

Create, Secure, and Publish APIs with IBM API Connect v2018

WD514 (Classroom)

ZD514 (Self-paced)

Course description

This course teaches you how to create, publish, and secure APIs with IBM API Connect V2018. You build Node.js API applications with the LoopBack framework. You define API interfaces according to the OpenAPI specification. You assemble message processing policies and define client authorization schemes, such as OAuth 2.0, in the API definition. You publish APIs and make them available on a secure gateway and on the Developer Portal.

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

**ibm.com**/training

General information

Delivery method

Classroom or self-paced virtual classroom (SPVC)

Course level

ERC 2.0

Product and version

IBM API Connect version 2018.4.1.1

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 in API Connect
* Implement APIs with the LoopBack Node.js framework
* Create message processing policies that transform API requests and responses
* Authorize client API requests with security definitions
* Enforce an OAuth flow with an OAuth 2.0 API security
* 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 | Not specified |
| GB RAM | Not specified |
| GB free disk space | Not Specified |
| Network requirements | LAN / Internet / DHCP |
| Other requirements | Browser-based access to the Internet for accessing labs on the IBM Remote Lab Platform |

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*.0.8 (WD508G)

Course agenda

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

|  |  |
| --- | --- |
| Unit 1. IBM API Connect V2018 overview  Duration: 1 hour | |
| Overview | This unit explains the scope and purpose of IBM API Connect V2018 from the perspective of an API developer and cloud administrator. You review the key capabilities of API Connect. You examine the nature of an on-premises cloud and how the cloud is configured in the API Cloud Manager user interface. You review the different gateway types for securing and managing APIs. Identify the requirements for installing an on-premises API Connect cloud. |
| Learning objectives | After completing this unit, you should be able to:   * Explain what is IBM API Connect * Describe the key capabilities of API Connect * Describe how API Connect manages API through the entire API lifecycle * Identify the components of an API Connect on-premises cloud * Describe the use of the Cloud Manager user interface to administer the cloud topology and resources * Describe the different gateway types for securing and managing APIs * Identify the requirements for installing an API Connect on-premises cloud |

|  |  |
| --- | --- |
| Exercise 1. Review the API Connect development and runtime environment  Duration: 45 minutes | |
| Overview | In the first part of the exercise, you test that you can access the Internet and that your private domain name service is working. You review and validate that the Kubernetes runtime environment and API Connect processes are running. Then, you sign on as the administrator to the Cloud Manager user interface. Review the services that are defined in the Cloud Manager. Review the provider organization that publishes the APIs. |
| Learning objectives | After completing this exercise, you should be able to:   * Test the operation of the private DNS on the image * Review the Kubernetes runtime components * Ensure that the API Connect pods are operational * Sign on to the Cloud Manager graphical interface * Review the provider organization in Cloud Manager * Review the settings in Cloud Manager. |

|  |  |
| --- | --- |
| Unit 2. API Connect development platform  Duration: 30 minutes | |
| Overview | This unit describes the purpose of the API Connect Developer Toolkit to create APIs and applications in the LoopBack framework. Learn the different approaches to creating API definitions and API applications. |
| Learning objectives | After completing this unit, you should be able to:   * Explain the purpose of the API Connect Developer Toolkit * Identify the different installation options for the API Connect Developer Toolkit * Identify the features of the apic lb command-line utility * Describe how to create APIs in API Connect |

|  |  |
| --- | --- |
| Unit 3. Creating an API definition  Duration: 1 hour | |
| Overview | This unit examines the API definition, a structured file that documents the API operations, parameters, and data types. You learn how to define your API interface. You also examine the role of the extensions that API Connect adds to the OpenAPI definition and how message processing policies are defined in the API assemble view. |
| 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 * Describe how to create an API definition * Describe the purpose of the target-url property * Define an API operation * Define query and path parameters * Define request and response messages * Describe the IBM API Connect extensions to the OpenAPI definition * Describe the message processing policy assembly * Identify the endpoint URL that gets called by the invoke message processing policy |

|  |  |
| --- | --- |
| Exercise 2. Create an API definition from an existing API  Duration: 1 hour | |
| Overview | This exercise covers how to define an API interface from an existing API. You review the API operations, parameters, and definitions from the API Manager web application. You also publish and test the API from the API Manager test feature. |
| Learning objectives | After completing this exercise, you should be able to:   * Review an existing API service endpoint * Create an API definition in API Manager * Review the operations, properties, and schema definitions in an API definition * Test the API GET operation in the assembly test facility * Create a POST operation for the existing service endpoint * Test the API POST operation in the assembly test facility. |

|  |  |
| --- | --- |
| Unit 4. Defining APIs that call REST and SOAP services  Duration: 1 hour | |
| Overview | With API Manager, 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 REST service in an API definition * Explain how to expose an existing SOAP service in an API definition |

|  |  |
| --- | --- |
| Exercise 3. Define an API that calls an existing SOAP service  Duration: 30 minutes | |
| 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 use the API Manager feature to create an API definition from an existing WSDL service. The imported WSDL defines the API paths and methods that map to SOAP web service operations, and map SOAP message types to API data types. You test the SOAP API in the test feature of API Manager. |
| Learning objectives | After completing this exercise, you should be able to:   * Review the SOAP sample * Download the WSDL file * Create an API definition that invokes an existing WSDL service * Review the assembly in API Manager * Test the SOAP API on the DataPower gateway. |

|  |  |
| --- | --- |
| 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 NodeJS. You examine the different types of LoopBack connectors and how they support the generation of application code. Learn how to generate an application scaffold with the Developer Toolkit. Implement an API that calls a REST back-end service with the LoopBack REST connector. |
| Learning objectives | After completing this unit, you should be able to:   * Describe what is LoopBack * Describe the function of LoopBack connectors and code generation * Create the application scaffold from the apic command-line utility * Implement an API with a NodeJS LoopBack application * Test the LoopBack application and operation with the LoopBack Explorer |

|  |  |
| --- | --- |
| Exercise 4. Create a LoopBack application  Duration: 45 minutes | |
| Overview | In this exercise, you build a LoopBack application to implement an API. You generate the application scaffold with the apic command-line utility. You examine the generated files, the model, and properties of the Loopback application. You run the generated Node application and test the operations with the Loopback API Explorer. |
| Learning objectives | After completing this exercise, you should be able to:   * Create an application scaffold with the apic command-line utility * Examine LoopBack models and properties * Test the application with the Loopback API Explorer browser-based web client. |

|  |  |
| --- | --- |
| 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, properties, and relationships in the API Connect Developer toolkit 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 an application that uses the built-in memory connector * Build model instances from unstructured data |

|  |  |
| --- | --- |
| Exercise 5. Define 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 LoopBack 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 LoopBack 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 6. Implement event-driven functions with remote and operation hooks  Duration: 45 minutes | |
| Overview | In this exercise, you add custom logic to the generated LoopBack models. You extend the inventory model with remote hooks: preprocessing and post-processing functions that change the behavior of API operation calls. You also create two operation hooks: event listeners that the LoopBack framework triggers when it accesses or modifies persisted model objects. |
| Learning objectives | After completing this exercise, you should be able to:   * Define an operation hook in a LoopBack application * Define a remote hook in a LoopBack application * Test remote and operation hooks in the LoopBack Explorer |

|  |  |
| --- | --- |
| Unit 9. Assembling message processing policies  Duration: 1 hour | |
| 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 Manager. |
| 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 * Identify the policies that the DataPower API gateway type supports |

|  |  |
| --- | --- |
| Exercise 7. Assemble message processing policies  Duration: 1 hour and 30 minutes | |
| Overview | This exercise explains how to create message processing policies. You define a sequence of policies in the assembly view of the API Manager. 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:   * Import an OpenAPI definition into API Manager * Test the OpenAPI definition in API Manager * 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 * Create an assembly with a switch that has a flow for each API operation * Define a gateway script with an API assembly that saves a variable and calls an external map service * Test DataPower gateway policies in the API Manager assembly 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: 45 minutes | |
| 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 and configure a Native OAuth Provider in either the Cloud Manager or API Manager graphical applications. |
| Learning objectives | After completing this unit, you should be able to:   * Explain the concept of an OAuth provider * Describe the steps to secure an API with OAuth 2.0 * Identify the OAuth Provider types * Explain how to create a Native OAuth Provider * Explain the OAuth flow and grant types * Explain the difference between public and confidential schemes * Describe how to configure security settings for an API |

|  |  |
| --- | --- |
| Exercise 8. Declare 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 and the API resource server. You define a Native OAuth 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 a Native OAuth provider in the API Manager graphical application * Configure the client ID and client secret security definition * Declare and enforce an OAuth 2.0 security definition with the API Manager graphical 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. You learn how to deploy API implementations to a Docker image and run the API in a Docker container. |
| 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 |

|  |  |
| --- | --- |
| Exercise 9. Deploy an API implementation to a container runtime environment  Duration: 30 minutes | |
| Overview | In this exercise, you deploy the inventory LoopBack application to a Docker container runtime environment. You test that the inventory application that runs in the Docker container. |
| 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 * Create a Docker image by using the docker build command with the Dockerfile * Verify that the API runs on the Docker image. |

|  |  |
| --- | --- |
| Unit 13. Publishing and managing products and APIs  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. The product 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 * Describe the steps to publish a product * Describe the Developer Toolkit interface for managing products and APIs * Explain the lifecycle states for products and APIs |

|  |  |
| --- | --- |
| Exercise 10. Define and publish an API product  Duration: 30 minutes | |
| Overview | This exercise examines how to publish APIs with plans and products. You create a product and a plan, and deploy the product in API Manager. |
| Learning objectives | After completing this exercise, you should be able to:   * Create a product and plan in the API Manager * Add the APIs to the product * Verify that the invoke URL for the inventory application routes to the Docker image * Publish the product to the Sandbox 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 creates 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 * Describe the Developer Portal self-registration process for development catalogs * Explain how to add an application in the Developer Portal * Describe the role of client ID and client secret for application identification * Describe how to subscribe to an API plan * Describe the subscription approval process * Explain the test client features in the Developer Portal |

|  |  |
| --- | --- |
| Exercise 11. Subscribe and test APIs  Duration: 1 hour | |
| Overview | In this exercise, you learn about the application developer experience in the Developer Portal. You review the consumer organization that is created for you. You sign on to the Developer Portal as the owner of the consumer organization. You review the published products and APIs. You register an application that uses the product and APIs. You review the client ID and client secret values, subscribe to an API plan, and test operations from an API product. Finally, you test all the APIs from a web-based consumer application. |
| Learning objectives | After completing this exercise, you should be able to:   * Review the consumer organizations of the Sandbox catalog in API Manager * Review the portal settings for the Sandbox catalog * Sign on to the Sandbox catalog Developer Portal as the owner of the consumer organization * Register an application in the Developer Portal * Review the client ID and client secret values * Test API operations in the Developer Portal * Test all the APIs that you created with a web-based consumer application |

|  |  |
| --- | --- |
| Exercise 12. Troubleshooting the case study - optional  Duration: 30 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. This exercise is only required if the student is not getting the correct results. |
| 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 * Troubleshoot general issues when running through the course exercises * Useful Kubernetes information and commands |

|  |  |
| --- | --- |
| Exercise 13. Assemble the shipping message processing policy - optional  Duration: 45 minutes | |
| Overview | This exercise explains how to create a message processing policy by mapping data from multiple APIs into a single response. You complete the shipping API that was imported in an earlier exercise. |
| Learning objectives | After completing this exercise, you should be able to:   * Review an existing API definition in the assembly editor * Map data from multiple API calls into an aggregate response * Publish the product that contains the API * Test the API in the Developer Portal * Test the API with the consumer web application |

|  |  |
| --- | --- |
| Unit 15. Course summary  Duration: 5 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: http://bit.ly/IBMTrainEN

YouTube: youtube.com/IBMTraining

Facebook: facebook.com/ibmtraining

Twitter: twitter.com/websphere\_edu