Anypoint Platform Development: Production-Ready Development Practices (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 Mule applications — applications that are ready to be used in a DevOps environment in professional software development projects. This includes securely provisioning API-related artifacts; automating the building, unit testing, and deployment of Mule applications; making Mule applications easily monitored; sharing development assets and resources between projects; and more.

Duration

3 days (in-person or online)

Objectives

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

- Interactively provision API-related artifacts and secure access to APIs.
- Enhance data protection using HTTPS.
- Configure Mule applications succinctly and securely for different deployment environments.
- Automate the building, unit testing, and deployment of Mule applications using Maven-based tooling.
- Write unit tests for Mule flows using MUnit that is optimized for reuse and maintainability.
- Improve monitorability of Mule applications through healthchecks and operational logging.
- Share assets and resources including code, connector configurations, and more between projects.

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 build production-ready Mule applications for professional software development projects

Prerequisites

- Experience developing Mule 4 applications as demonstrated by one of the following:
    - Passing the 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 solid understanding of essential Maven concepts

  Note: If you are new to Maven (a build automation tool), you need to learn Maven fundamentals BEFORE taking this course. Suggested tutorials include Apache Maven Tutorial, Maven in 5 Minutes, and Maven Getting Started Guide.

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: Interactively provisioning API-related artifacts

- Recap the fundamental API-related workflows in Anypoint Platform
- Export, import, and publish an API specification (OAS will be used)
- Manage an API instance in API Manager
- Implement an API as a Mule application
- Expose an HTTPS endpoint from a Mule application
- Register an API implementation to also be the API proxy using autodiscovery

Module 2: Applying basic software engineering principles

- Apply and follow coding conventions
- Securely parameterize Mule applications and its Maven build for different runtime environments
- Manage complex Maven dependency relationships of Mule applications
- Use Maven to automate application deployment to CloudHub
- Implement operational logging
- Expose healthcheck endpoints and monitor Mule applications from Anypoint Platform
- Extract reusable Mule application code into libraries

Module 3: Automating unit testing with MUnit

- Enable Mule applications for unit testing with MUnit

- Perform basic unit testing of integration functionality
- Mock external dependencies
- Spy on the data exchanged with external dependencies

Module 4: Sharing resources within domains

- Share libraries within a Mule runtime domain
- Share an HTTP Listener configuration and related resources within a Mule runtime domain