**Reading PDF Files by using Class mediator**

# Creating an Integration Project.

**Step 1:** Create a new Integration Project (For creating the new Integration Project Please refer the Basic Document to create Project. ).

# Adding Mediation Project.

**Step 2:** Right click on the project, click on new and then click on Mediation Project.

Graphical user interface, application

Description automatically generated

**Note :** The PDF file reading is only done by using Enterprise Integrator 6.6.0.

**Step 3:** Select Create New Mediator and then press next.Graphical user interface, text, application

Description automatically generated

**Step 4:** Give the Project Name, Package Name, Class Name and then Click on Finish.Graphical user interface, application, email

Description automatically generated

# Implementing Java

**Step 5:** Implement the Java Class and set a property with a value that has the base64 Encoded String.

Graphical user interface, text, application

Description automatically generated

# Exporting to JAR

**Step 6:** Right click on the Mediation Project and click on export.

Graphical user interface, text, application, email

Description automatically generated

**Step 7:** Click on JAR file and click Next.

Graphical user interface, application

Description automatically generated

**Step 8:** Select all the Classes that you are going to include in your JAR and give a destination folder and Click on Finish.Graphical user interface, text, application, email

Description automatically generated

# Adding JAR in Lib

**Step 9:** Add the Jar File in the <EI\_HOME>/repository/Lib folder.

Table

Description automatically generated

# Configuring axis2.xml

**Step 10:** Open the axis2.xml file of WSO2 Enterprise Integrator.

(Ex: In my system Location = “C:\Program Files\WSO2\Enterprise Integrator\6.6.0\conf\axis2”).

**Step 11:** Configure the Message Builders and Message Formatters in the File axis2.xml.

Add the following tag in the messageFormatters.

<messageFormatter contentType="application/pdf" class="org.wso2.carbon.relay.ExpandingMessageFormatter"/>

![Graphical user interface, text, application

Description automatically generated]()

Add the following tag in the messageBuilders.

<messageBuilder contentType="application/pdf" class="org.wso2.carbon.relay.BinaryRelayBuilder"/>

![Graphical user interface, text, application, email

Description automatically generated]()

# Creating API in Integration Studio

**Step 12:** Let us go to the Integration Studio and create an API. (For creating the APIs please refer to the Basic documents).

**Step 13:**Drag and drop the Log Mediator and just write the Hello Msg.

**Step 14:** Drag and drop the Property Mediatorgive a name and give the value as an expression so that we can give the file path in request body.

Graphical user interface, application

Description automatically generated

**Step 15:** Drag and drop the Class Mediator give the Class Name.

**Graphical user interface, application

Description automatically generated**

**Step 16:** Drag and drop the Payload Mediator, add the tag in the payload as

<ns:binary xmlns:ns="http://ws.apache.org/commons/ns/payload">$1</ns:binary>

**Step 17:** Click on the arguments and select expression the give the expression value.

**Graphical user interface, application

Description automatically generated**

**Step 18:** Drag and drop the Script Mediator, select Script Language as “js” and Script Type = Inline and add the Script Body as

**var** binaryNode = mc.getEnvelope().getBody().getFirstElement().getFirstOMChild();binaryNode.setBinary(**true) .**

Graphical user interface, text, application, email

Description automatically generated

**Step 19:** Add a property mediator and select the property name as messageType and give the value as application/pdf and add a respond mediator.Graphical user interface, application

Description automatically generated

**Step 20:** Save the File add the API artifact in the CAR file deploy on Enterprise Integrator server and Start the Server.

# Sample Source Code

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<api context=*"/pdfReadingUsingJAVA\_API"* name=*"PDFReadingUsingJAVA\_API"* xmlns=*"http://ws.apache.org/ns/synapse"*>

<resource methods=*"POST"*>

<inSequence>

<log description=*"Welcome"* level=*"custom"*>

<property name=*"API Hit"* value=*">>>Hello"*/>

</log>

<property description=*"File"* expression=*"json-eval($.File)"* name=*"File"* scope=*"default"* type=*"STRING"*/>

<class name=*"com.pdf.PDFtoBase64"*/>

<!-- <property description="Base 64" expression="get-property('base64String')" name="Base64" scope="default" type="STRING"/> -->

<payloadFactory media-type=*"xml"*>

<format>

<ns:binary xmlns:ns=*"http://ws.apache.org/commons/ns/payload"*>$1</ns:binary>

</format>

<args>

<arg evaluator=*"xml"* expression=*"get-property('base64String')"*/>

</args>

</payloadFactory>

<script language=*"js"*><![CDATA[var binaryNode = mc.getEnvelope().getBody().getFirstElement().getFirstOMChild();

binaryNode.setBinary(true);]]></script>

<property name=*"messageType"* scope=*"axis2"* type=*"STRING"* value=*"application/pdf"*/>

<respond/>

</inSequence>

<outSequence/>

<faultSequence/>

</resource>

</api>