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

<mule xmlns:json=*"http://www.mulesoft.org/schema/mule/json"* xmlns:cluster=*"http://www.mulesoft.org/schema/mule/ee/cluster"* xmlns:validation=*"http://www.mulesoft.org/schema/mule/validation"* xmlns:ws=*"http://www.mulesoft.org/schema/mule/ws"* xmlns:dw=*"http://www.mulesoft.org/schema/mule/ee/dw"* xmlns:metadata=*"http://www.mulesoft.org/schema/mule/metadata"* xmlns:context=*"http://www.springframework.org/schema/context"*

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

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-current.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/http http://www.mulesoft.org/schema/mule/http/current/mule-http.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/ee/dw http://www.mulesoft.org/schema/mule/ee/dw/current/dw.xsd*

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

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

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

<context:property-placeholder location=*"mule.properties"*/>

<http:listener-config name=*"HTTP\_Listener\_Configuration"* host=*"0.0.0.0"* port=*"8081"* doc:name=*"HTTP Listener Configuration"* />

<http:request-config name=*"United-http-request-config"* host=*"${united.host}"* port=*"80"* doc:name=*"HTTP Request Configuration"*>

<http:proxy host=*"172.17.2.218"* port=*"8085"* username=*"M1043142"* password=*"Shonit&amp;singh93"*/>

</http:request-config>

<http:request-config name=*"american-http-request-config"* host=*"localhost"* port=*"7070"* basePath=*"/api"* doc:name=*"HTTP Request Configuration"*>

<http:raml-api-configuration location=*"D:/american-api.raml"*/>

</http:request-config>

<ws:consumer-config name=*"delta-web-service-consumer"* wsdlLocation=*"http://mu.mulesoft-training.com/essentials/delta?wsdl"* service=*"TicketServiceService"* port=*"TicketServicePort"* serviceAddress=*"http://training-u.cloudhub.io/essentials/delta"* connectorConfig=*"United-http-request-config"* doc:name=*"Web Service Consumer"*/>

<payload-type-filter expectedType=*"java.util.List"* name=*"Payload"* doc:name=*"Payload"*/>

<flow name=*"muelsoft-week5Flow"*>

<http:listener config-ref=*"HTTP\_Listener\_Configuration"* path=*"/flights"* doc:name=*"HTTP"* allowedMethods=*"GET"*/>

<set-variable variableName=*"airline"* value=*"#[message.inboundProperties.'http.query.params'.airline]"* doc:name=*"Set airline name"*/>

<flow-ref name=*"setDestinationSubflow"* doc:name=*"setDestinationSubflow"*/>

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

<when expression=*"#[flowVars.airline == &quot;deltaflights&quot;]"*>

<flow-ref name=*"getDeltaFindFlights"* doc:name=*"getDeltaFindFlights"*/>

</when>

<when expression=*"#[flowVars.airline == &quot;united&quot;]"*>

<flow-ref name=*"getUnitedFlightsFlow"* doc:name=*"getUnitedFlightsFlow"*/>

</when>

<when expression=*"#[flowVars.airline == &quot;american&quot;]"*>

<flow-ref name=*"getAmericanFlightsFlow"* doc:name=*"getAmericanFlightsFlow"*/>

</when>

<when expression=*"#[flowVars.airline == &quot;delta&quot;]"*>

<flow-ref name=*"getDeltaFlightsFlow"* doc:name=*"getDeltaFlightsFlow"*/>

</when>

<otherwise>

<flow-ref name=*"getAllAirlineFlights"* doc:name=*"getAllAirlineFlights"*/>

</otherwise>

</choice>

<validation:is-not-empty message=*"Payload is empty."* value=*"#[payload]"* doc:name=*"Validation"*/>

<dw:transform-message doc:name=*"Transform Message"*>

<dw:set-payload><![CDATA[%dw 1.0

%output application/json

---

payload orderBy $.price]]></dw:set-payload>

</dw:transform-message>

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

<exception-strategy ref=*"ChoiceExceptionStrategy"* doc:name=*"Reference Exception Strategy"*/>

</flow>

<flow name=*"getUnitedFlightsFlow"*>

<http:request config-ref=*"United-http-request-config"* path=*"/essentials/united/flights/{destination}"* method=*"GET"* doc:name=*"United Rest Request"*>

<http:request-builder>

<http:uri-param paramName=*"destination"* value=*"#[flowVars.code]"*/>

</http:request-builder>

</http:request>

<dw:transform-message doc:name=*"Transform Message"* metadata:id=*"c5d017f0-3a54-471a-bd9f-816d0575366c"*>

<dw:input-payload mimeType=*"application/json"*/>

<dw:set-payload><![CDATA[%dw 1.0

%output application/java

---

payload.flights map ((flight , indexOfFlight) -> {

airlineName: flight.airlineName,

availableSeats: flight.emptySeats,

departureDate: flight.departureDate,

destination: flight.destination,

flightCode: flight.code,

origination: flight.origin,

planeType: flight.planeType,

price: flight.price

} as :object {

class : "com.mulesoft.training.Flight"

})]]></dw:set-payload>

</dw:transform-message>

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

<exception-strategy ref=*"ChoiceExceptionStrategy"* doc:name=*"Reference Exception Strategy"*/>

</flow>

<flow name=*"getAmericanFlightsFlow"*>

<http:request config-ref=*"american-http-request-config"* path=*"/flights"* method=*"GET"* doc:name=*"American Rest Request"*>

<http:request-builder>

<http:query-param paramName=*"code"* value=*"#[flowVars.code]"*/>

</http:request-builder>

</http:request>

<dw:transform-message doc:name=*"Transform Message"* metadata:id=*"f12cd90e-e81a-4836-ae48-87c197f6343c"*>

<dw:input-payload doc:sample=*"D:\american-response.json"* mimeType=*"application/json"*/>

<dw:set-payload><![CDATA[%dw 1.0

%output application/java

---

payload.flights map {

airlineName: $.'flight$$'.airline,

availableSeats: $.'flight$$'.availableSeats as :number,

departureDate: $.'flight$$'.departureDate,

destination: $.'flight$$'.destination,

flightCode: $.'flight$$'.code,

origination: $.'flight$$'.origin,

planeType: $.'flight$$'.plane.type,

price: $.'flight$$'.price

} as :object {

class : "com.mulesoft.training.Flight"

}]]></dw:set-payload>

<dw:set-variable variableName=*"americanResponse"*><![CDATA[%dw 1.0

%output application/java

---

{

}]]></dw:set-variable>

</dw:transform-message>

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

<exception-strategy ref=*"ChoiceExceptionStrategy"* doc:name=*"Reference Exception Strategy"*/>

</flow>

<flow name=*"getDeltaFlightsFlow"*>

<ws:consumer config-ref=*"delta-web-service-consumer"* operation=*"listAllFlights"* doc:name=*"Web Service Consumer"*/>

<dw:transform-message doc:name=*"Transform Message"* metadata:id=*"d2731cb9-1149-4631-b30a-4a76220e1eaa"*>

<dw:set-payload><![CDATA[%dw 1.0

%output application/java

%namespace ns0 http://soap.training.mulesoft.com/

---

payload.ns0#listAllFlightsResponse.\*return map ((return , indexOfReturn) -> {

airlineName: return.airlineName,

availableSeats: return.emptySeats,

departureDate: return.departureDate,

destination: return.destination,

flightCode: return.code,

origination: return.origin,

planeType: return.planeType,

price: return.price

} as :object {

class : "com.mulesoft.training.Flight"

})]]></dw:set-payload>

</dw:transform-message>

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

</flow>

<flow name=*"getDeltaFindFlights"*>

<dw:transform-message doc:name=*"Transform Message"*>

<dw:set-payload><![CDATA[%dw 1.0

%output application/xml

%namespace ns0 http://soap.training.mulesoft.com/

---

{

ns0#findFlight: {

destination: flowVars.code as :string

}

}]]></dw:set-payload>

</dw:transform-message>

<ws:consumer config-ref=*"delta-web-service-consumer"* operation=*"findFlight"* doc:name=*"Web Service Consumer"*/>

<dw:transform-message doc:name=*"Transform Message"* metadata:id=*"d2731cb9-1149-4631-b30a-4a76220e1eaa"*>

<dw:set-payload><![CDATA[%dw 1.0

%output application/java

%namespace ns0 http://soap.training.mulesoft.com/

---

payload.ns0#findFlightResponse.\*return map ((return , indexOfReturn) -> {

airlineName: return.airlineName,

availableSeats: return.emptySeats,

departureDate: return.departureDate,

destination: return.destination,

flightCode: return.code,

origination: return.origin,

planeType: return.planeType,

price: return.price

} as :object {

class : "com.mulesoft.training.Flight"

})]]></dw:set-payload>

</dw:transform-message>

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

<exception-strategy ref=*"ChoiceExceptionStrategy"* doc:name=*"Reference Exception Strategy"*/>

</flow>

<flow name=*"getAllAirlineFlights"*>

<scatter-gather doc:name=*"Scatter-Gather"*>

<processor-chain>

<flow-ref name=*"getAmericanFlightsFlow"* doc:name=*"getAmericanFlightsFlow"*/>

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

</processor-chain>

<processor-chain>

<flow-ref name=*"getUnitedFlightsFlow"* doc:name=*"getUnitedFlightsFlow"*/>

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

</processor-chain>

<processor-chain>

<flow-ref name=*"getDeltaFindFlights"* doc:name=*"getDeltaFindFlights"*/>

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

</processor-chain>

</scatter-gather>

<dw:transform-message doc:name=*"Transform Message"*>

<dw:set-payload><![CDATA[%dw 1.0

%output application/java

---

flatten payload

]]></dw:set-payload>

</dw:transform-message>

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

</flow>

<flow name=*"PostFlightFlow"*>

<http:listener config-ref=*"HTTP\_Listener\_Configuration"* path=*"/flights"* allowedMethods=*"POST"* doc:name=*"HTTP"*/>

<dw:transform-message doc:name=*"Transform Message to JSON"* metadata:id=*"09083658-dbef-4c94-b216-e5c734d987cb"*>

<dw:input-payload doc:sample=*"com\mule\samples\american-flights-example.json"* mimeType=*"application/json"*/>

<dw:set-payload><![CDATA[%dw 1.0

%output application/json

---

flights:

payload map

'flight$$': {

ID: $.ID,

code: $.code,

price: $.price as :number {class:"java.lang.Double"},

departureDate: $.departureDate as :date {format:"yyyy/MM/dd"}

as :string {format: "MMM dd, yyyy"},

origin: $.origin,

emptySeats: $.emptySeats as :number,

plane:

{

type: $.plane.type,

totalSeats: $.plane.totalSeats as :number

}

}]]></dw:set-payload>

<dw:set-variable variableName=*"xml"*><![CDATA[%dw 1.0

%output application/xml

---

flights:{ (payload map {

'flight$$':{

ID: $.ID,

code: $.code,

price: $.price as :number {class:"java.lang.Double"},

departureDate: $.departureDate as :date {format:"yyyy/MM/dd"}

as :string {format: "MMM dd, yyyy"},

origin: $.origin,

destination: $.destination,

emptySeats: $.emptySeats as :number,

plane: {

type: $.plane.type,

totalSeats: $.plane.totalSeats as :number

}

}

})

}

]]></dw:set-variable>

</dw:transform-message>

<set-payload value=*"#[flowVars.xml]"* doc:name=*"Set Payload to Xml"*/>

<logger message=*"XML data after dataweave: #[flowVars.xml]"* level=*"INFO"* doc:name=*"Logger"*/>

</flow>

<flow name=*"PostFlightFromXmlFlow"*>

<http:listener config-ref=*"HTTP\_Listener\_Configuration"* path=*"/flights/xml"* doc:name=*"HTTP"*/>

<dw:transform-message doc:name=*"Transform Message to JSON"* metadata:id=*"d554ead1-5970-436f-bc96-bcc9b621e3cf"*>

<dw:input-payload doc:sample=*"D:\MUFundamentals3.9\_studentFiles\examples\flights-example.xml"* mimeType=*"application/xml"*/>

<dw:set-payload><![CDATA[%dw 1.0

%output application/json

%namespace ns0 http://soap.training.mulesoft.com/

---

flights:{(payload.ns0#listAllFlightsResponse.\*return map {

'flight$$':{

code: $.code,

origin: $.origin,

destination: $.destination,

departureDate: $.departureDate as :date {format:"yyyy/MM/dd"}

as :string {format: "MMM dd, yyyy"},

price: $.price as :number {class:"java.lang.Double"},

emptySeats: $.emptySeats,

plane:{

type: $.planeType

}

}

}

)}

]]></dw:set-payload>

</dw:transform-message>

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

</flow>

<flow name=*"PostFlightFromXmlToJava"*>

<http:listener config-ref=*"HTTP\_Listener\_Configuration"* path=*"/flights/java"* doc:name=*"HTTP"*/>

<dw:transform-message doc:name=*"Transform Message"* metadata:id=*"1e35e4de-5586-4be3-9131-4e98384c071c"*>

<dw:input-payload doc:sample=*"D:\MUFundamentals3.9\_studentFiles\examples\flights-example.xml"* mimeType=*"application/xml"*/>

<dw:set-payload><![CDATA[%dw 1.0

%output application/java

%type flight = :object {class:"com.mulesoft.training.Flight"}

%namespace ns0 http://soap.training.mulesoft.com/

---

flights: payload.ns0#listAllFlightsResponse.\*return map {

flightCode: $.code,

availableSeats: $.emptySeats,

origination: $.origin,

price: $.price as :number {class:"java.lang.Double"},

destination: $.destination,

departureDate: $.departureDate as :date {format:"yyyy/MM/dd"}

as :string {format: "MMM dd, yyyy"},

airlineName: "American",

planeType: upper ($.planeType replace /(Boing)/ with "Boeing")

} as :flight orderBy $.departureDate orderBy $.price]]></dw:set-payload>

</dw:transform-message>

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

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

</flow>

<sub-flow name=*"setDestinationSubflow"*>

<set-variable variableName=*"code"* value=*"#[message.inboundProperties.'http.query.params'.code == empty ? 'SFO' : message.inboundProperties.'http.query.params'.code]"* doc:name=*"Set Destination"*/>

<logger message=*"The code is: #[flowVars.code]"* level=*"INFO"* doc:name=*"Logger"*/>

</sub-flow>

<choice-exception-strategy name=*"ChoiceExceptionStrategy"*>

<catch-exception-strategy when=*"#[exception.causeMatches('com.mulesoft.weave.\*')]"* doc:name=*"Catch Exception Strategy"*>

<set-payload value=*"{&quot;message&quot;: &quot;No flight to: #[flowVars.code ]&quot;, &quot;exception&quot;: &quot;#[exception]&quot;}"* mimeType=*"application/json"* doc:name=*"Set Payload"*/>

<set-property propertyName=*"http.status"* value=*"400"* doc:name=*"Property"*/>

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

</catch-exception-strategy>

<catch-exception-strategy when=*"#[exception.causedBy(org.mule.api.MessagingException)]"* doc:name=*"Catch Exception Strategy"*>

<set-payload value=*"{&quot;message&quot;: &quot;Some Exception Occurred&quot;, &quot;exception&quot;: &quot;#[exception]&quot;}"* doc:name=*"Set Payload"* mimeType=*"application/json"*/>

<set-property propertyName=*"http.status"* value=*"500"* doc:name=*"Property"*/>

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

</catch-exception-strategy>

</choice-exception-strategy>

</mule>