Anypoint Platform Development: DataWeave 2 (Mule 4)

Summary
This course is for developers interested in advancing their DataWeave 2 skills beyond those taught in the Anypoint Platform Development - Fundamentals (Mule 4) course.

Duration
2 days (in-person or online)

Objectives
At the end of this course, students should be able to:

- Write generalized and reusable transformations using variables, functions, DataWeave modules and mappings, and dynamic evaluation components.
- Build complex transformations from smaller testable steps.
- Build more robust and testable functions and expressions using strong typing, match operators, error handling, and logging.
- Create, transform, filter, combine, shuffle, select from, and reduce complex data structures that include nested arrays, objects, and arrays of objects.
- Recursively replace or format every element or a list of elements in a nested schema.

Audience
Mule 4 developers or architects who want to advance their DataWeave 2 skills so that they can build complex transformations

Prerequisites
- Experience developing Mule 4 applications as demonstrated by one of the following:
  - Passing the MuleSoft Certified Developer – Level 1 (Mule 4) exam
  - Completion of the Anypoint Platform Development: Fundamentals (Mule 4) course
  - Completion of the Anypoint Platform Development: Mule 4 for Mule 3 Users course
- A basic knowledge of functional programming

  Note: If new to functional programming, read An introduction to functional programming in JavaScript.

Setup requirements
- A computer with at least 8-16 GB (16 highly recommended) available RAM, 2GHz CPU, and 10GB available storage
- Internet access to port 80 (with > 5Mbps download and > 2Mbps upload)
• Anypoint Studio 7.7.0 or later with embedded Mule 4.3 runtime
• Advanced REST Client (or any other REST client application)
• (Optional) If no internet access to ports 80 and 3306, OpenJDK 8 (not 11 or a later version)

Get a detailed setup document here.

Outline

Module 1: Transforming data using metadata
• Apply DataWeave fundamentals as learned in the Development Fundamentals course
• Configure metadata for DataWeave transformation input and output
• Set example input for DataWeave transformations

Module 2: Organizing DataWeave code with variables and functions
• Organize DataWeave code into variables and functions
• Pass functions and lambda expressions as parameters to other DataWeave functions
• Chain DataWeave functions together
• Create and use reusable DataWeave modules
• Write more robust functions using the match operator to test for data types

Module 3: Constructing arrays and objects
• Add components to and remove elements from arrays and objects
• Construct objects from lists of DataWeave expressions using object constructor curly braces { }
• Troubleshoot common issues when using object constructor curly braces { }

Module 4: Iteratively transforming data using mapping operators
• Transform elements of arrays into a new array using the map operator
• Transform elements of objects into a new object using the mapObject operator
• Combine map and mapObject operators to transform complex schema
• Extract an array of keys and/or values from an object using the pluck operator
• Reduce and accumulate array elements to other output types using the reduce operator

Module 5: Recursively transforming complex structures
• Write recursive functions to transform complex schema
• Replace keys or values at any level of a nested data structure using a lookup object