EDUSUM

#1 Online Certification Guide

70-487

MCSD App Builder

A Success Guide to Prepare-
Developing Microsoft Azure and Web Services

edusum.com

Table of Contents

Introduction to 70-487 Exam on Developing Microsoft Azure and Web Services

Use this quick start guide to collect all the information about Developing Microsoft Azure and Web Services (70-487) Certification exam. This study guide provides a list of objectives and resources that will help you prepare for items on the 70-487 Developing Microsoft Azure and Web Services exam. The Sample Questions will help you identify the type and difficulty level of the questions and the Practice Exams will make you familiar with the format and environment of an exam. You should refer this guide carefully before attempting your actual Microsoft MCSD App Builder certification exam.

The Developing Microsoft Azure and Web Services certification is mainly targeted to those candidates who want to build their career in Microsoft Visual Studio domain. The Microsoft Certified Solutions Developer (MCSD) - App Builder exam verifies that the candidate possesses the fundamental knowledge and proven skills in the area of Microsoft MCSD App Builder.

Microsoft 70-487 Certification Details:

Exam Name	Microsoft Certified Solutions Developer (MCSD) - App Builder
Exam Code	70-487
Exam Price	$165 (USD)
Duration	120 min
Number of Questions	45-55
Passing Score	700 / 1000
Books / Training	20487B
Schedule Exam	Pearson VUE
Sample Questions	Developing Microsoft Azure and Web Services Sample Questions
Practice Exam	**Microsoft 70-487 Certification Practice Exam**

Microsoft 70-487 Exam Syllabus:

Topic	Details	Weights
Accessing data	Choose data access technologies - Choose a technology (ADO.NET, Entity Framework, WCF Data Services, Azure storage) based on application requirements Implement caching - Cache static data, apply cache policy (including expirations); use CacheDependency to refresh cache data; query notifications Implement transactions - Manage transactions by using the API from System.Transactions namespace; implement distributed transactions; specify transaction isolation level Implement data storage in Azure - Access data storage in Azure; choose data storage mechanism in Azure (blobs, tables, queues, SQL Database); distribute data by using the Content delivery network (CDN); handle exceptions by using retries (SQL Database); manage Azure Caching 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 Manipulate XML data structures - Read filter, create, modify XML data structures; Manipulate XML data by using XMLReader, XMLWriter, XMLDocument, XPath, LINQ to XML; transform XML by using XSLT transformations	20-25%
Querying and manipulating data by using Entity Framework	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; perform asynchronous operations using Entity Framework; map a stored procedure Query and manipulate data by using Data Provider for Entity Framework	20-25%

Topic	Details	Weights
	- Query and manipulate data by using Connection, DataReader, and Command from the System.Data.EntityClient namespace; perform synchronous and asynchronous operations; manage transactions (API); programmatically configure a Data Provider Query data by using LINQ to Entities - Query data by using LINQ operators (for example, project, skip, aggregate, filter, and join); log queries and database commands; implement query boundaries (IQueryable vs. IEnumerable); implement async query Query and manipulate data by using ADO.NET - Query and manipulate data by using Connection, DataReader, Command, DataAdapter, DataSet; perform synchronous and asynchronous operations; manage transactions (API) Create an Entity Framework data model - Structure the data model using table per type, table per class, table per hierarchy; choose and implement an approach to manage a data model (code first vs. model first vs. database first); implement POCO objects; describe a data model by using conceptual schema definitions, storage schema definition, mapping language (CSDL, SSDL, MSL), and Custom Code First Conventions	
Designing and implementing WCF Services	Create a WCF service - Create contracts (service, data, message, callback, and fault); implement message inspectors; implement asynchronous operations in the service Configure WCF services by using configuration settings - Configure service behaviors; configure service endpoints; configure bindings including WebSocket bindings; specify a service contract; expose service metadata (XSDs, WSDL, and metadata exchange endpoint); configure message compression and encoding Configure WCF services by using the API - Configure service behaviors; configure service endpoints; configure binding; specify a service contract; expose service metadata (XSDs, WSDL, and metadata exchange); WCF routing and discovery features	15-20%

Topic	Details	Weights
	Secure a WCF service - Implement message level security, implement transport level security; implement certificates; design and implement multiple authentication modes Consume WCF services - Generate proxies by using SvcUtil; generate proxies by creating a service reference; create and implement channel factories Version a WCF service - Version different types of contracts (message, service, data); configure address, binding, and routing service versioning Create and configure a WCF service on Azure - Create and configure bindings for WCF services (Azure SDK—extensions to WCF); relay bindings to Azure using service bus endpoints; integrate with the Azure service bus relay Implement messaging patterns - Implement one way, request/reply, streaming, and duplex communication; implement Azure Service Bus and Azure Queues Host and manage services - Manage services concurrency (single, multiple, reentrant); create service hosts; choose a hosting mechanism; choose an instancing mode (per call, per session, singleton); activate and manage a service by using AppFabric; implement transactional services; host services in an Azure worker role	
Creating and consuming Web API-based services	Design a Web API - Define HTTP resources with HTTP actions; plan appropriate URI space, and map URI space using routing; choose appropriate HTTP method (get, put, post, delete) to meet requirements; choose appropriate format (Web API formats) for responses to meet requirements; plan when to make HTTP actions asynchronous; design and implement routes Implement a Web API - Accept data in JSON format (in JavaScript, in an AJAX callback); use content negotiation to deliver different data formats to clients; define actions and parameters to handle data binding; use HttpMessageHandler to process client requests and	15-20%

Topic	Details	Weights
	server responses; implement dependency injection, along with the dependency resolver, to create more flexible applications; implement action filters and exception filters to manage controller execution; implement asynchronous and synchronous actions; implement streaming actions; implement SignalR; test Web API web services Secure a Web API - Implement HTTPBasic authentication over SSL; implement Windows Auth; prevent cross-site request forgery (XSRF); design, implement, and extend authorization and authentication filters to control access to the application; implement Cross Origin Request Sharing (CORS); implement SSO by using OAuth 2.0; configure multiple authentication modes on a single endpoint Host and manage Web API - Host Web API in an ASP.NET app; self-host a Web API in your own process (a Windows service) including Open Web Interface for .NET (OWIN); host services in an Azure worker role; restrict message size; configure the host server for streaming Consume Web API web services - Consume Web API services by using HttpClient synchronously and asynchronously; send and receive requests in different formats (JSON/HTML/etc.); request batching	
Deploying web applications and services	Design a deployment strategy - Create an IIS install package; deploy to web farms; deploy a web application by using XCopy; automate a deployment from TFS or Build Server Choose a deployment strategy for an Azure web application - Perform an in-place upgrade and VIP swap; configure an upgrade domain; create and configure input and internal endpoints; specify operating system configuration; deploy applications using Azure Web Site Configure a web application for deployment - Switch from production/release mode to debug mode; use SetParameters to set up an IIS app pool; set permissions and passwords; enable and monitor ASP.NET App Suspend; configure WCF endpoints	15-20%

Topic	Details	Weights
	(including HTTPS protocol mapping), bindings, and behaviors; transform web.config by using XSLT (for example, across development, test, and production/release environments); configure Azure configuration settings Manage packages by using NuGet - Create and configure a NuGet package; install and update an existing NuGet package; connect to a local repository cache for NuGet, set up your own package repository Create, configure, and publish a web package - Create an IIS InstallPackage; configure the build process to output a web package; apply pre- and post-condition actions to ensure that transformations are correctly applied; include appropriate assets (web content, certificates) Share assemblies between multiple applications and servers - Prepare the environment for use of assemblies across multiple servers (interning); 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 (for example, from MVC4 to MVC5)	

70-487 Sample Questions:

01. Which of the following is not a benefit of configuration?
a) Ability to change Endpoint URIs
b) Ability to modify security settings
c) Ability to redefine metadata exchange
d) Ability to add new method definitions

02. What are the advantages of a compiled query?
a) None. Every time you run the query you will hit the database.
b) The results of the query are cached making the querying a lot faster.
c) The translation of your query into SQL is cached.
d) None. You can't change any of the parameters you use in a query, rendering the compiled query useless.

03. Which of the following are true about SqlConnection objects?
a) They should be opened at the last possible moment and closed at the earliest opportunity.
b) They should be opened as early as possible and closed as early as possible.
c) They should be opened as late as possible and closed as late as possible.
d) As long as the Close statement is called in a catch block, the object will be closed and disposed of correctly.

04. What are the minimum steps that must be performed to configure a service using the WCF Service Configuration Editor?
a) Specify a new service.
b) Specify a MetadataExchange endpoint.
c) Specify a security configuration.
d) Specify a contract.

05. Which of the following are requirements for a valid and well-formed XML document?
a) Every element must have a corresponding closing element.
b) Every element must have at least one attribute.
c) Every attribute must have a corresponding closing attribute.
d) Elements and attributes can be used interchangeably as long as they have open and closing tags.

06. Which values are valid choices when defining the Priority property of the CacheItemPolicy using the System.Runtime.Caching version of the Cache?
a) Normal
b) High
c) NotRemovable
d) Low

07. Which of the following are valid transport binding elements when defining a custom binding?
a) TcpTransportBindingElement
b) TcpBinding
c) HttpTransportBindingElement
d) MTOMTransportBindingElement

08. Which of the following items must be provided to use the self-host option?
a) AspnetSelfHostConfiguration
b) HttpSelfHostConfiguration
c) AspnetSelfHostServer
d) HttpSelfHostServer

09. What happens if you attempt to Attach an entity to a context when it already exists in the context with an EntityState of unchanged?
a) A copy of the entity is added to the context with an EntityState of unchanged.
b) A copy of the entity is added to the context with an EntityState of Added.
c) Nothing happens and the call is ignored.
d) The original entity is updated with the values from the new entity, but a copy is not made. The entity has an EntityState of Unchanged.

10. Which elements do you need from the management portal when configuring your host service?
a) Namespace
b) NetTcpRelayBinding
c) ServiceHost
d) Default Key

Answers to 70-487 Exam Questions:

Question: 01 Answer: d	Question: 02 Answer: c	Question: 03 Answer: b	Question: 04 Answer: a, d	Question: 05 Answer: a
Question: 06 Answer: c	Question: 07 Answer: a, c	Question: 08 Answer: b, d	Question: 09 Answer: c	Question: 10 Answer: a, d

Note: If you find any typo or data entry error in these sample questions, we request you to update us by commenting on this page or write an email on feedback@edusum.com