global class EmailServices implements Messaging.InboundEmailHandler {

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

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

for (Messaging.Inboundemail.TextAttachment tAttachment : email.textAttachments) {

Attachment attachment = new Attachment();

attachment.Name = tAttachment.fileName;

Blob myBlob = Blob.valueOf(tAttachment.body);

XMLparse xmlParse= new XMLparse();

xmlParse.parseXML(myBlob.toString());

}

result.success = true;

return result;

}

}

XMLparse CLass:-

public class XMLparse{

string XMLString;

contact tempcont;

List<Contact> contactList;

public void parseXML(String XMLString ){

contactList = new list<Contact>();

DOM.Document doc=new DOM.Document();

try{

doc.load(XMLString);

DOM.XmlNode rootNode=doc.getRootElement();

parseXML(rootNode);

contactList.add(tempcont);

insert contactList;

}catch(exception e){

system.debug(e.getMessage());

}

}

private void parseXML(DOM.XMLNode node) {

if (node.getNodeType() == DOM.XMLNodeType.ELEMENT) {

if(node.getName()=='contact'){

if(tempcont!=null)

contactList.add(tempcont);

tempcont = new contact();

}

if(node.getName()=='fname')

tempcont.FirstName=node.getText().trim();

if(node.getName()=='lname')

tempcont.LastName=node.getText().trim();

}

for (Dom.XMLNode child: node.getChildElements()) {

parseXML(child);

}

}

}