

Creating, Publishing, and Securing APIs with IBM API Connect

WD508 (Classroom)

ZD508 (Self-paced)

Course description

This course teaches you how to create, publish, and secure APIs with IBM API Connect V5.0.8. You learn how to use the Developer Toolkit to define an API interface according to the OpenAPI specification. You assemble message processing policies in the API Designer web application and define client authorization schemes, such as OAuth 2.0, in the API definition. You build a Node.js API application with the LoopBack framework. You also package, stage, and publish an API with product and plan.

For information about other related courses, see the IBM Training website:

http://www.ibm.com/training

General information

Delivery method

Classroom or self-paced virtual classroom (SPVC)

Course level

ERC 1.0

Product and version

IBM API Connect version 5.0.8

Audience

This course is designed for API developers: software developers who define and implement API operations.

Learning objectives

After completing this course, you should be able to:

* Create APIs with the API Connect toolkit
* Implement APIs with the LoopBack Node.js framework
* Validate, filter, and transform API requests and responses with message processing policies
* Authorize client API requests with security definitions
* Enforce an OAuth flow with an OAuth 2.0 Provider API
* Stage, publish, and test APIs on the API Connect cloud

Prerequisites

Before taking this course, you should successfully complete *Developing REST APIs with Node.js for IBM Bluemix* (VY102G).

Duration

4 days

Skill level

Intermediate

|  |  |
| --- | --- |
| Classroom (ILT) setup requirements | |
| Processor | Intel Core i5 2.5 GHz or higher |
| GB RAM | 4 GB |
| GB free disk space | 60 GB |
| Network requirements | LAN / Internet / DHCP |
| Other requirements | None |

Notes

The following unit and exercise durations are estimates, and might not reflect every class experience. If the course is customized or abbreviated, the duration of unchanged units will probably increase.

This course is an update of the following previous course:

* *Create, Secure, and Publish APIs with IBM API Connect V5* (WD500G)

Course agenda

|  |
| --- |
| Course introduction  Duration: 15 minutes |

|  |  |
| --- | --- |
| Unit 1. IBM API Connect V5 overview  Duration: 1 hour and 15 minutes | |
| Overview | This unit explains the scope and purpose of IBM API Connect V5 from the perspective of an API developer. You review the role of APIs in the cloud network and enterprise architecture. You also examine the API create, run, manage, and secure features in an API Connect solution. |
| Learning objectives | After completing this unit, you should be able to:   * Explain the role of IBM API Connect in an enterprise architecture * Identify the components in the API Connect solution * Identify the deployment environments for API Connect * Identify the API create, run, manage, and secure features |

|  |  |
| --- | --- |
| Unit 2. API Connect Developer Toolkit  Duration: 1 hour | |
| Overview | This unit describes the features of the API development environment, and explains how to use the API Connect Developer Toolkit to design, develop, test, and publish APIs. You learn how to install and verify the installation of the Toolkit on your own workstation. |
| Learning objectives | After completing this unit, you should be able to:   * Explain the purpose of the API Connect Developer Toolkit * Identify the features of the API Designer * Identify the features of the apic command line utility * Describe the software and hardware requirements of the Developer Toolkit * Install the Developer Toolkit * Verify the installation of the Developer Toolkit |

|  |  |
| --- | --- |
| Unit 3. Creating an API definition  Duration: 1 hour and 15 minutes | |
| Overview | This unit examines the API definition, a structured file that documents the API operations, parameters, and data types. You learn how to document, version, and define your API interface. You also examine the role of environment variables as properties. |
| Learning objectives | After completing this unit, you should be able to:   * Explain the concept of an API definition * Explain the purpose of the OpenAPI specification * Document the API version and description * Define an API operation * Define query and path parameters * Define request and response headers * Explain the purpose of API properties * Define environment variables as properties * Modify the target URL as a property * Explain the concept of API templates |

|  |  |
| --- | --- |
| Exercise 1. Creating and publishing an API in API Designer  Duration: 1 hour | |
| Overview | This exercise covers how to define an API interface from an existing API. You review the API operations, parameters, and data types from the API Designer web application. You also publish and test the API from the API Designer web application. |
| Learning objectives | After completing this exercise, you should be able to:   * Create an API definition in the API Designer * Review the operations, properties, and data types in an API definition * Publish an API to the API Connect development environment * Start an API and the micro gateway * Test an API with the API Explorer |

|  |  |
| --- | --- |
| Unit 4. Defining APIs that call REST and SOAP services  Duration: 1 hour | |
| Overview | With API Designer, you can quickly expose your existing cloud and enterprise services in the API gateway. This unit examines how to define API operations that call existing REST or SOAP APIs. |
| Learning objectives | After completing this unit, you should be able to:   * Explain the role of the API gateway in exposing existing services * Explain how to expose an existing SOAP service in an API definition * Explain how to expose an existing REST service in an API definition |

|  |  |
| --- | --- |
| Exercise 2. Defining an API that calls an existing SOAP service  Duration: 1 hour | |
| Overview | With API Connect, you can define an API from existing enterprise services. In this exercise, you define an API that calls an existing SOAP service. You define API paths and methods that map to SOAP web service operations, and map SOAP message types to API data types. You then test the SOAP API in the API Explorer. |
| Learning objectives | After completing this exercise, you should be able to:   * Create an API definition with the API Designer web application * Define API operations that map to a SOAP web service * Map SOAP message types to API data types * Test the SOAP API on the DataPower gateway of the developer toolkit * Publish the API to API Manager * Test the SOAP API in the API Manager test client (optional) |

|  |  |
| --- | --- |
| Unit 5. Implementing APIs with the LoopBack framework  Duration: 1 hour | |
| Overview | This unit introduces the LoopBack framework, which is a model-driven approach to building REST APIs in Node.js. You examine the structure and components in a LoopBack application, and learn how to generate an application scaffold with the Developer Toolkit. |
| Learning objectives | After completing this unit, you should be able to:   * Explain the concept of model-driven design * Explain the purpose of the LoopBack Node.js framework * Identify the components of a LoopBack application * Create the application scaffold from the apic command line utility * Create a LoopBack application in the API Designer web application |

|  |  |
| --- | --- |
| Exercise 3. Creating a LoopBack application  Duration: 1 hour | |
| Overview | In this exercise, you build a LoopBack application to implement an API. You generate the application scaffold with the apic command line utility, and define the model and properties with the API Designer web application. |
| Learning objectives | After completing this exercise, you should be able to:   * Create an API definition with the API Designer web application * Create an application scaffold with the apic command line utility * Create LoopBack models and properties |

|  |  |
| --- | --- |
| Unit 6. LoopBack models, properties, and relationships  Duration: 1 hour and 15 minutes | |
| Overview | This unit explores the concepts of LoopBack models, properties, and relationships. You learn how to define models and properties to map business data structures to API resources. You also learn how to create relationships between models. |
| Learning objectives | After completing this unit, you should be able to:   * Explain the relationship between the model and the API * Explain the concepts of LoopBack models, properties, and relationships * Explain the features that each model base class inherits * Identify the property data types * List the model relationship types * Define models and properties in the API Designer * Define models, properties, and relationships in the apic utility |

|  |  |
| --- | --- |
| Unit 7. Defining data sources with connectors  Duration: 1 hour | |
| Overview | This unit explores how LoopBack data sources retrieve and save model data from data stores. You learn how to install LoopBack connectors in your application, and how to map model objects to data sources. |
| Learning objectives | After completing this unit, you should be able to:   * Explain the role of a LoopBack data source * Identify the role of database and non-database connectors * Install a LoopBack connector * Explain how to map data sources to models * Generate models and properties from a data source * Build model instances from unstructured data |

|  |  |
| --- | --- |
| Exercise 4. Defining LoopBack data sources  Duration: 1 hour | |
| Overview | In this exercise, you bind the model to relational and non-relational databases with data sources. You define relationships between models. Finally, you test API operations with the API Explorer. |
| Learning objectives | After completing this exercise, you should be able to:   * Install and configure the MySQL connector * Install and configure the MongoDB connector * Generate models and properties from a data source * Define relationships between models * Test an API with the API Explorer |

|  |  |
| --- | --- |
| Unit 8. Implementing remote methods and event hooks  Duration: 45 minutes | |
| Overview | This unit explores how to define custom remote methods and hooks. You learn how to extend the data-centric LoopBack model with remote methods. You also learn how to implement event-driven functions with remote and operation hooks. |
| Learning objectives | After completing this unit, you should be able to:   * Explain the purpose of a remote method * Define and call a remote method * Explain the purpose and use cases for a remote hook * Explain the purpose and use cases for an operation hook |

|  |  |
| --- | --- |
| Exercise 5. Implementing event-driven functions with remote and operation hooks  Duration: 45 minutes | |
| Overview | This unit explores how LoopBack data sources retrieve and save model data from data stores. You learn how to install LoopBack connectors in your application, and how to map model objects to data sources. |
| Learning objectives | After completing this exercise, you should be able to:   * Define a remote hook in a LoopBack application * Define an operation hook in a LoopBack application * Test remote and operation hooks in the assembly editor test client |

|  |  |
| --- | --- |
| Unit 9. Assembling message processing policies  Duration: 1 hour and 30 minutes | |
| Overview | In the API Gateway, message processing policies log, route, and transform API request and response messages. This unit explores the concept of message processing policies. You learn how to define a set of message processing policies in your API definition file with the API Designer. |
| Learning objectives | After completing this unit, you should be able to:   * Explain the concept of non-functional requirements * Identify use cases for message processing policies * Explain the relationship between message processing policies and the API application * List the policies that each API gateway type supports * Explain how to generate JSON Web Tokens with policies * Explain the concept and use case for custom user-created policies |

|  |  |
| --- | --- |
| Exercise 6. Assembling message processing policies  Duration: 2 hours | |
| Overview | This exercise explains how to create message processing policies. You define a sequence of policies in the assembly view of the API Designer. You define an API that exposes an existing SOAP service as a REST API. You also define an API that transforms responses from an existing service into a defined message format. |
| Learning objectives | After completing this exercise, you should be able to:   * Create an API with JSON object definitions and paths * Configure an API to call an existing SOAP service * Import an existing API definition into the source editor * Map data from multiple API calls into an aggregate response * Define and call a gateway script with an API assembly * Test DataPower gateway policies in the API Manager explore view |

|  |  |
| --- | --- |
| Unit 10. Declaring client authorization requirements  Duration: 1 hour | |
| Overview | This unit explores how to define client authorization requirements in the API definition. The client authorization requirements specify which authentication and authorization standards to enforce. You learn how to configure API keys, HTTP basic authentication, and OAuth 2.0 authorization schemes. |
| Learning objectives | After completing this unit, you should be able to:   * Identify the security definition options in API Connect * Explain the concept and use cases for API keys * Explain the concept and use cases for HTTP basic authentication * Explain the concept and use cases for OAuth 2.0 authorization * Explain the steps in the OAuth 2.0 message flow |

|  |  |
| --- | --- |
| Unit 11. Creating an OAuth 2.0 provider  Duration: 1 hour | |
| Overview | This unit examines the OAuth 2.0 provider. In an OAuth 2.0 message flow, the OAuth provider is an authorization server that issues access tokens to authorized clients. In an API Connect cloud, you can configure the API gateway to act as an OAuth 2.0 Provider. This unit explains how to create an OAuth 2.0 provider in an API definition from the API Designer web application. |
| Learning objectives | After completing this unit, you should be able to:   * Explain the concept of an OAuth provider * Identify the role of the authorization server in the OAuth flow * Identify the OAuth 2.0 roles, client types, and schemes * Explain the difference between public and confidential schemes * Explain the OAuth flow and grant types * Explain the concept of access tokens * Explain the concept of the token revocation service * Explain the purpose of the token introspection endpoint * Explain the concept of OAuth metadata * Explain the relationship between the OAuth provider and the API security definition |

|  |  |
| --- | --- |
| Exercise 7. Declaring an OAuth 2.0 provider and security requirement  Duration: 1 hour | |
| Overview | In this exercise, you examine two of the three parties in an OAuth 2.0 flow: the OAuth 2.0 provider API and the API resource server. You define an OAuth 2.0 API provider to authorize access and issue tokens. In the case study application, you declare an OAuth 2.0 security constraint that enforces access control with the OAuth 2.0 provider API. |
| Learning objectives | After completing this exercise, you should be able to:   * Define an OAuth 2.0 provider API in the API Designer web application * Configure the client ID and client secret security definition * Declare and enforce an OAuth 2.0 security definition with the API Designer web application |

|  |  |
| --- | --- |
| Unit 12. Deploying an API to a Docker container  Duration: 45 minutes | |
| Overview | This unit examines how to configure and deploy an API implementation to a Docker container. Docker is a lightweight architecture for managing portability and ease of deployment of the application to a runtime environment. Current versions of Docker include a swarm mode that is used for scalability and failover in distributed server runtime environments. You learn how to deploy API implementations to a Docker image and a Docker swarm. |
| Learning objectives | After completing this unit, you should be able to:   * Describe what Docker is * List the benefits of using Docker containers * Describe the differences between virtual machines and containers * Provide definitions for some of the terminology that is used with Docker * Explain how to deploy a LoopBack application to a Docker image * Describe the Docker swarm mode for managing a cluster of Docker engines |

|  |  |
| --- | --- |
| Exercise 8. Deploying an API implementation to a container runtime  Duration: 45 minutes | |
| Overview | In this exercise, you deploy and publish a LoopBack API application to a swarm of Docker container runtime environments. A swarm manager administers a cluster of Docker containers. At run time, the swarm manager routes API requests to a Docker cluster member that runs an instance of your API and its runtime dependencies. |
| Learning objectives | After completing this exercise, you should be able to:   * Test a local copy of a LoopBack API application * Examine the Dockerfile that is used to deploy a Loopback API * Build a Docker image by using the docker command with the Dockerfile * Verify that the API runs on the Docker image * Push the image to a local registry * Initialize the Docker swarm * Use the Docker stack command to deploy a swarm service * Test that the API runs on the swarm service |

|  |  |
| --- | --- |
| Unit 13. Staging, publishing, and deploying an API product  Duration: 1 hour | |
| Overview | This unit examines how to package and publish APIs to the API Connect cloud. A product defines a collection of APIs for deployment. It also contains a plan, which is a contract between the API provider and API consumer that specifies quality of service characteristics, such as the rate limit of API calls. |
| Learning objectives | After completing this unit, you should be able to:   * Explain the concept of a plan, a product, and a catalog * Explain the staging and publishing API lifecycle stages * Define an API product and a plan * Define product categories * Stage a product to a catalog |

|  |  |
| --- | --- |
| Exercise 9. Defining and publishing an API product  Duration: 45 minutes | |
| Overview | This exercise examines how to publish APIs with plans and products. You create a product and a plan, and deploy the product to the API Manager in the API Designer. |
| Learning objectives | After completing this exercise, you should be able to:   * Modify the invoke URL for the inventory application to route to the Docker image * Create a product in the API Designer * Modify the product properties and add the APIs to the product * Define an API plan * Define a publish target * Stage a product to a catalog |

|  |  |
| --- | --- |
| Unit 14. Subscribing and testing APIs  Duration: 1 hour | |
| Overview | This unit explores the application developer user experience. In the API Connect architecture, the application developer creates an application that calls published APIs. To use APIs, an application developer registers for an account in the Developer Portal. This unit explains how the application developer subscribes to a plan and tests API operations. |
| Learning objectives | After completing this unit, you should be able to:   * Explain the role of application developers in calling published APIs * Explain the difference between self-registration and registration approvals * Explain how to add an application in the Developer Portal * Explain the role of client ID and client secret * Explain how to subscribe to an API plan * Explain the test client features in the Developer Portal |

|  |  |
| --- | --- |
| Exercise 10. Subscribing and testing APIs  Duration: 1 hour | |
| Overview | In this exercise, you learn about the application developer experience in the Developer Portal. You create a developer account and add an application. You also review the client ID and client secret values, subscribe to an API plan, and test operations from an API product. |
| Learning objectives | After completing this exercise, you should be able to:   * Self-register an application developer account * Add an API consumer application in a developer account * Review and reset the client ID and client secret values * Test an API operation in the Developer Portal * Test an API operation with a consumer application |

|  |  |
| --- | --- |
| Unit 15. Troubleshooting  Duration: 30 minutes | |
| Overview | This unit reviews common deployment and run time issues with IBM API Connect. You learn how to troubleshoot and isolate an error in the API Connect cloud. You examine the relationship between the API Manager, Gateway server, and Developer Portal at run time. You also learn how to debug issues with an OAuth 2.0 message flow. |
| Learning objectives | After completing this unit, you should be able to:   * Identify the API message flow in the API Connect cloud * Explain the relationship between the API Manager, Gateway server, and Developer Portal at run time * Show the proper flow of the consumer web application * Explain how to troubleshoot errors with an OAuth 2.0 message flow * Explain how to troubleshoot LoopBack API implementation run time errors |

|  |  |
| --- | --- |
| Exercise 11. Troubleshooting the case study  Duration: 1 hour and 15 minutes | |
| Overview | In this exercise, you review and apply troubleshooting steps in the exercise case study. You apply the steps to verify the operation of the inventory application and correct the issues that you identify. |
| Learning objectives | After completing this exercise, you should be able to:   * Verify the key components in the exercise case study * Troubleshoot and correct common issues with the case study * Isolate implementation issues in your own API Connect cloud solution |

|  |  |
| --- | --- |
| Unit 16. Course summary  Duration: 15 minutes | |
| Overview | This unit summarizes the course and provides information for future study. |
| Learning objectives | After completing this unit, you should be able to:   * Explain how the course met its learning objectives * Access the IBM Training website * Identify other IBM Training courses that are related to this topic * Locate appropriate resources for further study |

For more information

To learn more about this course and other related offerings, and to schedule training, see **ibm.com**/training

To learn more about validating your technical skills with IBM certification, see **ibm.com**/certify

To stay informed about IBM training, see the following sites:

IBM Training News: ibm.com/blogs/ibm-training/

YouTube: youtube.com/IBMTraining

Facebook: facebook.com/ibmtraining

Twitter: twitter.com/IBMTraining