

我要—PASS



質 量 更 高 務 服 更 好

一年免費升級服務

<http://www.51-pass.com>

Exam : 70-513-Csharp

Title : TS: Windows
Communication Foundation
Development with
Microsoft .NET Framework 4

Version : DEMO

1.You create a Windows Communication Foundation (WCF) service and deploy it with wsHttpBinding and message security enabled.

You create an intermediate WCF service for logging messages sent to the primary service.The intermediate service is called via the clientVia endpoint behavior.

The primary service is receiving malformed data from a client application.

You need to enable inspection of the malformed data and prevent message tampering.

What should you do

A.Specify a protection level of None in the service contract for the intermediate service.Disable message and transport security from the client application configuration file.

B.Specify a protection level of Sign in the service contract for the intermediate service.Disable transport security from the client application configuration file.

C.Modify the binding on the intermediate service to use netNamedPipeBinding.

D.Modify the binding on the intermediate service to use webHttpBinding.

Answer: D

2.You are creating a Windows Communication Foundation (WCF) service.You do not want to expose the internal implementation at the service layer.

You need to expose the following class as a service named Arithmetic with an operation named Sum.

public class Calculator

```
{  
    public int Add(int x, int y)  
    {  
    }  
}
```

Which code segment should you use

A.[ServiceContract(Namespace="Arithmetic")]

public class Calculator

```
{  
    [OperationContract(Action="Sum")]  
    public int Add(int x, int y)  
    {  
    &  
    }  
}
```

B.[ServiceContract(ConfigurationName="Arithmetic")]

public class Calculator

```
{  
    [OperationContract(Action="Sum")]  
    public int Add(int x, int y)  
    {  
    &  
    }  
}
```

C.[ServiceContract(Name="Arithmetic")]

```
public class Calculator
{
    [OperationContract(Name="Sum")]
    public int Add(int x, int y)
    {
        &
    }
}
D.[ServiceContract(Name="Arithmetic")]
public class Calculator
{
    [OperationContract(ReplyAction="Sum")]
    public int Add(int x, int y)
    {
        &
    }
}
```

Answer: C

3.You are developing a data contract for a Windows Communication Foundation (WCF) service. The data in the data contract must participate in round trips.Strict schema validity is not required. You need to ensure that the contract is forward-compatible and allows new data members to be added to it.

Which interface should you implement in the data contract class

- A.ICommunicationObject
- B.IExtension<T>
- C.IExtensibleObject<T>
- D.IExtensibleDataObject

Answer: D

4.You are developing a Windows Communication Foundation (WCF) service that allows customers to update financial data.The service contract is defined as follows.(Line numbers are included for reference only.)

```
01[ServiceContract]
02public interface IDataUpdate
03{
04[OperationContract]
05[TransactionFlow(TransactionFlowOption.Mandatory)]
06void Update(string accountNumber, double amount);
07}
08
09class UpdateService : IDataUpdate
10{
11[OperationBehavior(TransactionScopeRequired = true,
```

```
TransactionAutoComplete = false)]  
12public void Update(string accountNumber,  
double amount)  
13{  
14&  
15}  
16}  
17
```

You need to ensure that the service is invoked within a transaction.

What should you do

A.Replace line 01 with the following code.

```
[ServiceContract(  
    SessionMode = SessionMode.NotAllowed)]
```

B.Replace line 01 with the following code.

```
[ServiceContract(  
    SessionMode = SessionMode.Required)]
```

C.Insert the following code at line 08.

```
[ServiceBehavior(  
    TransactionAutoCompleteOnSessionClose = false)]
```

D.Insert the following code at line 08.

```
[ServiceBehavior(  
    ReleaseServiceInstanceOnTransactionComplete = false)]
```

Answer: B

5.You are developing a client that sends several types of SOAP messages to a Windows Communication Foundation (WCF) service method named PostData.

PostData is currently defined as follows.

```
[OperationContract]  
void PostData(Order data);
```

You need to modify PostData so that it can receive any SOAP message.

Which code segment should you use

A.[OperationContract(IsOneWay = true, Action = "*", ReplyAction = "*")]

```
void PostData(Order data);
```

B.[OperationContract(IsOneWay = true, Action = "*", ReplyAction = "*")]

```
void PostData(BodyWriter data);
```

C.[OperationContract]

```
void PostData(BodyWriter data);
```

D.[OperationContract]

```
void PostData(Message data);
```

Answer: D

6.A class named TestService implements the following interface.

```
[ServiceContract]  
public interface ITestService {
```

```
[OperationContract]
DateTime GetServiceTime();
}
```

TestService is hosted in an ASP.NET application.

You need to modify the application to allow the GetServiceTime method to return the data formatted as JSON. It must do this only when the request URL ends in /ServiceTime.

What should you do

A. Add this attribute to the GetServiceTime method.

```
[WebInvoke(Method="POST")]
```

In the web.config file, add this element to system.serviceModel/behaviors/endpointBehaviors.

```
<behavior name="Json">
  <enableWebScript />
</behavior>
```

In the web.config file, configure TestService in the system.serviceModel/services collection as follows.

```
<service name="TestService">
  < endpoint address="/ServiceTime"
contract="TestService"
  behaviorConfiguration="Json"
  binding="webHttpBinding" />
</service>
```

B. Add this attribute to the GetServiceTime method.

```
[WebInvoke(Method = "GET",
  UriTemplate = "/ServiceTime",
  ResponseFormat = WebMessageFormat.Json)]
```

In the web.config file, configure TestService in the system.serviceModel/services collection as follows.

```
<service name="TestService">
  <endpoint address="/ServiceTime"
contract="TestService"
  binding="webHttpBinding" />
</service>
```

C. Add this attribute to the GetServiceTime method.

```
[WebGet(
  ResponseFormat= WebMessageFormat.Json,
  UriTemplate="/ServiceTime")]
```

Create a new .svc file named JsonVersion.svc with the following content.

```
<%@ ServiceHost Service="TestService"
  Factory="System.ServiceModel.Activation.WebServiceHostFactory" %>
```

D. Add this attribute to the GetServiceTime method.

```
[WebGet(UriTemplate = "{Json}/ServiceTime")]
```

Create a new .svc file named JsonVersion.svc with the following content.

```
<%@ ServiceHost Service="TestService"
  Factory="System.ServiceModel.Activation.WebServiceHostFactory" %>
```

Answer: C

7.You are creating a Windows Communication Foundation (WCF) service that implements operations in a RESTful manner.You need to add a delete operation.

You implement the delete method as follows.

```
void DeleteItems(string id);
```

You need to configure WCF to call this method when the client calls the service with the HTTP DELETE operation.

What should you do

A.Add the WebInvoke(UriTemplate = "/Items/{id}", Method="DELETE") attribute to the operation.

B.Add the HttpDelete attribute to the operation.

C.Replace the string parameter with a RemovedActivityAction parameter.

D.Replace the return type with RemovedActivityAction.

Answer: A

8.You are creating a Windows Communication Foundation (WCF) service that responds using plain-old XML (POX).

You have the following requirements:

You must enable the /catalog.svc/items operation to respond using the POX, JSON, or ATOM formats.You also must ensure that the same URL is used regardless of the result type.

You must determine the response format by using the Accepts HTTP header.

What should you do

A.Implement the IChannelInitializer interface in the service class.

B.Implement the System.Runtime.Serialization.IFormatterConverter interface in the service class.

C.Set the BodyStyle parameter of the WebGet attribute on the operation to WebMessageBodyStyle.WrappedResponse.

D.Set the return type of the operation to System.ServiceModel.Channels.Message.Use the current WebOperationContext methods to return the data in the required format.

Answer: D

9.A Windows Communication Foundation (WCF) solution uses two services to manage a shopping cart.Service A processes messages containing line items that total between \$0 and \$500.Service B processes messages containing line items that total more than \$500.

All messages are of equal importance to the business logic.

You need to route incoming messages to the appropriate services by using WCF routing.

Which two message filters should you add to the router (Each correct answer presents part of the solution.Choose two.)

A.a message filter with a priority of 100 that will forward messages that total between \$0 and \$500 to Service A

B.a message filter with a priority of 0 that will forward messages that total between \$0 and \$500 to Service A

C.a message filter with a priority of 0 that will forward all messages to Service B

D.a message filter with a priority of 100 that will forward all messages to Service B

Answer: AD

10.You have an existing Windows Communication Foundation (WCF) service.

You need to ensure that other services are notified when the service is started.

What should you do

A.Add the following standard endpoint to the service.

```
<endpoint name="udpAnnouncementEndpoint"
  kind="udpDiscoveryEndpoint" />
```

B.Add the following standard endpoint to the service.

```
<endpoint name="udpDiscoveryEndpoint"
  kind="udpAnnouncementEndpoint" />
```

C.Add a service behavior with the following element.

```
<serviceDiscovery>
  <announcementEndpoints>
    <endpoint kind="udpDiscoveryEndpoint" />
  </announcementEndpoints>
</serviceDiscovery>
```

D.Add a service behavior with the following element.

```
<serviceDiscovery>
  <announcementEndpoints>
    <endpoint kind="udpAnnouncementEndpoint" />
  </announcementEndpoints>
</serviceDiscovery>
```

Answer: D