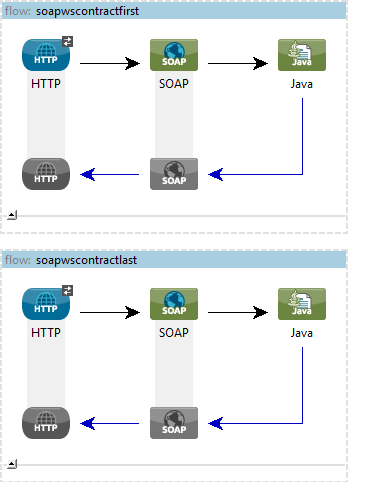
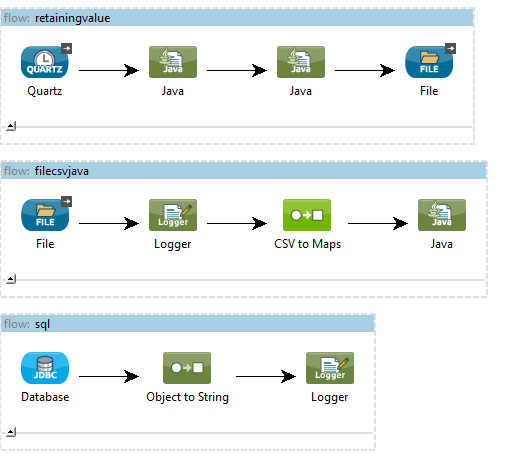
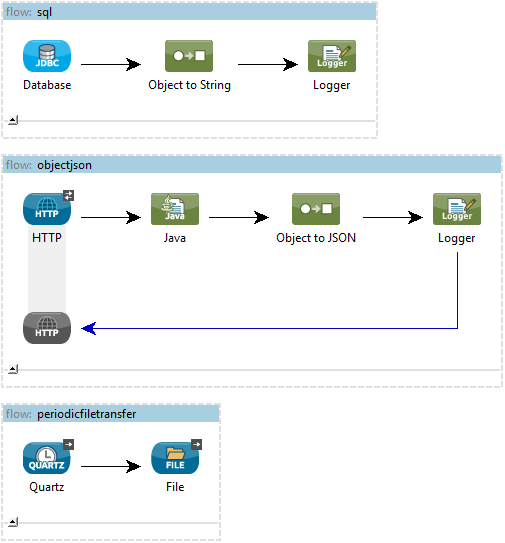
**Mule Demos :: Nivi Kundem 07102014**

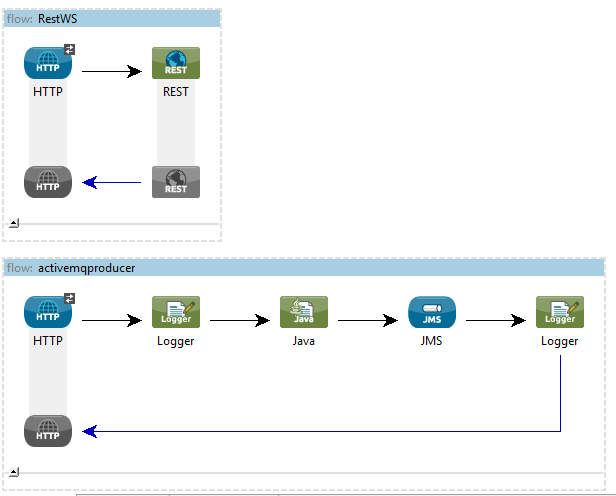
Demo for using :

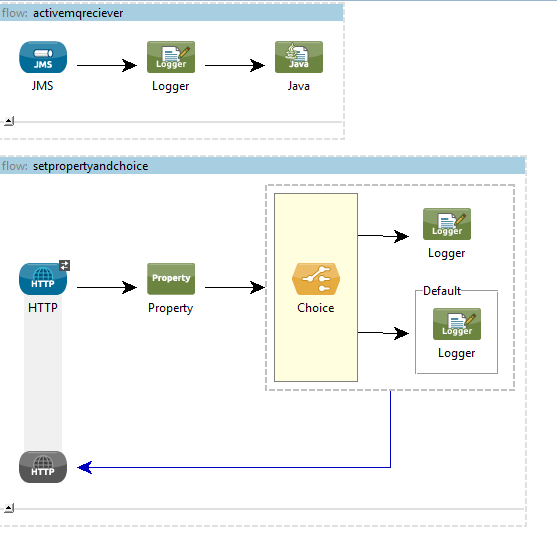
SOAP WS contract first, SOAP WS contract last, Quartz, Java, File, CSV to Maps, JDBC, Object to String, HTTP, Object to JSON, Logger, REST WS, JMS Producer, JMS Consumer, Set Property, Choice, Filter Reference, For each, variable, Expression, Cache, FTP, Datamapper

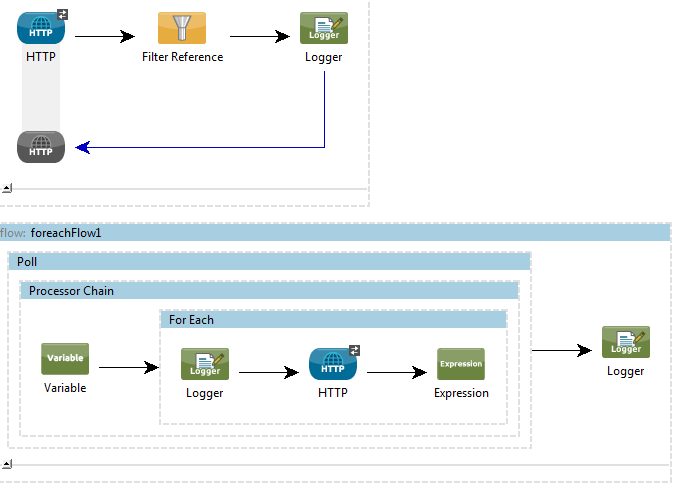


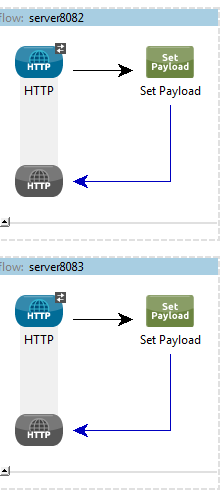


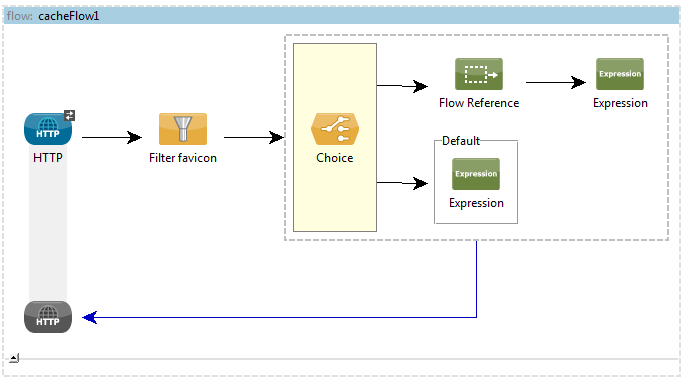


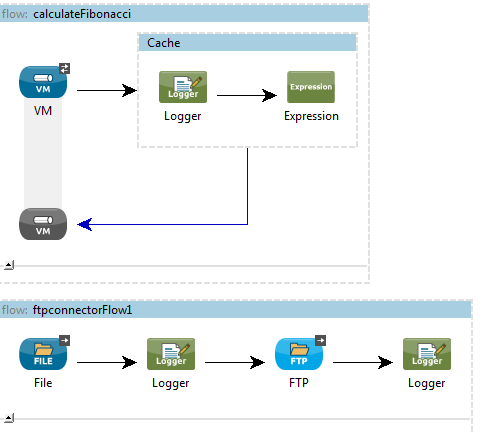


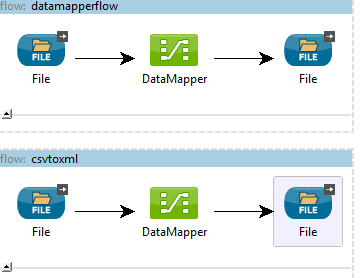












**Flow ::**

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

<mule xmlns:data-mapper=*"http://www.mulesoft.org/schema/mule/ee/data-mapper"* xmlns:ftp=*"http://www.mulesoft.org/schema/mule/ee/ftp"*

xmlns:jersey=*"http://www.mulesoft.org/schema/mule/jersey"*

xmlns:json=*"http://www.mulesoft.org/schema/mule/json"* xmlns:quartz=*"http://www.mulesoft.org/schema/mule/quartz"*

xmlns:http=*"http://www.mulesoft.org/schema/mule/http"*

xmlns:cxf=*"http://www.mulesoft.org/schema/mule/cxf"*

xmlns:jdbc-ee=*"http://www.mulesoft.org/schema/mule/ee/jdbc"* xmlns:file=*"http://www.mulesoft.org/schema/mule/file"* xmlns:ee=*"http://www.mulesoft.org/schema/mule/ee/core"* xmlns:tracking=*"http://www.mulesoft.org/schema/mule/ee/tracking"* xmlns:vm=*"http://www.mulesoft.org/schema/mule/vm"* xmlns:jms=*"http://www.mulesoft.org/schema/mule/jms"* xmlns=*"http://www.mulesoft.org/schema/mule/core"* xmlns:doc=*"http://www.mulesoft.org/schema/mule/documentation"* xmlns:spring=*"http://www.springframework.org/schema/beans"* version=*"EE-3.4.1"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://www.mulesoft.org/schema/mule/ee/ftp http://www.mulesoft.org/schema/mule/ee/ftp/current/mule-ftp-ee.xsd*

*http://www.mulesoft.org/schema/mule/jersey http://www.mulesoft.org/schema/mule/jersey/current/mule-jersey.xsd*

*http://www.mulesoft.org/schema/mule/quartz http://www.mulesoft.org/schema/mule/quartz/current/mule-quartz.xsd*

*http://www.mulesoft.org/schema/mule/http http://www.mulesoft.org/schema/mule/http/current/mule-http.xsd*

*http://www.mulesoft.org/schema/mule/cxf http://www.mulesoft.org/schema/mule/cxf/current/mule-cxf.xsd*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-current.xsd*

*http://www.mulesoft.org/schema/mule/core http://www.mulesoft.org/schema/mule/core/current/mule.xsd*

*http://www.mulesoft.org/schema/mule/vm http://www.mulesoft.org/schema/mule/vm/current/mule-vm.xsd*

*http://www.mulesoft.org/schema/mule/ee/core http://www.mulesoft.org/schema/mule/ee/core/current/mule-ee.xsd*

*http://www.mulesoft.org/schema/mule/file http://www.mulesoft.org/schema/mule/file/current/mule-file.xsd*

*http://www.mulesoft.org/schema/mule/ee/jdbc http://www.mulesoft.org/schema/mule/ee/jdbc/current/mule-jdbc-ee.xsd*

*http://www.mulesoft.org/schema/mule/jms http://www.mulesoft.org/schema/mule/jms/current/mule-jms.xsd*

*http://www.mulesoft.org/schema/mule/ee/tracking http://www.mulesoft.org/schema/mule/ee/tracking/current/mule-tracking-ee.xsd*

*http://www.mulesoft.org/schema/mule/json http://www.mulesoft.org/schema/mule/json/current/mule-json.xsd*

*http://www.mulesoft.org/schema/mule/ee/data-mapper http://www.mulesoft.org/schema/mule/ee/data-mapper/current/mule-data-mapper.xsd"*>

<spring:beans>

<spring:bean id=*"responseGenerator"* name=*"Bean"* class=*"com.mulesoft.mule.cache.FibonacciResponseGenerator"*/>

</spring:beans>

<jdbc-ee:mysql-data-source name=*"MySQL\_Data\_Source"* user=*"root"* password=*"root"* url=*"jdbc:mysql://localhost:3306/qe2\_bridge"* transactionIsolation=*"UNSPECIFIED"* doc:name=*"MySQL Data Source"*/>

<file:endpoint path=*"C:\JobSearch\MuleDemos\File\Outbound"* name=*"FileConnector"* responseTimeout=*"10000"* doc:name=*"File"*/>

<jms:activemq-connector name=*"Active\_MQ"* username=*"admin"* password=*"admin"* brokerURL=*"tcp://localhost:61616"* validateConnections=*"true"* doc:name=*"Active MQ"*/>

<expression-filter expression=*"#[payload=='/nivi']"* name=*"Expression"* doc:name=*"Expression"*/>

<ee:object-store-caching-strategy name=*"Caching\_Strategy"* keyGenerationExpression=*"#[message.inboundProperties['n']]"* responseGenerator-ref=*"responseGenerator"* doc:name=*"Caching Strategy"*/>

<vm:connector name=*"vmConnector"* validateConnections=*"true"* doc:name=*"VM"*/>

<configuration doc:name=*"Configuration"*>

<expression-language>

<global-functions>

def fibonacciRequest(n, cached)

{

import org.mule.DefaultMuleMessage;

import org.mule.RequestContext;

request = new DefaultMuleMessage("Fibonacci: " + n, app.registry['\_muleContext']);

request.setOutboundProperty("n", Integer.toString(n));

if (!cached)

{

request.setOutboundProperty("nocache", true);

}

RequestContext.getEventContext().sendEvent(request, "vm://fibonacci");

}

</global-functions>

</expression-language>

</configuration>

<jdbc-ee:connector name=*"Database"* dataSource-ref=*"MySQL\_Data\_Source"* validateConnections=*"true"* queryTimeout=*"-1"* pollingFrequency=*"0"* doc:name=*"Database"*/>

<data-mapper:config name=*"new\_mapping\_grf"* transformationGraphPath=*"new\_mapping.grf"* doc:name=*"DataMapper"*/>

<data-mapper:config name=*"new\_mapping\_1\_grf"* transformationGraphPath=*"new\_mapping\_1.grf"* doc:name=*"new\_mapping\_1\_grf"*/>

<!-- http://localhost:8081/BridgeCrossingWS?wsdl -->

<flow name=*"soapwscontractfirst"* doc:name=*"soapwscontractfirst"*>

<http:inbound-endpoint exchange-pattern=*"request-response"* host=*"localhost"* port=*"8081"* path=*"BridgeCrossingWS"* doc:name=*"HTTP"*/>

<cxf:jaxws-service serviceClass=*"com.crossing.www.BridgeCrossingWS.BridgeCrossingWS"* doc:name=*"SOAP"*/>

<component class=*"com.crossing.www.BridgeCrossingWS.BridgeCrossingWSImpl"* doc:name=*"Java"*/>

</flow>

<!-- http://localhost:8082/cxfwebservice?wsdl -->

<flow name=*"soapwscontractlast"* doc:name=*"soapwscontractlast"*>

<http:inbound-endpoint exchange-pattern=*"request-response"* host=*"localhost"* port=*"8082"* path=*"cxfwebservice"* doc:name=*"HTTP"*/>

<cxf:jaxws-service serviceClass=*"com.nivi.org.CXFWebservice"* doc:name=*"SOAP"*/>

<component class=*"com.nivi.org.CXFWebserviceImpl"* doc:name=*"Java"*/>

</flow>

<flow name=*"retainingvalue"* doc:name=*"retainingvalue"*>

<quartz:inbound-endpoint jobName=*"RetailValueQuartz"* repeatInterval=*"5000"* responseTimeout=*"10000"* doc:name=*"Quartz"*>

<quartz:event-generator-job/>

</quartz:inbound-endpoint>

<component class=*"com.nivi.events.EventCalculator"* doc:name=*"Java"*/>

<component class=*"com.datatofile.org.CurrentTime"* doc:name=*"Java"*/>

<file:outbound-endpoint path=*"C:\JobSearch\MuleDemos\File\output"* responseTimeout=*"10000"* doc:name=*"File"*/>

</flow>

<flow name=*"filecsvjava"* doc:name=*"filecsvjava"*>

<file:inbound-endpoint path=*"C:\JobSearch\MuleDemos\File\Inbound"* moveToDirectory=*"C:\JobSearch\MuleDemos\File\Outbound"* responseTimeout=*"10000"* doc:name=*"File"*/>

<logger message=*"#[string: output #[message]]"* level=*"INFO"* doc:name=*"Logger"*/>

<jdbc-ee:csv-to-maps-transformer delimiter=*","* mappingFile=*"person.xml"* doc:name=*"CSV to Maps"*/>

<component class=*"com.filecsv.org.CSVProcessor"* doc:name=*"Java"*/>

</flow>

<flow name=*"sql"* doc:name=*"sql"*>

<jdbc-ee:inbound-endpoint queryKey=*"VRN\_NUMBER"* queryTimeout=*"-1"* pollingFrequency=*"1000"* connector-ref=*"Database"* doc:name=*"Database"*>

<jdbc-ee:query key=*"VRN\_NUMBER"* value=*"SELECT VRN FROM qe2\_bridge.vehicle\_details;; "*/>

</jdbc-ee:inbound-endpoint>

<object-to-string-transformer doc:name=*"Object to String"*/>

<logger message=*"Payload from db ::: #[payload]"* level=*"INFO"* doc:name=*"Logger"*/>

</flow>

<!-- http://localhost:8083/ -->

<flow name=*"objectjson"* doc:name=*"objectjson"*>

<http:inbound-endpoint exchange-pattern=*"request-response"* host=*"localhost"* port=*"8083"* doc:name=*"HTTP"*/>

<component class=*"com.json.org.MyJavaObject"* doc:name=*"Java"*/>

<json:object-to-json-transformer doc:name=*"Object to JSON"*/>

<logger message=*"#[payload]"* level=*"INFO"* doc:name=*"Logger"*/>

</flow>

<flow name=*"periodicfiletransfer"* doc:name=*"periodicfiletransfer"*>

<quartz:inbound-endpoint jobName=*"myServiceJob"* repeatInterval=*"5000"* responseTimeout=*"10000"* doc:name=*"Quartz"*>

<quartz:endpoint-polling-job>

<quartz:job-endpoint ref=*"FileConnector"*/>

</quartz:endpoint-polling-job>

</quartz:inbound-endpoint>

<file:outbound-endpoint path=*"C:\JobSearch\MuleDemos\File\Out"* outputPattern=*"#[message.inboundProperties.originalFilename]"* responseTimeout=*"10000"* doc:name=*"File"*/>

</flow>

<!-- localhost:8086/rest/schools/1 -->

<flow name=*"RestWS"* doc:name=*"RestWS"*>

<http:inbound-endpoint exchange-pattern=*"request-response"* host=*"localhost"* port=*"8086"* path=*"rest"* doc:name=*"HTTP"*/>

<jersey:resources doc:name=*"REST"*>

<component class=*"com.nivi.integration.rest.RestImpl2"*/>

</jersey:resources>

</flow>

<!-- localhost:8087 -->

<!-- http://localhost:8161/admin/queues.jsp -->

<flow name=*"activemqproducer"* doc:name=*"activemqproducer"*>

<http:inbound-endpoint exchange-pattern=*"request-response"* host=*"localhost"* port=*"8087"* doc:name=*"HTTP"*/>

<logger message=*"Nivi :: #[payload]"* level=*"INFO"* doc:name=*"Logger"*/>

<component class=*"com.activemq.org.MessageConstructor"* doc:name=*"Java"*/>

<jms:outbound-endpoint queue=*"CROSSING.HO.PERFORMANCE"* connector-ref=*"Active\_MQ"* doc:name=*"JMS"*/>

<logger message=*"Nivi Kundem"* level=*"INFO"* doc:name=*"Logger"*/>

</flow>

<flow name=*"activemqreciever"* doc:name=*"activemqreciever"*>

<jms:inbound-endpoint queue=*"CROSSING.HO.PERFORMANCE"* connector-ref=*"Active\_MQ"* doc:name=*"JMS"*/>

<logger level=*"INFO"* doc:name=*"Logger"*/>

<component class=*"com.activemq.org.MessageReceiver"* doc:name=*"Java"*/>

</flow>

<!-- localhost:8089/a -->

<!-- localhost:8089/nivi -->

<flow name=*"setpropertyandchoice"* doc:name=*"setpropertyandchoice"*>

<http:inbound-endpoint exchange-pattern=*"request-response"* host=*"localhost"* port=*"8089"* doc:name=*"HTTP"*/>

<set-property propertyName=*"trial"* value=*"#[payload]"* doc:name=*"Property"*/>

<choice doc:name=*"Choice"*>

<when expression=*"#[message.outboundProperties['trial'] == '/a']"*>

<logger level=*"INFO"* doc:name=*"Logger"* message=*"Success"*/>

</when>

<otherwise>

<logger level=*"INFO"* doc:name=*"Logger"* message=*"error"*/>

</otherwise>

</choice>

</flow>

<!-- localhost:8090/nivi -->

<!-- localhost:8090/a -->

<flow name=*"FilterRef"* doc:name=*"FilterRef"*>

<http:inbound-endpoint exchange-pattern=*"request-response"* host=*"localhost"* port=*"8090"* doc:name=*"HTTP"*/>

<filter ref=*"Expression"* doc:name=*"Filter Reference"*/>

<logger message=*"conditionmet"* level=*"INFO"* doc:name=*"Logger"*/>

</flow>

<flow name=*"foreachFlow1"* doc:name=*"foreachFlow1"*>

<poll frequency=*"1000"* doc:name=*"Poll"*>

<processor-chain doc:name=*"Processor Chain"*>

<set-variable variableName=*"httpResponses"* value=*"#[[]]"* doc:name=*"Variable"*/>

<foreach collection=*"#[groovy:['localhost:8000', 'localhost:8001']]"* doc:name=*"For Each"*>

<logger level=*"INFO"* message=*"each payload :: #[payload]"* doc:name=*"Logger"*/>

<http:outbound-endpoint

exchange-pattern=*"request-response"* address=*"http://#[payload]"* method=*"GET"* doc:name=*"HTTP"*/>

<expression-component doc:name=*"Expression"*>httpResponses.add(message.payloadAs(java.lang.String)) </expression-component>

</foreach>

</processor-chain>

</poll>

<logger level=*"INFO"* message=*"Responses from :: #[httpResponses]"* doc:name=*"Logger"*/>

</flow>

<!-- Test server stubs -->

<flow name=*"server8082"* doc:name=*"server8082"*>

<http:inbound-endpoint exchange-pattern=*"request-response"* address=*"http://localhost:8000"* doc:name=*"HTTP"*/>

<set-payload value=*"This is 8000"* doc:name=*"Set Payload"*/>

</flow>

<flow name=*"server8083"* doc:name=*"server8083"*>

<http:inbound-endpoint exchange-pattern=*"request-response"* address=*"http://localhost:8001"* doc:name=*"HTTP"*/>

<set-payload value=*"This is 8001"* doc:name=*"Set Payload"*/>

</flow>

<!-- http://localhost:8084/fibonacci?n=10 -->

<flow name=*"cacheFlow1"* doc:name=*"cacheFlow1"*>

<http:inbound-endpoint exchange-pattern=*"request-response"* host=*"localhost"* port=*"8084"* path=*"fibonacci"* doc:name=*"HTTP"*/>

<message-filter doc:name=*"Filter favicon"*>

<not-filter>

<wildcard-filter pattern=*"/favicon.ico"* caseSensitive=*"true"*/>

</not-filter>

</message-filter>

<choice doc:name=*"Choice"*>

<when expression=*"message.inboundProperties['n'] &lt; 20"*>

<flow-ref name=*"calculateFibonacci"* doc:name=*"Flow Reference"*/>

<expression-component doc:name=*"Expression"*>payload= "Fibonacci(" + message.inboundProperties['n'] + ") = " + payload +"\nCOST: " + message.outboundProperties['cost']</expression-component>

</when>

<otherwise>

<expression-component doc:name=*"Expression"*>payload= "ERROR: n must be less than 20"</expression-component>

</otherwise>

</choice>

</flow>

<flow name=*"calculateFibonacci"* doc:name=*"calculateFibonacci"*>

<vm:inbound-endpoint path=*"fibonacci"* exchange-pattern=*"request-response"* doc:name=*"VM"*/>

<ee:cache cachingStrategy-ref=*"Caching\_Strategy"*

filterExpression=*"#[groovy:message.getInboundProperty('nocache') == null]"* doc:name=*"Cache"*>

<logger level=*"INFO"* message=*"#[payload]"* doc:name=*"Logger"*/>

<expression-component doc:name=*"Expression"*><![CDATA[

n = message.inboundProperties['n'];

if (n < 2)

{

payload = n;

message.outboundProperties["cost"] = 1;

} else {

boolean cached = message.inboundProperties['nocache'] == null;

import org.mule.api.MuleMessage;

MuleMessage fib1 = fibonacciRequest(n-1, cached);

MuleMessage fib2 = fibonacciRequest(n-2, cached);

message.outboundProperties["cost"] = fib1.getInboundProperty("cost") + fib2.getInboundProperty("cost") + 1;

payload = Long.parseLong(fib1.getPayload()) + Long.parseLong(fib2.getPayload());

}

]]>

</expression-component>

</ee:cache>

</flow>

<flow name=*"ftpconnectorFlow1"* doc:name=*"ftpconnectorFlow1"*>

<file:inbound-endpoint path=*"C:\JobSearch\MuleDemos\File\FtpTransferFiles"* responseTimeout=*"10000"* doc:name=*"File"*/>

<logger message=*"It is working"* level=*"INFO"* doc:name=*"Logger"*/>

<ftp:outbound-endpoint host=*"127.0.0.1"* port=*"21"* user=*"user1"* password=*"password"* responseTimeout=*"10000"* doc:name=*"FTP"*/>

<logger message=*"Nivi Kundem"* level=*"INFO"* doc:name=*"Logger"*/>

</flow>

<flow name=*"datamapperflow"* doc:name=*"datamapperflow"*>

<file:inbound-endpoint path=*"C:\JobSearch\MuleDemos\File\InXml"* responseTimeout=*"10000"* doc:name=*"File"*/>

<data-mapper:transform config-ref=*"new\_mapping\_grf"* doc:name=*"DataMapper"*/>

<file:outbound-endpoint path=*"C:\JobSearch\MuleDemos\File\outCsv"* outputPattern=*"nivi.csv"* responseTimeout=*"10000"* doc:name=*"File"*/>

</flow>

<flow name=*"csvtoxml"* doc:name=*"csvtoxml"*>

<file:inbound-endpoint path=*"C:\JobSearch\MuleDemos\File\inputcsv"* responseTimeout=*"10000"* doc:name=*"File"*/>

<data-mapper:transform config-ref=*"new\_mapping\_1\_grf"* doc:name=*"DataMapper"*/>

<file:outbound-endpoint path=*"C:\JobSearch\MuleDemos\File\outputxml"* outputPattern=*"output.xml"* responseTimeout=*"10000"* doc:name=*"File"*/>

</flow>

</mule>

**TEST**



