



MuleSoft®

Module 8: Consuming Web Services



Goal

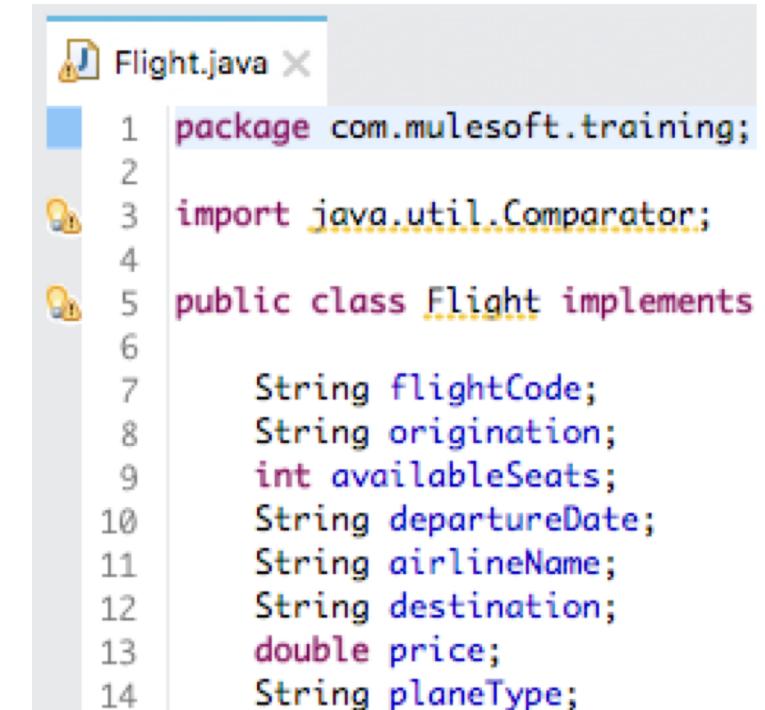
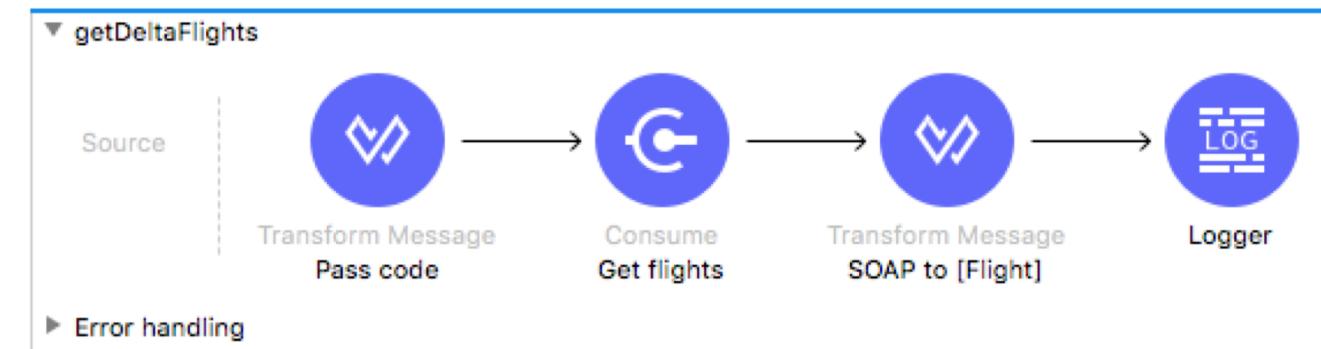
Call an operation
of a connector
in Exchange



Call a RESTful
web service



Call a SOAP
web service



```
Flight.java X
1 package com.mulesoft.training;
2
3 import java.util.Comparator;
4
5 public class Flight implements
6
7     String flightCode;
8     String origination;
9     int availableSeats;
10    String departureDate;
11    String airlineName;
12    String destination;
13    double price;
14    String planeType;
```

At the end of this module, you should be able to



- Consume web services that have a connector in Anypoint Exchange
- Consume RESTful web services
- Consume SOAP web services
- Pass parameters to SOAP web services using the Transform Message component
- Transform data from multiple services to a canonical format

Consuming web services that have a connector in Exchange



Connectors and modules



- **Modules** are extensions to the Mule runtime that you can use when building a Mule app
 - HTTP, Database, Salesforce, SAP, Slack, Validation, Java, and many more
- **Connectors** are modules that connect to an external server
 - HTTP, Database, Salesforce, SAP, Slack
- For module reference
 - <https://docs.mulesoft.com/connectors/>

✓ Connectors and Modules (for Mule 4)

- | | | |
|-----------------------------|--|----------------------------------|
| > Amazon DynamoDB Connector | > Java Module | > Salesforce Analytics Connector |
| > Amazon EC2 Connector | > JMS Connector | > Salesforce Composite Connector |
| > Amazon RDS Connector | > Kafka Connector | > Salesforce Connector |
| > Amazon S3 Connector | > LDAP Connector | > Salesforce Marketing Connector |
| > Amazon SNS Connector | > Microsoft Dynamics 365 Connector | SAP Connector |
| > Amazon SQS Connector | > Microsoft Dynamics 365 Operations Connecto | > SAP Concur Connector |
| > Anypoint MQ Connector | > Microsoft Dynamics AX Connector | > Scripting Module |
| > BMC Remedy Connector | > Microsoft Dynamics CRM Connector | > ServiceNow Connector |
| > Box Connector | > Microsoft Dynamics NAV Connector | > SFTP Connector |
| > Cassandra Connector | > Microsoft MSMQ Connector | > SharePoint Connector |
| > Database Connector | > Microsoft Powershell Connector | > Siebel Connector |
| > EDIFACT EDI Connector | > MongoDB Connector | > Spring Module |
| > Email Connector | > Neo4j Connector | > TRADACOMS EDI Connector |
| > File Connector | > NetSuite Connector | > Twilio Connector |
| > FTP Connector | OAuth Module Documentation Reference | > Validation Module |
| > FTPS Connector | > Object Store Connector | > VM Connector |
| > HDFS (Hadoop) Connector | > Oracle EBS 12.1 Connector | > Web Service Consumer Connector |
| > HL7 EDI Connector | > Oracle EBS 12.2 Connector | > Workday Connector |
| > HTTP Connector | > PeopleSoft Connector | > X12 EDI Connector |
| > IBM CTG Connector | > Redis Connector | > XML Module |
| | | > Zuora Connector |

Connectors in Anypoint Exchange



- Many connectors in Exchange package a much easier way to make calls to APIs

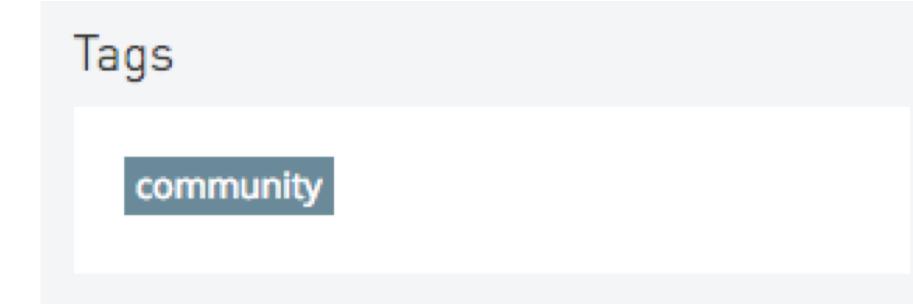
A screenshot of the MuleSoft Anypoint Exchange interface. The top navigation bar has 'Exchange' and 'Login' on the right. A sidebar on the left shows 'MuleSoft' selected under 'Assets'. The main area is titled 'Assets' with a 'Connectors' dropdown and a search bar. It displays a grid of connector cards. Each card includes a small icon, the connector name, its rating (5 stars), and the provider ('MuleSoft').

Category	Connector	Ratings	Provider
Connector	Mule Anypoint MQ Connector	★★★★★	MuleSoft
Module	Mule FTPS connector	★★★★★	MuleSoft
Module	FTP Connector	★★★★★	MuleSoft
Connector	Reltio Cloud Connector	★★★★★	MuleSoft
Connector	Splunk Connector	★★★★★	MuleSoft
Module	Microsoft Dynamics CRM Connector	★★★★★	MuleSoft

Connector types specify creator and support level



- The type of selector is specified in its tags on Exchange



	Premium	Select	MuleSoft Certified	Community
Additional cost	x			
Updated APIs	x	x		
Fully tested	x	x		
MuleSoft Support	Tier 1-3	Tier 1-3	Tier 1 (From developer: T2/T3)	Tier 1
Connector examples	HL7 SAP Siebel	Salesforce Workday	AS/400 Oracle JD Edwards Microsoft Azure Storage	LinkedIn Slack

Connector support levels



- Tier 1
 - MuleSoft will isolate the problem and diagnose it
- Tier 2
 - MuleSoft will find a workaround
- Tier 3
 - MuleSoft will fix the code

	Premium	Select	MuleSoft Certified	Community
Not included in Platform license	x			
Tier 2-3 Support	x	x		
Tier 1 Support	x	x	x	x

REST Connectors in Anypoint Exchange



- REST Connect converts a RAML 1.0 API specification added to Exchange to a connector
 - You did this in Module 4
- You can use the connector in Anypoint Studio or flow designer

A screenshot of the Anypoint Exchange web interface. The top navigation bar includes a menu icon, the 'Exchange' logo, 'Training' (which is highlighted in blue), a help icon, and a user icon. The left sidebar has sections for 'Assets', 'Organizations', 'MuleSoft' (which is highlighted in orange), and 'Training'. Below the sidebar is a 'My applications' section with a 'Public portal' link. The main content area is titled 'Assets' and features a search bar with 'All types' dropdown and a 'Search' input field. There are two cards displayed: 'American Flights API Connector' (Connector type, 5 stars, Max Mule) and 'American Flights API' (REST API type, 5 stars, Max Mule).

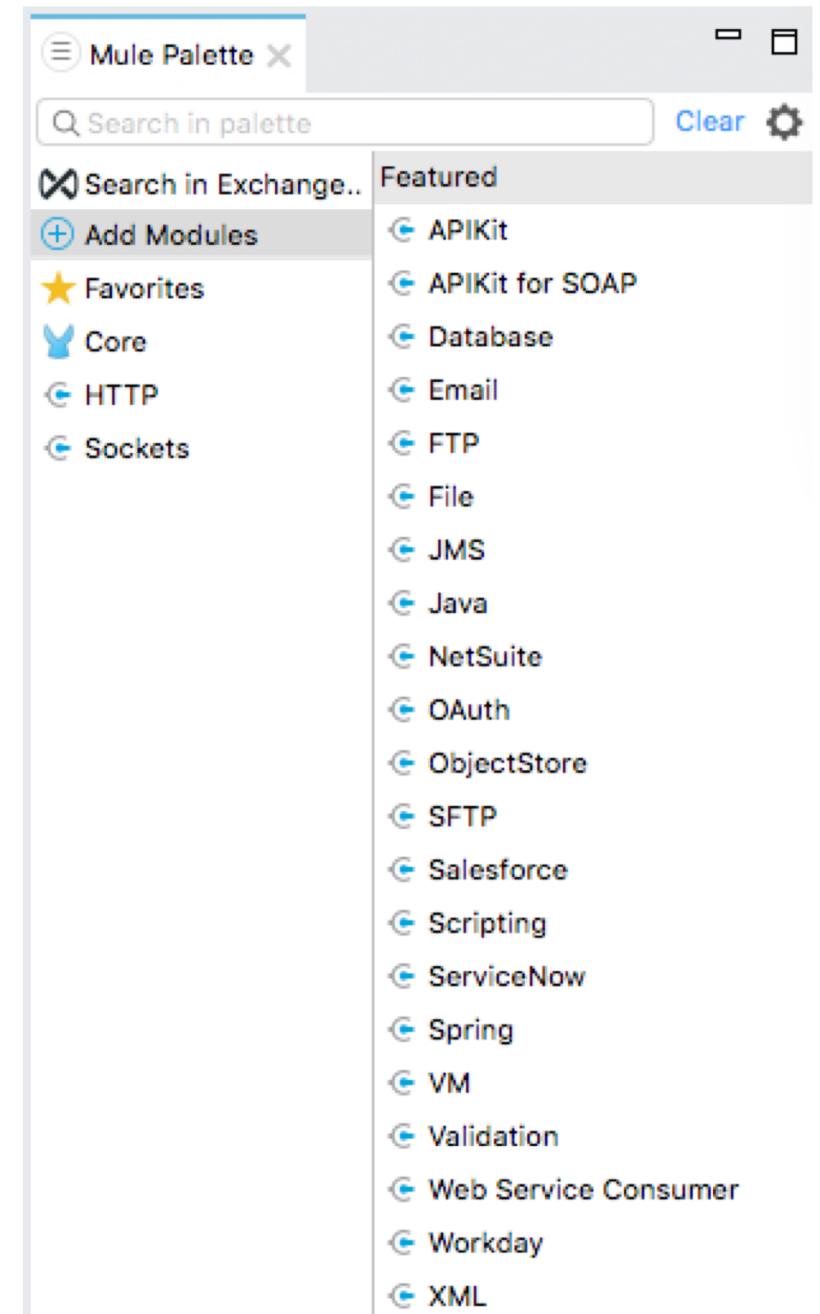
The screenshot shows the Anypoint Exchange interface with the 'Training' tab selected. The left sidebar shows 'MuleSoft' is the active organization. Two connectors are listed in the main content area:

- American Flights API Connector (Connector, 5 stars, Max Mule)
- American Flights API (REST API, 5 stars, Max Mule)

Connectors in Anypoint Studio



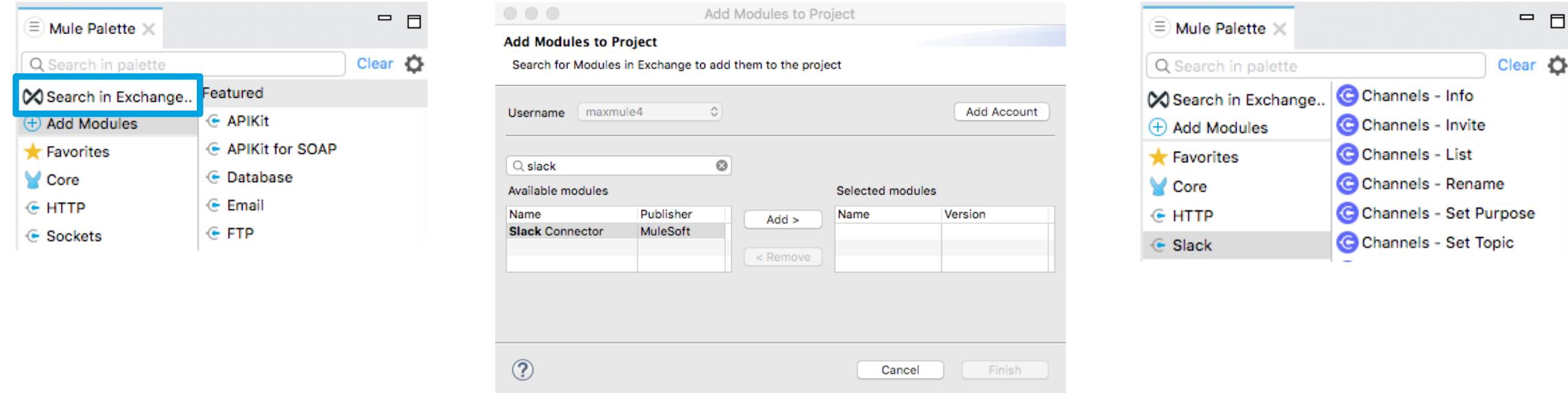
- Some modules are **pre-installed** in Studio
 - HTTP, Database, Salesforce, Validation, Java
- Some modules are **not pre-installed** in Studio
 - SAP, Slack



Adding connectors from Exchange



- If connectors are not pre-installed in Anypoint Studio, you can search Exchange and add them to a project



Walkthrough 8-1: Consume a RESTful web service that has a connector in Exchange



- Create a new flow to call the American RESTful web service
- Add a REST connector from Exchange to an Anypoint Studio project
- Configure and use a REST connector to make a call to a web service
- Dynamically set a query parameter for a web service call

The screenshot shows the Anypoint Studio interface with two main components:

- Flow Editor:** Displays a flow named "getAmericanFlights". The flow starts with a "Listener" icon (blue globe with a white arrow) configured for "GET /american". An arrow points to a "Flow Reference" icon (blue square with a white arrow). Another arrow points to a "Get flights" icon (blue circle with a white "C"). A return arrow originates from the "Get flights" icon and points back to the "Listener" icon. Below the flow is a "Error handling" section.
- American Flights API Connector Configuration:** This is a modal window with the following details:
 - Title:** Get flights
 - General Tab:** Shows "There are no errors." and a "Display Name" field containing "Get flights".
 - Basic Settings:** Shows "Connector configuration: American_Flights_API_Config".
 - General Tab (Details):** Shows "Client id: 7623dbcc2e1949d7a861160fe4a3a1e6", "Client secret: 52505680a6FB4d52892449d32F4D3a18", and "Destination: #[vars.code]".

Consuming RESTful web services



Consuming RESTful web services



- First check and see if there is an existing Anypoint Connector in Studio or Exchange to connect to the service provider
- If there is not, use the **HTTP** connector and its **Request** operation
 - Configure the operation and/or global element configuration
 - Specify any headers, query parameters, or URI parameters to pass to the call



The screenshot shows the MuleSoft Anypoint Studio interface with the 'Get flights' operation selected. The 'General' tab is active, displaying configuration options for the request. The 'Request' section includes fields for Method (set to GET), Path (/united/flights/{dest}), and URL. Below these, the 'URI Parameters' tab is selected, showing a table with a single entry: Name "dest" and Value "vars.code". The status bar at the top right indicates "There are no errors".

Name	Value
"dest"	vars.code

Walkthrough 8-2: Consume a RESTful web service



- Create a new flow to call the United RESTful web service
- Use the HTTP Request operation to call a RESTful web service
- Dynamically set a URI parameter for a web service call
- Add metadata for an HTTP Request operation's response

The screenshot shows the Mule Studio interface with two main panes. On the left, the 'getUnitedFlights' flow is displayed, consisting of a Listener (GET /united), a Flow Reference (setCode), and a Request (Get flights). On the right, the 'Get flights' configuration pane is open, showing the following details:

- General**:
 - MIME Type: `HTTP_Request_configuration_training`
 - Configuration: `http://${training.host}:${training.port}${training.basepath}/united/flights/{dest}`
 - Request:
 - Method: `GET (Default)`
 - Path: `/united/flights/{dest}`
 - URL:
 - URI Parameters:

Name	Value
<code>"dest"</code>	<code>vars.code</code>
 - Mule Message
 - Payload
 - Object : Object
 - flights : Array<Object>?
 - airlineName : String?
 - price : Number?
 - departureDate : String?
 - planeType : String?
 - origin : String?
 - code : String?
 - emptySeats : Number?
 - destination : String?- Output**:
 - Mule Message
 - Payload
 - Object : Object
 - flights : Array<Object>?
 - airlineName : String?
 - price : Number?
 - departureDate : String?
 - planeType : String?
 - origin : String?
 - code : String?
 - emptySeats : Number?
 - destination : String?

Consuming SOAP web services



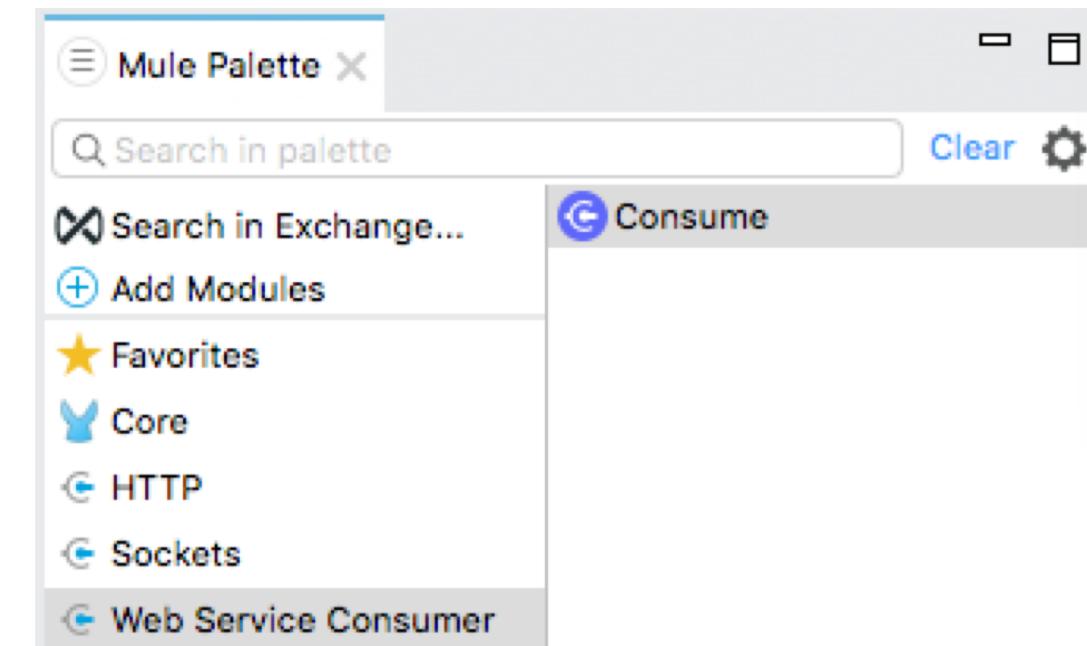
Consuming SOAP web services



- First check and see if there is an existing Anypoint Connector in Studio or Exchange to connect to the service provider
- If there is not, use the **Web Service Consumer** connector
 - Add the Web Service Consumer module to the project
 - Configure a global element configuration, which includes the location of the WSDL
 - Use the Consume operation
 - Select the SOAP operation to invoke



Consume



Passing data to a SOAP web service



- Use the **Transform Message** component to pass arguments to a SOAP web service
- When you add it before the Consume operation, DataSense is used to create metadata for the input that includes the arguments

The screenshot shows the Mule Studio interface with the following details:

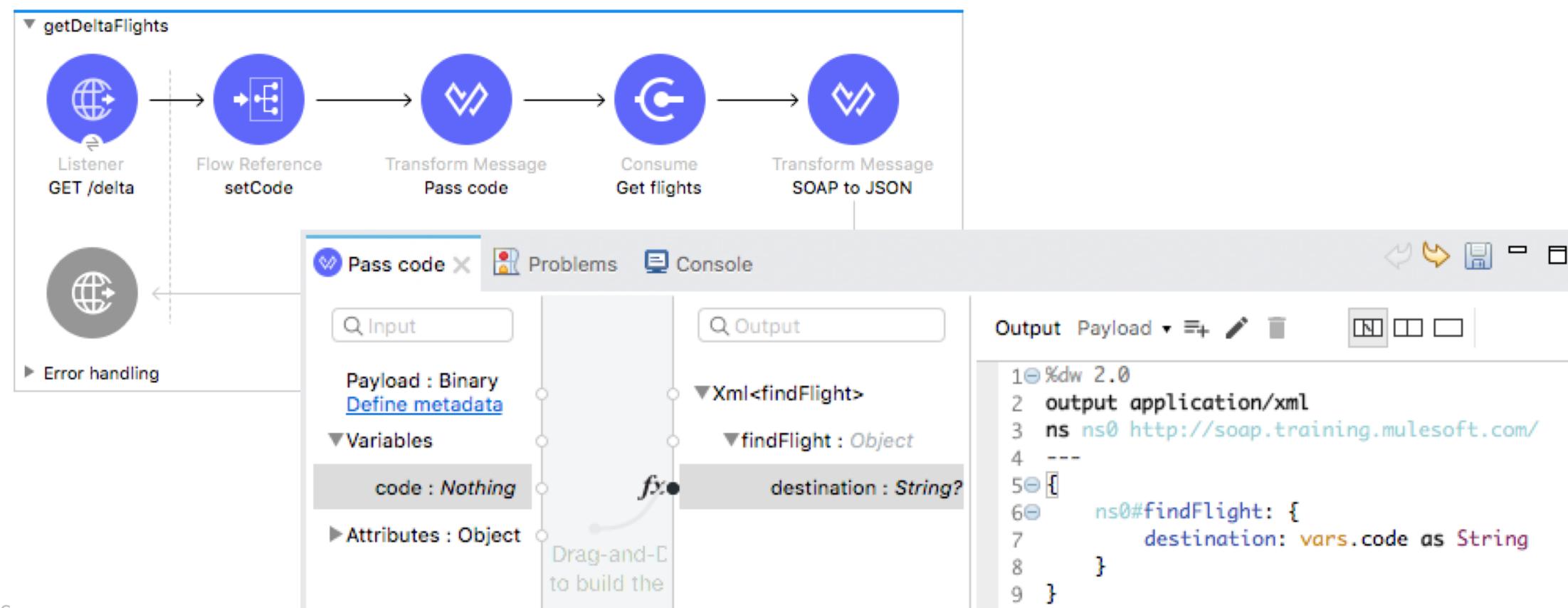
- Toolbar:** Includes tabs for "Pass code", "Problems", "Console", and "api APIkit Consoles (apdev-flights-ws)".
- Input Section:** Shows "Payload : Binary" and a link to "Define metadata".
- Variables Section:** Shows "code : Nothing".
- Attributes Section:** Shows "Attributes : Object" and "clientCertificate : C".
- Output Section:** Shows "Output Payload" with XML structure: `<?xml version="1.0"?><findFlight><destination>vars.code</destination></findFlight>`.
- Code View:** Displays the generated Java code:

```
1 %dw 2.0
2 output application/xml
3 ns ns0 http://soap.training.mulesoft.com/
4 ---
5 [
6   ns0#findFlight: {
7     destination: vars.code as String
8   }
9 ]
```

Walkthrough 8-3: Consume a SOAP web service



- Create a new flow to call the Delta SOAP web service
- Use the Web Service Consumer connector to call a SOAP web service
- Use the Transform Message component to pass arguments to a SOAP web service



Combining data from multiple services



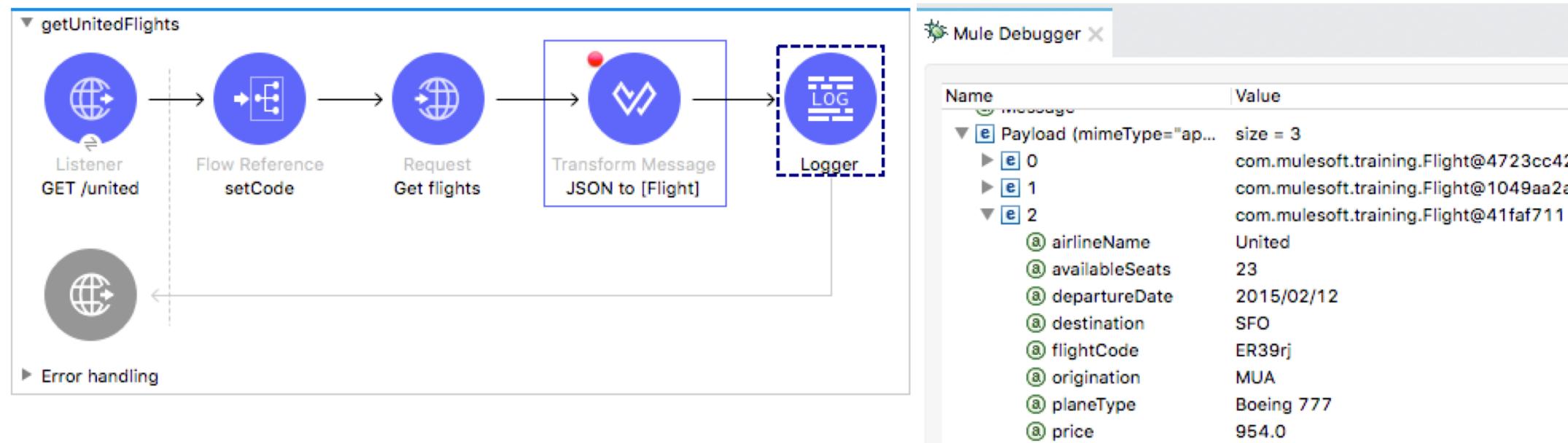
Combining data from multiple services



- Data from different services is pretty much always going to be in different formats
- To combine the data sets, you need to transform each of them to a canonical, or standard format
 - In this module, you will use a Java class as the canonical format
 - In module 11, you will learn to use the DataWeave format as a canonical format

Walkthrough 8-4: Transform data from multiple services to a canonical format

- Define a metadata type for the Flight Java class
- Transform the results from RESTful and SOAP web service calls to a collection of Flight objects



Summary



- To consume a web service, first look to see if it has a **connector in Anypoint Exchange**
 - Easiest way to consume a web service
- Use the **HTTP Request** operation to consume any REST web service
 - With or without URI parameters and query parameters
 - With or without a RAML definition
- Use the **Web Service Consumer** connector to consume any SOAP web service
- Use the Transform Message component to pass arguments to SOAP web services