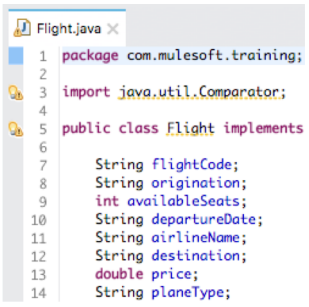
Module 8: Consuming Web Services

Goal

Call an operation of a connector in Exchange

Call a RESTful web service

Call a SOAP web service



```
Flight.java
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;
```

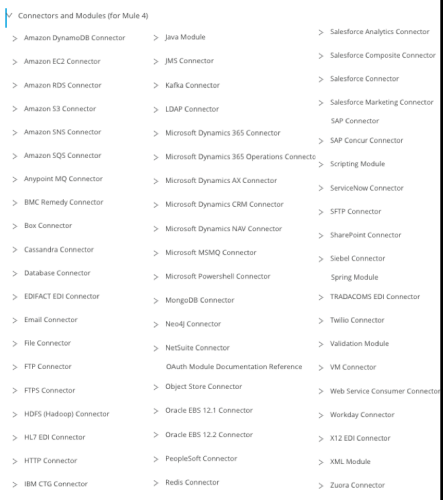
At the end of this module, you should be able to

- 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

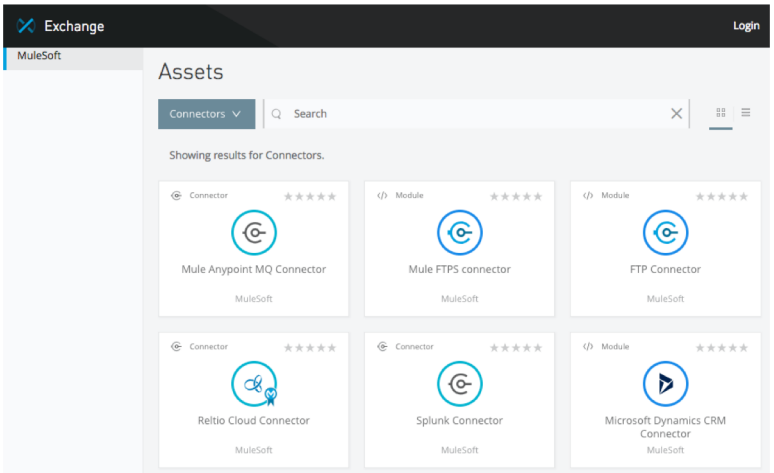
Connectors and modules

- **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 in Anypoint Exchange

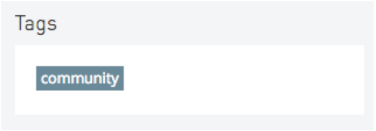
- Many connectors in Exchange package a much easier way to make calls to APIs



Connector types specify creator and support level

MuleSoft

- 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

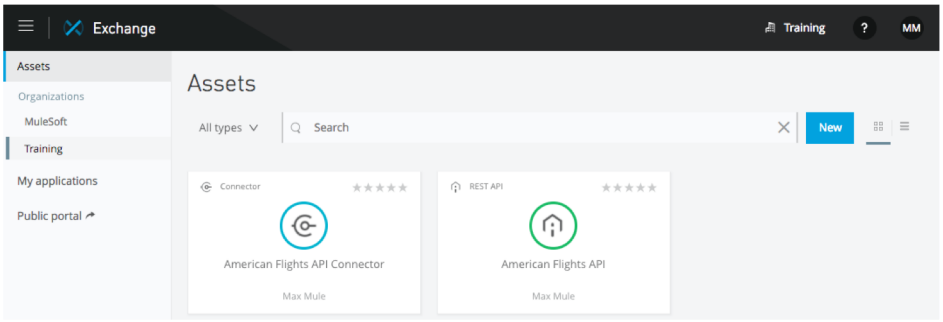
MuleSoft

- 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

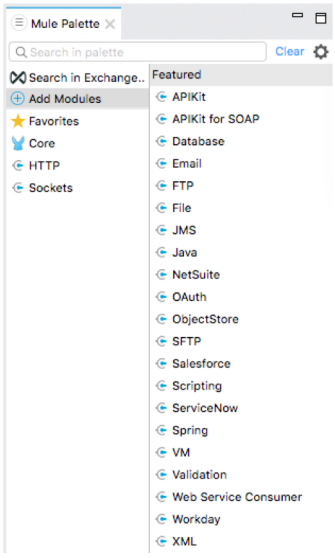
REST Connectors in Anypoint Exchange

- 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



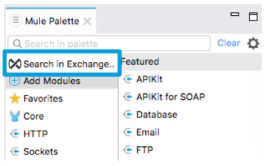
Connectors in Anypoint Studio

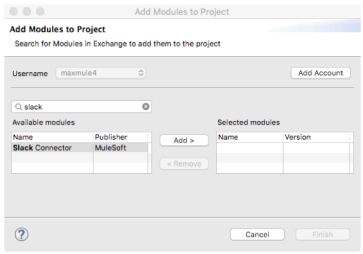
- Some modules are **pre-installed** in Studio
  - HTTP, Database, Salesforce, Validation, Java
- Some modules are **not pre-installed** in Studio
  - SAP, Slack

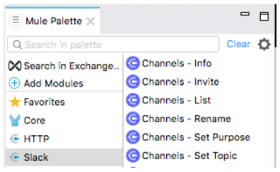


Adding connectors from Exchange

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

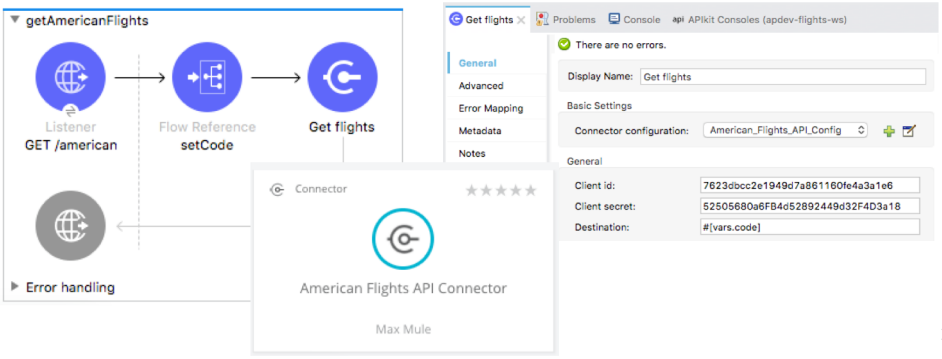






Walkthrough 8-1: Consume a RESTful web service that has a connector in Exchange

- 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

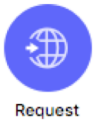


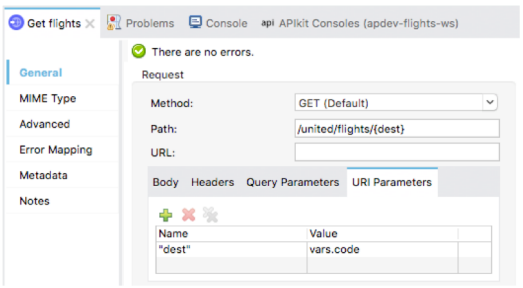
Consuming RESTful web services

Consuming RESTful web services

MuleSoft

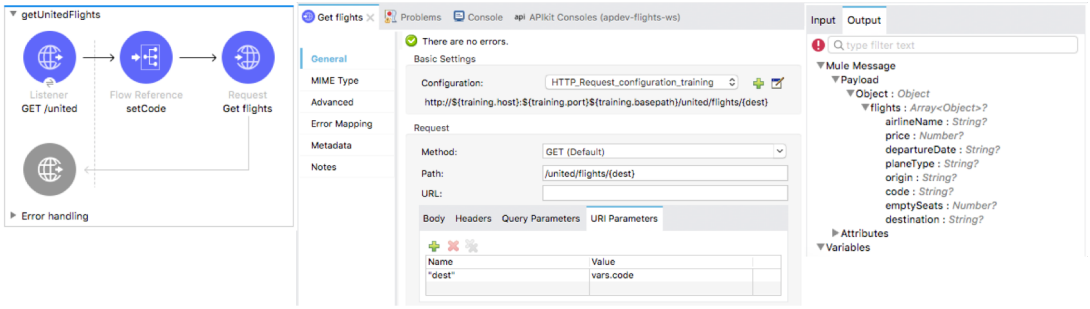
- First check and see if there is an existing Anypoint Connector in Studio or Exchange to connect to the service provider
- If there is not, use the **HTTP** connector and its **Request** operation
  - Configure the operation and/or global element configuration
  - Specify any headers, query parameters,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





Walkthrough 8-2: Consume a RESTful web service

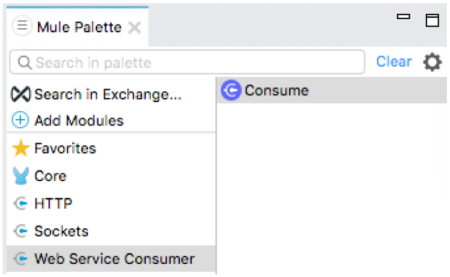
- Create a new flow to call the United RESTful web service
- Use the HTTP Request operation to call a RESTful web service
- Dynamically set a URI parameter for a web service call
- Add metadata for an HTTP Request operation's response



Consuming SOAP web services

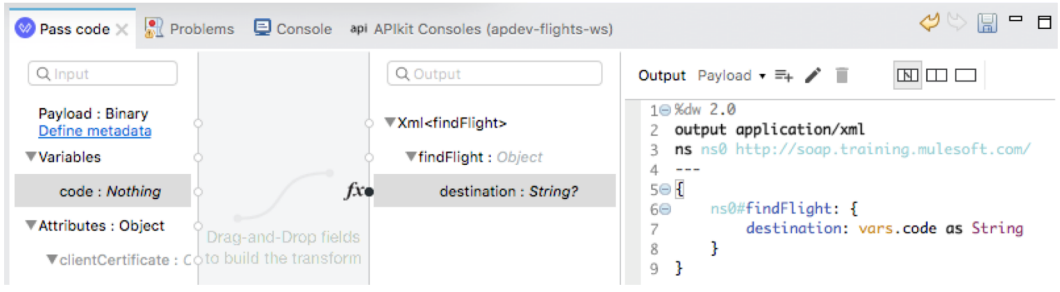
Consuming SOAP web services

- 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



Passing data to a SOAP web service

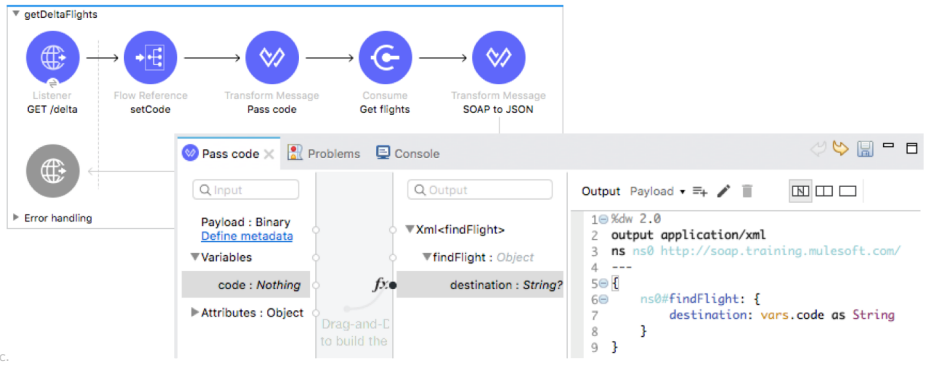
- Use the **Transform Message** component to pass arguments to a SOAP web service
- When you add it before the Consume operation, DataSense is used to create metadata for the input that includes the arguments



Walkthrough 8-3: Consume a SOAP web service

MuleSoft

- 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



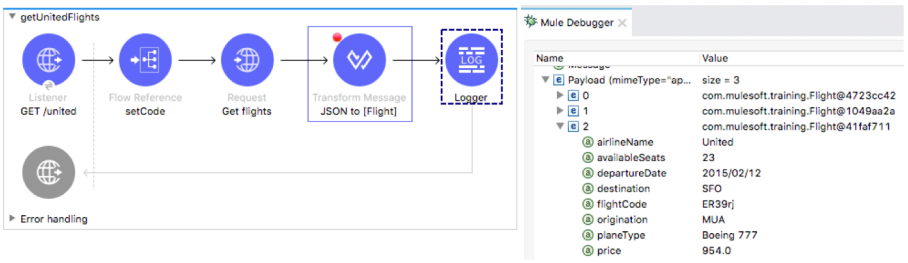
Combining data from multiple services

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

Summary

MuleSoft

- 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