Contents

Module 1

Fundamentals – Review++

WT 1-1 Import a basic Mule project into Anypoint Studio

Import the starter project

1. Start Anypoint Studio
2. Create a new workspace
3. Import the `apdw2-flights-starter.jar` project under the `studentFiles/mod01`

Create new project

4. Create a new project
   Creating a new project and copying only the files you minimally need for the class helps in containing the "noise" that is introduced with starter project. Additionally, there is the extra benefit of not having to deal with students who are having compilation issues with the starter project.
5. Create a new project and call it dataweave
6. From the apdw2-flights-starter copy the following files over to the new project:
   (a) `/main/resources/airportInfoTiny.csv` to `src/main/resources`
   (b) `src/main/resources/examples/mockdata/deltaSoapResponsesToAllDestinations.xml` to `src/test/resources`
   (c) `src/test/resources/flight-example.json` to `src/test/resources`

WT 1-2 Fundamentals review++

In this WT the goal is to attempt (I am saying attempt because often enough we have participants who don't meet the prerequisites) to bring everyone at the same level by (1) reviewing fundamentals and (2) illustrating features of DW that we will be using throughout the class

Create the flow, set the metadata

1. Rename the `dataweave.xml` to `mod1.xml`
2. Create a new flow named `mod1-review++`
   The reason for prefixing the flow name with the name of the flow is a best-practice one. Such a convention will improve the readability of your flows by identifying the Mule Configuration file a flow is defined under by just looking at a Flow Reference's display name.
3. Drop a DW (aka `Transform Message`) to the process area of the flow
4. Define the payload input metadata to the `src/test/resources/flight-example.json`, set the name of the type to `flight_json`
5. Edit the sample data
6. Turn on the preview
7. Change the output to JSON

Construction

Field access

String concatenation

Expression chaining

Conditional expressions

Array access and Ranges

Common functions we will be using

Transform XML to JSON

Transform JSON to XML

Module 2

Variables, Functions, Modules

WT 2-1 **Organize DataWeave code with variables and functions**

WT 2-2 **Reuse DataWeave transformations**

WT 2-3 **Create and use DataWeave modules**

Module 3

Defensive programming

Module 4

Operating on Arrays and Objects

Module 5

The Arrays and Objects Modules

Module 6

Flights and Airports

Module 7

Recursion