

**MuleSoft Development: A Comprehensive Foundation**

**Course Number:** MULE-100
**Duration:** 5 days

**Overview**

This live, online or in-person introductory MuleSoft training course is for anyone brand new to MuleSoft who would like a comprehensive hands-on introduction. Attendees learn how to build a system API and use it to create an API implementation. Students use Anypoint Platform components to deploy, secure, and manage APIs. Attendees also connect to databases, consume web services, and process large volumes of data using the batch features of MuleSoft.

**Prerequisites**

Some experience in XML and JSON data formats would be helpful but is not required.

**Materials**

All students receive comprehensive courseware.

**Software Needed on Each Student PC**

Students will not need to install any software on their computers for this class. The class will be conducted in a remote environment. Students need a local computer with a web browser (preferably Chrome), stable internet, two monitors, and a headset/microphone.

**Objectives**

* Define and explain the usage of application networks and API-led connectivity
* Explore Anypoint Platform and the full lifecycle of an API
* Distinguish between SOAP and RESTful web services
* Consume APIs using clients, status codes, resources, methods, and responses
* Identify JSON and XML data formats
* Define the functional elements of an API specification including headers, traits, examples, data types, security, resources, methods, and responses
* Create and document an API specification using Anypoint Design Center
* Test API specifications using API console and mocking service
* Make API specifications discoverable through Anypoint Exchange
* Build APIs and integrations using Anypoint Studio
* Transform data using DataWeave
* Build an API implementation and deploy it to CloudHub
* Create an API Proxy for your application using Anypoint Platform’s API Manager
* Apply policies and restrict access to your API from API Manager
* Organize processors and mule applications, and add metadata
* Connect APIs together to consume web services
* Use the Choice router, the Scatter-Gather router, and Validator module
* Connect systems to read and write files and records
* Process elements of records using For Each scope and Parallel For Each scope
* Evaluate transform load using the Batch Job scope
* Predict the error flows and error messages within a Mule application

**Outline**

* Packing Your Mule
	+ Introduction to MuleSoft
	+ API-led Connectivity
	+ Anypoint Platform
	+ API Lifecycle and Reusability
	+ Web Services and APIs
	+ JSON and XML
	+ REST and SOAP
	+ Consuming APIs
	+ URIs and Query Parameters
	+ Secured and Unsecured APIs
* Hello RAML
	+ RAML Structure and Syntax
	+ Diagram a RAML Specification
	+ Articulate the User Story
	+ Create an API Project
	+ Design the API Specification
	+ Test using API Console
	+ Publish to Exchange
	+ Document APIs in Exchange
	+ Share APIs to the Public Portal
* Welcome to Studio
	+ Navigate Anypoint Studio
	+ Create a Mule Project
	+ Design the Implementation
	+ Transform Data with DataWeave
	+ Scaffold the Interface using APIKit
	+ Link the Interface to the Implementation
	+ Update the RAML from Studio
	+ Sync Changes with Design Center
* Integrate with Mule
	+ Deploy to CloudHub from Studio
	+ Monitor on Runtime Manager
	+ Observe Deployment/Worker Logs
	+ Use API Manager to Apply Policies
	+ Add Service Level Agreement Tiers
	+ Add Client ID Enforcement
	+ Update RAML with Security Trait
	+ Update Version and Redeploy Proxy
* Methodical Mules
	+ The Persistence of Mule Event Data
	+ Observe Mule Event Data
	+ Create Flow, Subflows, Private Flows
	+ The HTTP Request
	+ The Flow Reference and VM Connector
	+ Synchronous and Asynchronous
	+ Create a Global.xml Configuration File
	+ Create a Properties File
	+ Parameterize Application Properties
	+ Organize Files and Folders
	+ Manage Metadata: Package Explorer
	+ Metadata in application-types.xml
* Connecting APIs Together
	+ Consume Web Services Individually
	+ Consume using a REST Connector
	+ Consume using an HTTP Connector
	+ Consume a SOAP Web Service
	+ Consume Web Services Collectively
	+ Route Based On Conditions
	+ Run All Routes at the Same Time
	+ Use the Validation Module
* DataWeave
	+ DataWeave in Studio
	+ DataWeave Structure
	+ DataWeave Coding
* Event-Driven Processors
	+ Reading and Writing Files
	+ Use a File Listener
	+ Use a Database Listener
	+ Use Watermarking and The Object Store
	+ Use Asynchronous Messaging
* Processing Collections
	+ Processing Elements
	+ Use the For Each Scope
	+ Use the Parallel For Each Scope
	+ Compare and Contrast
	+ Evaluate Transform Load
	+ The Batch Job Scope
	+ Variable Persistence
	+ Filter and Aggregate
	+ The On Complete Phase
* Error Handling
	+ Mule Default Error Handler
	+ Error Object Properties
	+ Types of Errors
	+ HTTP Default Settings
	+ The Application Level
	+ Global Error Handler
	+ On-Error Propagate On-Error Continue
	+ Create and Configure Application-Level Error Handlers
	+ Predict Flow and Error Message Response
	+ Modify HTTP Default Settings
	+ The Flow and Processor Levels
	+ Handle Specific Errors at Flow Level
	+ Global Error Handler Ignored?
	+ The Try Scope
	+ Create and Configure Error Handlers
* Conclusion
	+ Tactical Mission Summary
	+ Earning iQ Points
	+ A Quest Built for You
	+ Your Next Quest