36. Custom Settings – 16 July 2021

<https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_customsettings.htm>

Custom settings are similar to custom objects. Application developers can create custom sets of data and associate custom data for an organization, profile, or specific user. All custom settings data is exposed in the application cache, which enables efficient access without the cost of repeated queries to the database. Formula fields, validation rules, flows, Apex, and SOAP API can then use this data.

Custom Setting Types

1] List (Deprecate) 🡪 Custom Metadata Type

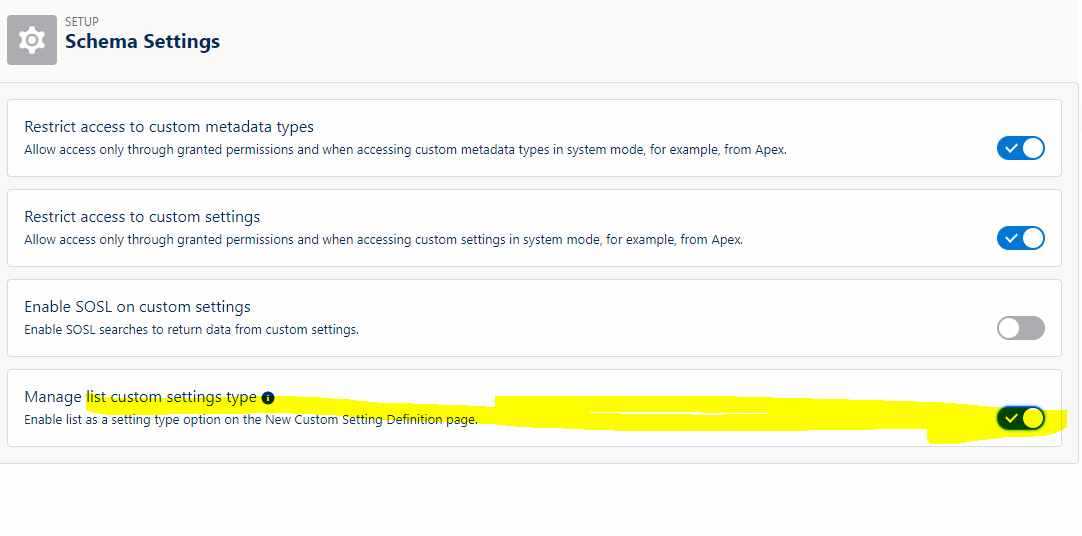
2] Hierarchy

How to Create a Custom Setting

Setup 🡪 Quick Find Box 🡪 Custom Setting 🡪 New

1] List 🡪 Enable 🡪

Quick find box 🡪 Schema Settings 🡪



1]

Map<String, Country\_Codes\_\_c> countryCodeMap = Country\_Codes\_\_c.getAll();

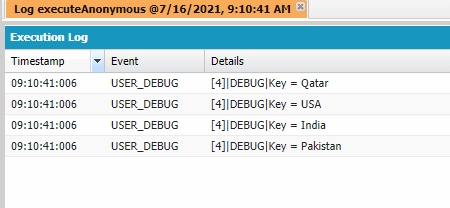
System.debug('Country = '+countryCodeMap);

2] Map<String, Country\_Codes\_\_c> countryCodeMap = Country\_Codes\_\_c.getAll();

for(String key : countryCodeMap.keySet()){

System.debug('Key = '+key);

}

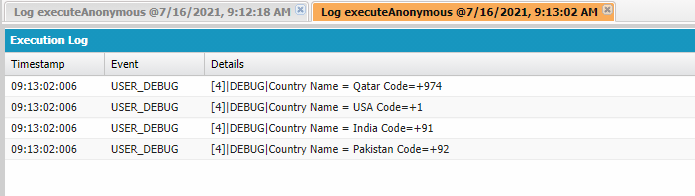


3] Map<String, Country\_Codes\_\_c> countryCodeMap = Country\_Codes\_\_c.getAll();

for(String countryName : countryCodeMap.keySet()){

System.debug('Country Name = '+countryName+' Code='+countryCodeMap.get(countryName).Code\_Number\_\_c);

}



4] Country\_Codes\_\_c country= Country\_Codes\_\_c.getInstance('India');

System.debug('Data='+country);

5] List<Country\_Codes\_\_c> countryList = [select id, Name,Code\_Number\_\_c from Country\_Codes\_\_c];

System.debug('Data = '+countryList);

6] List<Country\_Codes\_\_c> countryList = [select id, Name,Code\_Number\_\_c from Country\_Codes\_\_c];

for(Country\_Codes\_\_c objCustSe : countryList){

System.debug('County Name'+objCustSe.Name + 'Code'+objCustSe.Code\_Number\_\_c);

}

Scenario 1] Create a VF Page 🡪 Show a List of Country in a picklist.

When we select a country from the above picklist, then show its Country Code.

<https://www.sfdcpoint.com/salesforce/custom-picklist-in-visualforce-salesforce/>

<apex:page standardController="Customer\_\_c" extensions="VFCountryCode\_CX">

<apex:form >

<apex:pageblock title="Customer">

<apex:pageblockSection title="Customer Basics Information">

Country Code = {!countryCode}

<apex:selectList value="{!countryCode}" multiselect="false" size="1">

<apex:selectOptions value="{!countriesOptions}"/>

<apex:actionSupport event="onchange"/>

</apex:selectList>

</apex:pageblockSection>

</apex:pageblock>

</apex:form>

</apex:page>

public with sharing class VFCountryCode\_CX {

public String selectedCountry {get; set;}

public String countryCode {get; set;}

public VFCountryCode\_CX(ApexPages.StandardController controller) {

}

public List<SelectOption> getCountriesOptions() {

List<SelectOption> countryOptions = new List<SelectOption>();

Map<String, Country\_Codes\_\_c> countryCodeMap = Country\_Codes\_\_c.getAll();

countryOptions.add(new SelectOption('','--None--'));

for(String countryName: countryCodeMap.keySet()){

String cLabel =countryName;

String value= countryCodeMap.get(countryName).Code\_Number\_\_c;

countryOptions.add(new SelectOption(value,cLabel )); //1.Value, 2.Lable

}

return countryOptions;

}

}

Better Code:

if(!countryCodeMap.isEmpty()){

for(String countryName: countryCodeMap.keySet()){

countryOptions.add(new SelectOption(countryCodeMap.get(countryName).Code\_Number\_\_c , countryName)); //1.Value, 2.Lable

}

}