MuleSoft Certified Developer – Level 1 (Mule 4) DELTA Certification Exam

Summary

A MuleSoft Certified Developer – Level 1 (Mule 4) should be able to successfully work on basic Mule 4 projects with guidance and supervision. The MCD – Level 1 (Mule 4) DELTA exam validates that a certified MuleSoft Certified Developer – Integration and API Associate (Mule 3) has the required knowledge and skills to design, build, test and debug, deploy, and manage basic Mule 4 APIs and integrations. Certified candidates should be able to:

- Use MuleSoft-hosted Anypoint Platform to take a basic API through all the steps of its lifecycle: design, build, deploy, manage, and govern.
- Use Anypoint Studio to build, test, and debug basic Mule 4 integrations and API implementations.
- Use Mule 4 connectors to connect to a range of resources including databases, files, web services, SaaS applications, and JMS queues.
- Perform basic data transformations using DataWeave 2.0.
- Use Mule 4 event processors to control event flow, handle errors, and process batch records.

Format

- Format: Multiple-choice, closed book
- Length: 35 questions
- Duration: 75 minutes
- Pass score: 70%
- Language: English

You can take the exam a maximum of 2 times, with a 24-hour wait between attempts. After an unsuccessful second attempt, you can only take the full MuleSoft Certified Developer – Level 1 exam.

Cost

If you have a previous MuleSoft Certified Developer - Integration and API Associate (Mule 3) certification, you can purchase the exam with one of the following. Each includes one free retake.

- $250
- 1 Flexible Training Credit (FTC)

You can also get two exam attempts with the successful completion of the Anypoint Platform Development: Mule 4 for Mule 3 Users course.
Validity

The certification expires two years from the date you pass the exam.

Preparation

You can best prepare for the exam by leveraging the following resources:

- **Instructor-led training: Anypoint Platform Development: Mule 4 for Mule 3 Users**
  - Recommended as the most effective and efficient method of preparation
  - 3-day class
  - Private, public, onsite, and online classes available
  - Includes two attempts for this exam or the full MuleSoft Certified Developer - Level 1 exam

- **Self-study training: MuleSoft U Mule 4 for Mule 3 Users**
  - 30+ step-by-step exercises to teach you the basics
  - All content available instantly for you to complete at your own pace
  - Supported by the peer-to-peer MuleSoft training forum
  - Successful completion of the course includes two MuleSoft-sponsored attempts for this exam or the full MuleSoft Certified Developer - Level 1 exam

- **Self-assessment quiz**
  - 5+ multiple-choice questions for each knowledge section of the exam
  - Identifies strengths and weaknesses

- **Do-it-yourself exercises**
  - 10+ DIY exercises to get experience with and apply the knowledge required for the certification
  - Starting code and solutions provided
  - Can be completed in any order
## Topics

The exam validates that the candidate can perform the following tasks.

*Note: DEV:443 is the acronym for the Anypoint Platform Development: Mule 4 for Mule 3 Users course. DEV:DIY4 is the acronym for the MCD - Level 1 / Development Fundamentals (Mule 4) Self-Assessment Quiz & DIY Exercises materials.*

<table>
<thead>
<tr>
<th>Accessing and modifying Mule events</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Describe the Mule event data structure.</td>
<td>• DEV:443 Module 1</td>
</tr>
<tr>
<td>• Use transformers to set event payloads, attributes, and variables.</td>
<td>• DEV:DIY4 Exercise 6-1, 7-1, and 7-2</td>
</tr>
<tr>
<td>• Write DataWeave expressions to access and modify event payloads, attributes, and variables.</td>
<td>• About Mule Event</td>
</tr>
<tr>
<td>• Enrich Mule events using target parameters.</td>
<td>• Set Payload</td>
</tr>
<tr>
<td></td>
<td>• Set Variable</td>
</tr>
<tr>
<td></td>
<td>• About the Transform Component</td>
</tr>
<tr>
<td></td>
<td>• About DataWeave</td>
</tr>
<tr>
<td></td>
<td>• Logger Component Example</td>
</tr>
<tr>
<td></td>
<td>• Enriching Data with Target Parameters</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Structuring Mule applications</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Parameterize an application using property placeholders.</td>
<td>• DEV:443 Module 1</td>
</tr>
<tr>
<td>• Define and reuse global configurations in an application.</td>
<td>• DEV:443 Module 2</td>
</tr>
<tr>
<td>• Break an application into multiple flows using private flows, subflows, and the Flow Reference component.</td>
<td>• DEV:DIY4 Exercise 7-1 and 7-2</td>
</tr>
<tr>
<td>• Specify what data (payload, attributes, variables) is persisted between flows when a Flow Reference is used.</td>
<td>• To Configure Property Placeholders for Mule Apps</td>
</tr>
<tr>
<td>• Specify what data is persisted between flows when a Mule event crosses a connection boundary.</td>
<td>• Global Elements</td>
</tr>
<tr>
<td>• Specify what data exists in a flow before and after a call in the middle of a flow to an external resource.</td>
<td>• Flow and Subflow Components</td>
</tr>
<tr>
<td></td>
<td>• About Flow Ref</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Building API implementation interfaces</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Manually create a RESTful interface for a Mule application.</td>
<td>• DEV:DIY4 Exercise 4-1</td>
</tr>
<tr>
<td>• Generate a REST Connector from a RAML specification.</td>
<td>• Converting a RAML to a Connector Using REST Connect</td>
</tr>
<tr>
<td>• Describe the features and benefits of APIkit.</td>
<td>• Workflow: Creating Testing and Implementing an API</td>
</tr>
<tr>
<td>• Use APIkit to create implementation flows from a RAML file.</td>
<td>• HTTP Request Configuration Reference: Adding Custom Parameters</td>
</tr>
<tr>
<td>• Describe how requests are routed through flows generated by APIkit.</td>
<td></td>
</tr>
</tbody>
</table>
Routing events

- Use the Choice router to route events based on conditional logic.
- Use the Scatter-Gather router to multicast events.
- Validate data using the Validation module.

Handling errors

- Describe the default error handling in a Mule application.
- Define a custom global default error handler for an application and identify in what situations it will be used.
- Compare and contrast how the On Error Continue and On Error Propagate scopes work.
- Create one or more error handlers for a flow.
- Use the Try scope to specify error handlers for one or more event processors.
- Describe the data structure of the Mule Error object.
- Map errors to custom application errors.

Transforming data with DataWeave

- Write DataWeave scripts to convert JSON, XML, and Java data structures to different data structures and data types.
- Use DataWeave functions.
- Define and use DataWeave variables, functions, and modules.
- Define and use custom data types.
- Apply correct DataWeave syntax to coerce data types.
- Apply correct DataWeave syntax to format strings, numbers, and dates.
- Call Mule flows from a DataWeave script.
### Using Connectors

- Retrieve data from a Database using the Database connector.
- Create parameterized SQL queries for the Database connector.
- Retrieve data from a REST service using the HTTP Request operation or a REST Connector.
- Use a Web Service Consumer connector to consume a SOAP web service.
- Use the Transform Message component to pass arguments to a SOAP web service.
- List, read, and write local files using the File connector.
- List, read, and write remote files using the FTP connector.
- Use the JMS connector to publish and listen for JMS messages.

### Processing records

- List and compare and contrast the methods for processing individual records in a collection.
- Explain how Mule events are processed by the For Each scope.
- Use the For Each scope to process records.
- Explain how Mule events are processed by the Batch Job scope.
- Use a Batch Job with Batch Steps and a Batch Aggregator to process records.
- Use the Scheduler component to trigger a flow.
- Use connector listeners to trigger flows.
- Describe the features, benefits, and process to use automatic watermarking vs. manual watermarking.
- Use connectors with automatic watermarking capabilities.
- Persist data between flow executions using the Object Store.

### Debugging and troubleshooting Mule applications

- Use breakpoints to inspect a Mule event during runtime.
- Install missing Maven dependencies.
- Read and decipher Mule log error messages.
### Deploying and managing APIs and integrations

| • Package Mule applications for deployment. | • DEV: DIY4 Exercise 5-1 and 5-2 |
| • Deploy applications to CloudHub.          | • Deploying to CloudHub          |
| • Use CloudHub properties to ensure deployment success. | • Configuring API Autodiscovery in a Mule 4 Application |
| • Create and deploy API proxies.            | • About Policies                 |
| • Connect an API implementation to API Manager using autodiscovery. | • To Apply a Policy and SLA Tier |
| • Use policies, including client ID enforcement, to secure an API. |                       |
| • Create SLA tiers and apply SLA based policies. |                       |

### More information

For more information, visit [http://help.learn.mulesoft.com](http://help.learn.mulesoft.com).