IBM Systems

Create, Secure, and Publish APIs with IBM API Connect 10

WD515 (Classroom)
ZD515 (Self-paced)

Course description

This course teaches you how to configure a newly built API Connect 10 environment. You are taught how to configure a catalog with the gateway, portal, and analytics services and set up the environment for API development. You then define API interfaces according to the OpenAPI specification. You build SOAP and REST based APIs along with a GraphQL API. You assemble message processing policies and define client authorization schemes, such as OAuth 2.0, in the API definition. You verify the proper sequencing of policies in the assembly tester and further test your APIs in the new Test tab and Local Test Environment. After building and testing your APIs, you publish them and make them available on the Developer Portal. You manage all aspects of the provider organization in the API Manager user interface to create, publish, version, and retire API artifacts such as products, plans and APIs themselves. You also learn how to manage consumer organizations who use the APIs that are made available on the Developer Portal. You learn how to add members to the consumer organization that provides access to the APIs on the Developer Portal. You learn how the layout of the Developer Portal can be customized. Finally, you call the APIs on the secure gateway and you view the graphs and metrics of API usage.

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 3.0

Product and version

IBM API Connect 10.0.1

Audience

This course is designed for API developers.

Learning objectives

After completing this course, you should be able to:

- Configure services in Cloud Manager for an on-premises installation of API Connect
- Create a catalog and Developer Portal
- Create consumer and provider organizations
- Create, test, and publish SOAP, REST, and GraphQL APIs
- 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 provider
- Perform advanced testing of APIs by using the Test tab and the Local Test Environment
- Define products and plans in API Manager
- Stage, publish, version, migrate, deprecate, and retire products and APIs
- Manage member roles and permissions in the Developer Portal
- Create an application and subscribe to a plan
- Review API analytics in the Developer Portal
- Review analytics dashboards and visualizations in API Manager
- Customize the Developer Portal

Prerequisites

- Basic understanding of web services and protocols
- Basic understanding of application programming
- Conceptual knowledge of APIs
- Basic understanding of Red Hat Linux

Duration

5 days

Skill level

Intermediate

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 is a new course.

Course agenda

Course introduction
Duration: 15 minutes

Unit 1. Introduction to IBM API Connect 10
Duration: 1 hour and 30 minutes

Overview	This unit explains the scope and purpose of IBM API Connect 10 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. You also learn how to manage security, configure the cloud topology, register services and set up organizations and catalogs for an API Connect installation.
Learning objectives	After completing this unit, you should be able to: • Describe the key capabilities of API Connect • Describe what an API is and the different classifications of APIs • Explain the key benefits and use cases of API management • Describe how API Connect manages APIs 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 • Review the topology of an API Connect cloud • Explain how DataPower secures the API Gateway • Describe the roles and activities involved in the development of an API • Identify the requirements for installing an API Connect on-premises cloud • Describe the API Connect user interfaces by function • Identify deployment options for API Connect at installation • Describe the function of the installation assist utility • Identify the components of the runtime environment

Exercise 1. Reviewing the API Connect development and runtime environments
Duration: 1 hour

Overview	In this 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 and review the cloud topology.
Learning objectives	After completing this exercise, you should be able to: • Review the network connectivity and domains • Review the Kubernetes certificates • Review the Kubernetes runtime environment • Review the API Connect installation file • Review how notifications are configured • Review the configured services in Cloud Manager Console • Review the provider and consumer organization settings and user registries

Unit 2. Managing catalogs and organizations
Duration: 1 hour

Overview	Users in consumer organizations subscribe to products, plans, and APIs that you create in API Connect. In this unit, you learn how to define a catalog and Developer Portal in API Manager. You see where the Developer Portal user registry is defined. You then create a consumer organization in the API Manager and review the Developer Portal user interface.
Learning objectives	After completing this unit, you should be able to: • Describe the interaction between organizations and catalogs • Explain the concept of a provider organization • Explain how to create a catalog and a Developer Portal • Describe the use of spaces within a catalog • Configure a Developer Portal for the catalog • Identify the administration menu options in the Developer Portal • Describe the relationship between the provider organization owner and the owner of the consumer organization • Describe how to create a consumer organization • Describe the management options that are available to the owner of a consumer organization in the Developer Portal • Describe how to add a member in the Developer Portal • Describe the consumer roles that are defined in API Manager • Identify the roles that are defined in the Developer Portal • Explain the password lockout criteria

Exercise 2. Managing catalogs and consumer organizations
Duration: 2 hours

Overview	This exercise shows you how to manage consumer organizations through the API Manager and Developer Portal web interfaces. You review the role of the provider organization owner in creating a consumer organization. You also learn how to manage members and configure member roles and permissions in the Developer Portal. You first create a Staging catalog and configure the settings for the Developer Portal. You sign on to the Developer Portal with the admin user to validate the Portal installation. You then create a consumer organization and sign on to the Developer Portal as the owner of the consumer organization and manage the resources.
Learning objectives	After completing this exercise, you should be able to: • Create a catalog • Configure settings for the Developer Portal • Define a Developer Portal and user registry in API Manager • Activate the admin user for the Developer Portal • Configure modules in the Developer Portal • Create a consumer organization in API Manager • Add a member to the consumer organization • Respond to the email message to activate the app developer member • Manage member roles and permissions in the Developer Portal

Unit 3. Defining APIs in API Manager
Duration: 1 hour and 30 minutes

Overview	This unit provides an overview of APIs and API types. It describes the structure of an API definition and how to create a new API definition in API Manager. It explains the role of the DataPower gateway in exposing existing web services. It also covers how to edit and test an API definition in API Manager. Options for defining SOAP APIs are covered in more detail.
Learning objectives	After completing this unit, you should be able to: • Explain the concept of an API definition • Describe how to create an API definition • Define an API operation • Identify the endpoint URL that gets called by the invoke message processing policy • Describe the purpose of the Assemble view in API Manager • Explain how to test API operations in API Manager

Exercise 3. Defining an API that calls an existing SOAP service
Duration: 1 hour and 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 • 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 4. Defining a REST API in API Manager
Duration: 1 hour

Overview	This unit describes the options for defining REST APIs in API Manager and examines the API definition file for an OpenAPI specification. You learn how to define a REST API interface for a target service endpoint. You 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. You learn the HTTP methods for REST operations and how to create a GET and POST operation in API Manager.
Learning objectives	After completing this unit, you should be able to: • Examine the OpenAPI 2.0 definition file • Explain the purpose of the OpenAPI definition • Describe the IBM API Connect extensions to the OpenAPI definition • Explain how to create a REST API for a target service • Describe the purpose of the target-url property • Define query and path parameters • Define request and response messages • Describe the message processing policy assembly • List the HTTP methods in REST architecture • Add a SWITCH policy to an API assembly • Define a GET operation • Define a POST operation

Exercise 4. Defining a REST API from a target service
Duration: 1 hour and 30 minutes

Overview	This exercise covers how to define a REST API interface from a target service. First, you review the structure of the operations you call in your API on the target service web site. Then, you build the API operations, parameters, and definitions in 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 endpoint • Create an API definition in API Manager • Review the operations, properties, and schema definitions in an API definition • Create a GET operation for the existing service endpoint • Test the API GET operation in the assembly test facility • Create an assembly with a switch that has a flow for each API operation • Create a POST operation for the existing service endpoint • Test the API POST operation in the assembly test facility

Unit 5. 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 • Explain the difference between a global-scoped user-defined policy and a catalog-scoped user-defined policy • Describe when and how to change the version of an API

Exercise 5. Assembling message processing policies
Duration: 2 hours 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: • Create a new the version of an API • Configure an API to call a SOAP service and return the result as a JSON object • Define input and output parameters in a map policy • Map responses from multiple API calls into a single response • Redact specific fields from the response body to obfuscate sensitive data

Unit 6. 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 • Describe the purpose of user registries • Identify the types of supported user registries in Cloud Manager • Describe the role of Transport Layer Security (TLS) profiles • 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 7. 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 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 6. Implementing OAuth 2.0 security
Duration: 1 hour and 30 minutes

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: • Create a user registry for use with an OAuth provider • Create a native OAuth provider and make it available within the catalog • Add OAuth security to an API • Update the Sandbox Test App (client application) to provide an OAuth redirect for testing • Test the OAuth security by invoking the secured API

Unit 8. Testing and debugging APIs
Duration: 30 minutes

Overview	Before you publish an API where customers can access it, you need to test it and ensure that it is defined and implemented correctly. IBM API Connect offers tools for running both simple and complex tests, in different environments. Up to this point in this course, you have been testing your APIs by using the Assembly tab so that you can ensure that your APIs are defined and implemented correctly. This unit covers more extensive testing and debugging options in IBM API Connect.
Learning objectives	After completing this unit, you should be able to: • Explain the testing and debugging features of API Manager • Describe what is required to test an API in the Test tab • Define the steps to test an API in the Test tab • Explain how to activate an API • Explain the purpose of the Endpoints tab

Exercise 7. Introduction to the Test tab
Duration: 1 hour

Overview	This exercise covers the use of the Test tab to test your APIs. Up to this point in this course, you have been using the Assembly tab to perform simple testing of your APIs.In this exercise, you test APIs you built in prior exercises. As an introduction to the Test tab, you use the Test tab to test a SOAP API (InventoryService) and REST API (petstore). In a later exercise, you use the Test tab to test a GraphQL API.
Learning objectives	After completing this exercise, you should be able to: • Use the Test tab to test and debug a SOAP API • Use the Test tab to test and debug a REST API

Unit 9. Creating and testing a GraphQL API
Duration: 1 hour

Overview	This unit describes the process of creating and testing a GraphQL API. You examine the definition of a GraphQL API, its advantages and disadvantages, and the differences between GraphQL APIs and REST APIs. You learn how to create a GraphQL API in API Manager. You learn the definition of a GraphQL schema and how to query a GraphQL API by using queries and mutations. You learn how to test a GraphQL API by using the Test tab in API Manager.
Learning objectives	After completing this unit, you should be able to: • Explain what is GraphQL API • Advantages and Disadvantages of GraphQL API • Compare and Contrast REST and GraphQL • Describe how to create a GraphQL API • Define a GraphQL API query • Describe how to test a GraphQL API with the Test tab • Develop a GraphQL schema

Exercise 8. Creating and testing a GraphQL API
Duration: 3 hours

Overview	In this exercise, you build a GraphQL API that proxies a backend GraphQL server. You query and modify the schema with the GraphQL Playground. You modify types and fields of the GraphQL schema in the API Connect GraphQL schema tab. You test the GraphQL API with the GraphiQL editor in the API Connect Test tab.
Learning objectives	After completing this exercise, you should be able to: • Create a GraphQL API that proxies a back end GraphQL server • Send POST and GET operations in the Test tab GraphiQL editor to test the GraphQL API • Trace the response of a GraphQL query in the Test tab • Modify settings for types and fields in a GraphQL schema • Apply an @remove directive to types and fields in a GraphQL schema • Replace and download GraphQL schemas

Unit 10. Testing an API in the Local Test Environment
Duration: 1 hour

Overview	This unit describes the process of testing an API in the Local Test Environment. You learn the definition and uses of a Local Test Environment, as well as how to start and install the Local Test Environment. You learn how to test a REST API in the Local Test Environment by using a cURL call. You learn the definition and uses of a TLS Client profile in the Local Test Environment, as well as how to create a TLS Client profile.
Learning objectives	After completing this unit, you should be able to: • Explain what is the Local Test Environment • Describe how to install the Local Test Environment • Describe how to start the API Designer in the Local Test Environment • Describe how to test an API in the Local Test Environment • Describe how to create a TLS Client profile in the Local Test Environment

Exercise 9. Testing an API in the Local Test Environment
Duration: 2 hours

Overview	In this exercise, you test an API on the Local Test Environment (LTE) on your local machine. You invoke a REST API from the API Designer UI application running in Online mode. You call the API in the Local Test Environment with a cURL command. You create a TLS profile to securely authenticate your API.
Learning objectives	After completing this exercise, you should be able to: • Install the Local Test Environment • Start the API Designer in the Local Test Environment • Test the API in the Local Test Environment with a cURL command • Create a TLS Client profile in the Local Test Environment

Unit 11. Publishing and managing products and APIs
Duration: 45 minutes

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 • Explain the lifecycle states for products and APIs

Exercise 10. Define and publish an API product
Duration: 45 minutes

Overview	This exercise examines how to publish APIs with plans and products. You create a product 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 • Publish the product to the Staging catalog

Unit 12. The product lifecycle
Duration: 2 hours

Overview	This unit explains the concept of the Product lifecycle. The lifecycle management feature controls the staging of a Product version to a catalog. Lifecycle management continues through publishing to make the Product version available to your application developers. The lifecycle governance eventually controls retiring and archiving of the Product and APIs.
Learning objectives	After completing this unit, you should be able to: • Describe provider organization roles and permissions • Explain the product lifecycle stages • Describe how staging and publishing differ in development and production catalogs • Describe how lifecycle events are managed in API Manager • Explain the product availability and visibility settings • Describe how to create versions of products • Explain the concept of replacing and superseding published products • Explain how to migrate application subscriptions to a new product version and plan • Explain how application subscriptions are created in API Manager • Describe the state changes that occur when approvals are enabled

Exercise 12. Subscribing and testing APIs in the Developer Portal
Duration: 45 minutes

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 and Staging catalogs • Review the portal settings for the Sandbox and Staging catalogs • Register an application in the Developer Portal • Review the client ID and client secret values • Test API operations in the Developer Portal

Exercise 11. Managing and approving API Products
Duration: 1 hour and 15 minutes

Overview	This exercise shows you how the Product lifecycle is managed in API Manager. You review Product and API availability and visibility settings and create and plans. You configure lifecycle settings and approval settings for a catalog. You examine how to define a user for the provider organization. You manage Product and API versions. You publish artifacts to the Staging catalog, and then review and approve the lifecycle stage for a published Product.
Learning objectives	After completing this exercise, you should be able to: • Review product availability and visibility settings in API Manager • Create and configure plans • Review the roles and members of the provider organization • Create a provider organization member with the developer role • Sign into API Manager with the owner role • Configure lifecycle and approval settings • Publish a Product and APIs to the Staging catalog • Create a version of the API and Product • Approve a published Product

Unit 13. Subscribing and testing APIs in the Developer Portal
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

Unit 14. API Analytics
Duration: 1 hour

Overview	This unit describes the API analytics features in IBM API Connect. API analytics is built on the Kibana open source analytics and visualization platform. You review some default dashboards and visualizations that are provided with the API Connect analytics service
Learning objectives	After completing this unit, you should be able to: • Describe what is API Connect analytics • Describe the role of the Kibana open source platform in the API Connect API analytics feature • Describe where analytics are configured and captured • Identify which user interfaces in API Connect provide access to analytical data • Role defaults required to view analytics in the Developer Portal • Describe the purpose of default dashboards • Review the features of default visualizations • Create a visualization • Describe API events and event records • Describe how to export analytics and API event data

Exercise 13. Calling an API on the gateway and monitoring API usage
Duration: 1 hour and 15 minutes

Overview	This exercise covers....
Learning objectives	After completing this exercise, you should be able to: • Run the test feature in the Developer Portal • Identify the API endpoints in the gateway • Run a script to generate multiple calls to the API gateway • View the analytics dashboard for the catalog • Change the time period filter for a visualization • View API event data • Export API event data

Unit 15. Customizing the Developer Portal
Duration: 1 hour

Overview	As the administrator, you can change the appearance and layout of the Developer Portal. This unit describes the customization options that are available to you. You learn how to customize the Developer Portal through the administration menu and examine the options for using themes and sub-themes on the Developer Portal.
Learning objectives	After completing this unit, you should be able to: • Briefly explain the purpose of the Developer Portal • Explain the role of the Drupal open source project in the Developer Portal • Explain the concept of modules and themes • List the roles that are defined in the Developer Portal • Describe the Drupal terminology that is used when administering the portal • Describe the various ways to create a theme for the Developer Portal • Describe the use of sub-themes for customizing the standard API Connect Developer Portal theme

Exercise 14. Customizing the Developer Portal
Duration: 1 hour and 30 minutes

Overview	This exercise shows you the customization options in the Developer Portal. You sign in to the Developer Portal with a Portal administrator account, add and configure a Drupal sub-theme, and review some of the standard features of the Developer Portal.
Learning objectives	After completing this exercise, you should be able to: • Generate a Developer Portal sub-theme • Review and customize the sub-theme • Install the sub-theme on the Developer Portal • Review the forum features in the Developer Portal • Create a topic and add a new comment in a forum

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