



TI-Capital Humano

Desarrollador .Net

ADO



Contenido

1	ADO.NET	3
1.1	CARACTERISTICAS.....	¡Error! Marcador no definido.
1.2	ARQUITECTURAS DE CONEXIÓN.....	4
1.3	Comprender ADO.NET y su biblioteca de clases.....	5
1.4	Clases importantes en ADO.NET	6
1.5	Cómo conectarse a una base de datos usando ADO.NET	7
1.6	Código para conectarse a la base de datos Oracle	9

1 ADO.NET

ADO es el acrónimo de **ActiveX Data Objects**. Nos permite conectarnos a datos o bases de datos subyacentes. Tiene clases y métodos para recuperar y manipular datos.

1.1 Características

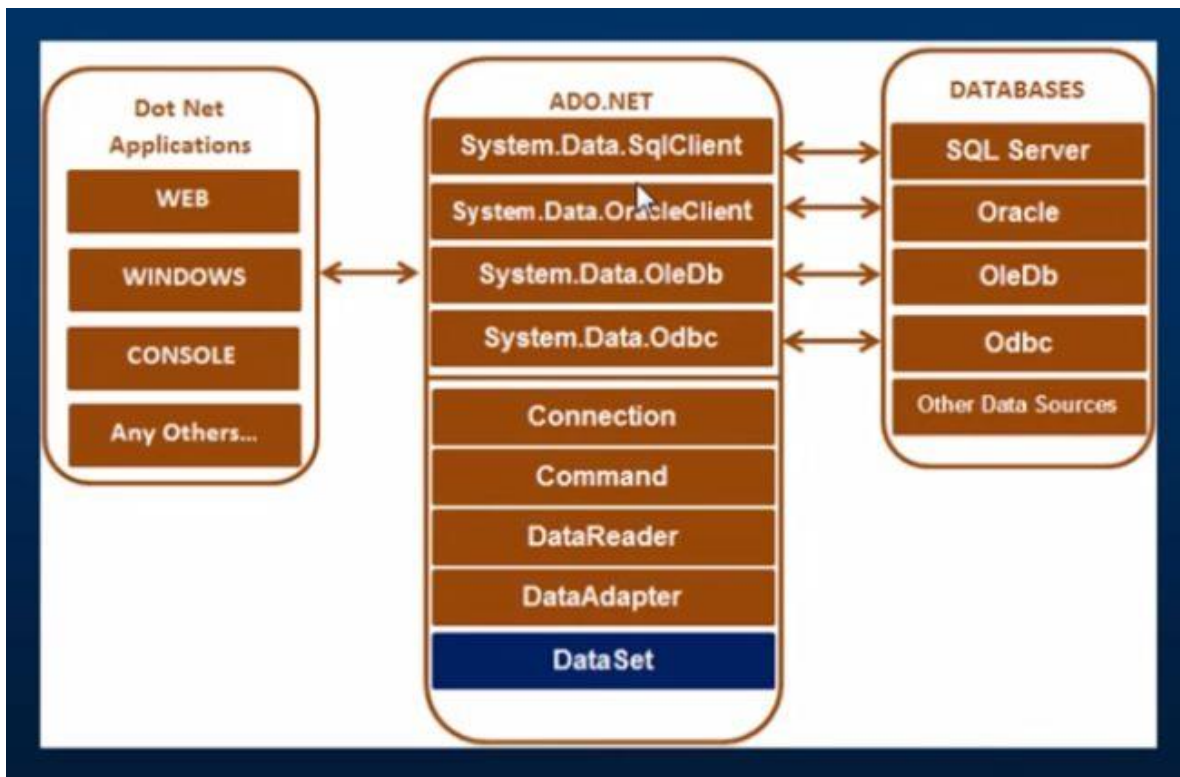
- Es un conjunto de clases que exponen servicios de acceso a datos para programadores de .NET Framework.
- Separa el acceso a datos de la manipulación de datos en componentes discretos que se pueden usar por separado o en conjunto
- Proporciona un amplio conjunto de componentes para crear aplicaciones distribuidas para compartir datos.
- Es una parte integral de .NET Framework, que proporciona acceso a datos relacionales, XML y de aplicaciones. ADO.NET
- Admite una variedad de necesidades de desarrollo, incluida la creación de clientes de base de datos de front-end y objetos comerciales de nivel medio utilizados por aplicaciones, herramientas, lenguajes o navegadores de Internet.
- ADO.NET proporciona acceso consistente a fuentes de datos como SQL Server y XML, y a fuentes de datos expuestas a través de OLE DB y ODBC.
- Aplicaciones .NET que usan ADO.NET para conectarse a una base de datos, ejecutar comandos y recuperar datos de la base de datos.
 - Aplicaciones web ASP.NET
 - Aplicaciones de consola
 - Aplicaciones de Windows.
- Las clases ADO.NET se encuentran en System.Data.dll y están integradas con las clases XML que se encuentran en System.Xml.dll.

1.2 Arquitecturas de conexión

Existen los dos tipos siguientes de arquitecturas de conexión:

- Arquitectura conectada: la aplicación permanece conectada con la base de datos durante todo el procesamiento.
- Arquitectura desconectada: la aplicación se conecta / desconecta automáticamente durante el procesamiento. La aplicación utiliza datos temporales en el lado de la aplicación llamados DataSet.

1.3 Comprender ADO.NET y su biblioteca de clases



En este diagrama, podemos ver que hay varios tipos de aplicaciones (aplicación web, aplicación de consola, aplicación de Windows, etc.) que usan ADO.NET para conectarse a bases de datos (SQL Server, Oracle, OleDb, ODBC, archivos XML, etc. en).

1.4 Clases importantes en ADO.NET

También podemos observar varias clases en el diagrama anterior. Son:

1. Clase connection
2. Clase command
3. Clase DataReader
4. Clase DataAdaptor
5. Clase DataSet

1. Clase connection

En ADO.NET, usamos estas clases de conexión para conectarnos a la base de datos. Estas clases de conexión también administran transacciones y agrupaciones de conexiones.

2. Clase command

La clase Command proporciona métodos para almacenar y ejecutar sentencias SQL y procedimientos almacenados. Los siguientes son los diversos comandos que ejecuta la clase Command.

ExecuteReader: devuelve datos al cliente como filas. Por lo general, sería una instrucción select de SQL o un procedimiento almacenado que contiene una o más instrucciones de selección. Este método devuelve un objeto DataReader que puede usarse para llenar un objeto DataTable o usarse directamente para imprimir informes y así sucesivamente.

ExecuteNonQuery: ejecuta un comando que cambia los datos en la base de datos, como una instrucción de actualización, eliminación o inserción, o un procedimiento almacenado que contiene una o más de estas declaraciones. Este método devuelve un número entero que es el número de filas afectadas por la consulta.

ExecuteScalar: este método solo devuelve un valor único. Este tipo de consulta devuelve un recuento de filas o un valor calculado.

3. Clase DataReader

El DataReader se utiliza para recuperar datos. Se usa junto con la clase Command para ejecutar una instrucción SQL Select y luego acceder a las filas devueltas.

4. Clase DataAdapter

El DataAdapter se utiliza para conectar DataSets a bases de datos. El DataAdapter es más útil cuando se usan controles enlazados a datos en Windows Forms, pero también se puede usar para proporcionar una manera fácil de administrar la conexión entre su aplicación y las tablas de base de datos subyacentes, vistas y procedimientos almacenados.

5. Clase DataSet

El DataSet es el corazón de ADO.NET. El DataSet es esencialmente una colección de objetos DataTable. A su vez, cada objeto contiene una colección de objetos DataColumn y DataRow. El DataSet también contiene una colección de Relaciones que se puede usar para definir relaciones entre Objetos de Tabla de Datos.

1.5 Cómo conectarse a una base de datos usando ADO.NET

Ahora aprendamos cómo conectarnos a una base de datos usando ADO.NET. Para crear una conexión, debe estar familiarizado con las cadenas de conexión. Se requiere una cadena de conexión como parámetro para SqlConnection. Un ConnectionString es una variable de cadena (no distingue entre mayúsculas y minúsculas).

Este contiene pares de clave y valor, como proveedor, servidor, base de datos, ID de usuario y palabra, como se muestra a continuación:

Server = "nombre del servidor o dirección IP del servidor"

Database = "nombre de la base de datos"

userid = "nombre de usuario que tiene permiso para trabajar con la base de datos"

password = "la contraseña de userid"

Ejemplo

Autenticación SQL

String constr = "servidor = .; base de datos = instituto; id de usuario = rakesh;
palabra = abc @ 123";

O:

```
String constr = "fuente de datos = .; catálogo inicial = instituto; uid = rakesh; pwd = abc @ 213";
```

Autenticación de Windows

```
String constr = "servidor = .; base de datos = instituto; conexión_de_confianza = verdadero"
```

O:

```
String constr = "servidor = .; catálogo inicial = instituto; seguridad integrada = verdadero"
```

Cómo recuperar y mostrar datos de una base de datos

1. Cree un objeto SqlConnection utilizando una cadena de conexión.
2. Manejar excepciones.
3. Abra la conexión.
4. Cree un SqlCommand. Para representar un SqlCommand como (seleccione * de los detalles del alumno) y adjunte la conexión existente a él. Especifique el tipo de SqlCommand (texto / procedimiento almacenado).
5. Ejecute el comando (use executereader).
6. Obtenga el resultado (use SqlDataReader). Este es un objeto de datos de solo lectura / avance.
7. Cerrar la conexión
8. Procesar el resultado
9. Mostrar el resultado

Debe utilizar el espacio de nombres System.Data.SqlClient para conectarse a una base de datos SQL. En el código anterior usamos la clase SqlConnection, la clase SqlCommand y la clase SqlDataReader porque nuestra aplicación está hablando con SQL Server. SQL Server solo comprende SQL.

1.6 Código para conectarse a la base de datos Oracle

Si desea conectarse a una base de datos de Oracle, todo lo que necesita hacer es cambiar el nombre de la clase de conexión de SqlConnection a OracleConnection, el nombre de la clase de SqlCommand a OracleCommand y SqlDataReader a OracleDataReader y también al principio usar el espacio de nombres System.Data.OracleClient.

Esto también funciona para OleDb y ODBC.