

Exam 70-487: Developing Microsoft Azure and Web Services – Skills Measured

Audience Profile

Candidates for this certification are professional developers that use Visual Studio 2017 and the Microsoft® .NET Core Framework to design and develop Web solutions. Candidates should have a minimum of three to five years of experience developing ASP.NET MVC-based solutions. Additionally, candidates should be able to demonstrate the following:

- Experience designing and developing Web applications that access various (local and remote) data and services including Windows Azure
- Experience with the full software development life cycle of data and service solutions
- Experience developing and deploying to multi-tier environments, including Windows Azure
- Experience designing and developing asynchronous solutions
- Experience creating and consuming HTTP services

Skills Measured

NOTE: The bullets that appear below each of the skills measured are intended to illustrate how we are assessing that skill. This list is not definitive or exhaustive.

NOTE: In most cases, exams do NOT cover preview features, and some features will only be added to an exam when they are GA (General Availability).

Accessing Data (20-25%)

Choose data access technologies

- choose a data access technology, including ADO.NET, Entity Framework, WCF Data Services, and Azure Cosmos DB, based on application requirements

Implement caching

- cache static data, apply cache policies, including policy expirations; use CacheDependency to refresh cache data; query notifications; implement caching using Redis

Implement transactions

- manage transactions by using the API from System.Transactions namespace; implement distributed transactions including distributed transaction on SQL Azure; specify a transaction isolation level

Implement data storage in Microsoft Azure

- access data storage in Windows Azure; choose a data storage mechanism in Microsoft Azure, including blobs, tables, queues, Azure SQL, and Cosmos DB; distribute data by using the Content Delivery Network (CDN) and Azure File Sync; handle exceptions by using retries; use Elastic client library with Azure SQL

Create and implement a WCF Data Services service

- address resources; implement filtering; create a query expression; access payload formats, including JSON; use data service interceptors and service operators; version a data service; implement data services providers; host the dataservice; use actions to implement server-side behavior

Querying and Manipulating Data by Using the Entity Framework (20-25%)

Query and manipulate data by using the Entity Framework

- query, update, and delete data by using DbContext; build a query that uses deferred execution; implement lazy loading and eager loading; create and run compiled queries; query data by using Entity SQL; create global query filters

Query and manipulate data by using Data Provider for Entity Framework

- query and manipulate data by using Connection, DataReader, and Command objects from the System.Data.EntityClient namespace; perform synchronous and asynchronous operations; manage transactions (API)

Query data by using LINQ to Entities

- query data by using LINQ operators, including project, skip, aggregate, filter, and join; log queries; implement query boundaries

Query and manipulate data by using ADO.NET

- query and manipulate data by using Connection, DataReader, and Command objects; perform synchronous and asynchronous operations; manage transactions

Create an Entity Framework data model

- structure a data model using Table per type, table per class, and table per hierarchy; select and implement an approach to manage a data model, including code first, model first, and database first; implement POCO objects; describe a data model by using conceptual schema definitions, storage schema definitions, and mapping languages, including CSDL, SSDL, and MSL

Implement Entity Framework with third party databases

- implement Entity Framework using MySQL and SQLite databases; design a strategy to manage differences between database capabilities; leverage database specific technologies, including ON DUPLICATE KEY using Entity Framework

Creating and Consuming Web API-based services (20-25%)

Design a Web API

- define HTTP resources with HTTP actions; plan appropriate URI space, and map URI space using routing; choose appropriate HTTP method to meet requirements; choose appropriate Web API formats for responses to meet requirements; plan when to make HTTP actions asynchronous

Implement a Web API

- accept data in JSON format; use content negotiation to deliver different data formats to clients; define actions and parameters to handle data binding; implement dependency injection to create more flexible applications; implement action filters and exception filters to manage controller execution; implement asynchronous and synchronous actions; implement streaming actions; implement middleware

Secure a Web API

- implement Identity for authentication; implement authorization using roles, claims, and custom authorization; implement Data Protection APIs; enable cross-domain requests (CORS); prevent cross-site request forgery (XSRF); implement and extend authorization filters to control access to applications

Host and manage Web API

- host Web API in IIS; self-host a Web API in your own process; host Web API in Kestrel; host services in a Windows Azure Web App; host services in Docker containers; configure the host server for streaming; work with a hosting environment

Consume Web API web services

- consume Web API services by using HttpClient; send and receive requests in different formats; handle retry logic; implement content negotiation; use Swagger to construct Uris and payloads; use AutoRest to build clients

Designing and Implementing Web Services (15-20%)

Consume a WCF service

- generate proxies by using SvcUtil; generate proxies by creating a service reference; create and implement channel factories; configure WCF services by using configuration settings; create and configure bindings for WCF services; relay bindings to Azure using service bus endpoints; integrate with the Azure service bus relay

Implement serverless Azure Web Services

- host web services using App Services, including Logic Apps and API Apps; design and implement Azure Function based services; design and implement Azure Web Jobs; design and implement Service Fabric based web services; implement schedule-based processing in a serverless environment

Implement traffic management in Azure

- implement Azure Load Balancer, including scaling; implement Azure Application Gateway; implement Azure Traffic Manager; design for multiple regions; leverage Azure CDN for caching web services; implement Log Analytics

Implement Azure API Management

- secure Web Services using certificates, Azure Active Directory, and OAuth; define and implement policies, including secrets, caching, external services, monitoring and throttling; define API interface using the Azure Portal and Swagger; manage running services using logging, disaster recovery, and multiple regions

Monitor web services

- collect logs and metrics using Azure Event Hubs; process logs and metrics using Azure Event Hubs, Stream Analytics, and Machine Learning; use Azure App Insights to monitor and troubleshoot web services

Deploying Web Applications and Services (15-20%)

Design a deployment strategy

- deploy using Web Deploy; deploy using Web Publishing in Visual Studio; deploy a web application by using XCopy; automate a deployment from VSTS or TFS Build Server; deploy a web application to a Docker container; design a continuous deployment pipeline; deploy using cloud sync

Choose a deployment strategy for Azure

- determine appropriate service; perform an in-place upgrade and deployment slot swap; create staging environments; configure an upgrade domain; create and configure input and internal endpoints; specify operating system configuration; implement ARM templates to customize deployment; deploy to Service Fabric; deploy to Azure Stack

Configure a web application for deployment

- switch from production or release mode to debug mode; transform configuration files ; configure Azure configuration settings; configure Azure Key Vault for application secrets; configure deployment credentials for Azure App Service

Manage packages by using NuGet

- create and configure a NuGet package; install and update an existing NuGet package; resolve versioning conflict issues; connect to a local repository cache for NuGet, set up your own package repository; manage NuGet dependencies

Share assemblies between multiple applications and servers

- prepare the environment for use of assemblies across multiple servers; sign assemblies by using a strong name; deploy assemblies to the global assembly cache; implement assembly versioning; create an assembly manifest; configure assembly binding redirects; target netstandard for cross platform libraries