes">
    <aura:handler name="init" value="{!this}" action="{!c.doInit}"/>
    <aura:attribute name="accounts" type="list"/>
    <aura:attribute name="products" type="list"/>
    <aura:attribute name="selectedAccounts" type="String[]"/>
    <aura:attribute name="selectedProducts" type="String[]"/>
    <div aria-labelledby="newexpenseform">
        <fieldset class="slds-box slds-theme--default
slds-container--small">
        <div class="slds-form--stacked">
            <div class="slds-form-element">
                <label class="slds-form-element__label"
for="accountInput">Related Account (Single Select Demo)</label>
            options="{!v.accounts}" aura:id="accountInput"
                                selectedOptions="{!v.selectedAccounts}"
showSearchBox="true"/>
            </div>
                <div class="slds-form-element">
                <label class="slds-form-element__label"
for="productsInput">Select Products(Multi-Select Demo)</label>
                <ASTPC:PicklistWidget options="{!v.products}"
aura:id="productsInput" multiselect="true"
selectedOptions="{!v.selectedProducts}" showSearchBox="true"/>
            </div>
            <lightning:button label="Create Order"
                                class="slds-m-top--medium"
                                variant="brand"
                                onclick="{!c.clickCreate}"/>
        </div>
        <!-- Selected Accounts -->
        Following are selected Account (Single Select)
        <ul>
           <aura:iteration items="{!v.selectedAccounts}" var="accID">
```

```
                <li>{!accID}</li>
        </aura:iteration>
           </ul>
            <br/>
        Following are selected Products (Multi-Select)
        <ul>
           <aura:iteration items="{!v.selectedProducts}" var="pdID">
                <li>{!pdID}</li>
        </aura:iteration>
           </ul>
      </fieldset>
    </div>
</aura:component>
```

Aura Component Javascript controller

```
({
    doInit : function(component, event, helper) {
          var action = component.get("c.queryProducts");
          action.setCallback(this, function(data) {
              var result = data.getReturnValue()
                    var opts = [];
               for (var i in result){
                   opts.push({
                       class: "optionClass",
                       label: result[i],
                       value: i
                   });
               }
               component.set("v.products", opts);
       });
       $A.enqueueAction(action);

       action = component.get("c.queryAccounts");
       action.setCallback(this, function(data) {
           var result = data.getReturnValue()
           var opts = [];
           for (var i in result){
               opts.push({
                   class: "optionClass",
                   label: result[i],
                   value: i
               });
           }
           component.set("v.accounts", opts);
       });
       $A.enqueueAction(action);
    }
})
```

Apex Controller

```
public class {

    public QuickOrderController() {

    }

    @AuraEnabled
      public static Map<String, String> queryProducts() {
         Map<String, String> products = new Map<String, String>();
         for(Product2 pd : [select ID, Name from Product2 Limit 1000])
{
            products.put(pd.id, pd.Name);
         }
         return products;
    }

    @AuraEnabled
    public static Map<String, String> queryAccounts() {
         Map<String, String> accounts = new Map<String, String>();
         for(Account ac : [select ID, Name from Account Limit 1000]) {
            accounts.put(ac.id, ac.Name);
         }
         return accounts;
    }

}
```