**src/main/resources/script/script3.groovy**

import com.sap.gateway.ip.core.customdev.util.Message;
import java.util.HashMap;
def Message processData(Message message) {
def body = message.getBody(java.lang.String) as String;
def messageLog = messageLogFactory.getMessageLog(message);
if(messageLog != null){
messageLog.setStringProperty("Logging#1", "Printing Payload As Attachment")
messageLog.addAttachmentAsString("Log Output:", body, "text/plain");
}
return message;
}

**src/main/resources/script/script5.groovy**

import com.sap.gateway.ip.core.customdev.util.Message;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathFactory;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import org.w3c.dom.Element;
import java.util.ArrayList;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.w3c.dom.Document;
import groovy.util.slurpersupport.GPathResult;
import java.util.HashMap;
import java.util.Date;
import static java.util.Calendar.\*
import groovy.time.TimeCategory
import com.sap.it.spi.ITApiHandler;
import com.sap.it.api.ITApiFactory;
import com.sap.it.api.mapping.ValueMappingApi;
import java.util.regex.Pattern;
import groovy.transform.AutoClone
import com.sap.gateway.ip.core.customdev.util.Message;
import java.util.HashMap;
def Message processData(Message message) {
//Body
def body = message.getBody().toString();
messageLog.addAttachmentAsString("Before", body.toString(), "text/plain")
body = body.replaceAll("\\<\\?xml(.+?)\\?\\>", "").trim()
messageLog.addAttachmentAsString("After", body.toString(), "text/plain")
def slurper = new XmlSlurper().parseText(body)
message.setBody(nodes)
return message;
}

**src/main/resources/script/script2.groovy**

import com.sap.gateway.ip.core.customdev.util.Message;
import java.util.HashMap;
def Message processData(Message message) {
def body = message.getBody(java.lang.String) as String;
def messageLog = messageLogFactory.getMessageLog(message);
if(messageLog != null){
messageLog.setStringProperty("Logging#1", "Printing Payload As Attachment")
messageLog.addAttachmentAsString("Log Output:", body, "text/plain");
}
return message;
}

**src/main/resources/script/script4.groovy**

import com.sap.gateway.ip.core.customdev.util.Message;
import static javax.xml.xpath.XPathConstants.\*
import javax.xml.\*
import groovy.xml.\*
import groovy.xml.QName
import groovy.xml.dom.DOMCategory
import org.w3c.dom.Document;
import java.util.HashMap;
def Message processData(Message message) {
//take the body
def xml = message.getBody(java.lang.String) as String;
//take property with Student ID
def reader = new StringReader(xml)
def doc = DOMBuilder.parse(reader)
def records = doc.documentElement
def text=""
use(DOMCategory) {
records.EmpEmployment.each{
ex-> text = text + ex.userId.text() +", "
}
}
message.setProperty('Text',text.toString());
return message;
}

**src/main/resources/script/script1.groovy**

import com.sap.gateway.ip.core.customdev.util.Message;
import java.util.HashMap;
def Message processData(Message message) {
def body = message.getBody(java.lang.String) as String;
def messageLog = messageLogFactory.getMessageLog(message);
if(messageLog != null){
messageLog.setStringProperty("Logging#1", "Printing Payload As Attachment")
messageLog.addAttachmentAsString("Log Output:", body, "text/plain");
}
return message;
}