Module 8

Accessing Remote Data

Module Overview

- Accessing Data Across the Web
- Accessing Data in the Cloud

Lesson 1: Accessing Data Across the Web

- Overview of Web Connectivity in the .NET Framework
- Defining a Data Contract
- Creating a Request and Processing a Response
- Authenticating a Web Request
- Sending and Receiving Data
- Demonstration: Consuming a Web Service

Overview of Web Connectivity in the .NET Framework

- Use the request and response pattern
- Use the classes in the **System.Net** namespace:
  - **WebRequest** (abstract base class)
  - **WebResponse** (abstract base class)
  - **HttpWebRequest**
  - **HttpWebResponse**
  - **FtpWebRequest**
  - **FtpWebResponse**
  - **FileWebRequest**
  - **FileWebResponse**

Defining a Data Contract

Use the **DataContract** and **DataMember** attributes to expose types from a web service

```
[DataContract()]
public class SalesPerson
{
   [DataMember()]
   public string FirstName { get; set; }

   [DataMember()]
   public string LastName { get; set; }

   [DataMember()]
   public string Area { get; set; }

   [DataMember()]
   public string EmailAddress { get; set; }
}
```

Creating a Request and Processing a Response

- Get a URI

```
var uri =
    "http://sales.fourthcoffee.com/SalesService.svc/GetSalesPerson";
```

- Create a request object

```
var request = WebRequest.Create(uri) as HttpWebRequest;
```

- Get a response object from the request object

```
var response = request.GetResponse() as HttpWebResponse;
```

- Read the properties in the response object

```
var status = response.StatusCode;
// Returns OK if a response is received.
```

Authenticating a Web Request

- Create the request object

```
var uri =
    "http://sales.fourthcoffee.com/SalesService.svc/GetSalesPerson";
var request = WebRequest.Create(uri) as HttpWebRequest;
```

- Use the **NetworkCredential** class

```
var username = "jespera";
var password = "Pa$$w0rd";
request.Credentials = new NetworkCredential(username, password);
```

- Use the **CredentialCache** class

```
request.Credentials = CredentialCache.DefaultCredentials;
```

- Use the **X509Certificate2** class

```
var certificate = FourthCoffeeCertificateServices.GetCertificate();
request.ClientCertificates.Add(certificate);
```

Sending and Receiving Data

- Send data

```
var uri =
  "http://sales.fourthcoffee.com/SalesService.svc/GetSalesPerson";
var rawData = Encoding.Default.GetBytes(
  "{\"emailAddress\":\"jespera@fourthcoffee.com\"}");
var request = WebRequest.Create(uri) as HttpWebRequest;
request.Method = "POST";
request.ContentType = "application/json";
request.ContentLength = rawData.Length;
var dataStream = request.GetRequestStream();
dataStream.Write(rawData, 0, rawData.Length);
dataStream.Close();
```

- Process the response

```
var response = request.GetResponse() as HttpWebResponse;
var stream = new StreamReader(response.GetResponseStream());
// Code to process the stream.
stream.Close();
```

Demonstration: Consuming a Web Service

In this demonstration, you will use the **HttpWebRequest** and **HttpWebResponse** classes to consume a web service over HTTP.

Lesson 2: Accessing Data in the Cloud

- What Is WCF Data Services?
- Defining a WCF Data Service
- Exposing a Data Model by Using WCF Data Services
- Exposing Web Methods by Using WCF Data Services
- Referencing a WCF Data Source
- Retrieving and Updating Data in a WCF Data Service
- Demonstration: Retrieving and Modifying Grade Data in the Cloud Lab

What Is WCF Data Services?

- WCF Data Services:
  - Enables you to create highly flexible data services
  - Uses the REST model for data access

```
http://FourthCoffee.com/SalesService.svc/SalesPersons
http://FourthCoffee.com/SalesService.svc/SalesPersons/99
http://FourthCoffee.com/SalesService.svc/SalesPersons?top=10
```

  - Enables you to query and modify data by using URIs with standard HTTP verbs: GET, PUT, POST, and DELETE

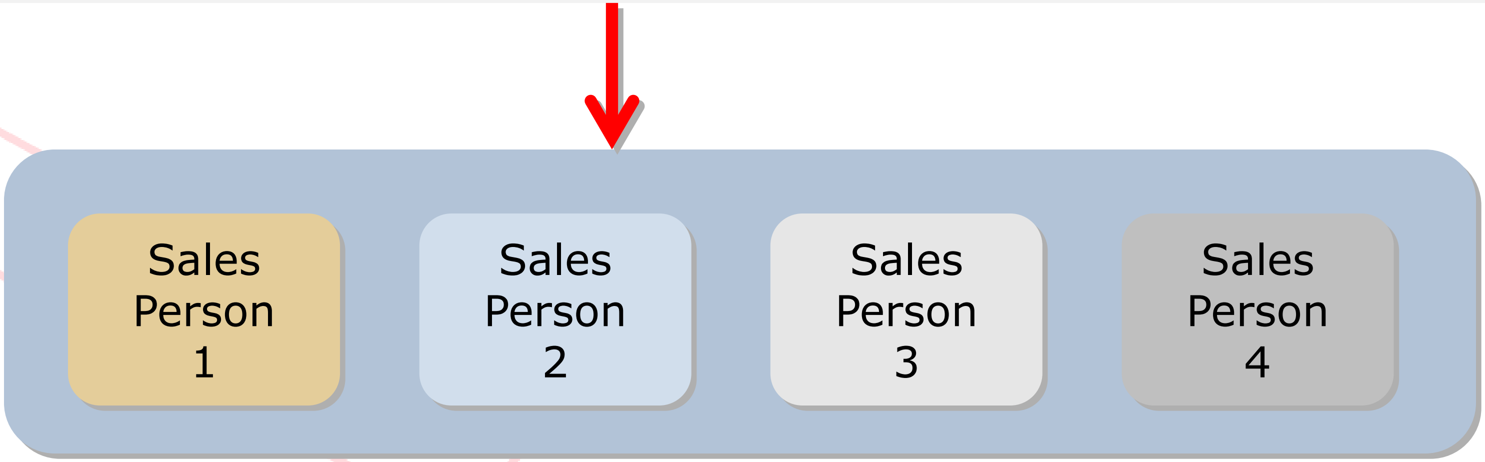
Defining a WCF Data Service

- WCF Data Services is based on the **System.Data.Services.DataService** generic class

```
public class FourthCoffeeDataService : DataService<FourthCoffee>
{
   ...
}
```

- URIs are mapped to entity sets by a data service:

```
http://FourthCoffee.com/SalesService.svc/SalesPersons
```



Sales Person 1

Sales Person 2

Sales Person 3

Sales Person 4

Exposing a Data Model by Using WCF Data Services

Configure the access rules on the WCF Data Service by using the **SetEntitySetAccessRule** method

```
public class FourthCoffeeDataService : DataService<FourthCoffee>
{
   public static void InitializeService(
      DataServiceConfiguration config)
   {
      config.SetEntitySetAccessRule("*", EntitySetRights.All);
   }
}
```

Exposing Web Methods by Using WCF Data Services

Expose operations by using the **WebGet** and **WebInvoke** attributes

```
public class FourthCoffeeDataService : DataService<FourthCoffee>
{
    public static void InitializeService(DataServiceConfiguration config)
    {
       config.SetServiceOperationAccessRule("SalesPersonByEmail",
            ServiceOperationRights.ReadMultiple);
    }

   [WebGet]
   [SingleResult]
   public SalesPerson SalesPersonByEmail(string emailAddress)
   {
       ...
    }
}
```

Referencing a WCF Data Source

Client libraries:

- Are derived from the **DataServiceContext** class
- Expose entities that the **DataServiceQuery** collection contains

Create a client library by using:

- The **Add Service Reference** function in Visual Studio
- The **DataSvcUtil** command line utility

Retrieving and Updating Data in a WCF Data Service

- Retrieve entities:
  - Use the properties that are exposed by the context
  - Invoke custom service operations
  - Use eager or explicit loading to get related entities

- Modify entities:
  - Use the **AddTo***XXXX* method to add a new entity
  - Use the **DeleteObject** method to remove an existing entity
  - Use the **UpdateObject** method to modify an existing entity

Demonstration: Retrieving and Modifying Grade Data in the Cloud Lab

In this demonstration, you will learn about the tasks that you will perform in the lab for this module.

Lab: Retrieving and Modifying Grade Data in the Cloud

- Exercise 1: Creating a WCF Data Service for the SchoolGrades Database
- Exercise 2: Integrating the Data Service into the Application
- Exercise 3: Retrieving Student Photographs Over the Web (If Time Permits)

Logon Information

- Virtual Machine: 20483B-SEA-DEV11, MSL-TMG1
- User Name: Student
- Password: Pa$$w0rd

Estimated Time: 60 minutes

Lab Scenario

- Currently, the application retrieves data from a local database. However, you have decided to store the data in the cloud and must configure the application so that it can retrieve data across the web.
- You must create a WCF Data Service for the **SchoolGrades** database that will be integrated into the application to enable access to the data.
- Finally, you have been asked to write code that displays student images by retrieving them from across the web.

Module Review and Takeaways

- Review Question(s)