Vendor: Microsoft

Exam Code: 70-487

Exam Name: Developing Windows Azure and Web Services

Version: Demo

Topic 1, Flight Information

Background
You are developing a flight information consolidation service. The service retrieves flight information from a number of sources and combines them into a single data set. The consolidated flight information is stored in a SQL Server database. Customers can query and retrieve the data by using a REST API provided by the service.

The service also offers access to historical flight information. The historical flight information can be filtered and queried in an ad hoc manner.

The service runs on a Windows Azure Web Role. SSL is not used.

Business Requirements
A new data source for historical flight information is being developed by a contractor located on another continent.
If a time zone is not specified, then it should be interpreted as Coordinated Universal Time (UTC).
When you upgrade a service from a staging deployment to a production deployment, the time that the service is unavailable must be minimized.
The default port must be used for HTTP.

Technical Requirements
The existing sources of flight information and the mechanism of exchange are listed below.

Blue Yonder Airlines provides flight information in an XML file.
Consolidated Messenger provides flight information in a Microsoft Access database that is uploaded every 12 hours to the service using SFTP. The company uses port 22 for SFTP.
Margie's Travel provides and consumes flight information using serialized ADO.NET DataSets. Data is periodically synced between the service and Margie's Travel.
Trey Research provides data from multiple sources serialized in proprietary binary formats. The data must be read by using .NET assemblies provided by Trey Research. The assemblies use a common set of dependencies. The current version of the Trey Research assemblies is 1.2.0.0. All assemblies provided by Trey Research are signed with a key pair contained in a file named Trey.snk, which Trey Research also supplies.
The application specification requires that any third-party assemblies must have strong names.

Application Structure

FlightInfo.cs

```
public class FlightInfo
{
  string DataSource { get; set; }
  public string Airline { get; set; }
  public string Flight { get; set; }
  public DateTimeOffset Arrival { get; set; }
  public int Seats { get; set; }
  public bool WasLate { get; set; }
}
```

BlueYonderLoader.cs

```
public class BlueYonderLoader
{
  public IEnumerable<RawFlightData> LoadFlights(XDocument feed)
  {
    ...
  }

  private RawFlightData Parse(XElement flightElement)
  {
    ...
  }
}
```

HistoricalDataLoader.cs

```
public class HistoricalDataLoader
{
  public static IEnumerable<HistoricalFlightInfo> LoadHistoricalFlights()
  {
    ...
  }

  public void StreamHistoricalFlights(XmlWriter responseWriter, string airline)
  {

    ...
  }

  private XElement ConvertToHistoricalFlight(XElement flight)
  {
    return new XElement("Flight", flight);
  }

  private string GetAirline(XElement flightName)
  {
    return flightName.Value.Substring(0, 2);
  }

  IEnumerable<XElement> RemoteDataStream()
  {
    return XDocument.Load("").Elements();
  }
}
```

MargiesTravelSync.cs

```
public class MargiesTravelSync
{
  public void Sync()
  {
    ...
  }

  private DataSet LoadLocal()
  {
    var dataSet = new DataSet();
    dataSet.ReadXml("local");
    return dataSet;
  }

  private StreamWriter SendStream()
  {
    return new StreamWriter("SendStream");
  }

  private StreamReader ReceiveStream()
  {
    return new StreamReader("ReceiveStream");
  }
}
```

FlightInfoContext.cs

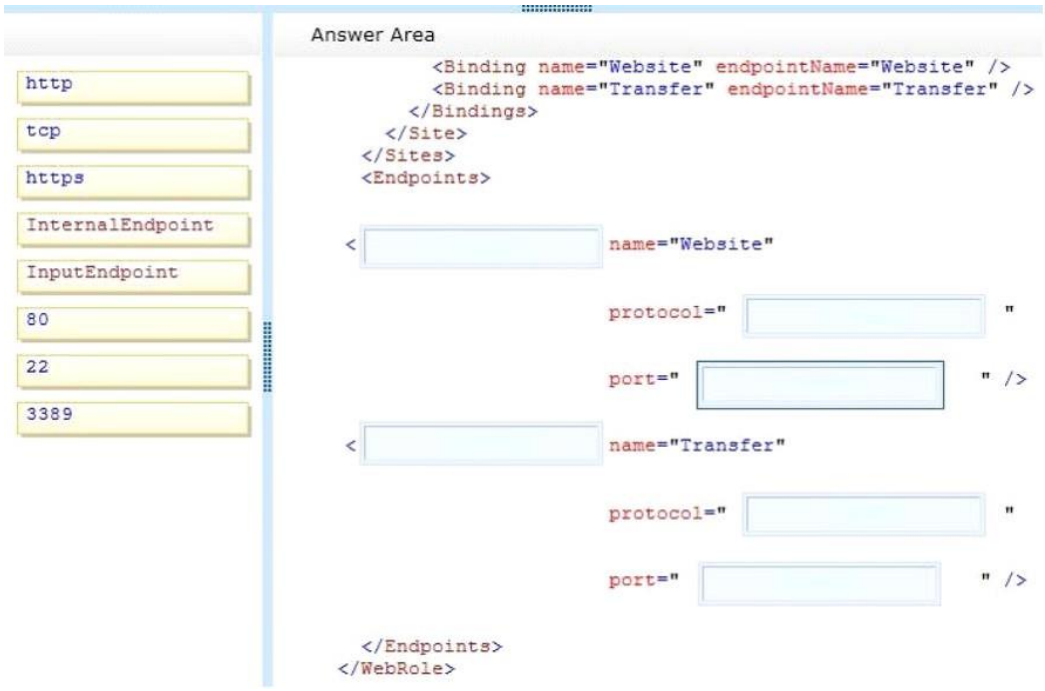
```
public class FlightInfoContext : DbContext
{
  public DbSet<FlightInfo> FlightInfo { get; set; }

  public override int SaveChanges()
  {
    return base.SaveChanges();
  }

  private bool IsTransient(int ex)
  {
    var errors = new[] { 10053, 10054, 64 };
    return errors.Contains(ex);
  }
}
```

FlightDataController.cs

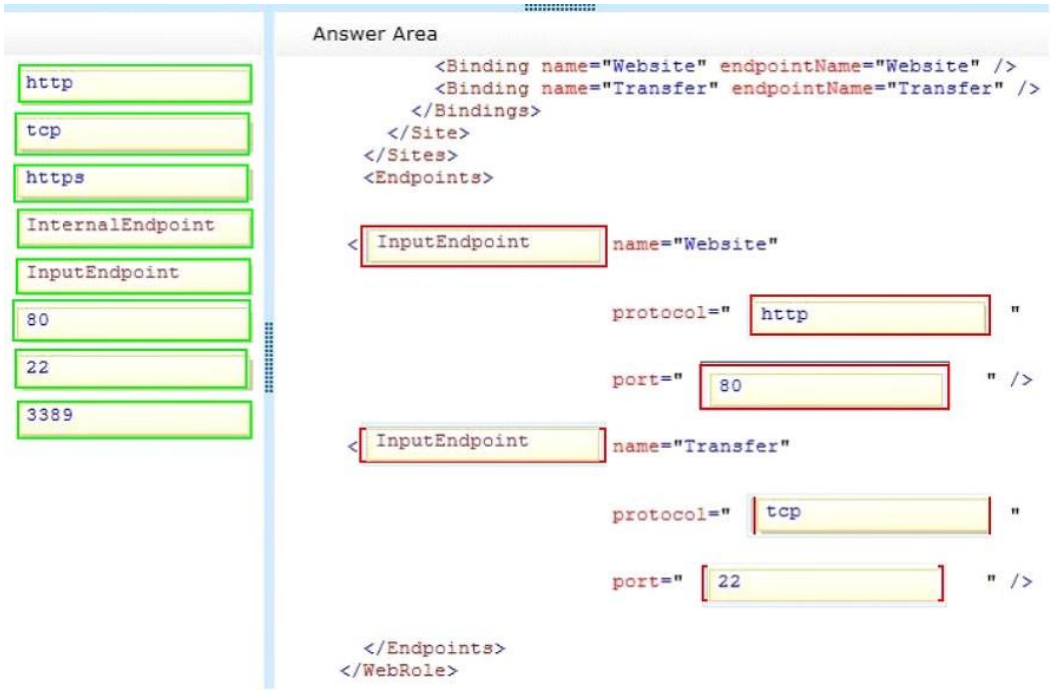
```
public class FlightDataController : ApiController
{
  FlightInfoContext _Context;

  public FlightDataController()
  {
    _Context = new FlightInfoContext();
  }

  [HttpGet]
  public IEnumerable<FlightInfo> GetFlightInfo()
  {
    return _Context.FlightInfo.Select(x => x).AsEnumerable();
  }

  private IEnumerable<HistoricalFlightInfo> LoadHistorical()
  {
    return HistoricalDataLoader.LoadHistoricalFlights();
  }
}
```

QUESTION 1
DRAG DROP
You need to configure the Windows Azure service definition to enable Consolidated Messenger to upload files. What should you do?

To answer, drag the appropriate configuration items to the correct location or locations. Each configuration item may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.



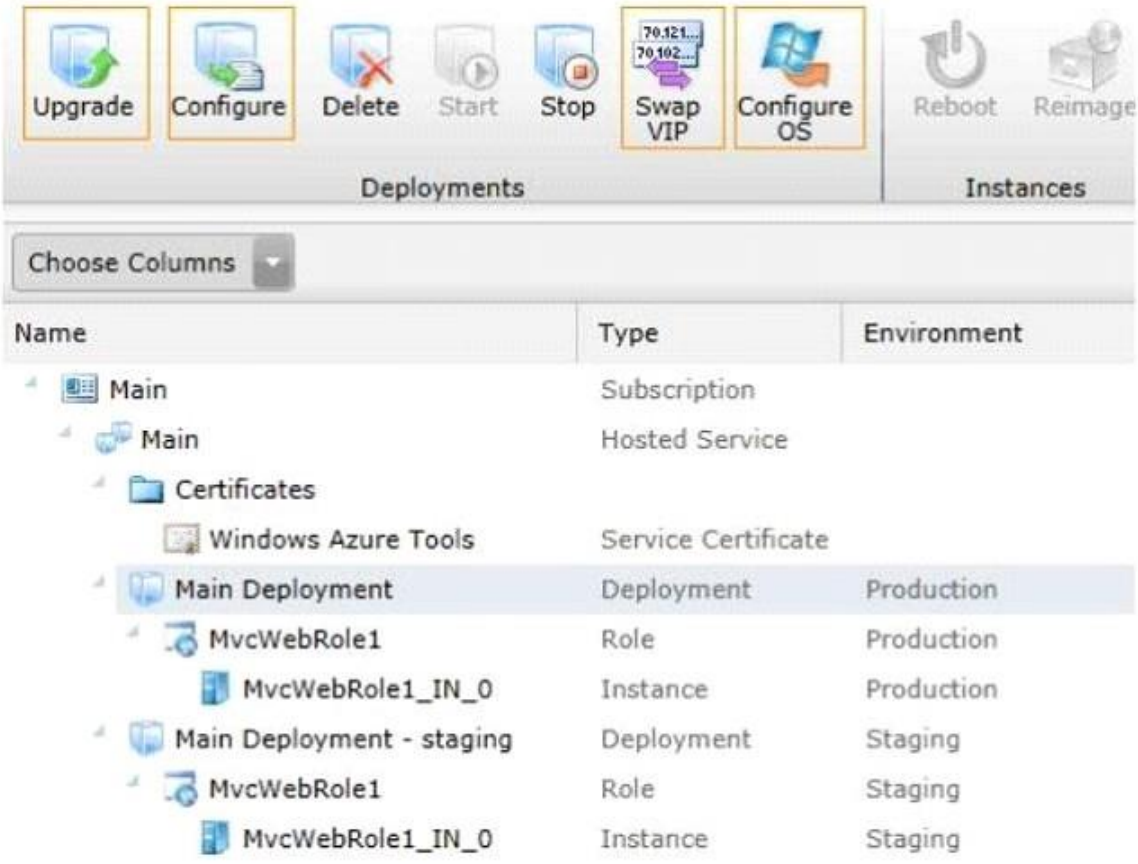
Correct Answer:



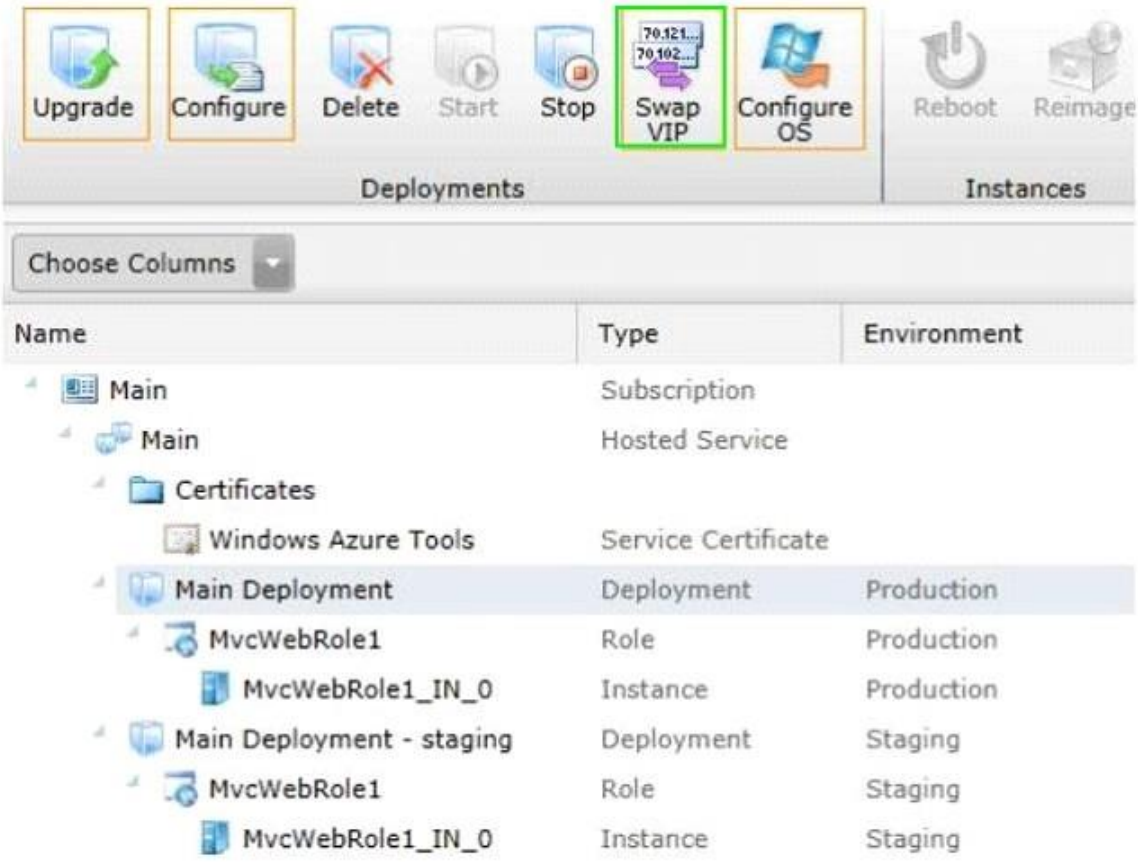
QUESTION 2
HOTSPOT
You need to deploy the application to the Windows Azure production environment to meet the business requirements.

What should you do?

To answer, select the appropriate button in the answer area.



Correct Answer:



QUESTION 3
You need to recommend a data access technology to the contractor to retrieve data from the new data source. Which data access technology should you recommend?

A. LINQ to XML
B. ADO.NET Entity Framework
C. ADO.NET DataSets
D. WCF Data Services

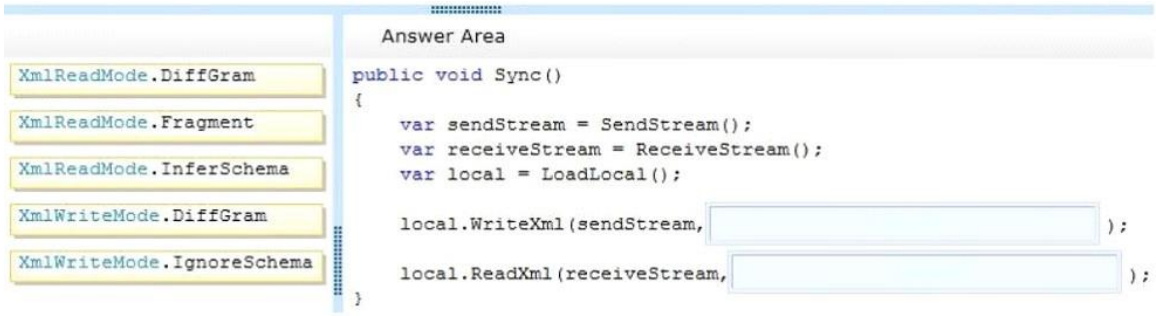
Correct Answer: D

QUESTION 4
DRAG DROP
Flight information data provided by Margie's Travel is updated both locally and remotely. When the data is synced, all changes need to be merged together without causing any data loss or corruption.

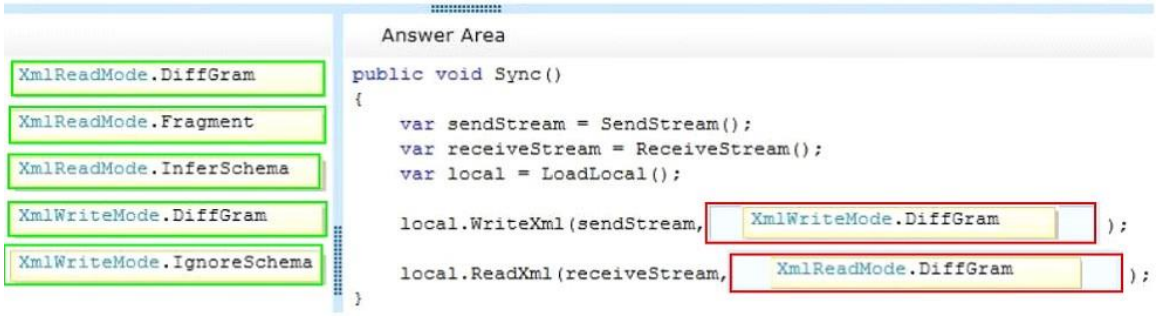
You need to implement the Sync() method in the MargiesTravelSync.es file.

What should you do?

To answer, drag the appropriate code segments to the correct location or locations in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.



Correct Answer:



QUESTION 5
DRAG DROP
Historical flight information data will be stored in Windows Azure Table Storage using the FlightInfo class as the table entity.

There are millions of entries in the table. Queries for historical flight information specify a set of airlines to search and whether the query should return only late flights. Results should be ordered by flight name.

You need to specify which properties of the FlightInfo class should be used at the partition and row keys to ensure that query results are returned as quickly as possible.

What should you do?

To answer, drag the appropriate properties to the correct location or locations in the answer area. Each property may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Correct Answer:

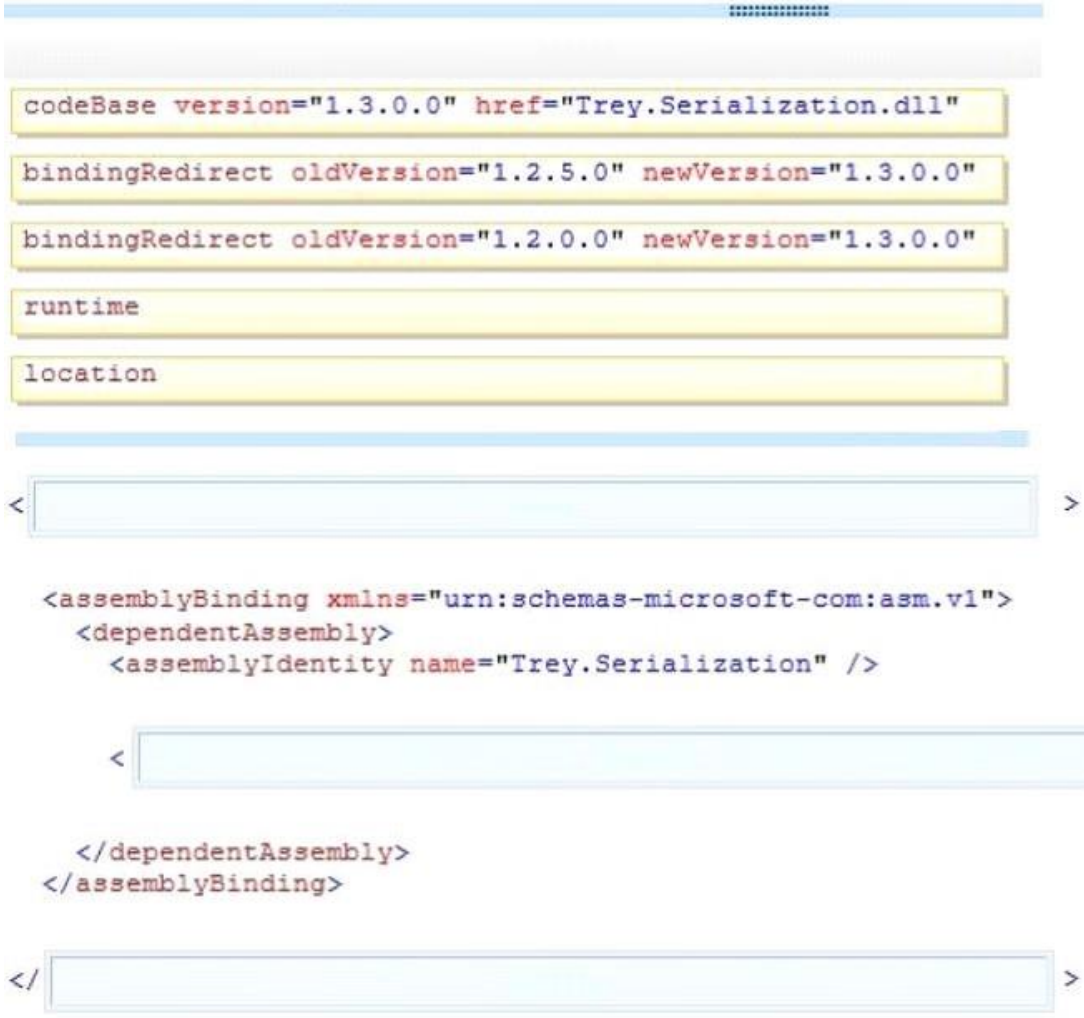
QUESTION 6
DRAG DROP
The service has been deployed to Windows Azure.

Trey Research has provided version 1.3.0.0 of the assembly to support a change in the serialization format. The service must remain available during the transition to the new serialization format.

You need to ensure that the service is using the new assembly.

Which configuration setting should you add to the web.config?

To answer, drag the appropriate configuration elements to the correct location or locations in the answer area. Each configuration element may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.



Correct Answer:


codeBase version="1.3.0.0" href="Trey.Serialization.dll"
bindingRedirect oldVersion="1.2.5.0" newVersion="1.3.0.0"
bindingRedirect oldVersion="1.2.0.0" newVersion="1.3.0.0"
runtime
location
< runtime >
<assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
<dependentAssembly>
<assemblyIdentity name="Trey.Serialization" />
< codeBase version="1.3.0.0" href="Trey.Serialization.dll"
</dependentAssembly>
</assemblyBinding>
</ runtime >

QUESTION 7
Errors occasionally occur when saving data using the FlightInfoContext ADO.NET Entity Framework context. Updates to the data are being lost when an error occurs. You need to ensure that data is still saved when an error occurs by retrying the operation. No more than five retries should be performed. Which code segment should you use as the body of the SaveChanges() method in the FlightInfoContext.es file?

A.
```
for (var i = 0; i < 5; i++)
{
  try
  {
    return base.SaveChanges();
  }
  catch (SqlException ex)
  {
    if (IsTransient(ex.Number))
    {
      continue;
    }
  }
}
return base.SaveChanges();
```

B.
```
var exception = new EntitySqlException();
while (exception.Data != 0 && exception.Data.Count < 5)
{
  try
  {
    return base.SaveChanges();
  }
  catch (EntitySqlException ex)
  {
    if (IsTransient(ex.HResult))
    {
      exception = ex;
    }
  }
}
return base.SaveChanges();
```

C.
```
for (var i = 0; i < 5; i++)
{
  try
  {
    return base.SaveChanges();
  }
  catch (SqlException ex)
  {
    if (IsTransient(ex.Number))
    {
      break;
    }
  }
}
return base.SaveChanges();
```

D.
```
for (var i = 0; i < 5; i++)
{
  try
  {
    return base.SaveChanges();
  }
  catch (SqlException ex)
  {
    if (!IsTransient(ex.Number))
    {
      continue;
    }
  }
}
return base.SaveChanges();
```

A. Option A
B. Option B
C. Option C
D. Option D

Correct Answer: A

QUESTION 8
You are adding a new REST service endpoint to the FlightDataController controller. It returns flights from the consolidated data sources only for flights that are late. You need to write a LINQ to Entities query to extract the required data. Which code segment should you use?

A.
```
var historical = LoadHistorical();
var query = _Context.FlightInfo.AsQueryable()
 .Join(historical, x => x.Flight, y => y.Flight, (x, y) => new { Current = x,
Historical = y })
 .Where(x => x.Historical.WasLate)
 .Select(x => x.Current);
```

B.
```
var historical = LoadHistorical();
var query = _Context.FlightInfo.AsEnumerable()
 .Where(x => historical.All(y => y.WasLate && x.Flight == y.Flight))
 .Select(x => x);
```

C.
```
var historical = LoadHistorical();
var query = _Context.FlightInfo.AsQueryable()
 .Where(x => historical.Select(y => y.Flight).Contains(x.Flight))
 .Where(x => historical.Any(y => y.WasLate))
 .Select(x => x);
```

D.
```
var historical = LoadHistorical();
var query = _Context.FlightInfo.AsEnumerable()
 .Join(historical, x => x.Flight, y => y.Flight, (x, y) => new { Current = x,
Historical = y })
 .Where(x => x.Historical.WasLate)
 .Select(x => x.Current);
```

A. Option A
B. Option B
C. Option C
D. Option D

Correct Answer: D
Explanation:
D is right because you send result as REST so if you use "AsQueryable" the result is deferred to the next enumeration of your result.
D is not optimized but will works.
A will break at runtime.
Credits to Rem

QUESTION 9
Data provided by Consolidated Messenger is cached in the HttpContext.Cache object. You need to ensure that the cache is correctly updated when new data arrives. What should you do?

A. Ensure that the EffectivePrivateBytesLimit value is greater than the size of the database file.
B. Change the sliding expiration of the cache item to 12 hours.
C. Use the SqlCacheDependency type configured with a connection string to the database file.
D. Use the CacheDependency type configured to monitor the SFTP target folder.

Correct Answer: D

QUESTION 10
You need to load flight information provided by Consolidated Messenger. Which should you use?

A. SQL Server Data Transformation Services (DTS)
B. EntityTransaction and EntityCommand
C. Office Open XML
D. OleDbConnection and OleDbDataReader

Correct Answer: D

QUESTION 11
DRAG DROP
You need to parse flight information from Blue Yonder Airlines. The content of the XML file is shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<AirlineFeed>
  <Flight xmlns="urn:CFI" name="AS515">
    <Seats>123</Seats>
    <Arrival>5/2/2011 12:01:13</Arrival>
  </Flight>
  <Flight name="UN24">
    <Seats>123</Seats>
    <Arrival>5/1/2012 10:17:57 PM +02:00</Arrival>
  </Flight>
  <FlightManifest>
    ...
  </FlightManifest>
</AirlineFeed>
```

Some airlines do not specify the timezone of the arrival time. If the timezone is not specified, then it should be interpreted per the business requirements.

You need to implement the LoadFlights() and Parse() methods of the BlueYonderLoader class.

What should you do?

To answer, drag the appropriate code segments to the correct location in the answer area. Each segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

```
var flights = feed.Elements(
 feed.Root.GetPrefixOfNamespace("{urn:CFI}") + "Flight");
```

```
var flights = feed.Descendants().Where(x =>
 x.NodeType != XmlNodeType.XmlDeclaration && (string)x ==
"Flight");
```

```
var flights = feed.Descendants("{urn:CFI}Flight")
 .Concat(feed.Descendants("Flight"));
```

```
fi.Arrival = DateTimeOffset.Parse(arrivalRaw,
 null, System.Globalization.DateTimeStyles.AssumeUniversal);
```

```
fi.Arrival = DateTimeOffset.Parse(arrivalRaw,
 null, System.Globalization.DateTimeStyles.AdjustToUniversal);
```

```
fi.Arrival = XmlConvert.ToDateTimeOffset(arrivalRaw,
 new[] { "Local", "Universal" });
```

```
public IEnumerable<FlightInfo> LoadFlights(XDocument feed)
{
  return flights.Select(x => Parse(x));
}

private FlightInfo Parse(XElement flightElement)
{
  var fi =  new FlightInfo();
  fi.Flight = flightElement.Attribute("name").Value;
  var arrivalRaw = flightElement.Element("Arrival").Value;

  fi.Seats = XmlConvert.ToInt32(flightElement.Element("Seats").Value);
  return fi;
}
```

Correct Answer:

```
var flights = feed.Elements(
 feed.Root.GetPrefixOfNamespace("{urn:CFI}") + "Flight");
```

```
var flights = feed.Descendants().Where(x =>
 x.NodeType != XmlNodeType.XmlDeclaration && (string)x ==
"Flight");
```

```
var flights = feed.Descendants("{urn:CFI}Flight")
 .Concat(feed.Descendants("Flight"));
```

```
fi.Arrival = DateTimeOffset.Parse(arrivalRaw,
 null, System.Globalization.DateTimeStyles.AssumeUniversal);
```

```
fi.Arrival = DateTimeOffset.Parse(arrivalRaw,
 null, System.Globalization.DateTimeStyles.AdjustToUniversal);
```

```
fi.Arrival = XmlConvert.ToDateTimeOffset(arrivalRaw,
 new[] { "Local", "Universal" });
```

```
public IEnumerable<FlightInfo> LoadFlights(XDocument feed)
{
  var flights = feed.Descendants("{urn:CFI}Flight")
   .Concat(feed.Descendants("Flight"));
  return flights.Select(x => Parse(x));
}

private FlightInfo Parse(XElement flightElement)
{
  var fi =  new FlightInfo();
  fi.Flight = flightElement.Attribute("name").Value;
  var arrivalRaw = flightElement.Element("Arrival").Value;
  fi.Arrival = DateTimeOffset.Parse(arrivalRaw,
   null, System.Globalization.DateTimeStyles.AssumeUniversal);
  fi.Seats = XmlConvert.ToInt32(flightElement.Element("Seats").Value);
  return fi;
}
```

QUESTION 12
You are adding a new REST service endpoint to the FlightDataController controller that returns the total number of seats for each airline. You need to write a LINQ to Entities query to extract the required data. Which code segment should you use?

A.
```
var query = from flight in _Context.FlightInfo
 group flight by flight.Seats into agg
 let airline = agg.First()
 select new
 {
   TotalSeats = agg.Key,
   Airline = airline,
 };
```

B.
```
var query = from flight1 in _Context.FlightInfo
 from flight2 in _Context.FlightInfo
 where flight1.Airline == flight2.Airline
 select new
 {
   Airline = flight1.Airline,
   TotalSeats = Math.BigMul(flight1.Seats, flight2.Seats),
 };
```

C.
```
var query = from flight in _Context.FlightInfo
 from airline in flight.Airline
 group airline by airline into agg
 select new
 {
   Airline = agg.Key,
   TotalSeats = agg.Sum(x => Convert.ToInt32(x)),
 };
```

D.
```
var query = from flight in _Context.FlightInfo
 group flight by flight.Airline into agg
 select new
 {
   Airline = agg.Key,
   TotalSeats = agg.Sum(x => x.Seats),
 };
```

A. Option A
B. Option B
C. Option C
D. Option D

Correct Answer: D

QUESTION 13
Transformed historical flight information provided by the RemoteDataStream() method must be written to the response stream as a series of XML elements named Flight within a root element named Flights. Each Flight element has a child element named FlightName that contains the flight name that starts with the two-letter airline prefix. You need to implement the StreamHistoricalFlights() method so that it minimizes the amount of memory allocated. Which code segment should you use as the body of the StreamHistoricalFlights() method in the HistoricalDataLoader.es file?

A.
```
responseWriter.WriteStartElement("Flights");
var flights = RemoteDataStream()
 .OrderBy(x => GetAirline(x.Element("FlightName")));
var filteredFlights = flights
 .SkipWhile(x => GetAirline(x.Element("FlightName")) != airline);
foreach (var f in filteredFlights)
{
  var flight = ConvertToHistoricalFlight(f);
  flight.WriteTo(responseWriter);
}
responseWriter.WriteEndElement();
```

B.
```
responseWriter.WriteStartElement("Flights");
var flights = RemoteDataStream().Select(x =>
 {
   if (GetAirline(x) == airline)
   {
     return ConvertToHistoricalFlight(x);
   }
   return null;
 });
flights.TakeWhile(x =>
 {
   x.WriteTo(responseWriter);
   return x != null;
 });
responseWriter.WriteEndElement();
```

C.
```
var data = RemoteDataStream().ToDictionary(x =>
 GetAirline(x.Element("FlightName")),
 x => new XStreamingElement("Flights", ConvertToHistoricalFlight(x).Descendants()));
data[airline].WriteTo(responseWriter);
```

D.
```
var flights = new XStreamingElement("Flights",
 from flight in RemoteDataStream()
 where GetAirline(flight.Element("FlightName")) == airline
 select ConvertToHistoricalFlight(flight));
flights.WriteTo(responseWriter);
```

A. Option A
B. Option B
C. Option C
D. Option D

Correct Answer: D
Explanation:
http://msdn.microsoft.com/en-us/library/system.xml.linq.xstreamingelement.aspx
http://msdn.microsoft.com/en-us/library/bb551307.aspx

QUESTION 14
Errors occasionally occur when saving data using the FlightInfoContext ADO.NET Entity Framework context. Updates to the data are being lost when an error occurs. You need to ensure that data is still saved when an error occurs by retrying the operation. No more than five retries should be performed. With which code segment should you replace the body of the SaveChanges() method in the FlightInfoContext.es file?

A.
```
var result = FlightInfo.SqlQuery("UPDATE WITH RETRY", FlightInfo, "IsTransient", 5);
if (result.Count() > 5)
{
  result.AsNoTracking();
  return -1;
}
return 0;
```

B.
```
try
{
  return base.SaveChanges();
}
catch (EntityCommandExecutionException ex)
{
  if (ex.Data.Keys.Cast<int>().Any(x => IsTransient(x)))
  {
    return 5 & SaveChanges();
  }
  return -1;
}
```

C.
```
for (var i = 0; i < 5; i++)
{
  try
  {
    return base.SaveChanges();
  }
  catch (SqlException ex)
  {
    if (IsTransient(ex.Number))
    {
      continue;
    }
  }
}
return base.SaveChanges();
```

D.
```
var exception = new EntitySqlException();
while (exception.HResult != 0 && exception.Data.Count < 5)
{
  try
  {
    return base.SaveChanges();
  }
  catch (EntitySqlException ex)
  {
    if (IsTransient(ex.HResult))
    {
      exception = ex;
    }
  }
}
return base.SaveChanges();
```

A. Option A
B. Option B
C. Option C
D. Option D

Correct Answer: C
Explanation:
EntitySqlException:
Represents errors that occur when parsing Entity SQL command text. This exception is thrown when syntactic or semantic rules are violated.
SqlException:
The exception that is thrown when SQL Server returns a warning or error. This class cannot be inherited.
EntityCommandExecutionException:
Represents errors that occur when the underlying storage provider could not execute the specified command. This exception usually wraps a provider-specific exception.

QUESTION 15
You need to load flight information provided by Consolidated Messenger. What should you use?

A. Office Open XML
B. COM interop
C. OleDbConnection and OleDbDataReader
D. EntityConnection and EntityDataReader

Correct Answer: C

EnsurePass.com Members Features:

1. Verified Answers researched by industry experts.
2. Q&As are downloadable in PDF and VCE format.
3. 98% success Guarantee and **Money Back** Guarantee.
4. Free updates for **180** Days.
5. **Instant Access to download the Items**

View list of All Exam provided:
http://www.ensurepass.com/certfications?index=A

To purchase Lifetime Full Access Membership click here:
http://www.ensurepass.com/user/register

Valid Discount Code for 2015: JREH-G1A8-XHC6

To purchase the HOT Microsoft Exams:

Microsoft			
70-243	70-347	70-466	70-515
70-246	70-410	70-467	70-516
70-247	70-411	70-480	70-519
70-321	70-412	70-483	70-583
70-331	70-413	70-484	70-640
70-332	70-414	70-485	70-649
70-336	70-417	70-486	70-668
70-337	70-461	70-487	70-680
70-341	70-462	70-488	70-687
70-342	70-463	70-489	70-688
70-346	70-464	70-513	70-689