



SAP Community > Products and Technology > Technology > Technology Blogs by Members > My adventure in learning Groovy Script for Cloud I...

Technology Blogs by Members

Explore a vibrant mix of technical expertise, industry insights, and tech buzz in member blogs covering SAP products, technology, and events. Get in the mix!

Blog



What are you looking for today?

My adventure in learning Groovy Script for Cloud Integration (CI) – Part 5



raviteja_satuluri

Contributor



2023 Feb 08 7:57 AM



11 Kudos

👁 11,490

SAP Managed Tags: SAP HANA Cloud, SAP Integration Suite, Cloud Integration, SAP BTP, Cloud Foundry runtime and environment, SAP Cloud Integration for data services, SAP Cloud Platform, industry edition, client libraries

Introduction

Here I am just explaining the Importance of Groovy script in the Real time scenarios when we are dealing with Cloud Integration.

Groovy scripting is **an integral and important feature of SAP Cloud Platform Integration (CPI)**. The goals of this repository are: Providing templates when you are implementing a new script. Easily finding Groovy functions related to the CPI topic at hand. Minimizing search engine time for common tasks.

Case 1:

Business Requirement:

Create an Integration Scenario to log the Incoming Payload

Solution: The below Groovy Script will help you to capture the Incoming payload whenever you want in the E2E Execution process.

Code:

```
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("Payload", body, "text/plain");

}

return message;

}
```

Case 2:

Business Requirement:

Create an Integration Scenario to record logs or capture the Incoming Payload only if there is an Exception.

Solution: The below Groovy Script will help you to capture the Record logs or capture the Incoming Payload only if there is an exception occurs during E2E Execution process.

Groovy Functions used: getMessage() & getMessageLog(message)

Code:

```
import com.sap.gateway.ip.core.customdev.util.Message;

import java.util.HashMap;

def Message processData(Message message)
```

```
{  
  
def map = message.getProperties();  
  
def ex = map.get("CamelExceptionCaught");  
  
if (ex != null) {  
  
exceptionText = ex.getMessage();  
  
def messageLog = messageLogFactory.getMessageLog(message);  
  
messageLog.addAttachmentAsString("Exception", exceptionText,"application/text");  
  
}  
  
return message;  
  
}
```

Case 3:

Business Requirement:

Create an Integration Scenario to remove XML tags from your Payload.

Solution: The below Groovy Script will help you to remove XML tags from your Incoming payload.

Groovy Function Involved: `replace()`.

Code:

Input	Script Copy Share URL	Output
<pre> 1 <?xml version="1.0" encoding="UTF-8"?> 2 <catalog> 3 <book id="bk101"> 4 <author>Gambardella, Matthew</author> 5 <title>XML Developer's Guide</title> 6 <genre>Computer</genre> 7 <price>44.95</price> 8 <publish_date>2000-10-01 9 </publish_date> 10 <description>An in-depth look at 11 creating applications 12 with XML.</description> 13 </book> 14 </catalog> </pre>	<pre> 1 import com.sap.gateway.ip.core.customdev.util.Message; 2 import java.util.HashMap; 3 def Message processData(Message message) 4 { 5 def removal=message.getBody(java.lang.String) as String; 6 removal=removal.replace(<?xml version="1.0" encoding="UTF-8"?>/, ""); 7 message.setBody(removal); 8 return message; 9 } 10 11 </pre>	<pre> 1 2 <catalog> 3 <book id="bk101"> 4 <author>Gambardella, Matthew</author> 5 <title>XML Developer's Guide</title> 6 <genre>Computer</genre> 7 <price>44.95</price> 8 <publish_date>2000-10-01 9 </publish_date> 10 <description>An in-depth look at 11 creating applications 12 with XML.</description> 13 </book> 14 </catalog> </pre>

Case 4:

Business Requirement:

Create an Integration Scenario, which will return all the Headers and Exchange Properties

Solution: The below Groovy script will help you to capture or fetch all the Headers and Exchange properties that you are created in your Integration Process.

Groovy Functions Involved: `getHeaders()` & `getProperties()`

Code:

```
import com.sap.gateway.ip.core.customdev.util.Message;
```

```
import java.util.HashMap;
```

```
def Message processData(Message message) {
```

```
    //Body
```

```
    def body = message.getBody();
```

```
    //Get Headers
```

```
    def map = message.getHeaders();
```

```
    def value = map.get("FNAME");
```

```
    //Get Properties
```

```
    map = message.getProperties();
```

```
    value = map.get("Company");
```

```
    return message;
```

```
}
```

Input		Script Copy Share URL	Output							
Headers <div>add header</div> <table border="1"> <tr> <td>FNAME</td> <td>Raviteja Satuluri</td> <td>remove</td> </tr> </table>		FNAME	Raviteja Satuluri	remove	<pre> 1 import com.sap.gateway.ip.core.customdev.util.Message; 2 import java.util.HashMap; 3 4 def Message processData(Message message) { 5 //Body 6 def body = message.getBody(); 7 8 //Get Headers 9 def map = message.getHeaders(); 10 def value = map.get("FNAME"); 11 12 //Get Properties 13 map = message.getProperties(); 14 value = map.get("Company"); 15 16 return message; 17 }</pre>	Headers <table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>FNAME</td> <td>Raviteja Satuluri</td> </tr> </tbody> </table>	Key	Value	FNAME	Raviteja Satuluri
FNAME	Raviteja Satuluri	remove								
Key	Value									
FNAME	Raviteja Satuluri									
Properties <div>add property</div> <table border="1"> <tr> <td>Company</td> <td>Knack</td> <td>remove</td> </tr> </table>		Company	Knack	remove	Properties <table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Company</td> <td>Knack</td> </tr> </tbody> </table>	Key	Value	Company	Knack	
Company	Knack	remove								
Key	Value									
Company	Knack									

Case 5:

Business Requirement:

Create an Integration Scenario to capture a property in the content modifier using XPATH, use single IF condition, create and update a new Property.

Solution: The below Groovy script will help you to update the value of custom Header or Exchange Property based on some condition. In below Code we are creating a Custom Header 'Result' with the value either True or False based on condition on existing Exchange Property 'ID'.

Case 1: if (ID== Blank) then Result 'false'

Code:

Input

Headers

add header

Properties

add property

ID	Value	remove
----	-------	--------

Script

Copy Share URL

```

1 import com.sap.gateway.ip.core.customdev.util.Message;
2 import java.util.HashMap;
3
4 def Message processData(Message message)
5 {
6     map = message.getProperties();
7     def zID = map.get("ID");
8     def Indicator;
9     if(zID == "")
10 {
11     Indicator = 'false';
12 } else {
13     Indicator = 'true';
14 }
15 message.setProperty("Result", Indicator);
16 return message;
17 }

```

Output

Headers

empty

Properties

Key	Value
ID	
Result	false

Case 2: if (ID!= Blank) then Result 'true'

Input

Headers

add header

Properties

add property

ID	Knack	remove
----	-------	--------

Script

Copy Share URL

```

1 import com.sap.gateway.ip.core.customdev.util.Message;
2 import java.util.HashMap;
3
4 def Message processData(Message message)
5 {
6     map = message.getProperties();
7     def zID = map.get("ID");
8     def Indicator;
9     if(zID == "")
10 {
11     Indicator = 'false';
12 } else {
13     Indicator = 'true';
14 }
15 message.setProperty("Result", Indicator);
16 return message;
17 }

```

Output

Headers

empty

Properties

Key	Value
ID	Knack
Result	true

Case 6:

Business Requirement:

Create an Integration Scenario to remove special characters from the Incoming Payload

Solution: The below Groovy Script will help you to remove unwanted special characters in the Incoming Payload during E2E Execution process.

Groovy Functions Involved: replace()

Code:

Input	Script Copy Share URL	Output
<pre>1 <?xml version="1.0" encoding="UTF-8"?> 2 <catalog> 3 <book id="bk101"> 4 <author>Gambardella@, Matthew</author> 5 <title>XML Developer's Guide!</title> 6 <genre>Computer%</genre> 7 <price>#44.95</price> 8 <publish_date>2000-10-01 9 </publish_date> 10 <description>An in-depth look at 11 creating applications 12 with XML.</description> 13 </book> 14 </catalog></pre>	<pre>1 import com.sap.gateway.ip.core.customdev.util.Message; 2 import java.util.HashMap; 3 def Message processData(Message message) 4 { 5 def configXml=message.getBody(java.lang.String) as String; 6 configXml=configXml.replace("@",""); 7 configXml=configXml.replace("#",""); 8 configXml=configXml.replace("%",""); 9 configXml=configXml.replace("!",""); 10 configXml=configXml.replace("-",""); 11 message.setBody(configXml); 12 return message; 13 } 14</pre>	<pre>1 <?xml version="1.0" encoding="UTF8"?> 2 <catalog> 3 <book id="bk101"> 4 <author>Gambardella, Matthew</author> 5 <title>XML Developer's Guide</title> 6 <genre>Computer</genre> 7 <price>44.95</price> 8 <publish_date>20001001 9 </publish_date> 10 <description>An indepth look at 11 creating applications 12 with XML.</description> 13 </book> 14 </catalog></pre>

Case 7:

Business Requirement:

Create an Integration Scenario to make your IFLOW sleep or stop or pause for some time.

Solution: The below Groovy Script will help you to stop or Pause your entire IFLOW execution for some time, so that the message will deliver to target system with some delay.

Groovy Functions Involved: sleep()

Code:

```
import com.sap.gateway.ip.core.customdev.util.Message;
```

```
import java.util.HashMap;
```

```
def Message processData(Message message) {
```

```
    def body = message.getBody();
```

```
    sleep(60000); [60000 represents Milli seconds]
```

```
    message.setBody(body);
```

```
    return message;
```

```
}
```

Case 8:

Business Requirement:

Create an Integration Scenario to Concat two fields.

Solution: The below Groovy Script will help you to concat 2 fields.

Code:

The screenshot displays the SAP Integration Suite interface for creating an integration scenario. It is divided into three main sections: Input, Script, and Output.

Input Section:

- Headers:** A button labeled "add header".
- Properties:** A table with three rows: "First Name" (Satuluri), "Middle Name" (Value), and "Last Name" (Raviteja). Each row has a "remove" button.

Script Section:

- Script:** A code editor showing the following Groovy script:

```
1 import com.sap.gateway.ip.core.customdev.util.Message;
2 import java.util.HashMap;
3 import java.lang.*;
4 def Message processData(Message message)
5 {
6     map = message.getProperties();
7     def ZF1 = map.get("First Name");
8     def ZF2 = map.get("Middle Name");
9     def ZF3 = map.get("Last Name");
10    String Name = ZF1.concat(ZF2).concat(ZF3);
11    message.setProperty("Full Name",Name);
12
13    return message;
14 }
```
- Script Function:** A dropdown menu showing "processData".
- Console:** A button labeled "Click to Execute Script (Alt+Enter)".

Output Section:

- Headers:** The text "empty".
- Properties:** A table with four rows: "First Name" (Satuluri), "Full Name" (Satuluri Raviteja), "Middle Name" (Value), and "Last Name" (Raviteja). The "Full Name" row is highlighted with a green border.

Case 9:

Business Requirement:

Create an Integration Scenario to count the total number of attachments from an Email Message

Solution: The below Groovy Script will help you to count the Number of attachments which are attached to the message.

For example, if there is an Email message which is having multiple attachments and where we need to count the number of attachments, the count value has to be populated to the Target System.

Groovy Functions Involved: **getAttachments()** & **attachments.size()**

Code:

```
import com.sap.gateway.ip.core.customdev.util.Message
```

```
import java.util.Map
```

```
import java.util.Iterator
```

```
import javax.activation.DataHandler
```

```
def Message processData(Message message)
```

```
{
```

```
// get attachments
```


```
Map<String, DataHandler> attachments = message.getAttachments();
```


```
if (attachments.isEmpty()) {  
  
    // in case of no attachment  
  
    message.setBody('<warning>Attachment is missing</warning>');  
  
} else {  
  
    // get overall number of attachments  
  
    message.setHeader('NumberOfAttachments', attachments.size());  
  
}  
  
return message  
  
}
```

From Sender:

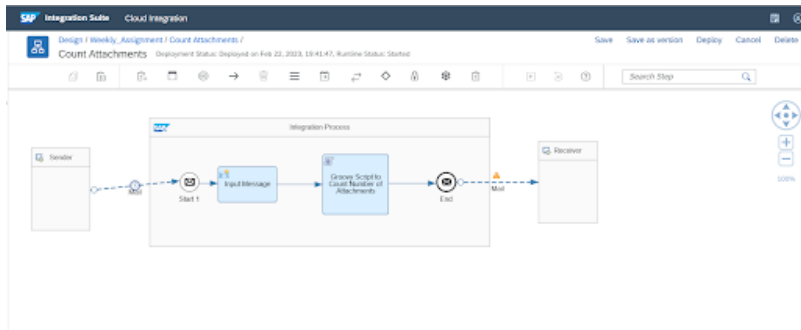
We are sending 2 attachments.

Attachment count - Hello team, please find the below attachments.

 book1.xml

 MATMAS IDOC....

IFLOW Created:



From Message log:

Message before Step	
<div>Header Exchange Properties Payload</div>	
Name	Value
NumberOfAttachments	2
SAP_MessageProcessingLog ID	AGP2I_MgtLvcL_XYgNjv5HJpDtyS

Case 10:

Business Requirement:

Create an Integration Scenario to raise an exception when there are no attachments found from an Email Message.

Solution: The below Groovy Script will help you to throw a custom exception when there are no attachments to the email message.

Code:

```
import com.sap.gateway.ip.core.customdev.util.Message

import java.util.Map

import java.util.Iterator

import javax.activation.DataHandler

def Message processData(Message message) {

    Map<String, DataHandler> attachments = message.getAttachments();

    if (attachments.isEmpty()) {

        throw new Exception ("No Attachment found");

    } else {

        //Handling of available attachments

        attachments.values().each { attachment ->

            if(attachment.getName().contains("book1")) {

                message.setBody(attachment.getContent());

            }

        }

    }

}
```

```
}  
  
return message;  
  
}
```

Read Attachments Last Updated at: Feb 22, 2023, 20:15:24

[Status](#) [Properties](#) [Logs](#) [Artifact Details](#)

Message processing failed.

Processing Time: 1 sec 526 ms

Error Details

javax.script.ScriptException: java.lang.Exception: java.lang.Exception: No Attachment found@
line 10 in script1.groovy, cause: java.lang.Exception: No Attachment found

Case 11:

Business Requirement:

Create an Integration Scenario to create custom Header and Exchange Properties

Solution: The below Groovy script will help you to know how to create a Custom Exchange Property. In this case, we are dealing with IDOC Payload where IDOC Number is an unique identifier for the Message. To capture the IDOC Number we are creating a custom Property.

The IDOC Number will be appear in the Message logs.

Code:

Input

Headers

add header		
IDOCNUM	500007	remove

Properties

add property	
--------------	--

Script

[Copy](#) [Share URL](#)

```

1 import com.sap.gateway.ip.core.customdev.util.Message;
2
3 def Message processData(Message message)
4 {
5     def messageLog = messageLogFactory.getMessageLog(message);
6     if(messageLog != null) {
7         //Read IDoc number from Header
8         def IDOCNUM = message.getHeaders().get("IDOCNUM");
9         //Set IDoc number as Custom Header
10        if(IDOCNUM!=null)
11            messageLog.addCustomHeaderProperty("IDOCNUM", IDOCNUM);
12        }
13    return message;
14 }

```

Output

Headers

Key	Value
IDOCNUM	500007

Properties

empty

Message Processing Logs

Log Properties

empty

Custom Header Properties

Name	Values
IDOCNUM	["500007"]

Case 12:

Business Requirement:

Create an Integration Scenario to perform the JSON to XML conversion.

Solution: The below Groovy script will help you to understand, how to convert JSON to XML.

Groovy Function Involved: **JsonSlurper()**.

Code:

```
import com.sap.gateway.ip.core.customdev.util.*;
```

```
import groovy.xml.*;
```

```
import groovy.json.*;
```

```
def Message processData(Message message)
```

```
{
```

```
//Body
```

```
def body = message.getBody(String)
```

```
def json = new JsonSlurper().parseText(body)
```

```
def writer = new StringWriter()
```

```
def builder = new MarkupBuilder(writer)
```

```
builder
```

```
{
```

```
"employees"
```

```
{
```

```
firstName (json.employees.firstName )
```

```
lastName (json.employees.lastName )
```

```
}
```

```
}
```

```
message.setBody(writer.toString())
```

```
return message;
```

```
}
```

Input of JSON:

```
{"employees":[ { "firstName":"Jhansi", "lastName":"rani" }, { "firstName":"Samiksha", "lastName":"raut" }, {  
"firstName":"bala", "lastName":"chandra" } ]}
```

Output:

```
<call> <employees> <firstName>[Jhansi, Samiksha, bala]</firstName> <lastName>[rani, raut, chandra]</lastName>  
</employees> </call>
```

```

1 {"employees":[
2   { "firstName":"Jhansi", "lastName":"rani" }
3   { "firstName":"Samiksha", "lastName":"raut" }
4   { "firstName":"bala", "lastName":"chandra" }
5 ]}

```

Headers

add header

```

1 import com.sap.gateway.ip.core.customdev.util.*;
2 import groovy.xml.*;
3 import groovy.json.*;
4 def Message processData(Message message)
5 {
6   //Body
7   def body = message.getBody(String)
8   def json = new JsonSlurper().parseText(body)
9   def writer = new StringWriter()
10  def builder = new MarkupBuilder(writer)
11  builder
12  { "employees"
13  {
14    firstName (json.employees.firstName )
15    lastName (json.employees.lastName )
16  }
17  }
18  message.setBody(writer.toString())
19  return message;
20 }

```

```

1 <call>
2   <employees>
3     <firstName>[Jhansi, Samiksha, bala]</firstName>
4     <lastName>[rani, raut, chandra]</lastName>
5   </employees>
6 </call>

```

Properties

Key	Value
-----	-------

Case 13:

Business Requirement:

Create an Integration Scenario to read the IDOC Number extracted in Content Modifier and set it as a Custom Header.

Solution: The below Groovy Script will help you to Read IDOC Number from Header and setting up the same IDOC number as a Custom Header.

Code:

```
import com.sap.gateway.ip.core.customdev.util.*;
```

```
def Message processData(Message message)
```

```
{  
  
def messageLog = messageLogFactory.getMessageLog(message);  
  
if(messageLog != null)  
  
{  
  
//Read IDoc number from Header  
  
def IDOCNUM = message.getHeaders().get("SAP_ApplicationID");  
  
//Set IDoc number as Custom Header  
  
if(IDOCNUM!=null)  
  
messageLog.addCustomHeaderProperty("IDOCNUM", IDOCNUM);  
  
def SERIAL = message.getHeaders().get("SAP_MessageType");  
  
//Set IDoc number as Custom Header  
  
if(SERIAL!=null)  
  
messageLog.addCustomHeaderProperty("SERIAL", SERIAL);  
  
}  
  
return message;
```

```
}
```

Groovy script for set Body:

```
import com.sap.gateway.ip.core.customdev.util.Message;
```

```
import java.util.HashMap;
```

```
import java.lang.*;
```

```
def Message processData(Message message)
```

```
{
```

```
    map = message.getHeaders();
```

```
    message.setBody(map);
```

```
    return message;
```

```
}
```

To read the IDoc Number set as a Custom Header Last updated at: Feb 22, 2023, 11:06:09

Status Properties Logs Attachments Artifact Details

Message processing completed successfully.

Processing Time: 523 ms

Properties

Message ID: AGP1qXvKSvVh0BEeAnv7R1cX3UJ
 Correlation ID: AGP1qXv3CWUUK5KhgJUN1Cqe42
 Sender: SAP ERP
 Receiver: HTTP BIN
 Application Message ID: 0000000042766395
 Application Message Type: 20220111183833

Custom Headers (2)

Name	Value
IDOCNUM	0000000042766395
SERIAL	20220111183833

Logs

Case 14:

Business Requirement:

Create an Integration Scenario for a Custom message whenever there is an exception raised by CPI

For Example: if any message fails either due to data issue or connection issue, we will capture the Error log details. Here in this case, From the Payload we want to show Title & Price details for the failed message in the Error log.

Solution: The below Groovy Script will help you to show the Title and Price details in Error Message logs. Whichever source fields that you want to show in the Message error log those details can be highlighted using below Script.

Code:

```
import com.sap.gateway.ip.core.customdev.util.*;
```

```
import java.util.HashMap;

def Message processData(Message message)

{

def cus_properties = message.getProperties();

def error_message = cus_properties.get("exception_msg");

def error_stacktrace = cus_properties.get("exception_stacktrace");

def price = cus_properties.get("price");

def title = cus_properties.get("title");

def messageLog = messageLogFactory.getMessageLog(message);

if (messageLog != null)

{

messageLog.addAttachmentAsString("Error Stacktrace",error_stacktrace,"text/plain");

}

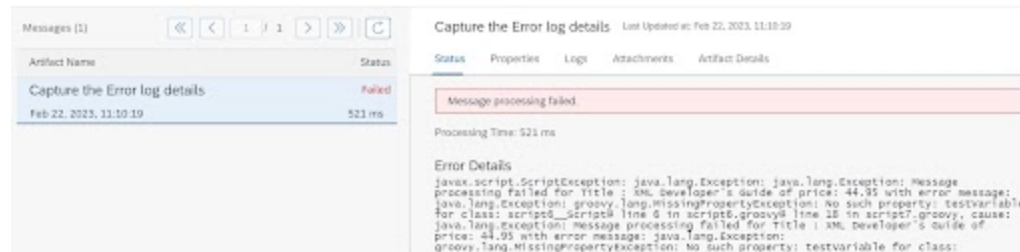
String exceptionBody = "Message processing failed for Title : "+title+" of price: "+price+" with error message:
"+error_message;

throw new Exception(exceptionBody);
```



```
return message;
```

```
}
```



Case 15:

Create an Integration Scenario which removes duplicate records Create an Integration Scenario which describes all Data Store Operations using Groovy Scrip Perform all Data Store Operations (Get, Write) using Groovy Script.

Solution:

Groovy Script for Data store Write operation:

```
import com.sap.gateway.ip.core.customdev.util.Message
```

```
//Imports for DataStoreService-class
```

```
import com.sap.it.api.asdk.datastore.*
```

```
import com.sap.it.api.asdk.runtime.*

Message processData(Message message)

{

//Data to be stored in datatore

def map = message.getHeaders();

//def idoc = map.get("COND_A04");

def payload = map.get("idoc");

//Get service instance

def service = new Factory(DataStoreService.class).getService()

//Check if valid service instance was retrieved

if( service != null)

{

def dBean = new DataBean()

dBean.setDataAsArray(payload.getBytes("UTF-8"))

//Class model offers headers, but for me it didn't work out
```

```
//Map<String, Object> headers = ["headerName1":"me", "anotherHeader": false]

//dBean.setHeaders(headers)

//Define datatore name and entry id

def dConfig = new DataConfig()

dConfig.setStoreName("Data")

dConfig.setId("1")

dConfig.setOverwrite(true)

//Write to data store

result = service.put(dBean,dConfig)

}

return message

}
```

Groovy Script for Data store Get operation:

```
import com.sap.gateway.ip.core.customdev.util.Message

//Imports for DataStoreService-class
```

```
import com.sap.it.api.asdk.datastore.*

import com.sap.it.api.asdk.runtime.*

Message processData(Message message)

{

//Get service instance

def service = new Factory(DataStoreService.class).getService()

//Check if valid service instance was retrieved

if( service != null)

{

//Read data store entry via id

def dsEntry = service.get("Data","1")

def result = new String(dsEntry.getDataAsArray())

message.setBody(result)

}

return message
```

```
}
```

Case 16:

Business Requirement:

Create an Integration scenario to extract text data only from PDF file.

Code:

```
import com.sap.gateway.ip.core.customdev.util.Message;
```

```
import java.util.HashMap;
```

```
import org.apache.pdfbox.pdmodel.PDDocument;
```

```
import org.apache.pdfbox.util.PDFTextStripper;
```

```
def Message processData(Message message)
```

```
{
```

```
    def body = message.getBody();
```

```
    InputStream IS = body;
```

```
    String res = readFromPDF(IS)
```

```
message.setBody(res);

return message;

}

public static String readFromPDF(InputStream input)

{

    PDDocument pd;

    try {

        pd = PDDocument.load(input);

        PDFTextStripper stripper = new PDFTextStripper();

        stripper.setStartPage(1); // Start Page

        String text = stripper.getText(pd);

        if (pd != null) {

            pd.close();

        }

        return text.toString()

    }

}
```

```
} catch (Exception e) {  
  
e.printStackTrace();  
  
}  
  
return null  
  
}
```

Input:

Please find the below table which we are used in the PDF file

First name	Last name	location	Date of birth
Raviteja	Satuluri	mysore	06/09/1990
samiksha	raut	mumbai	09/7/1999
Bala	tirumani	Andhra pradesh	06/04/1998
jhansi	rani	hyderabad	09/08/1999

Output:

Content extracted from PDF:

First name Last name location Date of birth

Raviteja Satuluri mysore 06/09/1990

samiksha raut mumbai 09/7/1999

Bala tirumani Andhra pradesh 06/04/1998

jhansi rani hyderabad 09/08/1999

Case 17:

Business Requirement:

Create an Integration Scenario to pick up only PDF files from the list of attachments from the Email Message.

Solution: The below groovy script will help you to filter the attachments.

Code:

```
import com.sap.gateway.ip.core.customdev.util.Message
```



```
import java.util.Map

import java.util.Iterator

import javax.activation.DataHandler

def Message processData(Message message)

{

    //message.getContent();

    //message.setBody(message.getContent());

    Map<String, DataHandler> attachments = message.getAttachments();

    if (attachments.isEmpty()) {

        throw new Exception("No Attachments !!"); // Handling of missing attachment goes here

    } else {

        //Handling of available attachments

        attachments.values().each{ attachment ->

            if(attachment.getContentType().contains("pdf")){

                message.getAttachments();

            }

        }

    }

}
```

```
}  
  
}  
  
}  
  
return message;  
  
}
```

Case 18:

Business Requirement:

Create an Integration scenario to convert JSON String UserID to Integer using groovy script.

Solution: The below groovy script will help you to convert string to Integer.

Code:

```
import com.sap.gateway.ip.core.customdev.util.Message;  
  
import java.util.HashMap;  
  
import groovy.json.*;
```

```
def Message processData(Message message) {  
  
    //Body  
  
    def body = message.getBody(String.class);  
  
    def jsonSlurper = new JsonSlurper()  
  
    def list = jsonSlurper.parseText(body)  
  
    list.EmployeeData.Record.each  
  
    {  
  
        it.PersonID.toString()  
  
        if ( it.UserID != "")  
  
        {  
  
            it.UserID=Integer.parseInt(it.get("UserID").toString());  
  
        }  
  
        else  
  
        {  
  
            it.UserID=it.UserID
```

```

    }

    it.EmploymentID.toString()

}

def jsonOP = JsonOutput.toJson(list)

message.setBody(jsonOP)

return message;

}

```

```

1 {
2   "EmployeeData": {
3     "Record": [{
4       "PersonID": "P11",
5       "UserID": "31",
6       "EmploymentID": "E221"
7     }, {
8       "PersonID": "P12",
9       "UserID": "32",
10      "EmploymentID": "E222"
11     }, {
12      "PersonID": "P13",
13      "UserID": "33",
14      "EmploymentID": "E223"
15     }
16   ]
17 }
18 }

```

```

1 import com.sap.gateway.ip.core.customdev.util.Message
2 import java.util.HashMap;
3 import groovy.json.*;
4 def Message processData(Message message) {
5   //Body
6   def body = message.getBody(String.class);
7   def jsonSlurper = new JsonSlurper()
8   def list = jsonSlurper.parseText(body)
9   list.EmployeeData.Record.each
10  {
11    it.PersonID.toString()
12    if ( it.UserID != "" )
13    {
14      it.UserID=Integer.parseInt(it.get("UserID").toString())
15    }
16    else
17    {
18      it.UserID=it.UserID
19    }
20    it.EmploymentID.toString()
21  }
22  def jsonOP = JsonOutput.toJson(list)
23  message.setBody(jsonOP)
24  return message;
25 }
26

```

```

1 {"EmployeeData":{"Record":
2 [{"PersonID":"P11","UserID":31,
3 "EmploymentID":"E221"},{"PersonID":"P12",
4 "UserID":32,"EmploymentID":"E222"},
5 {"PersonID":"P13","UserID":33,
6 "EmploymentID":"E223"}
7 ]}}

```

Case 19:**Business Requirement:**

Write a Groovy Script and execute a Return Message Processing ID (MPL ID) and use it in a Mapping Expression.

Code:

```
import com.sap.gateway.ip.core.customdev.util.Message;

import java.util.HashMap;

import com.sap.it.api.mapping.*;

def String getMessageProcessingLogID(String header, MappingContext context)

{

    String mplId = context.getHeader("SAP_MessageProcessingLogID").toString();

    return mplId;
```

```
}
```

Case 20:

Business Requirement:

Write a Groovy Script and execute to Return null for a field in Message Mapping's Mapping Expression.

Code:

```
import com.sap.it.api.mapping.*;

def String custom_Func(String arg1) {

return null;

}
```

Case 21:

Business Requirement:

Create an Integration Scenario to read a mail message, which is having an attachment from Inbox folder. Send the Mail CSV attachment only to the target system.

Code:

```
import com.sap.gateway.ip.core.customdev.util.Message

import java.util.Map

import java.util.Iterator

import javax.activation.DataHandler

def Message processData(Message message) {

    Map<String, DataHandler> attachments = message.getAttachments();

    if (attachments.isEmpty()) {

        // Handling of missing attachment goes here

    } else {

        //Handling of available attachments

        attachments.values().each{ attachment ->

            if(attachment.getContentType().contains("CSV")){

                message.setBody(attachment.getContent());

            }

        }

    }

}
```

```
}
```

Case 22:

Business Requirement:

Create an Integration Scenario to read a mail message, which is having an attachment from Inbox folder. Send the Mail XML attachment only to the target system.

Code:

```
import com.sap.gateway.ip.core.customdev.util.Message
```

```
import java.util.Map
```

```
import java.util.Iterator
```

```
import javax.activation.DataHandler
```

```
def Message processData(Message message) {
```

```
    Map<String, DataHandler> attachments = message.getAttachments();
```

```
    if (attachments.isEmpty()) {
```

```
        // Handling of missing attachment goes here
```

```
    } else {
```



```
//Handling of available attachments

attachments.values().each{ attachment ->

if(attachment.getContentType().contains("XML"))

{

message.setBody(attachment.getContent());

}

}
```

Case 23:

Business Requirement:

Create an Integration Scenario to read a mail body and populate mail body message to Target System.

Code:

```
import com.sap.gateway.ip.core.customdev.util.Message

import java.util.Map
```

```
import java.util.Iterator

import javax.activation.DataHandler

def Message processData(Message message) {

    Map<String, DataHandler> attachments = message.getAttachments();

    if (attachments.isEmpty()) {

        // Handling of missing attachment goes here

    } else {

        //Handling of available attachments

        attachments.values().each{ attachment ->

            if(attachment.getContentType().contains("text/plain"))

            {

                message.setBody(attachment.getContent());

            }

        }

    }

}
```

Case 24:**Business Requirement:**

Create an Integration Scenario to read an JSON attachment from Email Inbox.

Code:

```
import com.sap.gateway.ip.core.customdev.util.Message

import java.util.Map

import java.util.Iterator

import javax.activation.DataHandler

def Message processData(Message message) {

    Map<String, DataHandler> attachments = message.getAttachments();

    if (attachments.isEmpty()) {

        // Handling of missing attachment goes here

    } else {

        //Handling of available attachments

        attachments.values().each{ attachment ->
```

```
    if(attachment.getContentType().contains("json")){  
  
        message.setBody(attachment.getContent());  
  
    }  
  
}  
  
}  
  
return message;  
  
}
```

Case 25:

Business Requirement:

Create an Integration Scenario to remove duplicate records.

Input:

```
<?xml version="1.0" encoding="UTF-8"?>
<header>
<Record>
<id>"bk101"</id>
<author>Gambardella, Matthew</author>
<title>XML Developer's Guide</title>
<genre>Computer</genre>
<price>44.95</price>
<publish_date>2000-10-01</publish_date>
<description>An in-depth look at creating applications
with XML.</description>
</Record>
<Record>
<id>"bk101"</id>
<author>Gambardella, Matthew</author>
<title>XML Developer's Guide</title>
<genre>Computer</genre>
<price>44.95</price>
<publish_date>2000-10-01</publish_date>
<description>An in-depth look at creating applications
with XML.</description>
</Record>
</header>
```

XSLT Code:

```
<xsl:stylesheet version="2.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

```
<xsl:output method="xml" indent="yes"/>
<xsl:template match="node()|@*">
  <xsl:copy>
    <xsl:apply-templates select="node()|@*" />
  </xsl:copy>
</xsl:template>
<xsl:template match="header">
  <header>
    <xsl:for-each-group select="Record"
      group-by="id">
      <xsl:apply-templates select="."/>
    </xsl:for-each-group>
  </header>
</xsl:template>
</xsl:stylesheet>
```

Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<header>
  <Record>
    <id>"bk101"</id>
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish_date>
    <description>An in-depth look at creating applications
with XML.</description>
  </Record>
```

</header>

Case 26:

Business Requirement:

Fetch the data which is in the Body section for SOAP Envelope.

Input:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <Action xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none"
      s:mustUnderstand="1">http://sap.com/xi/WebService/soap1.1</Action>
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <_BAPI_PO_GETDETAILS xmlns="urn:sap-com:document:sap:rfc:functions">
      <PO xmlns=""/>
    </_BAPI_PO_GETDETAILS>
  </s:Body>
</s:Envelope>
```

Expected Output:

```
<_BAPI_PO_GETDETAILS xmlns="urn:sap-com:document:sap:rfc:functions">
  <PO xmlns=""/>
</_BAPI_PO_GETDETAILS>
```

Code:

```
import com.sap.gateway.ip.core.customdev.util.Message;

import java.util.HashMap;

def Message processData(Message message)

{

    def removal=message.getBody(java.lang.String) as String;

    removal=removal.replace("http://sap.com/xi/WebService/soap1.1","");

    removal=removal.replace("xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"","");

    removal=removal.replace("xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none"","");

    removal=removal.replace("xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"","");

    removal=removal.replace("xmlns:xsd="http://www.w3.org/2001/XMLSchema"","");

    removal=removal.replace("s:mustUnderstand="1"","");

    removal=removal.replace("<s:Envelope >","");

    removal=removal.replace("<s:Header>","");
```



```
removal=removal.replace("</s:Header>", "");  
  
removal=removal.replace("<s:Body >", "");  
  
removal=removal.replace("<Action ></Action>", "");  
  
removal=removal.replace("</s:Body>", "");  
  
removal=removal.replace("</s:Envelope>", "");  
  
message.setBody(removal);  
  
return message;  
  
}
```

Script

[Copy](#) [Share URL](#)

```
1 import com.sap.gateway.ip.core.customdev.util.Message;
2 import java.util.HashMap;
3 def Message processData(Message message)
4 {
5     def removal=message.getBody(java.lang.String) as String;
6     removal=removal.replace("http://sap.com/xi/WebService/soap1.1","");
7     removal=removal.replace("xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"", "");
8     removal=removal.replace("xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none""", "");
9     removal=removal.replace("xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance""", "");
10    removal=removal.replace("xmlns:xsd="http://www.w3.org/2001/XMLSchema""", "");
11    removal=removal.replace("s:mustUnderstand="1""", "");
12    removal=removal.replace("<s:Envelope >", "");
13    removal=removal.replace("<s:Header>", "");
14    removal=removal.replace("</s:Header>", "");
15    removal=removal.replace("<s:Body >", "");
16    removal=removal.replace("<Action ></Action>", "");
17    removal=removal.replace("</s:Body>", "");
18    removal=removal.replace("</s:Envelope>", "");
19    message.setBody(removal);
20    return message;
21 }
```

Output

```
1
2
3
4
5
6 <_BAPI_PO_GETDETAILS xmlns="urn:sap-com:document:sap:rfc:functions">
7 <PO xmlns=""/>
8 </_BAPI_PO_GETDETAILS>
9
10
```

Headers

empty

Conclusion –

Hope this document will help to beginners to understand CPI concept of Groovy Script. Happy Learning 😊

Follow Like RSS Feed

6 Comments



pieterjanssens

Active Participant



2023 Feb 09 6:18 PM



2 Kudos

Please use "insert code sample" to format your code and make it more easy to read through.

**revanth55**

Newcomer



2023 Feb 16 8:54 AM

**0 Kudos**

Hi Raviteja ,

Please domenstrate with Cpi example with the groovy code /Please domenstrate with a groovy ide example.

So that we can undertstand easily .

Thank you

**raviteja_satuluri**

Contributor



2023 Feb 17 12:16 PM

**1 Kudo**

Hi Revanth,

We can't demonstrate all the scenarios using Groovy IDE. You can refer my other blogs where I have tested and demonstrated, documented in blog.

Thank You !!



meerhyderali

Explorer



2023 Feb 20 6:44 PM



Hi Raviteja,

I tried to extract the Body data from the XML payload, in both ways, but couldn't succeed. Hence I am writing this blog to make the script/Content Modifier based on the requirement.

Requirement

I need to fetch the data which is in the Body, I declared the **xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"** in main and tried to fetch the data using content modifier. But the body is coming as blank after the Content Modifier step execution by giving the path as **//s:Envelope/s:Body/**.

I tried using the Groovy script using the removal statement by below code using your method but its failing.

```
import com.sap.gateway.ip.core.customdev.util.Message;

import java.util.HashMap;

def Message processData(Message message) {

def removal=message.getBody(java.lang.String) as String;

removal=removal.replace(/s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" /, "");

message.setBody(removal);

return message;

}
```

Payload.

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
<s:Header>
<Action xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none"
s:mustUnderstand="1">http://sap.com/xi/WebService/soap1.1</Action>
</s:Header>
<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<_BAPI_PO_GETDETAILS xmlns="urn:sap-com:document:sap:rfc:functions">
<PO xmlns="" />
</_BAPI_PO_GETDETAILS>
</s:Body>
```

</s:Envelope>

Can you all please help me on this using the content modifier or Groovy Script where i am wrong. I have highlighted in bold which i required in payload.

Thank you in advance.

Regards,

Meer Hyder Ali



raviteja_satuluri
Contributor



2023 Feb 21 5:42 PM



0 Kudos

Hi Hyder Ali,

Please find below code:

```
import com.sap.gateway.ip.core.customdev.util.Message;

import java.util.HashMap;

def Message processData(Message message)

{

    def removal=message.getBody(java.lang.String) as String;

    removal=removal.replace("http://sap.com/xi/WebService/soap1.1","");

    removal=removal.replace("xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"","");

    removal=removal.replace("xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none"","");

    removal=removal.replace("xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"","");

    removal=removal.replace("xmlns:xsd="http://www.w3.org/2001/XMLSchema"","");

    removal=removal.replace("s:mustUnderstand="1"","");
```



```
removal=removal.replace("<s:Envelope >", "");  
  
removal=removal.replace("<s:Header>", "");  
  
removal=removal.replace("</s:Header>", "");  
  
removal=removal.replace("<s:Body >", "");  
  
removal=removal.replace("<Action ></Action>", "");  
  
removal=removal.replace("</s:Body>", "");  
  
removal=removal.replace("</s:Envelope>", "");  
  
message.setBody(removal);  
  
return message;  
  
}
```

Tested in Groovy IDE:

Script

[Copy Share URL](#)

```

1 import com.sap.gateway.ip.core.customdev.util.Message;
2 import java.util.HashMap;
3 def Message processData(Message message)
4 {
5     def removal=message.getBody(java.lang.String) as String;
6     removal=removal.replace("http://sap.com/xi/WebService/soap1.1","");
7     removal=removal.replace("xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"","");
8     removal=removal.replace("xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none"","");
9     removal=removal.replace("xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"","");
10    removal=removal.replace("xmlns:xsd="http://www.w3.org/2001/XMLSchema"","");
11    removal=removal.replace("s:mustUnderstand="1"","");
12    removal=removal.replace("<s:Envelope >","");
13    removal=removal.replace("<s:Header>","");
14    removal=removal.replace("</s:Header>","");
15    removal=removal.replace("<s:Body >","");
16    removal=removal.replace("<Action ></Action>","");
17    removal=removal.replace("</s:Body>","");
18    removal=removal.replace("</s:Envelope>","");
19    message.setBody(removal);
20    return message;
21 }

```

Output

```

1
2
3
4
5
6 <_BAPI_PO_GETDETAILS xmlns="urn:sap-com:document:sap:rfc:functions">
7 <PO xmlns=""/>
8 </_BAPI_PO_GETDETAILS>
9
10

```

Headers

empty

**David_Davis**

Participant



2023 Sep 02 7:58 AM



0 Kudos

Good blog. Do we need to use exception subprocess for the case 2?

You must be a registered user to add a comment. If you've already registered, sign in. Otherwise, register and sign in.

Comment**Labels In This Area**

["as_written_by_Marian_Zeis" 1](#)
["automatische backups" 1](#)
["Best Practices" 1](#)
["Data Source Migration" 1](#)

["Integration Challenges" 2](#)
["regelmäßige sicherung" 1](#)
["SAP BW 7.4" 1](#)
["SAP BW" 5](#)

["SAP VARIANT CONFIGURAITION 2](#)
["SAP_BUILD_APPS" 1](#)
["SAPDatasphe" 3](#)

["TypeScript" "Development" "FeedBack" 1](#)
[*SAP" 1](#)
[-147 Get CurrentUserInfo failed 1](#)
[2YM 1](#)
[3-TIER Extensibility 3](#)

[30 examples 1](#)
[505 Technology Updates 53 1](#)
[@expertsap 1](#)
[@hanasizing 1](#)
[@RetroDate_HireDateCorrection 1](#)

[@sapilm @archiving @sapiq 1](#)
[@SAPSupport 1](#)
[@SCPI 2](#)
[A Comprehensive Guide to Using OLE Objects in SAP ABAP 1](#)

[aATP 1](#)
[ABAP 44](#)
[ABAP 7.4 2](#)
[ABAP API 1](#)
[ABAP BAPI BAPI_FIXEDASSET_CREATE1 1](#)
[ABAP BTP 1](#)

[ABAP CDS VIEW 2](#)
[ABAP CDS Views 13](#)
[ABAP CDS Views - BW Extraction 3](#)

[ABAP CDS Views - CDC \(Change Data Capture\) 3](#)
[ABAP Class 3](#)
[ABAP Cloud 9](#)
[ABAP Cloud Developer Trial 1](#)

[ABAP Custom 1](#)
[ABAP DDIC CDS view 1](#)
[ABAP development 14](#)
[ABAP Editor 1](#)
[ABAP Environment & RAP 2](#)

[ABAP Extensibility 4](#)
[ABAP for EWM 1](#)
[ABAP in Eclipse 3](#)
[ABAP Interface 1](#)
[ABAP New Syntax 1](#)
[ABAP ODATA 2](#)

[ABAP on HANA 1](#)
[ABAP OOABAP 1](#)
[ABAP PLATFORM 1](#)
[ABAP Platform Trial 2](#)
[ABAP Programming 8](#)

[ABAP Push Channels 1](#)
[ABAP Query 1](#)
[ABAP RAP 4](#)
[ABAP RAP custom action 3](#)

[ABAP RAP\(RESTful Application Programming\) 6](#)
[ABAP RESTFul API 1](#)
[ABAP RESTful Application Programming Model 2](#)

[ABAP String functions 1](#)
[abap technical 1](#)
[ABAP test cockpit 1](#)
[abap to xml 1](#)
[abapGit 1](#)
[absl 2](#)

[Access data from datasphere to ADF Azure Data Factory 5](#)
[access data from SAP Datasphe directly from Snowflake 1](#)

[Access data from SAP datasphere to QlikSense 2](#)
[Accessibility 1](#)
[Accessibility in SAPUI5 1](#)
[Accrual 1](#)

[Acquire SAC Knowledge 3](#)
[acquired 1](#)
[action 1](#)
[actions 1](#)
[Activity 1](#)
[Adaptation Project 1](#)
[adapter 2](#)

adapter modules 2

ADDING LEAN SERVICES 2

Addon 2

Adobe Document Services 1

Adobe forms 1

ADS 1

ADS Config 1

ADS with ABAP 1

ADS with Java 1

ADT 4

Advance Shipping and Receiving 1

Advanced ABAP 1

Advanced Event Mesh 4

Advanced formula 1

Advanced Formulas 2

Advanced Metric 1

Advanced SAP Techniques 1

Advanced Scripting in SAC 3

Advanced Workflow 2

AEM 1

Related Content

SAP's Pioneering Leap in Artificial Intelligence: Transforming Enterprise Processes with Business AI

in Technology Blogs by SAP yesterday

SAP Integrated Business Planning (IBP): A Machine Learning-Driven Framework for Supply Chain

in Technology Blogs by Members yesterday

Part 1: SAP Revenue Accounting and Reporting (RAR): Foundation for Modern Revenue Management

in Technology Blogs by SAP yesterday

Part 2: Revenue Accounting and Reporting (RAR) with GenAI and RAG

in Technology Blogs by SAP yesterday

Integration Suite: What is the advantage of using B2B TPM?

in Technology Q&A Sunday

Popular Blog Posts



SAP PI for Beginners



former_member200339

Participant

👁 728599 💬 155 👍 388



ABAP 7.40 Quick Reference



jeffrey_towell2

Participant

👁 1222368 💬 75 👍 345



Difference between SAP S/4HANA :Public Vs Private edition : RISE with SAP



rajarajeswari_kaliyaperum

Active Participant

👁 189941 💬 47 👍 310

Top Kudoed Authors



Phil_from_Madrid

👍 10



hsonnenahalli

👍 9



dallasmrks

👍 6



saroopreddy88

👍 5



christian_willi

👍 4



LucasMagriniRigo

👍 3



Jayabalaji_Sakthivel

👍 3



mickaelquesnot

👍 3



Jinal_Patel

👍 3



junwu

👍 3

View all

Follow



Privacy

Terms of Use

Copyright

Legal Disclosure

Trademark

Support

Cookie Preferences