Mulesoft Developer Course

Fundamentals

Jagadishwar Reddy

Contents

[Prerequisites 2](#_Toc30446013)

[Software Required to Complete this Course 2](#_Toc30446014)

[Introduction 3](#_Toc30446015)

[Concept of Integration (or) Middleware 3](#_Toc30446016)

[Leading Technologies for Integration (or) Middleware 3](#_Toc30446017)

[Types of Integrations 3](#_Toc30446018)

[What is Server-Client Model 4](#_Toc30446019)

[What is a Webservice 4](#_Toc30446020)

[Types of Webservices 4](#_Toc30446021)

[What is Mulesoft 5](#_Toc30446022)

[What is Anypoint Platform 5](#_Toc30446023)

[Tools Used in Mulesoft 5](#_Toc30446024)

[Why Mulesoft? 5](#_Toc30446025)

[Who Uses Mulesoft? 6](#_Toc30446026)

[What is Mulesoft Development Lifecycle? 6](#_Toc30446027)

[Design 8](#_Toc30446028)

[RAML: What is RAML? 8](#_Toc30446029)

[RAML: What is the Purpose of RAML? 8](#_Toc30446030)

[RAML: Where do you create RAML and How do you create RAML? 8](#_Toc30446031)

[RAML: What are traits? 10](#_Toc30446032)

[RAML: What are Schemas and how do you use them? 10](#_Toc30446033)

[RAML: What are types? 10](#_Toc30446034)

[RAML: What is the difference between Types/Schema? 10](#_Toc30446035)

[RAML: What are resource types? 11](#_Toc30446036)

[RAML: What do you after Designing RAML? 11](#_Toc30446037)

[What is Exchange/Anypoint Exchange? 12](#_Toc30446038)

[Design Conclusion 12](#_Toc30446039)

# Prerequisites

* Core Java Basics makes it easier to learn this course, else No Problem! 😊  
  <https://www.tutorialspoint.com/java/java_quick_guide.htm>
* [Knowledge on ***Computer Networks***]  
  <https://www.javatpoint.com/computer-network-tutorial>
* [Knowledge on ***HTTP*** ***Protocol***]  
  <https://www.javatpoint.com/computer-network-http>
* [Knowledge on ***FTP Protocol***]  
  <https://www.javatpoint.com/computer-network-ftp>
* [Knowledge on ***Transport Layer Protocol***]  
  <https://www.javatpoint.com/computer-network-transport-layer-protocols>

# Software Required to Complete this Course

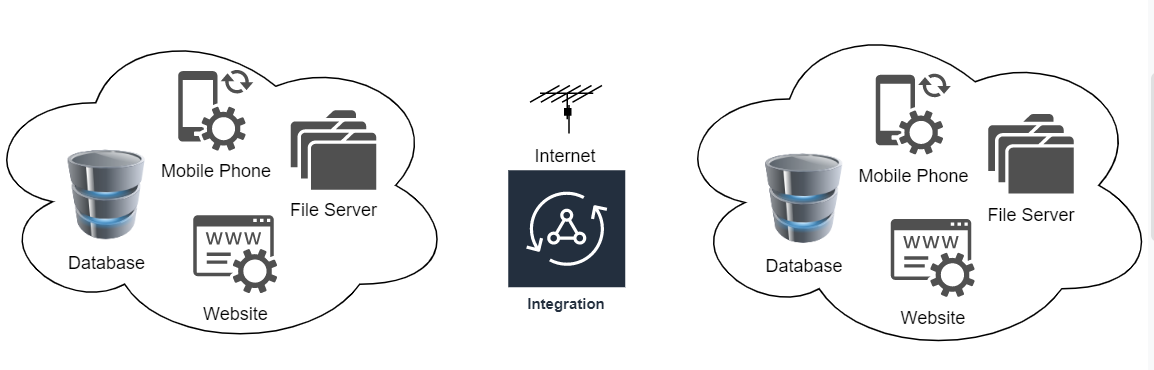
|  |  |  |
| --- | --- | --- |
| **Category** | **Software / Site Name** | **Download / Access Link** |
| Java | JDK 8 | <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> |
| Source  Code Management | GitHub | <https://github.com/> |
| Git.exe | <https://git-scm.com/downloads> |
| Tortoise Git | <https://tortoisegit.org/download/> |
| Mulesoft | Anypoint Platform | <http://anypoint.mulesoft.com/> |
| Anypoint Studio6 (Mule 3) | <https://www.mulesoft.com/lp/dl/studio/previous> |
| Anypoint Studio7 (Mule 4) | <https://www.mulesoft.com/lp/dl/studio> |
| REST API Testing | Postman | <https://www.getpostman.com/downloads/> |
| Messaging Service | Active MQ | <https://activemq.apache.org/download.html> |
| Build (CI/CD) Tools | Apache Maven | <https://maven.apache.org/download.cgi> |
|  | Jenkins | <https://jenkins.io/download/> |

# Introduction

Before learning Mulesoft there few questions those needs to be answered.

****Concept of Integration (or) Middleware****

Integration is a Technology used to TRANSPORT and TRANSFORM, DATA between multiple End Systems.



****Leading Technologies for Integration (or) Middleware****

* Mulesoft (Highly Customizable, Less Cost, Cloud based)
* Dell Boomi (Browser - Too Much Costly)
* Web Methods
* TIBCO (BW, BPM, EMS, Hawk) – Too Much Costly, High Maintenance

****Types of Integrations****

* Webservice Based Integration (HTTP/HTTPS = SOAP/REST)
* Point-to-Point Integration (On-Premise Integrations)

****What is Server-Client Model****

1. **Server**: Resource (File, Database, Code, Message etc.) provider
2. **Client**: Resource Requester

|  |  |  |
| --- | --- | --- |
|  |  |  |

****What is a Webservice****

The term Web service (WS) is either:

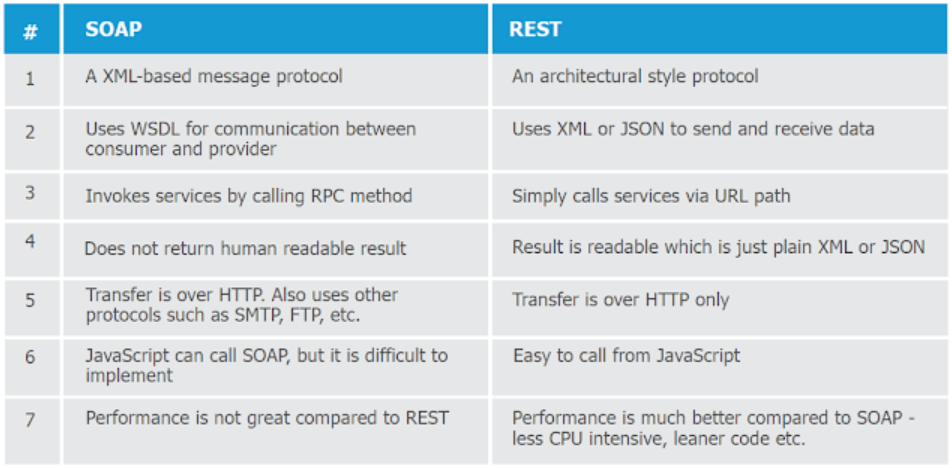
* (generic) a service offered by an electronic device to another electronic device, communicating with each other via the World Wide Web, or
* (specific) a Web service is a server running on a computer device, listening for requests at a port over a network, serving web documents (HTML, JSON, XML, Images), and creating web applications services, which serve in solving specific domain problems over the web (www, internet, **HTTP**)

**Web Service Architecture:**

* The provider: is used to create web service which makes it accessible for client applications who want to utilize it.
* The requestor: is not anything but the client app which requires contacting a web service. That client app can be ".Net", "Java", or any other language-based application that seeks functionality through a web service.

****Types of Webservices****

* **SOAP (Simple Object Access Protocol) Web Services**It is an XML-based protocol having the main benefit of implementing the SOAP Web Service as its security.
* **REST (Representational State Transfer) Web Services**It is not a Protocol it is instead the style architecture for software.   
  It uses normal protocol (like HTTP's: GET, PUT, POST, DELETE, etc.)



What is Mulesoft

Mulesoft is a Java based integration tool, which can be run in both On-Premise (Private server, Standalone) and Cloud (Cloudhub).

What is Anypoint Platform

Anypoint Platform is a unified portal, to Design, Manage and Run Mule Applications from Single portal.

Tools Used in Mulesoft

* Anypoint Platform:  
  a. Design Center: Design RAML (API Specification)

1. Runtime Manager: Deploy Applications (Cloudhub or Standalone)
2. API Manager: Register Runtime Manager Instance to Apply Policies
3. Access Management: Control Who can Access Anypoint Platform as part of Business Group (Company)

* Anypoint Studio:  
  Code Editor for Mulesoft Applications

Why Mulesoft?

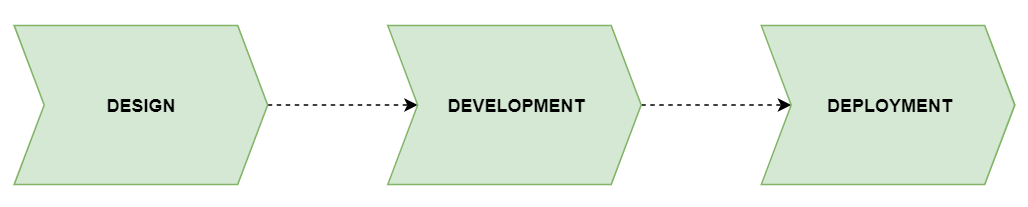
* Cloud based Integration Tool, where all products like (Runtime Manager, Alert Tools, Data Gateway etc.) are grouped under single platform.  
  Purely Integration Platform as a Service (iPaaS)
* Highly Customizable, if you can’t achieve any task using Mulesoft provided Components, it allows you to write custom components, transformers and connectors.
* Easy maintainability
* Low Cost compared to other integration tools
* Faster Development because of the Look and Feel (Drag and Drop) type of Development

Who Uses Mulesoft?



What is Mulesoft Development Lifecycle?

Mulesoft follows a standard procedure to Develop **RESTful** webservices, which appears as below:



1. **Design**

**Tools Used:** Anypoint Platform 🡪 Design Center 🡪 Anypoint Exchange  
In this stage, Developer needs to create a template for REST API he is about to develop

This is carried out in **Design Center** of Anypoint Platform.

[https://anypoint.mulesoft.com/designcenter](https://anypoint.mulesoft.com/designcenter/)

Template that is created for the REST API which is to be developed is called RAML.

**RAML (RESTful API Modelling Language):** It is a YAML-based language (space intended) for describing RESTful APIs. It provides all the information necessary to describe RESTful or practically RESTful APIs.

**Once RAML design is complete, it should be Published to Anypoint Exchange for stake holder’s Approval.**

**Anypoint Exchange:** This is a Central Design exchange service provide by Mulesoft in Anypoint Platform. People who publish content to this portal must be part of Business Group, so that other people who are part of the same Business Group can view the published content.

1. **Development  
   Tools Used:** Anypoint Studio (Code Editor)

In this stage, based on RAML created in previous step, Developer needs to Generate flows out of RAML and implement the Business Logic in **Anypoint Studio.**

1. **Deployment**

In the stage, once the Development is complete as part of previous step, Mulesoft Project (Code) needs to Be Packaged (Generate Deployable Archive) and needs to be Run in an environment so that End User/Client Applications can use the same as a real-time app.

Mulesoft Projects need Mule Runtime to Deploy, and Mule Runtime is available at two platforms.  
CloudHub: Anypoint Platform 🡪 Runtime Manager

On-Premise: MULE\_HOME 🡪 Apps folder.  
***[Note: Deployment Guide will be shared as a Separate Document]***

# Design

RAML: What is RAML?

RAML (RESTful API Modelling Language) is a template created in Anypoint Platform used to Describe the REST API with list of Resource Paths, Methods and Sample Request-Responses associated with each Method 🡪 Resource Path

RAML: What is the Purpose of RAML?

RAML serves two purposes:

1. **Template:** Describing the REST API implementation explaining how to consume resources from a REST API and what are sample requests and responses available for temporary usage.
2. **Temporary Service (Mocking Service):**

Instead of Client Applications (Mobile App or Website) to wait till Mulesoft REST API development is complete, they can make use of RAML provided Mocking Service which will have HTTP endpoint for client applications to configure, so that they can proceed with UI screens development.

RAML: Where do you create RAML and How do you create RAML?

RAML is created in Anypoint Platform 🡪 Design Centre ([https://anypoint.mulesoft.com/designcenter](https://anypoint.mulesoft.com/designcenter/))

|  |
| --- |
| **Example RAML with all types of assets used** |
| #%RAML 1.0  title: students-api  version: v1  protocols:  - HTTPS  - HTTP  mediaType:  - application/json  - application/xml  baseUri: http://mulebytes.com/api/v1  securitySchemes:  myOwnSecurityScheme: !include securitySchemes/mySecurityScheme.raml  securedBy:  - myOwnSecurityScheme  types:  student:  properties:  name:  type: string  minLength: 3  required: true  age:  type: number  minimum: 10  default: 10  traits:  myCommonHeaders: !include traits/headers.raml  resourceTypes:  getStudentsResouceTypeReference:  usage: This API is used by <<resourcePathName>>  description: API is used ti retrieve and create <<resourcePathName>>  get:  queryParameters:  studentId:  type: number  required: true  responses:  200:  body:  application/json:  example: |  <<getStudentsResponseExample>>  /getStudentsInfo:  type:  getStudentsResouceTypeReference:  getStudentsResponseExample: !include samples/output/getStudentsInfoResponse.json  /getStudentsClass:  type:  getStudentsResouceTypeReference:  getStudentsResponseExample: !include samples/output/getSudentsClass.json    /registerStudent:  post:  body:  application/json:  example: !include samples/input/registerStudentRequest.json  schema: !include schemas/input/registerStudentsRequestSchema.json |

Post RAML design is complete.

We publish to Exchange for Reviews from Clients.

Once the RAML is reviewed completely, proceed for development.

RAML: What are traits?

Traits are used to define common attributes like headers or query parameters for HTTP method or resource paths in same RAML file or in an external Trait RAML and use like as above.

|  |
| --- |
| #%RAML 1.0 Trait  queryParameters:  username:  type: string  required: true  example: "admin" |

RAML: What are Schemas and how do you use them?

Schema is set of Types, those describe a message along with its properties and validations.

RAML: What are types?

Types are custom data types those can be defined to validate payload.

RAML: What is the difference between Types/Schema?

**Types**: Custom Data Types **Schema**: Predefined template in native data format

|  |  |
| --- | --- |
| **Types Example** | **Schema Example** |
| types:  student:  properties:  name:  type: string  minLength: 3  age:  type: number  minimum: 10 | {  "$schema": "http://json-schema.org/draft-04/schema#",  "type": "object",  "properties": {  "name": {  "type": "string"  },  "age": {  "type": "integer"  }  },  "required": [  "name"  ]  } |

RAML: What are resource types?

ResourceType is basically a template that is used to define the descriptions, methods, and parameters that can be used by multiple resources without writing the duplicate code or repeating code.

If you have multiple resource paths have same set of methods but with dynamic set of examples and schemas, then it’s recommended to use resource types.

**Please refer above RAML example to understand resource types.**

RAML: What do you after Designing RAML?

Publish RAML to Anypoint Exchange post completing RAML design for Stake holder’s Approval.

|  |
| --- |
|  |
|  |

What is Exchange/Anypoint Exchange?

Exchange is a Asset Distribution service provided by Mulesoft, used to share RAML/Connectors/Templates/Sample APIs across Teams in an Organization (Business Groups)

|  |
| --- |
|  |
|  |

Design Conclusion

In Design phase, based on the business requirements,

1. RAML is designed
2. Publish to Exchange
3. Get Stakeholder’s / Client Approval
4. Mock service is enabled
5. Mock URL is shared with Client Applications
6. Get ready to start the Development

**GitHub:** <https://github.com/mulebytes/raml-practice/tree/master/students-api-sample-raml>

# Development

What are Scopes?

A Mule Scope is a code block which contains a series of message processors. Mule provides some very important scopes, e.g. Async, Cache, Message Enricher, and For Each.

1. Where to write Code in Mulesoft
2. Flows (Main Flow (Connector, Scheduler), Private Flow (Flow Reference))
3. Sub Flows (Flow Reference, If any error report it back to caller)
4. Exception Strategies (How and When it will be executed: When there is an error, and via internal exception strategy scope)