

# Salesforce Lightning Developer Training

Working Session – Flows

## UC1

A Multinational Product company uses **Shipping Country** on **Account** Object to store Account's base Country. Contacts have two country fields : **Mailing Country** and **Previous Country**(custom field).

When an **Account's Shipping Country Is Changed**, we need to update all related Contacts of the Account and copy the value of latest Shipping country in the Mailing Country field of Contacts. Also, we need a way to copy and store old Shipping Country value of Account on all related Contact records in a custom field named "Previous Country".

# UC1

## Solution Design

1. Create a custom field to store previous country name on Contact

Label: Previous Country

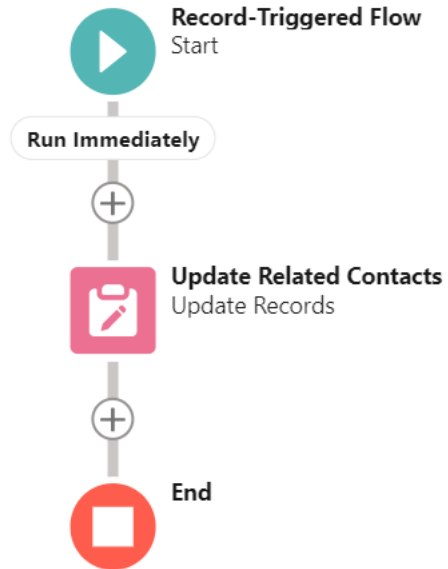
Datatype: Text

2. If Account's Shipping Country Changed

→ Update Mailing Country of all Child Contact Records with latest value of Account's Shipping Country

→ Also Store old Shipping Country Value of Account in a Custom Field (Previous Country) on all Contact Records.

# UC1 Screenshot



## UC2

A Multinational Product company wants to send the contact details to “Go Rest” system using a button on Contact Record Detail Page

Here is the sample JSON format in which the data needs to be sent to

```
{  
  "name" : "Pran Khan",  
  "email" : "khan_pran@graham.name",  
  "gender" : "male",  
  "status" : "active"  
}
```

## UC2

# Solution Design

1. Create 2 custom picklist fields on contact object.  
Status (Active, In Active)  
Gender (Male, Female, Prefer Not To Say)
2. Create a post callout to with the following details

```

@future(callout=true)
public static void sendDataGo(List<Id> contactIds) {
    //get the data and prepare json
    List<Contact> con = [Select Name, Email, Gender__c, Status__c FROM Contact WHERE Id = :contactIds[0]];
    UserInfo info = new UserInfo();
    //populate info with the details from soql
    String jsonString = JSON.serialize(info);
    System.debug(jsonString);

    //prepare request
    String endpoint = 'https://gorest.co.in/public/v2/users';
    String token = '70e1f461fd475135683473e734663791d988ca49336b3a1002405be2ff00ed29';
    HttpRequest request = new HttpRequest();
    //set endpoint, http method and request body
    request.setHeader('Content-Type', 'application/json; charset=utf-8');
    request.setHeader('Authorization', 'Bearer ' + token);

    Http http = new Http();
    HttpResponse response = http.send(request);
    //create a log => It can be info in case of success, error in case of error
}

```

## UC2 Solution Design

3. Create an invocable method so that you can call it from flow. That invocable method should be calling the callout
4. Here is the UserInfo wrapper class which we have used in the callout. FYI

```
public class UserInfo {  
    public String name;  
    public String email;  
    public String gender;  
    public String status;  
  
    public UserInfo() {  
        this.name = '';  
        this.email = '';  
        this.gender = '';  
        this.status = '';  
    }  
}
```



# UC2 Solution Design

Optional Part : A debug message can be printed for response or a new Application Log record can be created



SETUP > OBJECT MANAGER

## Application Log

Details

**Fields & Relationships**

Page Layouts

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Restriction Rules

Scoping Rules

Triggers

### Fields & Relationships

10 Items, Sorted by Field Label

Quick Find

FIELD LABEL	FIELD NAME	DATA TYPE
Application Log Name	Name	Auto Number
Created By	CreatedById	Lookup(User)
Last Modified By	LastModifiedById	Lookup(User)
Log Type	Log_Type__c	Picklist
Message	Message__c	Text Area(255)
Owner	OwnerId	Lookup(User,Group)
Stack Trace	Stack_Trace__c	Long Text Area(32768)
Status	Status__c	Picklist
Target Name	Target_Name__c	Text(80)
Target Type	Target_Type__c	Picklist

## UC2

Optional Part : the following code blocks can be used

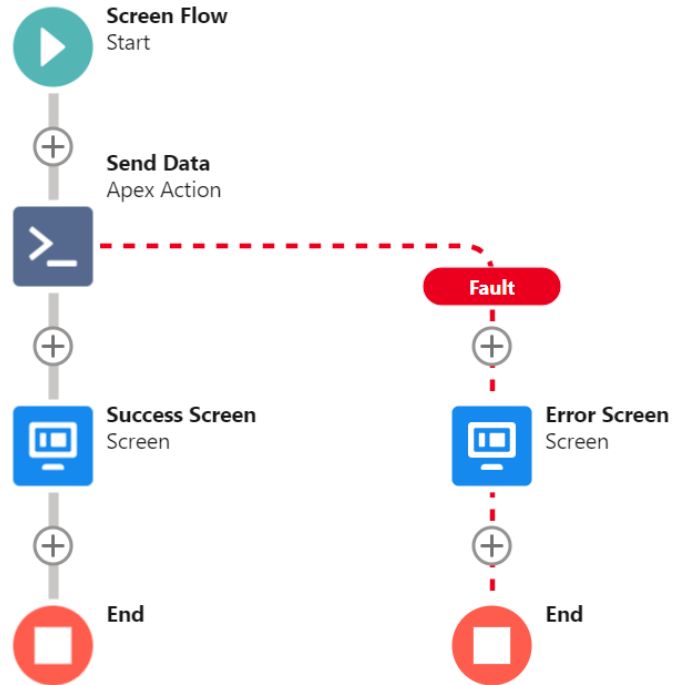
```
.....Http http = new Http();
.....HttpResponse response = http.send(request);
.....System.debug('Status Code: ' + response.getStatusCode());
.....System.debug('Response: ' + response.getBody());
.....//create a log => it can be info incase of success, error incase of error
.....if(response.getStatusCode() == 201) {
.....    //create success log
.....    CommonUtilities.createLog('Success', 'ContactCallouts.sendData', 'Integration', response.getStatus(), response.getBody());
.....} else {
.....    //create error log
.....    CommonUtilities.createLog('Error', 'ContactCallouts.sendData', 'Integration', response.getStatus(), response.getBody());
.....}
.....}
```

```
1 public with sharing class CommonUtilities {
2     public static void createLog(String logType, String targetName, String targetType, String message, String stackTrace) {
3         .....Application_Log__c log = new Application_Log__c();
4         .....log.Log_Type__c = logType;
5         .....log.Target_Name__c = targetName;
6         .....log.Target_Type__c = targetType;
7         .....log.Message__c = message;
8         .....log.Stack_Trace__c = stackTrace;
9         .....insert log;
10    .....}
11 }
```

## UC2 Solution Design

4. Create an action which invokes the flow. Make the button available on contact record detail page

# UC2 Screenshot



## UC3

Create a screen flow which lets the user to choose loan type. Based on the loan type, the system should show relevant fields. Ex: when we choose home loan, home loan related fields should be visible and others should be hidden

## UC3

# Solution Design

Create a loan object with the following fields:

1. Loan Number (Auto Number: LOAN-{0000})
2. Loan Type (Picklist - Home Loan, Car Loan, Personal Loan)
3. Person (MD-Contact)
4. Home Type (Text)
5. Home Location (Text)
6. Home Cost (Currency)
7. Car Type (Text)
8. Car Vendor (Text)
9. Car Cost (Currency)
10. Personal Need (Text)
11. Personal Amount (Currency)

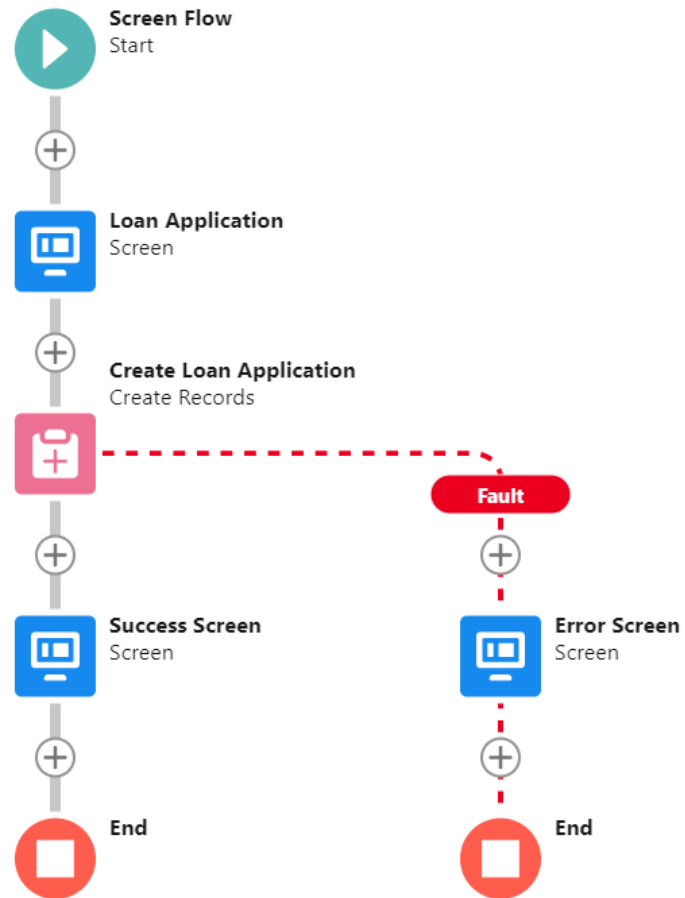
## UC3

# Solution Design

Create a screen flow.

1. Show the field loan type.
2. If user chooses home loan in step 1, The flow should show up Home Type, Home Location & Home Cost for the user to enter data.
3. Else, if the user chooses car loan step 1, the flow should show up car type, car vendor & car cost
4. Else, if the user chooses personal loan in step 1, the flow should show up personal need and amount fields
5. Create an action on contact object and place the action in the contact detail page

# UC3 Screenshot





Thanks  
Questions





# Great!

To follow up recent updates from development world,  
you can find us on [www.softinnovas.com](http://www.softinnovas.com)