1.

<!-- ---------------------------------------------------------------------- -->

<!-- NOTE: Please add the following <META> element to your page <HEAD>. -->

<!-- If necessary, please modify the charset parameter to specify the -->

<!-- character set of your HTML page. -->

<!-- ---------------------------------------------------------------------- -->

<META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=UTF-8">

<!-- ---------------------------------------------------------------------- -->

<!-- NOTE: Please add the following <FORM> element to your page. -->

<!-- ---------------------------------------------------------------------- -->

<form action="https://webto.salesforce.com/servlet/servlet.WebToLead?encoding=UTF-8" method="POST">

<input type=hidden name="oid" value="00D6F000001OsOC">

<input type=hidden name="retURL" value="https://coderinme.com">

<!-- ---------------------------------------------------------------------- -->

<!-- NOTE: These fields are optional debugging elements. Please uncomment -->

<!-- these lines if you wish to test in debug mode. -->

<!-- <input type="hidden" name="debug" value=1> -->

<!-- <input type="hidden" name="debugEmail" value="saifi@coderinme.com"> -->

<!-- ---------------------------------------------------------------------- -->

<label for="first\_name">First Name</label><input id="first\_name" maxlength="40" name="first\_name" size="20" type="text" /><br>

<label for="last\_name">Last Name</label><input id="last\_name" maxlength="80" name="last\_name" size="20" type="text" /><br>

<label for="email">Email</label><input id="email" maxlength="80" name="email" size="20" type="text" /><br>

<label for="company">Company</label><input id="company" maxlength="40" name="company" size="20" type="text" /><br>

<label for="city">City</label><input id="city" maxlength="40" name="city" size="20" type="text" /><br>

<label for="state">State/Province</label><input id="state" maxlength="20" name="state" size="20" type="text" /><br>

SFId:<input id="00N6F00000HqnP2" maxlength="20" name="00N6F00000HqnP2" size="20" type="text" /><br>

<input type="submit" name="submit">

</form>

2.

global class myHandler implements Messaging.InboundEmailHandler {

global Messaging.InboundEmailResult handleInboundEmail(Messaging.InboundEmail email, Messaging.InboundEnvelope envelope) {

Messaging.InboundEmailResult result = new Messaging.InboundEmailresult();

return result;

}

}

3.

global class batchFormatClass implements Database.Batchable{

global Database.QueryLocator start(Database.BatchableContext BC){

string query='';

return Database.getQueryLocator(query);

}

global void execute(Database.BatchableContext BC, List<sObject> scope){

for(sobject s : scope){

// operation or process

}

update scope;

}

global void finish(Database.BatchableContext BC){

}

}

global class batchAccountUpdate implements Database.Batchable<sObject> {

//start

global Database.QueryLocator start(Database.BatchableContext BC){

// query from an object for records

String query='SELECT Id, Name FROM Account';

return Database.QueryLocator(query);

}

//execute

global void execute(Database.BatchableContext BC,List<Account > Scope){

// here we will write logic for that what we want to do

integer i=1;

for(Account a: Scope){

a.name=a.name+i;

i++;

}

update ac;

system.debug('mine'+ac);

}

//finish

global void finish(Database.BatchableContext BC){

/\* after the completion of batch anything we want to do like a confirmation of batch execution/\*

}

}

//scheduler

global class scheduledBatchable implements Schedulable {

global void execute(SchedulableContext sc) {

// batch class name do you want to call

batchAccountUpdate b = new batchAccountUpdate();

// size of batch how much record in single batch

database.executebatch(b,200);

}

}

batchAccountUpdate b = new batchAccountUpdate();

// size of batch how much record in single batch

database.executebatch(b,200);

4.

Http ht = new Http();

HttpRequest req = new HttpRequest ();

String key =''; // you can get the google api key from google services

String url ='https://maps.googleapis.com/maps/api/geocode/json?key='+key+'&latlng=28.5810215,77.3152004&sensor=true';

req.setEndpoint(url);

req.setMethod('GRT');

//req.setBody('body');

HttpResponse res = ht.send(req);

if(res.getStatusCode()==200){

string str= res.getBody().split('"formatted\_address" : "')[1].split('",')[0];

system.debug(str);

}

5.

trigger updateMap on Contact (after insert, after update) {

for(Contact c: Trigger.new){

if(trigger.isUpdate){

// for update if location change we will call apiMap Class

if(c.Latitude\_\_c != Trigger.oldMap.get(c.id).Latitude\_\_c || c.Longitude\_\_c != Trigger.oldMap.get(c.id).Longitude\_\_c)

apiMap.chckMap(String.valueOf(c.Latitude\_\_c), string.valueOf(c.Longitude\_\_c), c.id);

}

if(trigger.isInsert)

apiMap.chckMap(String.valueOf(c.Latitude\_\_c), string.valueOf(c.Longitude\_\_c), c.id);

// we are passing three args longitude, latitude and contact Id.

}

}

public class apiMap {

/\* if we are using api call we will use future callout then trigger will support API Callout \* /

@future (callout=true)

public static void chckMap(string lat, string lon, string conId){

// we will use these lat and lon on api url

Http h = new Http();

//28.5810215,77.3152004

HttpRequest req = new HttpRequest ();

String url ='https://maps.googleapis.com/maps/api/geocode/json?key=AIzaSyDCJfSJhXuKJlffbFfB57yOO\_iQK4kAmio&latlng='+lat+','+lon+'&sensor=true';

req.setEndpoint(url);

req.setMethod('GET');

HttpResponse res = h.send(req);

if(res.getStatusCode()==200){

string str= res.getBody().split('"formatted\_address" : "')[1].split('",')[0];

system.debug(str);

// update contact with address from API response

contact c= new Contact();

c.id=conId;

c.address\_\_c=str;

update c;

}

// making a log for every API Hit for best practice

httpRequest\_\_c log1= new httpRequest\_\_c();

log1.Request\_\_c='MapAPI';

log1.response\_\_c=str;

log1.status\_\_c=String.valueOf(res.getStatusCode());

insert log1;

}

6.

RestRequest req = RestContext.request;

RestResponse res = RestContext.response;

ApplicantDetails appDetails = (ApplicantDetails)JSON.deserialize(req.requestBody.toString(),ApplicantDetails.class);

public class ApplicantDetails {

public String Session\_Key;

public String Method\_Type;

public String Application\_Id;

public String Applicant\_Id;

}

//Code to call

If we want to send the request to any other software, then our code will be like this:

htttp hreq = new Http();

HttpRequest req = new HttpRequest();

//req.setHeader('userid','mukeshlms');

req.setHeader('Content-type', 'application/json');

req.setEndpoint(url);

req.setMethod('POST');

req.setBody(body);

HttpResponse res = hreq.send(req);

System.debug(‘>>>’+req.getbody())