Mulesoft Topic

Explaining application network basics

- Explain MuleSoft's proposal for closing the IT delivery gap.
- Describe the role and characteristics of the "modern API."
- Describe the purpose and roles of a Center for Enablement (C4E).
- Define and describe the benefits of API-led connectivity and application networks.
- Define and correctly use the terms API, API implementation, API interface, API consumer, and API invocation.
- Describe the basics of the HTTP protocol and the characteristics of requests and responses.
- Describe the capabilities and high-level components of Anypoint Platform for the API lifecycle.

Designing and consuming APIs

- Describe the lifecycle of the "modern API."
- Use RAML to define API resources, nested resources, and methods.
- Identify when and how to define query parameters vs URI parameters.
- Use RAML to define API parameters, requests, and responses.
- Use RAML to define reusable data types and format-independent examples.
- Read a RAML spec and formulate RESTful requests with query parameters and/or headers as appropriate.

Accessing and modifying Mule events

- Describe the Mule event data structure.
- Use transformers to set event payloads, attributes, and variables.
- Write DataWeave expressions to access and modify event payloads, attributes, and variables.
- Enrich Mule events using target parameters.

Enrich Data with Target Parameters

- Structuring Mule applications
- Parameterize an application using property placeholders.
- Define and reuse global configurations in an application.

- Break an application into multiple flows using private flows, subflows, and the Flow Reference component.
- Specify what data (payload, attributes, variables) is persisted between flows when a Flow Reference is used.
- Specify what data (payload, attributes, variables) is persisted between flows when a Mule event crosses a connection boundary.
- Specify what data (payload, attributes, variables) exists in a flow before and after a call in the middle of a flow to an external resource.

Building API implementation interfaces

- Manually create a RESTful interface for a Mule application.
- Generate a REST Connector from a RAML specification.
- Describe the features and benefits of APIkit.
- Use APIkit to create implementation flows from a RAML file.
- Describe how requests are routed through flows generated by APIkit.

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 HTTP Request or a REST Connector.
- Use a Web Service Consumer connector to consume SOAP web services.
- 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 watermarking.
- 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.
- Deploy applications to CloudHub.
- Use CloudHub properties to ensure deployment success.
- Create and deploy API proxies.
- Connect an API implementation to API Manager using autodiscovery.
- Use policies, including client ID enforcement, to secure an API.
- Create SLA tiers and apply SLA based policies.