

**API Design using MuleSoft**

**Course Number:** MULE-104
**Duration:** 1.5 days

**Overview**

This live, online MuleSoft API Design training course teaches attendees how to diagram, analyze, and create effective API specifications using RAML to support the design-first approach to API development.

**Prerequisites**

Students must be acquainted with API-led connectivity and the REST architectural style. Some experience with RAML or OAS is helpful but not required.

**Materials**

All students will 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**

* Diagram an API specification identifying resources, methods, parameters, responses, and data types
* Define the functional elements of an API specification including headers, traits, examples, data types, security, resources, methods, and responses
* Create an API specification in RAML 1.0 using Anypoint Design Center
* Document API specifications using display names and descriptions
* Test API specifications using API console and mocking service
* Make API specifications discoverable through Anypoint Exchange
* Semantic versioning for APIs

**Outline**

* Hello RAML
	+ RAML Structure and Syntax
	+ Diagram RAML Specifications
* Planning Our API
	+ Deconstruct API Requirements
	+ Articulate the User Story
	+ Create an API Project
	+ Design the API Specification
	+ Test using API Console
* Planning Your API
	+ Deconstruct API Requirements
	+ Articulate the User Story
	+ Create an API Project
	+ Design the API Specification
	+ Test using API Console
* RAML Fragments
	+ Discover, import, and reuse
	+ Create, publish, import, and use
* Publish Both APIs to Exchange
	+ Test Using API Console
	+ Document APIs in Exchange
	+ Share APIs to Public Portal
* Versioning APIs
	+ Semantic Versioning for APIs
	+ Update Both APIs
	+ Publish Changes to Exchange
	+ Update Portal Documentation
* Conclusion