Anypoint Platform Development: Production-Ready Integrations (Mule 4)

Summary

This course is for developers who have mastered the fundamentals of creating Mule applications with Anypoint Studio and Anypoint Platform™ and now want to learn the skills and best practices to implement production-ready integrations — applications that apply essential integration patterns to take on the production challenges of distributed communication.

Note: Prior completion of the Anypoint Platform Development: Production-Ready Development Practices course is recommended but not required.

Duration

4 days (in-person or online)

Objectives

At the end of this course, students should be able to use best practices to:

- Invoke REST APIs and SOAP web services using various client components taking into consideration the non-functional properties of API invocations.
- Pass messages asynchronously reliably between Mule flows and Mule applications.
- Use various modules and techniques to implement message validation across Mule flows and Mule applications.
- Apply essential Enterprise Integration Patterns to orchestrate multiple Mule applications and API invocations.
- Use various techniques and Anypoint Platform components to store and manage state in an application network to increase performance and resilience.
- Identify and extract reusable Mule application code into different Mule runtime extensions.

Audience

Software developers who have mastered the fundamentals of creating Mule applications with Anypoint Studio and Anypoint Platform and now want to learn the skills and best practices to implement production-ready integrations that address the challenges of distributed communication

Prerequisites

- Experience developing Mule 4 applications as demonstrated by one of the following:
    - Attainment of the *MuleSoft Certified Developer – Level 1 (Mule 4)* certification
    - Completion of the *Anypoint Platform Development: Fundamentals (Mule 4)* course
    - Completion of the *Anypoint Platform Development: Mule 4 for Mule 3 Users* course
- Experience using Maven to build Mule 4 applications

  Note: If you do not have Mule Maven experience, you should first complete the Anypoint Platform Development: Production-Ready Development Practices course.

Setup requirements

- A computer with:
    - At least 8-16 GB (16 highly recommended) available RAM, 2GHz CPU, and 10GB available storage
    - A minimum screen resolution of 1024x768
- Internet access to ports 80 and 443 (with > 5Mbps download and > 2Mbps upload)
- The latest version of Chrome, Safari, Firefox, or Edge
- An Anypoint Platform account
- OpenJDK 8 (not 11 or a later version)
- Anypoint Studio 7.11.0 or later with embedded Mule 4.4 runtime
- Advanced REST Client 16.0.1 or later (or any other REST client application)
- Apache Maven 3.5.4 or later

Get a detailed setup document here.

Outline

Module 1: Invoking web APIs and services

- Invoke HTTP APIs using the HTTP connector paying attention to the non-functional properties of API invocations
- Invoke HTTP APIs with an API specification using REST connectors
- Improve the non-functional properties of API invocations with REST connectors
- Enable an API client for OAuth 2.0
- Invoke a SOAP web service with full control over TLS certificates
- Implement an HTTP callback

Module 2: Passing messages asynchronously

- Publish messages to a VM queue
- Listen for messages in a VM queue
- Publish messages to an Anypoint MQ exchange
- Subscribe to messages in an Anypoint MQ queue

Module 3: Validating messages

- Assert flow pre/post conditions and invariants
- Validate XML messages
- Validate JSON messages

Module 4: Orchestrating integration functionality

- Parallelize integration logic with the Scatter-Gather router
- Trace transactions across an application network using correlation IDs
- Retry failed API invocations

Module 5: Storing objects for persistence, performance, and resilience

- Temporarily persist data in an Object Store
- Avoid expensive operations with the Cache scope
- Apply a caching API policy to an API implementation

Module 6: Componentizing reusable integration functionality

- Create an XML SDK component
- Create a custom API policy