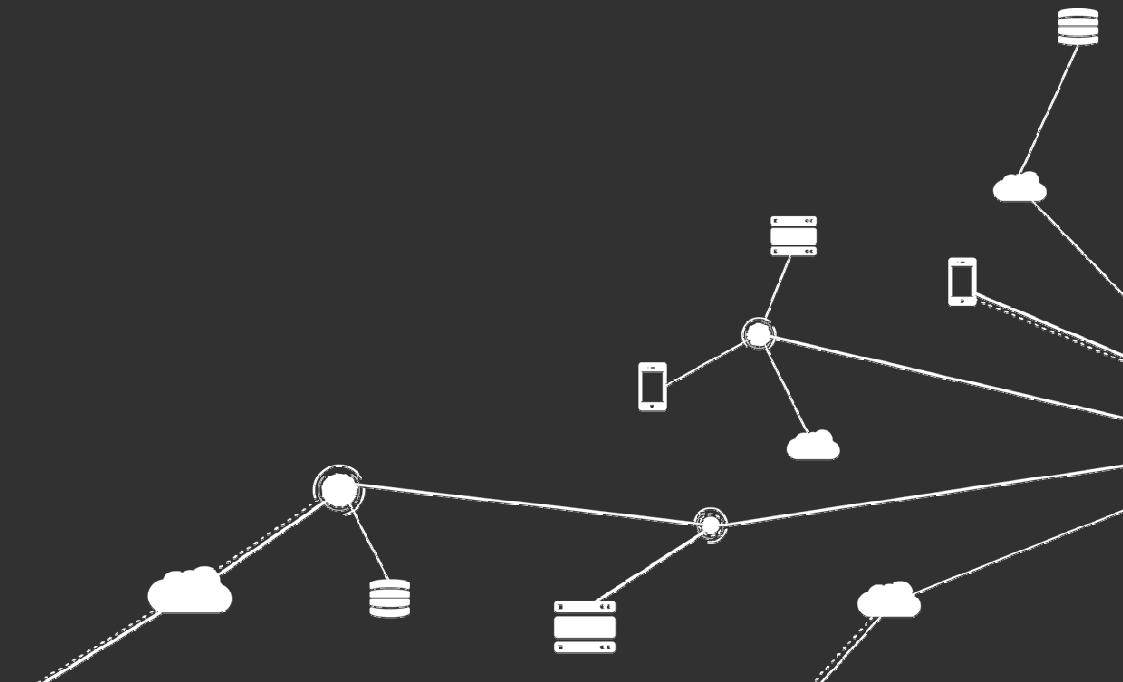




MuleSoft®

Module 10: Building RESTful Interfaces with RAML and APIkit



Goal

The screenshot displays the Anypoint Platform interface, specifically the API administration section for the "MUA Flights API - 1.0".

Left Panel (API Administration):

- Shows the RAML file for the "MUA Flights API".
- The RAML code defines a root resource "/flights" with a GET method that takes a query parameter "destination". The "example" section shows sample JSON responses for different airline filters.

Middle Panel (API Requests):

- A browser window shows a successful GET request to `localhost:8081/api/flights/PDX?airline=delta`, returning a JSON array of flight records.
- An MuleSoft flow diagram titled "get/flights/{destination}:mua-config" is displayed. It consists of the following components:
 - Source
 - Set Payload (with configuration "airlineName": "Delta")
 - getFlightsFlow (represented by a dashed box)
 - Logger

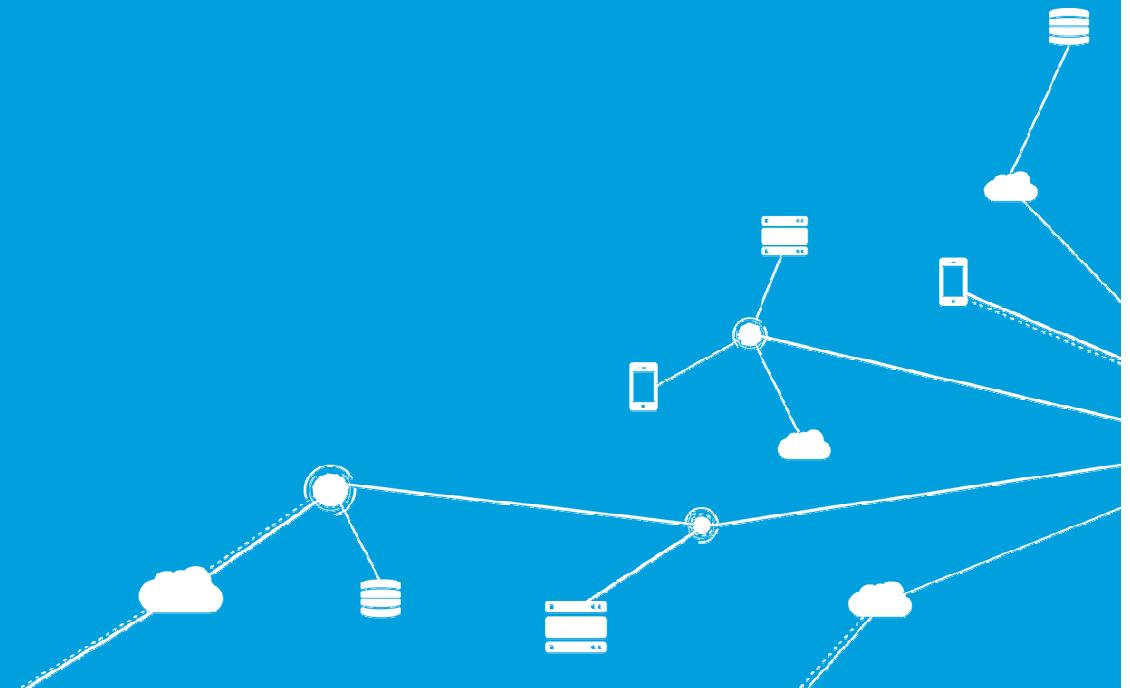
Bottom Right (MuleSoft Logo):

MuleSoft

Objectives

- In this module, you will learn:
 - To define an API with RAML
 - To create RAML files with Anypoint Designer
 - To implement a RAML file as a RESTful web service with Anypoint Studio and APIkit

Using API Designer to define APIs with RAML



RAML: RESTful API Modeling Language



- A simple and succinct way of describing RESTful APIs
 - Resources, schema, parameters, responses, and more
- Developed to help out the current API ecosystem
 - Encourages reuse, enables discovery and pattern-sharing, and aims for merit-based emergence of best practices
- A non-proprietary, vendor-neutral open spec
- Built on broadly-used standards such as YAML and JSON
 - YAML A'int a Markup Language
 - A human-readable data serialization format where data structure hierarchy is maintained by outline indentation
- <http://raml.org>

RAML example

```
1  #%RAML 0.8
2
3  title: World Music API
4  baseUri: http://example.api.com/{version}
5  version: v1
6  traits:
7      - paged:
8          queryParameters:
9              pages:
10                 description: The number of pages to return
11                 type: number
12      - secured: !include http://raml-example.com/secured.yml
13  /songs:
14      is: [ paged, secured ]
15      get:
16          queryParameters:
17              genre:
18                  description: filter the songs by genre
19      post:
20      /{songId}:
21          get:
22              responses:
23                  200:
24                      body:
25                          application/json:
26                              schema: |
```

RAML example: Resources and methods

```
8 mediaType: application/json
9 /flights: ← Resource
10 /{destination}: ← URI parameter / nested
11      get: ← Method returning JSON
12      description: Makes a call to ...
13      responses:
14          200: ← Describes a 200
15              body: response
16              application/json:
17                  Defines the example: | Example JSON
18                  response type [ {           }
19                      "key": "value"
20                  } ]
```

Anypoint Platform RAML tooling

- API Designer
 - A web-based API development tool for creating RAML API definitions
- API Manager
 - A web-based tool for making the APIs manageable and discoverable
- API Console
 - A graphical representation of the API that exposes the API's structure and provides access to interactive documentation
- API Notebook
 - A web-based persistent JavaScript console that enables live testing and exploring of APIs and saving API use cases

The API Designer

The screenshot shows the Anypoint Platform API Designer interface. On the left, there is a **File Browser** panel. In the center, the **Editor** panel displays the RAML code for the "Mythical Logistics Inc Shipping API". The code includes details about the API title, version, base URI, documentation, media type, and various resource definitions like /quote, /shipment, and /shipment/{shipment_id}. Below the editor is a **Shelf** panel containing links to protocols, base URI parameters, security options, resources, traits and types, and schemas. On the right, the **API Console** panel shows the "Mythical Logistics Inc Shipping API" with three listed endpoints: /quote (GET), /shipment (POST), and /shipment/{shipment_id} (GET). The MuleSoft logo is visible at the bottom right.

File Browser

API Designer

* /api.raml

```
1  #%RAML 0.8
2  title: Mythical Logistics Inc Shipping API
3  version: 1.0
4  baseUri: http://mythical.cloudhub.io/api/b2bshipping/{version}
5  documentation:
6    - title: Mythical Logistics Shipping Quote and Shipment API Documentation
7      content: Use this API to 1) get shipping quotes for your shipments
8        from mobile or web apps 2) initiate shipment requests from quote
9        (premium account holders must provide the corporate account number)
10       and 3) track the status of a shipment
11
12      mediaType: application/json
13
14      /quote:
15        get:
16          description: Get a live quote for your shipment package. Mobile apps
17          can use the 'current location' feature on the device to send the
18          optional GPS co-ordinates. If GPS co-ordinates are used, the shipment
19          'origin' port is looked up based on proximity.
20          queryParameters:
21            object:
22              displayName: Shipped object packaging
23              type: string
24              enum: ["Box", "Parcel", "Pallet", "Crate"]
25              description: The type of shipment - parcel/box/pallet/crate
```

ROOT (1) PARAMETERS (1) SECURITY (2)

protocols baseUriParameters secureday
Shelf securitySchemes

RESOURCES (1) TRAITS AND TYPES (2) SCHEMAS (1)

Mythical Logistics Inc Shipping API

expand / collapse all

/quote GET

/shipment POST

/shipment/{shipment_id} GET

DOCUMENTATION →

Mocking Service OFF

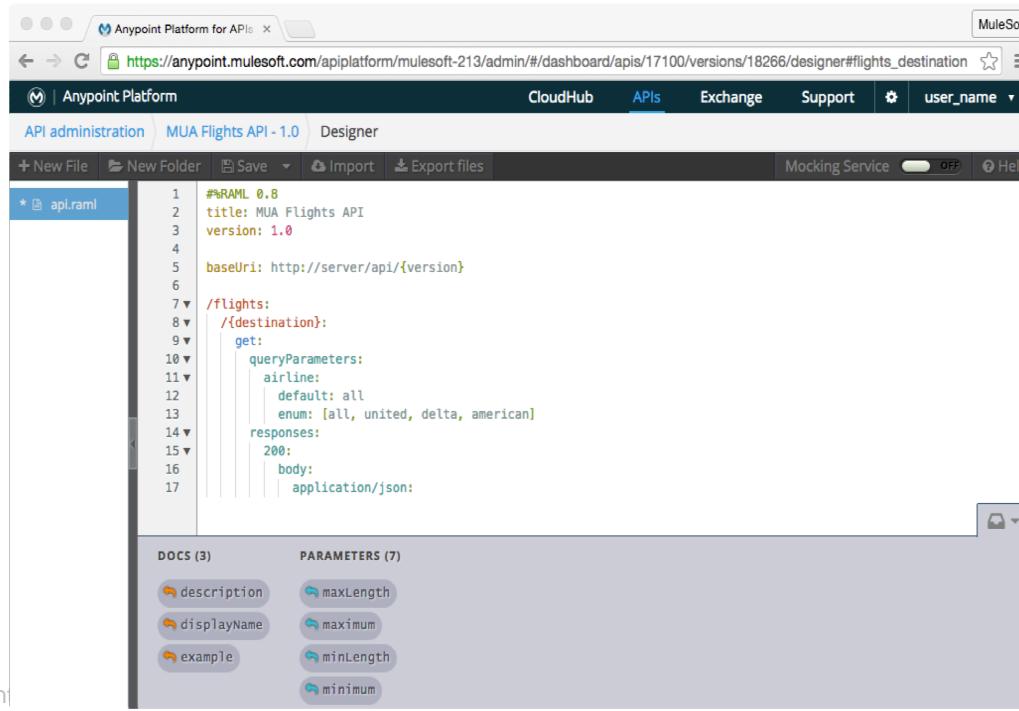
BETA

Powered by MuleSoft™

API Console

Walkthrough 10-1: Use API Designer to define an API with RAML

- In this walkthrough, you will:
 - Add a new API to the Anypoint Platform
 - Use the API Designer to create a RAML file
 - Use a nested resource for the destination and a query parameter for the airline



The screenshot shows the MuleSoft Anypoint Platform API Designer interface. The URL in the browser is https://anypoint.mulesoft.com/apiplatform/mulesoft-213/admin/#/dashboard/apis/17100/versions/18266/designer#flights_destination. The page title is "MUA Flights API - 1.0". The RAML code in the editor is:

```
#RAML 0.8
title: MUA Flights API
version: 1.0

baseUri: http://server/api/{version}

/flights:
  /{destination}:
    get:
      queryParameters:
        airline:
          default: all
          enum: [all, united, delta, american]
      responses:
        200:
          body:
            application/json:
```

Below the code editor, there are sections for "DOCS (3)" and "PARAMETERS (7)".

RAML resources

- RAML has a lot more to offer than our example in class
 - raml.org – Website
 - raml.org/docs – Documentation / tutorial
 - github.com/raml-org/raml-spec - Full spec
 - youtube.com/watch?v=5o_nExedezw – RAML overview with Uri Sarid (CTO of MuleSoft)

Simulating an API

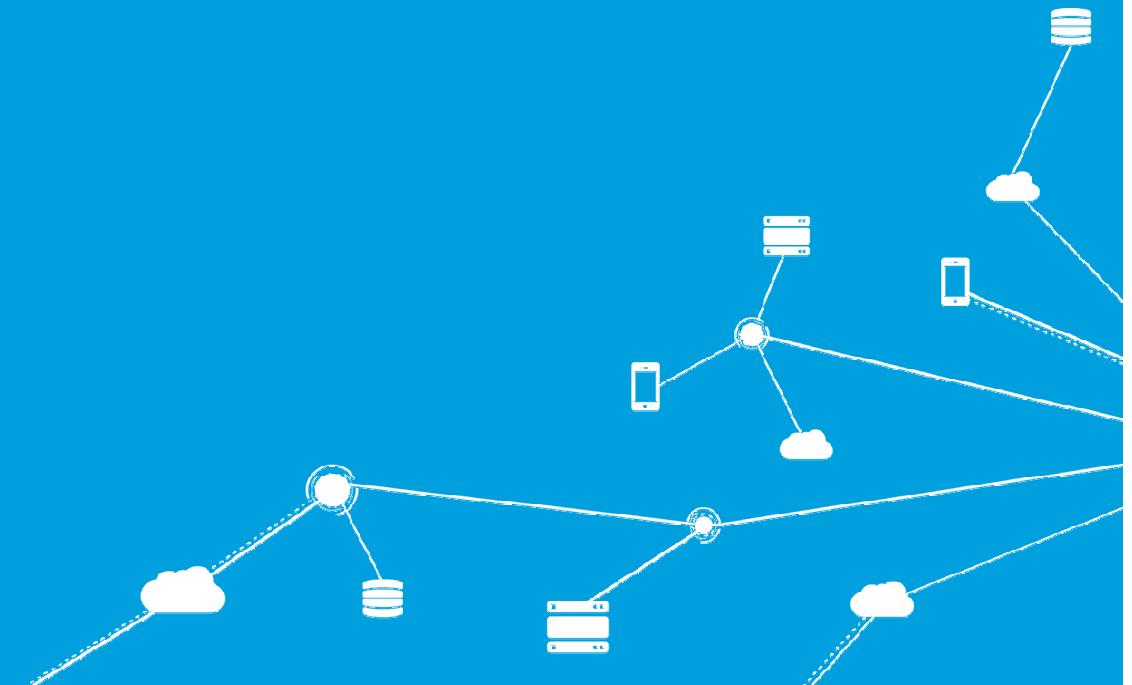
- You can use simulate an API to test it before it is implemented
 - Useful to get early feedback from developers
- Use the API console and the mocking service to run a live simulation
 - Use RAML to specify example responses for your API
 - Can be used in the API Designer and in API portals

Walkthrough 10-2: Use API Designer to simulate an API

- In this walkthrough, you will:
 - Use the API console in API Designer
 - Use RAML to specify example responses for your API
 - Use the API Designer mocking service to run a live simulation of your API

The screenshot shows the MuleSoft Anypoint Platform for APIs Designer interface. On the left, there is a code editor window titled "mua.raml" containing RAML 0.8 code. The code defines a "flights" resource with a "get" method that takes a query parameter "destination". The "responses" section for the "200" status includes an "example" with a JSON object representing a flight. On the right, the "Request URL" is displayed as <http://mocksvc.mulesoft.com/mocks/5abf473c-7577-4735-9ef8-de90b0b09003/flights/SFO?airline=all>. Below the URL, the "Response" pane shows the "Status" (200), "Headers", and "Body". The "Body" pane displays the JSON response from the mock service, which includes fields like "airlineName", "price", "departureDate", "planeType", "origin", "code", "emptySeats", and "destination".

Using Anypoint Studio to create RESTful web services from RAML files

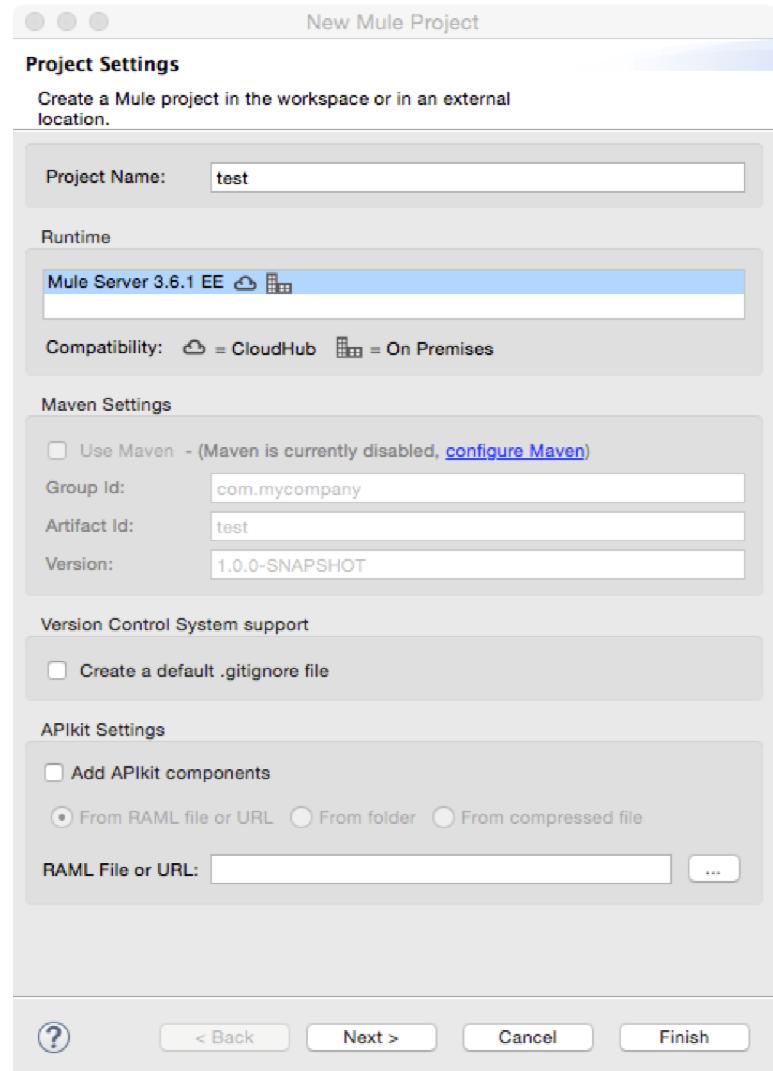


Creating RESTful web services

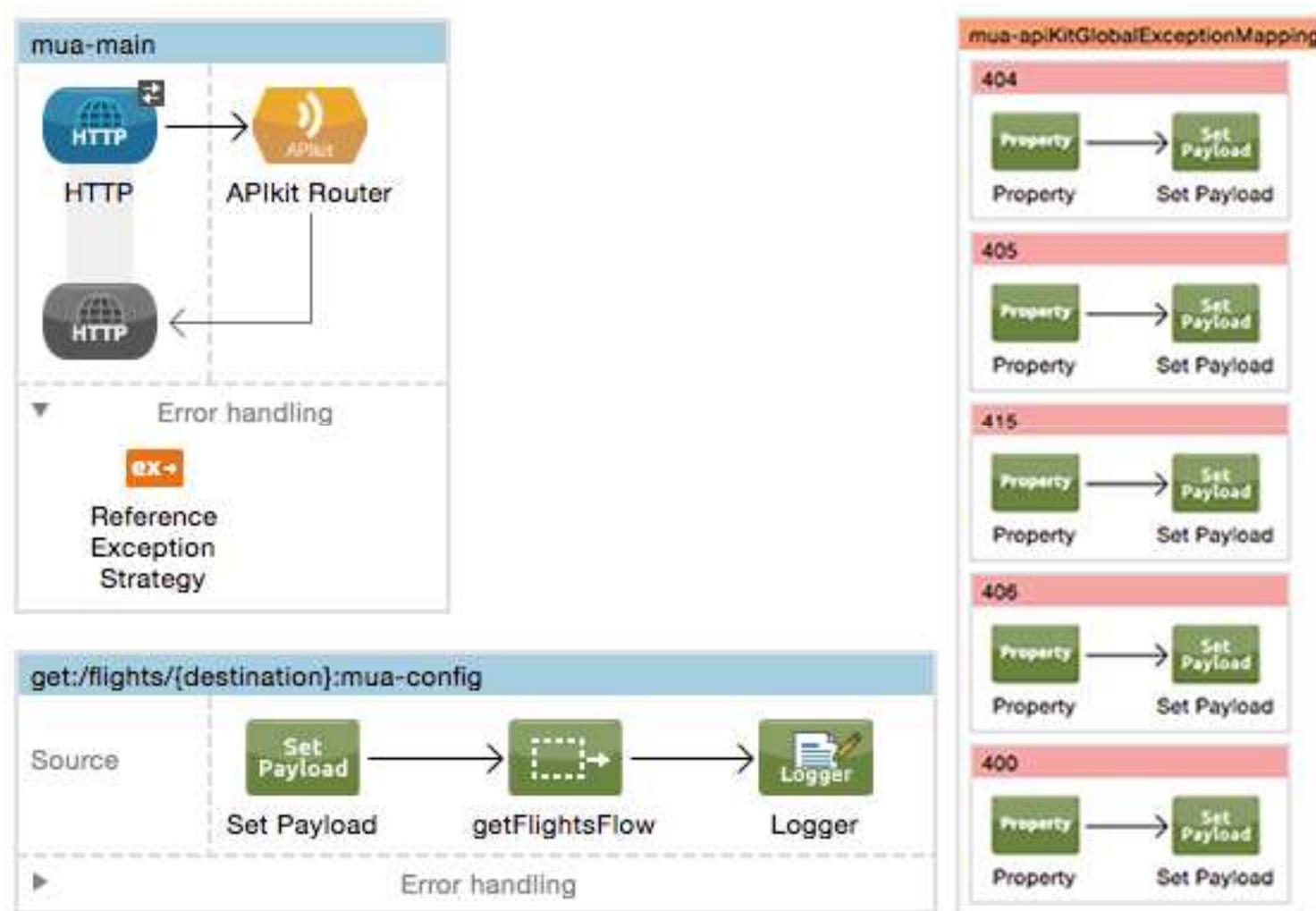
- Use Anypoint Studio and APIkit to generate a RESTful web services from a RAML file
- APIkit is an open-source, declarative toolkit created to facilitate REST API implementation
 - Integrated with Anypoint Studio to create a RESTful interface based on a RAML file
 - It generates a main routing flow and flows for each of the API resources
 - You add processors to the resource flows (often Flow References) to hook up to your backend logic

Using Anypoint Studio to create RESTful interfaces

- In a new project
 - Specify a RAML file when you create the project
- In an existing project
 - Add a RAML file to the project
 - Right-click and select APIkit > Generate Flows



The generated application

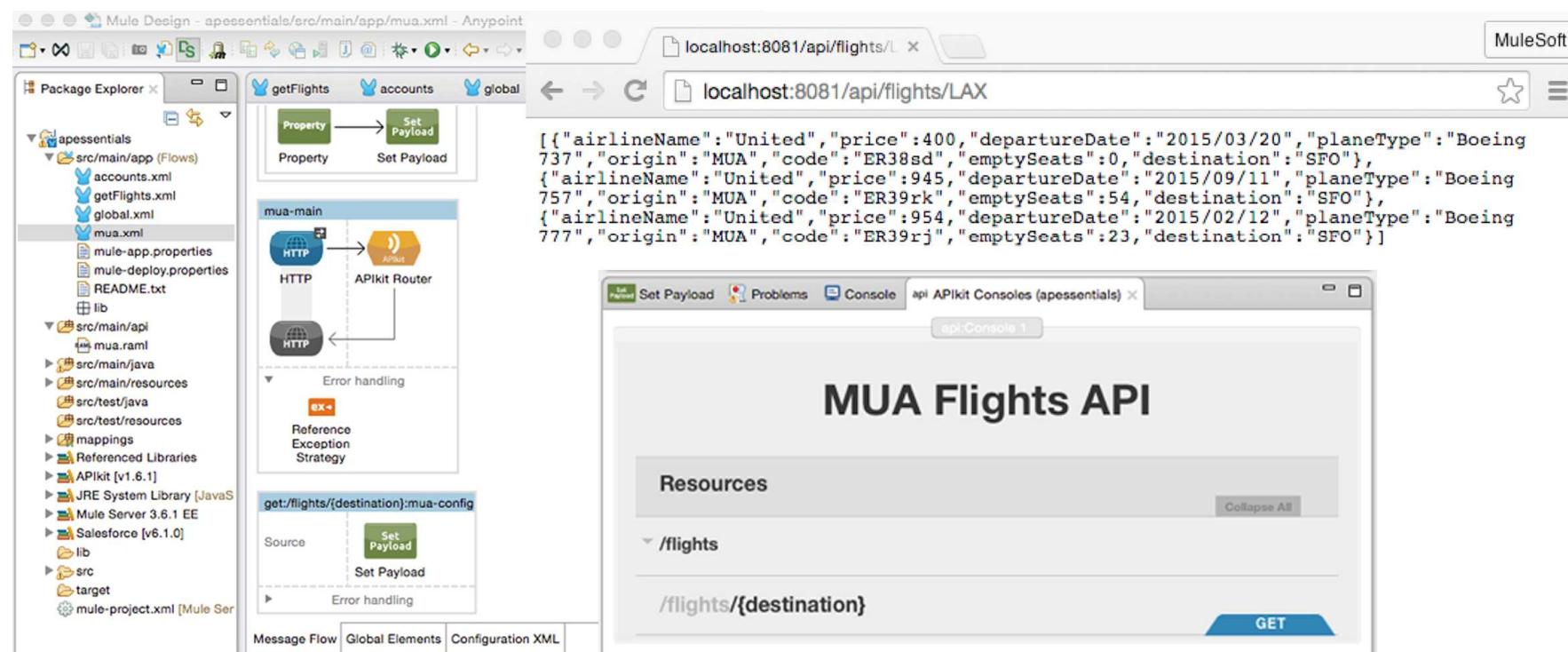


APIkit anatomy

- Three main parts that together form a RESTful API
 - Interface
 - The mediator between the service exposed to the world and the internal assets that need to be exposed
 - Defines the data to which end users have access, and specifies the actions against the data (GET, PUT, etc.)
 - Main flow
 - Exposes the API via HTTP or Jetty
 - Routes requests between the interface (defined in RAML) and the backend flows (defined in XML)
 - References exception strategies specially designed to produce HTTP-status-code-friendly responses
 - Backend flows
 - Do the "heavy lifting", performing the actions the interface defined

Walkthrough 10-3: Use Anypoint Studio to create a RESTful interface from a RAML file

- In this walkthrough, you will:
 - Add a RAML file to your apessentials project
 - Use Anypoint Studio and APIkit to generate a RESTful web service interface from a RAML file
 - Test the web service in the APIkit Consoles view and a browser



Walkthrough 10-4: Use Anypoint Studio to implement a RESTful web service

- In this walkthrough, you will:
 - Split the getFlightsFlow into two flows: one that transforms the form post data into a FlightRequest object and the other that gets all the flights
 - Determine how to reference the web service destination and airline values
 - Call the backend flow
 - Test the web service in the APIkit Consoles view and a browser

The screenshot shows a browser window and a MuleSoft Anypoint Studio interface.

Browser View:

- Address bar: localhost:8081/api/flights/
- Address bar: localhost:8081/api/flights/PDX?airline=delta
- Content: A JSON array of flight records. One record is highlighted with a red box and a cursor, showing: {"departureDate": "2015/02/20", "airlineName": "Delta", "destination": "PDX", "price": 283.0, "planeType": "Boing 777", "code": "AFFFC4", "origin": "MUA", "emptySeats": 30}, {"departureDate": "2015/02/12", "airlineName": "Delta", "destination": "PDX", "price": 283.0, "planeType": "Boing 777", "code": "A1B3D4", "origin": "MUA", "emptySeats": 10}, {"departureDate": "2015/02/13", "airlineName": "Delta", "destination": "PDX", "price": 283.0, "planeType": "Boing 777", "code": "A1FGF4", "origin": "MUA", "emptySeats": 80}

Anypoint Studio Flow:

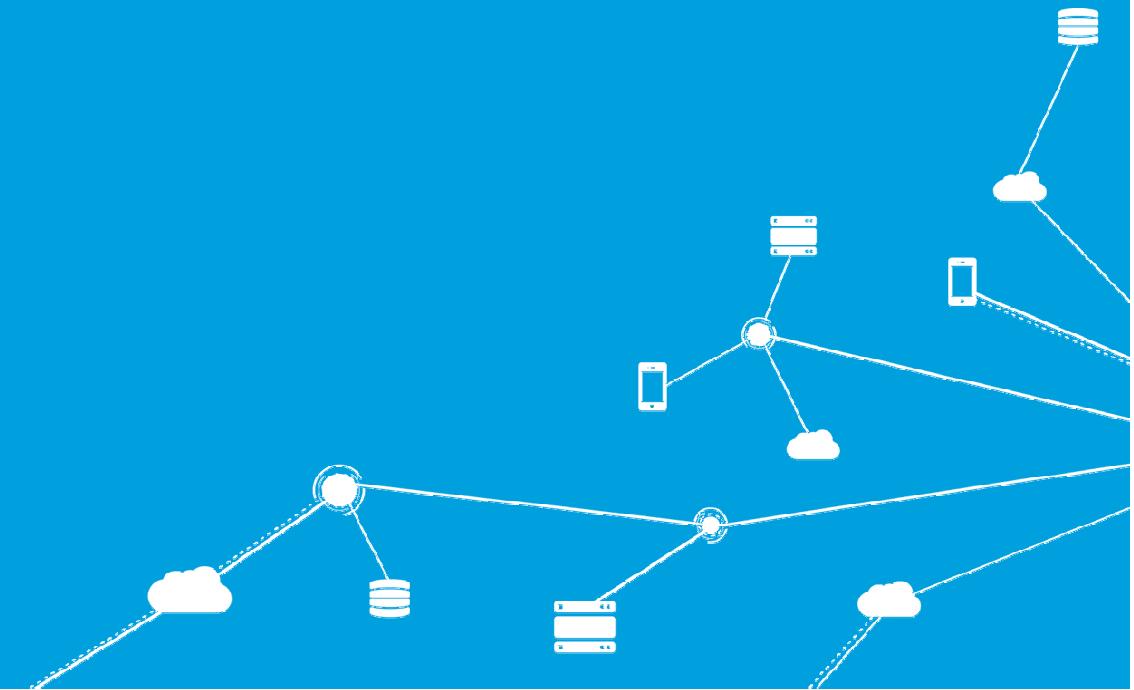
```
graph LR; Source[Source] --> SetPayload1[Set Payload]; SetPayload1 --> getFlightsFlow1[getFlightsFlow]; getFlightsFlow1 --> Logger1[Logger];
```

The flow consists of the following components:

- Source
- Set Payload (highlighted with a red circle)
- getFlightsFlow (highlighted with a red box and a cursor)
- Logger

Below the flow, there is an "Error handling" section.

Summary



Summary

- In this module, you learned to build a RESTful web service from a RAML API definition with Anypoint Platform for APIs
- Anypoint Platform for APIs includes web applications, tools, and frameworks for designing, testing, building, and managing RESTful APIs
 - API Designer, Anypoint Studio, APIkit, API Portal, API Manager
- Anypoint Platform RAML tooling includes API Designer, API Console, API Manager, and API Notebook
- Use the Anypoint Designer and API Console to define and simulate an API with RAML

Summary

- Use Anypoint Studio and APIkit to generate a RESTful web services from a RAML file
- APIkit is an open-source, declarative toolkit
 - Created to facilitate REST API implementation
 - Integrated with Anypoint Studio to create a RESTful interface based on a RAML file
 - It generates a main routing flow and flows for each of the API resources
 - You add processors to the resource flows (often Flow References) to hook up to your backend logic