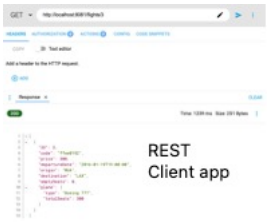
MuleSoft

Module 4: Building APIs

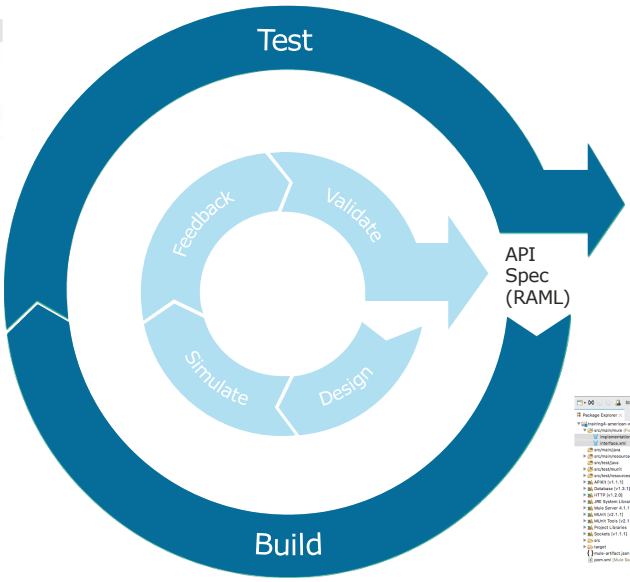
1

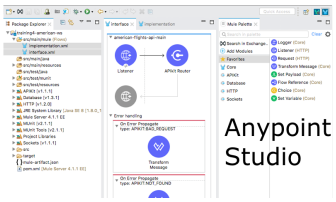
Goal

MuleSoft



REST
Client app





Anypoint
Studio

2

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

MuleSoft

- Use Anypoint Studio to build, run, and test Mule applications
- Use a connector to connect to databases
- Use the graphical DataWeave editor to transform data
- Create RESTful interfaces for applications from API specifications
- Connect API interfaces to API implementations
- Synchronize changes to API specifications between Anypoint Studio and Anypoint Platform

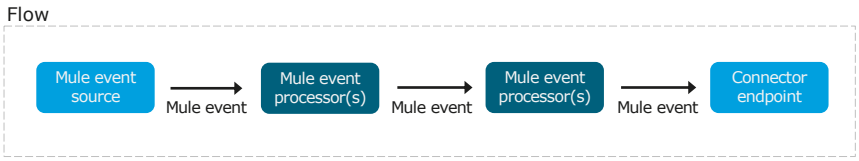
3

Reviewing Mule 4 applications

4

Review: Mule 4 applications and flows

- Mule applications receive events, process them, and route them to other endpoints
- Mule applications accept and process a Mule event through a series of Mule event processors plugged together in a flow with
  - A **Mule event source** that initiates the execution of the flow
  - **Mule event processors** that transform, filter, enrich, and process the event and its message



5

Review: Mule 4 event structure

The data that passes through flows in the app

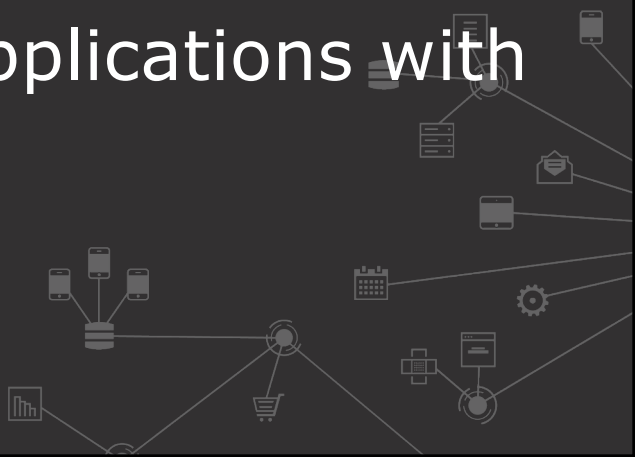
Metadata contained in the message header

The core info of the message - the data the app processes

Metadata for the Mule event - can be defined and referenced in the app processing the event

6

Creating Mule applications with Anypoint Studio



7

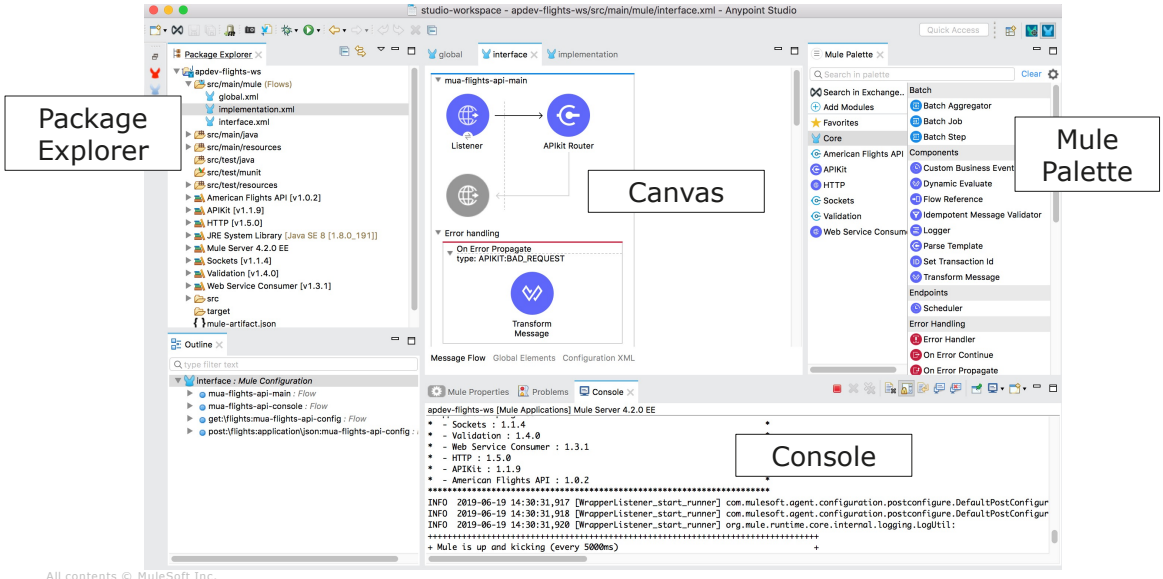
Introducing Anypoint Studio

MuleSoft

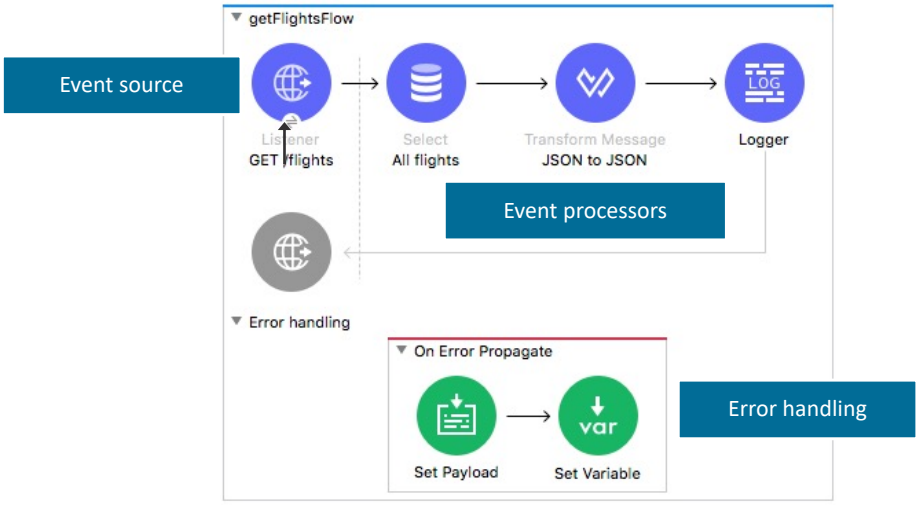
- Based on Eclipse, a common Java integrated development environment
- Features include
  - Two-way editing between graphical and XML views
  - Pre-built tooling to connect to APIs (REST, SOAP), protocols (HTTP, FTP, SMTP, more), and popular services (Salesforce, Workday, more!)
  - A data transformation framework and language
  - An embedded Mule runtime to test applications without leaving it
  - Visual debugging
  - Round-trip editing of API specifications with Anypoint Platform
  - One-click deployment of applications to CloudHub
  - Templates for common integration patterns
  - Integration with Maven for continuous build processes

8

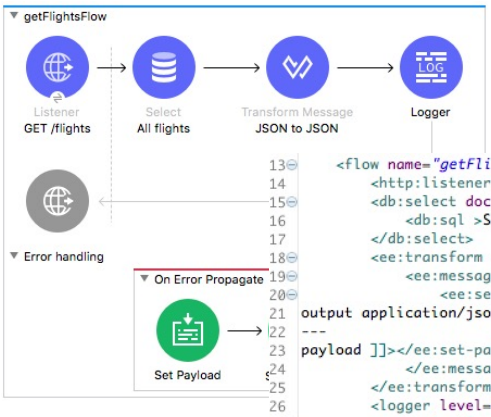
Anypoint Studio anatomy



Anatomy of a flow: Visual



Anatomy of a flow: XML

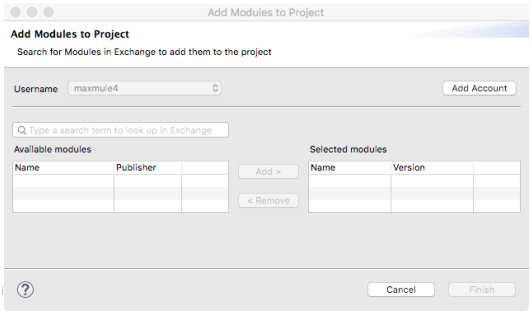


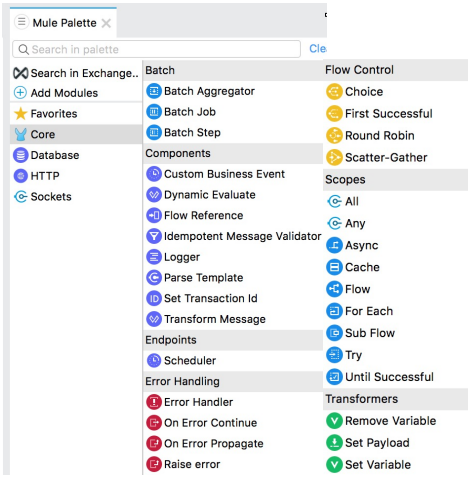

```
<flow name="getFlightsFlow" doc:id="fd1b7165-5456-476a-818f-4b25016b4a26" >
    <http:listener doc:name="GET /flights" doc:id="97ab8f0f-2cb3-4809-ab61-
    <db:select doc:name="All flights" doc:id="13978339-a263-4512-9b6b-114f1
        <db:sql >SELECT * FROM american</db:sql>
    </db:select>
    <ee:transform doc:name="JSON to JSON" doc:id="8a319aca-486a-487b-840a-c
        <ee:message >
            <ee:set-payload ><![CDATA[%dw 2.0
output application/json
---
payload ]]></ee:set-payload>
        </ee:message>
    </ee:transform>
    <logger level="INFO" doc:name="Logger" doc:id="5b1628cf-ec96-4beb-81d6-
    <error-handler >
        <on-error-propagate enableNotifications="true" logException="true"
            <set-payload value="#[payload]" doc:name="Set Payload" doc:id="
            <set-variable value="test" doc:name="Set Variable" doc:id="f40e
        </on-error-propagate>
    </error-handler>
</flow>
```


11

Mule application building blocks

- Are separated into categories in the Core section of the Mule Palette
- By default, projects include HTTP and Sockets modules
- Can add additional modules



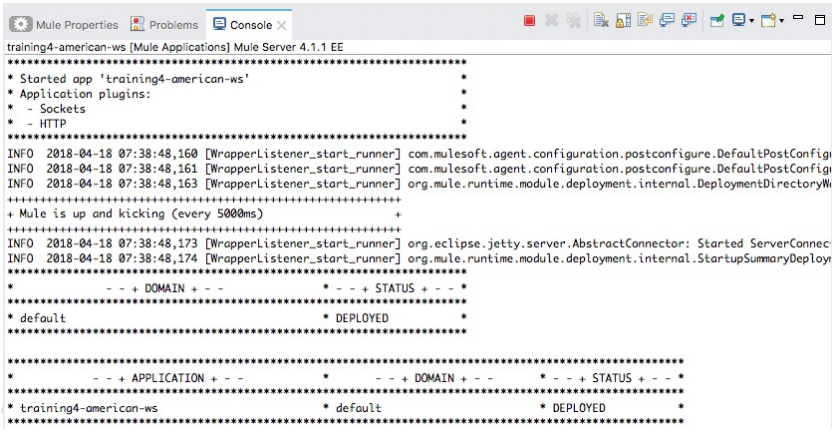


12

Running applications

MuleSoft

- Anypoint Studio comes with an embedded Mule runtime to test applications without leaving it
- The console outputs application logs and information

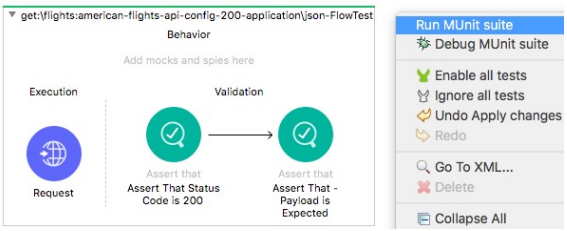


13

Automating testing of applications

MuleSoft

- You can automate testing of Mule applications using MUnit
- MUnit is a Mule app testing framework for building automated tests
- MUnit is fully integrated with Anypoint Studio
  - You can create, design, and run MUnit tests and suites of tests just like you do Mule applications



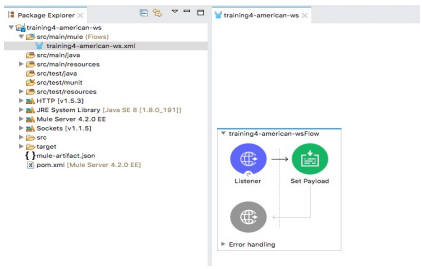
- MUnit is covered in *Anypoint Platform Development: Production-Ready Development Practices*

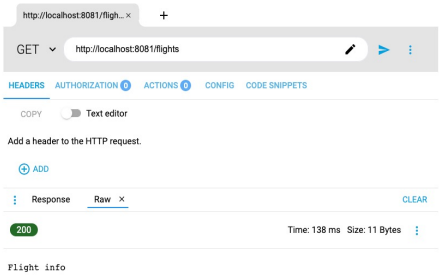
14

Walkthrough 4-1: Create a Mule application with Anypoint Studio

MuleSoft

- Create a new Mule project with Anypoint Studio
- Add a connector to receive requests at an endpoint
- Set the event payload
- Run a Mule application using the embedded Mule runtime
- Make an HTTP request to the endpoint using ARC





15

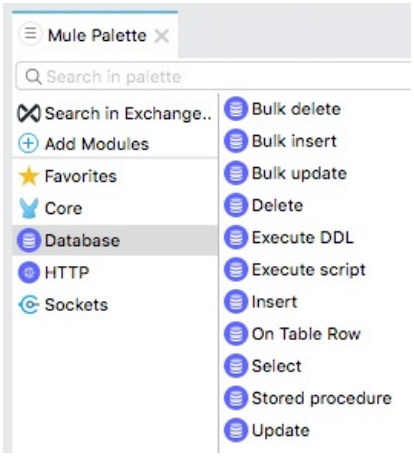
Connecting to data

16

The Database connector

MuleSoft

- Can connect to almost any JDBC relational database
  - Any database engine for which you have a driver
- To use
  - Add the Database module to your project
  - Add a database operation to a flow
  - Configure the connection to the database

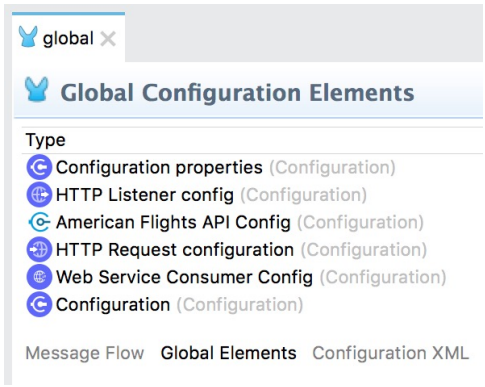


17

Global configuration elements

MuleSoft

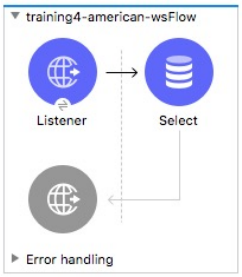
- For most operations, a lot of the configuration is encapsulated in a separate global element
  - A reusable configuration that can be used by many operations
  - Defines a connection to a network resource
- This is a connector configuration
  - Though it is sometimes referred to simply as the connector

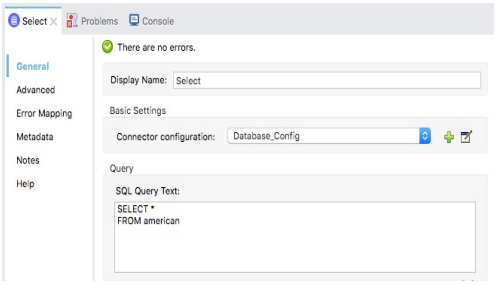


18

Walkthrough 4-2: Connect to data (MySQL database)

- Add a Database Select operation
- Configure a Database connector that connects to a MySQL database
  - Or optionally an in-memory Derby database if you do not have access to port 3306
- Configure the Database Select operation to use that Database connector
- Write a query to select data from a table in the database





19

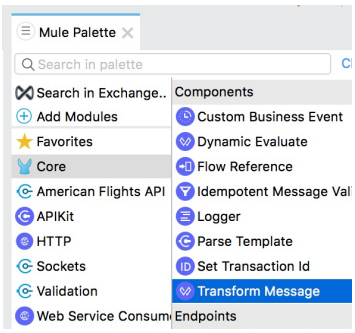
19

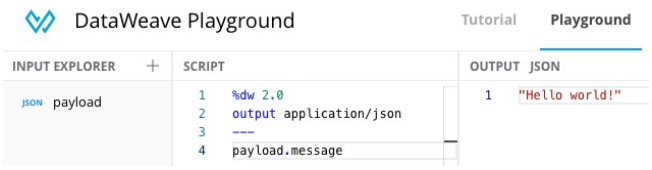
Transforming data

20

Transforming data

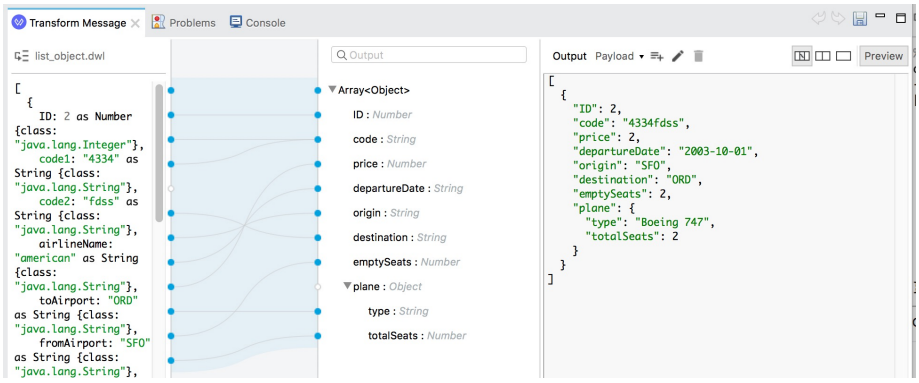
- **DataWeave 2.0** is the expression language for Mule to access, query, and transform **Mule 4** event data
  - DataWeave was introduced and used in Module 2
- In Studio, use the **Transform Message** component for transformations
  - Graphical interface with payload-aware development
- **DataWeave Playground**
  - Interactive browser environment for mocking
  - developer.mulesoft.com/learn/dataweave





Walkthrough 4-3: Transform data

- Use the Transform Message component
- Use the DataWeave visual mapper to change the response to a different JSON structure



Creating RESTful interfaces manually for Mule applications

23

Creating RESTful interfaces

MuleSoft

- A RESTful interface for an application will have listeners for each resource / method pairing defined by the API
  - GET: /flights
  - POST: /flights
  - GET: /flights/{ID}
  - DELETE: /flights/{ID}
  - PUT: /flights/{ID}

/flights

GET POST

/flights/{ID}

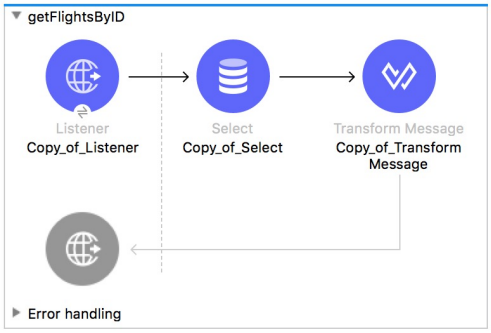
GET DELETE PUT

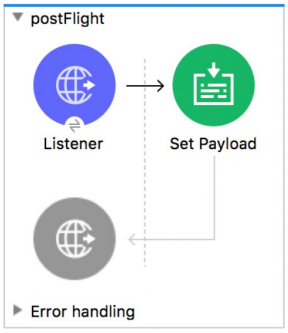
- You can create the interface manually or have it generated from the API definition
  - We will do both in the next two walkthroughs

24

Walkthrough 4-4: Create a RESTful interface for a Mule application

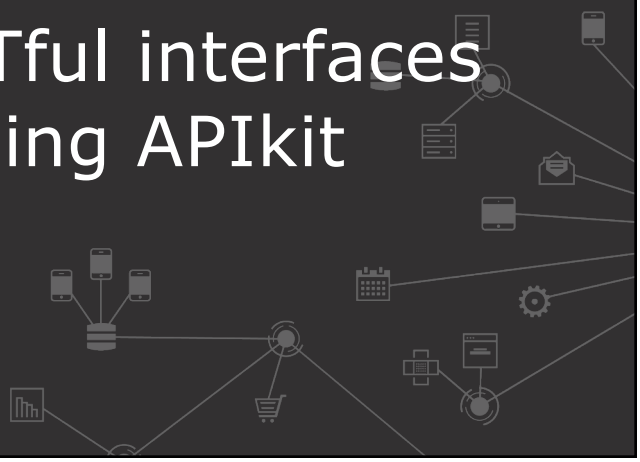
- Route based on path
- Use a URI parameter in the path of a new HTTP Listener
- Route based on HTTP method





25

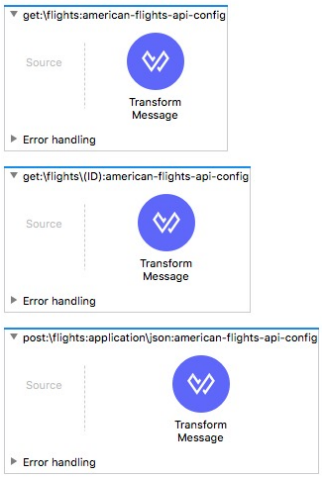
Generating RESTful interfaces automatically using APIkit



26

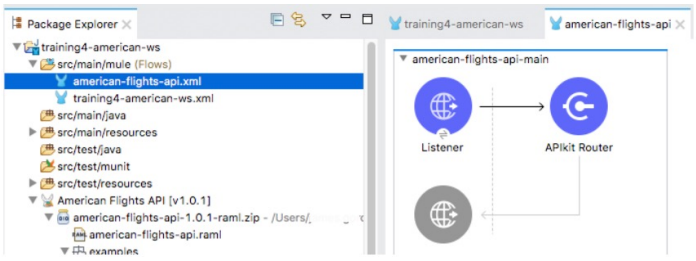
Creating RESTful interfaces automatically using APIkit

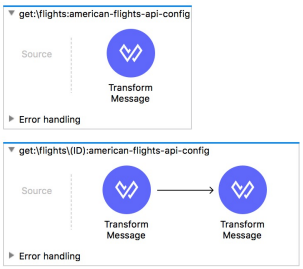
- **APIkit** is an open-source toolkit that includes an Anypoint Studio plugin
- The Anypoint Studio **APIkit plugin** can generate an interface automatically from a RAML API definition
  - For new or existing projects
- It generates a main routing flow and flows for each of the API resource / method pairs
- You add processors to the resource flows to hook up to your backend logic



Walkthrough 4-5: Use Anypoint Studio to create a RESTful API interface from a RAML file

- Add Anypoint Platform credentials to Anypoint Studio
- Import an API from Exchange into an Anypoint Studio project
- Use APIkit to generate a RESTful web service interface from an API
- Test a web service using APIkit console and Advanced REST Client





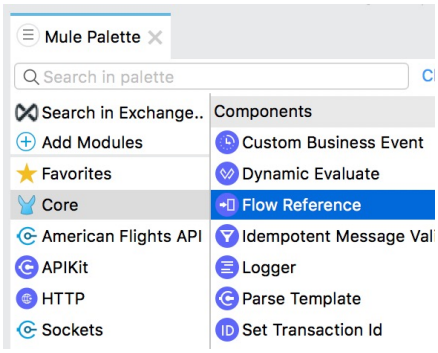
Connecting interfaces to implementations

29

Passing messages to other flows

MuleSoft

- Flows can be broken into multiple flows
  - Makes the graphical view more intuitive and the XML code easier to read
  - Promotes code reuse
- All flows are identified by name and can be called via **Flow Reference** components in other flows



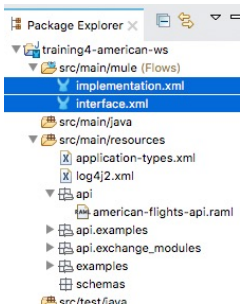
30

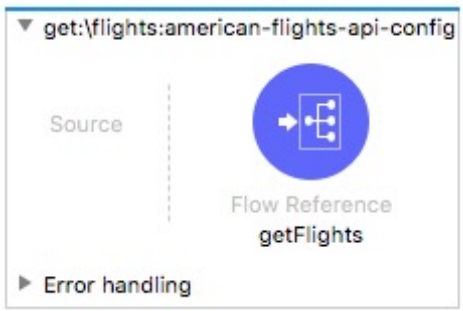
30

Walkthrough 4-6: Implement a RESTful web service

MuleSoft

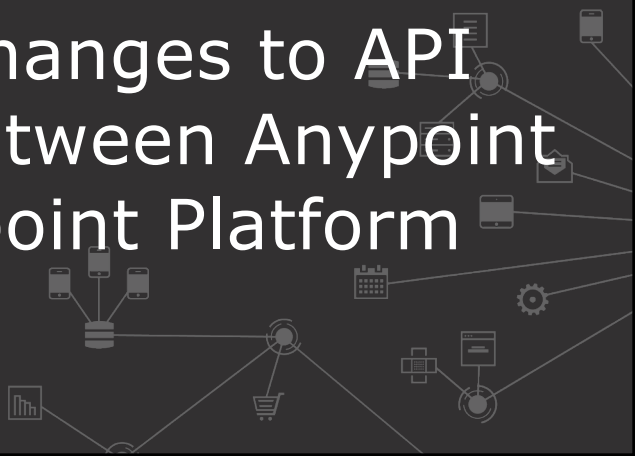
- Pass an event from one flow to another
- Call the backend flows
- Create new logic for the nested resource call
- Test the web service using Advanced REST Client





31

Synchronizing changes to API specifications between Anypoint Studio and Anypoint Platform



32

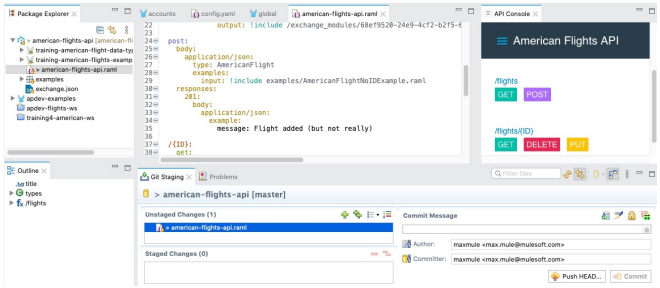
Synchronizing API specifications

- **API Sync** feature of Anypoint Studio enables you to
  - Pull specifications from Design Center into Studio
    - You already did this!
    - You can also initiate the creation of API specifications from scratch in Studio
  - Edit the specification offline in Anypoint Studio
  - Push the updates back to Design Center
  - Publish the new API asset version to Exchange
- This lets you develop Mule applications while following API lifecycle development practices from within Anypoint Studio
- If an API specification changes in Exchange, the generated API interface in Anypoint Studio can be updated
  - Flows that have already been modified are not overwritten

33

Handling change conflicts

- The **API Design perspective** includes a Git Staging tab



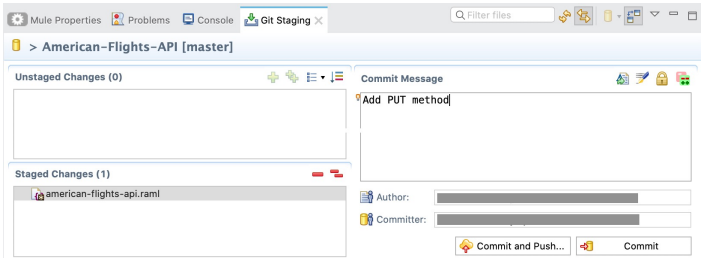
- Under the hood, a **Git** version control system is used to pull, push, and merge branches made to API specifications
- If someone modifies the version in Design Center while you are modifying the same version locally, a **conflict** is triggered in Git
  - You must tell Git how to apply your changes over a modified version

34

Walkthrough 4-7: Synchronize changes to an API specification between Studio and Anypoint Platform

MuleSoft

- Create an editable version of an API specification in Anypoint Studio
- Make changes to an API specification in Anypoint Studio
- Push the changes from Anypoint Studio to Design Center
- Publish the modified API specification from Studio to Exchange
- Update the version of an API specification used in a Mule project
- Rescaffold an API interface from an updated API specification



35

Summary

36

Summary

MuleSoft

- **Anypoint Studio** can be used to build Mule applications for integrations and API implementations
  - Two-way editing between graphical and XML views
  - An embedded Mule runtime for testing applications
- **Mule applications** accept and process events through a series of event processors plugged together in a flow
  - Use the **HTTP Listener** as an inbound endpoint to trigger a flow with an HTTP request
  - Use the **Set Payload** transformer to set the payload
  - Use the **Database** connector to connect to JDBC databases
  - Use DataWeave and the **Transform Message** component to transform messages from one data type and structure to another

37

Summary

MuleSoft

- Create RESTful interfaces for applications
  - **Manually** by creating flows with listeners for each resource/method pairing
  - **Automatically** using Anypoint Studio and **APIkit**
- Connect web service interfaces to implementations using the **Flow Reference** component to pass messages to other flows
- Synchronize changes to API specifications between Anypoint Studio and Anypoint Platform using **API Sync**

38