ADO.NET

Documentation

Abstract

In this module I learn how I use database with C# using ADO.NET.

Dhruvil Dobariya
dhruvildobariya21@gmail.com

INDEX

DataReader

1.1 Introduction

- This class is used to read data from database.
- It read data in only forward direction of rows.
- it is sealed class so that cannot be inherited.
- It inherits DbDataReader class and implements IDisposable interface.

1.2 Properties

Property	Description
Connection	It is used to get the SqlConnection associated with the SqlDataReader.
Depth	It is used to get a value that indicates the depth of nesting for the current row.
FieldCount	It is used to get the number of columns in the current row.
HasRows	It is used to get a value that indicates whether the SqlDataReader contains one or more rows.
IsClosed	It is used to retrieve a boolean value that indicates whether the specified SqlDataReader instance has been closed.
Item[String]	It is used to get the value of the specified column in its native format given the column name.
Item[Int32]	It is used to get the value of the specified column in its native format given the column ordinal.
RecordsAffected	It is used to get the number of rows changed, inserted or deleted by execution of the Transact-SQL statement.
VisibleFieldCount	It is used to get the number of fields in the SqlDataReader that are not hidden.

1.3 Methods

Method	Description
Close()	It is used to closes the SqlDataReader object.
GetBoolean(Int32)	It is used to get the value of the specified column as a Boolean.
GetByte(Int32)	It is used to get the value of the specified column as a byte.
GetChar(Int32)	It is used to get the value of the specified column as a single character.
GetDateTime(Int32)	It is used to get the value of the specified column as a DateTime object.
GetDecimal(Int32)	It is used to get the value of the specified column as a Decimal object.
GetDouble(Int32)	It is used to get the value of the specified column as a double-precision floating point number.

GetFloat(Int32)	It is used to get the value of the specified column as a single-precision floating point number.
GetName(Int32)	It is used to get the name of the specified column.
GetSchemaTable()	It is used to get a DataTable that describes the column metadata of the SqlDataReader.
GetValue(Int32)	It is used to get the value of the specified column in its native format.
GetValues(Object[])	It is used to populate an array of objects with the column values of the current row.
NextResult()	It is used to get the next result, when reading the results of SQL statements.
Read()	It is used to read record from the SQL Server database.

DATAADAPTER

2.1 INTRODUCTION

- The DataAdapter works as a bridge between a DataSet and a database.
- It is used to transfer data between DataSet and database.
- DataAdapter is a class that represents a set of SQL commands and a database connection.
- It can be used to fill the DataSet from data source.
- It is also used to update the data source.

2.2 CONSTRUCTORS

Constructors	Description
DataAdapter()	It is used to initialize a new instance of a DataAdapter class.
DataAdapter(DataAdapter)	It is used to initializes a new instance of a DataAdapter class from an existing object of the same type.

2.3 METHODS

Method	Description
CloneInternals()	It is used to create a copy of this instance of DataAdapter.
Dispose(Boolean)	It is used to release the unmanaged resources used by the DataAdapter.
Fill(DataSet)	It is used to add rows in the DataSet to match those in the data source.
FillSchema(DataSet, SchemaType, String, IDataReader)	It is used to add a DataTable to the specified DataSet.
GetFillParameters()	It is used to get the parameters set by the user when executing an SQL SELECT statement.
ResetFillLoadOption()	It is used to reset FillLoadOption to its default state.
ShouldSerializeAcceptChangesDuringFill()	It determines whether the AcceptChangesDuringFill property should be persisted or not.
ShouldSerializeFillLoadOption()	It determines whether the FillLoadOption property should be persisted or not.
ShouldSerializeTableMappings()	It determines whether one or more DataTableMapping objects exist or not.

Update(DataSet)	It is used to call the respective INSERT, UPDATE, or DELETE statements.

DataSet

3.1 Introduction

- It is a collection of data tables that contain the data.
- It is used to fetch data without interacting with a Data Source that's why, it also known as disconnected data access method.
- It is an in-memory data store that can hold more than one table at the same time.
- We can use DataRelation object to relate these tables.
- The DataSet can also be used to read and write data as XML document.

3.2 Constructors:

Constructor	Description
DataSet()	It is used to initialize a new instance of the DataSet class.
DataSet(String)	It is used to initialize a new instance of a DataSet class with the given name.
DataSet(SerializationInfo, StreamingContext)	It is used to initialize a new instance of a DataSet class that has the given serialization information and context.
DataSet(SerializationInfo, StreamingContext, Boolean)	It is used to initialize a new instance of the DataSet class.

3.3 Properties

Properties	Description
CaseSensitive	It is used to check whether DataTable objects are case-sensitive or not.
DataSetName	It is used to get or set name of the current DataSet.
DefaultViewManager	It is used to get a custom view of the data contained in the DataSet to allow filtering and searching.
HasErrors	It is used to check whether there are errors in any of the DataTable objects within this DataSet.
IsInitialized	It is used to check whether the DataSet is initialized or not.
Locale	It is used to get or set the locale information used to compare strings within the table.
Namespace	It is used to get or set the namespace of the DataSet.
Site	It is used to get or set an ISite for the DataSet.
Tables	It is used to get the collection of tables contained in the DataSet.

3.4 METHODS

Method	Description
BeginInit()	It is used to begin the initialization of a DataSet that is used on a form.
Clear()	It is used to clear the DataSet of any data by removing all rows in all tables.
Clone()	It is used to copy the structure of the DataSet.
Copy()	It is used to copy both the structure and data for this DataSet.
CreateDataReader(DataTable[])	It returns a DataTableReader with one result set per DataTable.
CreateDataReader()	It returns a DataTableReader with one result set per DataTable.
EndInit()	It ends the initialization of a DataSet that is used on a form.
GetXml()	It returns the XML representation of the data stored in the DataSet.
GetXmlSchema()	It returns the XML Schema for the XML representation of the data stored in the DataSet.
Load(IDataReader, LoadOption, DataTable[])	It is used to fill a DataSet with values from a data source using the supplied IDataReader.
Merge(DataSet)	It is used to merge a specified DataSet and its schema into the current DataSet.
Merge(DataTable)	It is used to merge a specified DataTable and its schema into the current DataSet.
ReadXml(XmlReader, XmlReadMode)	It is used to read XML schema and data into the DataSet using the specified XmlReader and XmlReadMode.
Reset()	It is used to clear all tables and removes all relations, foreign constraints, and tables from the DataSet.
WriteXml(XmlWriter, XmlWriteMode)	It is used to write the current data and optionally the schema for the DataSet using the specified XmlWriter and XmlWriteMode.

DATATABLE

4.1 INTRODUCTION

- DataTable represent relational data in tabular form.
- ADO.NET provides a DataTable class to create and use data table independently.
- Initially, when we create DataTable, it does not have table schema.
- We can create table schema by adding columns and constraints to the table.
- After defining table schema, we can add rows to the table.

Namespace: "System.Data"

4.2 CONSTRUCTORS

Constructors	Description
DataTable()	It is used to initialize a new instance of the DataTable class with no arguments.
DataTable(String)	It is used to initialize a new instance of the DataTable class with the specified table name.
DataTable(SerializationInfo, StreamingContext)	It is used to initialize a new instance of the DataTable class with the SerializationInfo and the StreamingContext.
DataTable(String, String)	It is used to initialize a new instance of the DataTable class using the specified table name and namespace.

4.3 PROPERTIES

Property	Description
Columns	It is used to get the collection of columns that belong to this table.
Constraints	It is used to get the collection of constraints maintained by this table.
DataSet	It is used to get the DataSet to which this table belongs.
DefaultView	It is used to get a customized view of the table that may include a filtered view.
HasErrors	It is used to get a value indicating whether there are errors in any of the rows in the table of the DataSet.
MinimumCapacity	It is used to get or set the initial starting size for this table.
PrimaryKey	It is used to get or set an array of columns that function as primary keys for the data table.
Rows	It is used to get the collection of rows that belong to this table.
TableName	It is used to get or set the name of the DataTable.

4.4 METHODS

Method	Description
AcceptChanges()	It is used to commit all the changes made to this table.
Clear()	It is used to clear the DataTable of all data.
Clone()	It is used to clone the structure of the DataTable.
Copy()	It is used to copy both the structure and data of the DataTable.
CreateDataReader()	It is used to returns a DataTableReader corresponding to the data within this DataTable.
CreateInstance()	It is used to create a new instance of DataTable.
GetRowType()	It is used to get the row type.
GetSchema()	It is used to get schema of the table.
ImportRow(DataRow)	It is used to copy a DataRow into a DataTable.
Load(IDataReader)	It is used to fill a DataTable with values from a data source using the supplied IDataReader.
Merge(DataTable, Boolean)	It is used to merge the specified DataTable with the current DataTable.
NewRow()	It is used to create a new DataRow with the same schema as the table.
Select()	It is used to get an array of all DataRow objects.
WriteXml(String)	It is used to write the current contents of the DataTable as XML using the specified file.

CONNECTION

5.1 INTRODUCTION

- It is used to establish an open connection to the SQL Server database.
- It is a sealed class so that cannot be inherited.
- SqlConnection class uses SqlDataAdapter and SqlCommand classes together to increase performance when connecting to a Microsoft SQL Server database.
- Connection does not close explicitly even it goes out of scope.
- Therefore, you must explicitly close the connection by calling Close() method.

5.2 CONSTRUCTOR

Constructors	Description
SqlConnection()	It is used to initializes a new instance of the SqlConnection class.
SqlConnection(String)	It is used to initialize a new instance of the SqlConnection class and takes connection string as an argument.
SqlConnection(String, SqlCredential)	It is used to initialize a new instance of the SqlConnection class that takes two parameters. First is connection string and second is sql credentials.

5.3 METHODS

Method	Description
BeginTransaction()	It is used to start a database transaction.
ChangeDatabase(String)	It is used to change the current database for an open SqlConnection.
ChangePassword(String, String)	It changes the SQL Server password for the user indicated in the connection string.
Close()	It is used to close the connection to the database.
CreateCommand()	It enlists in the specified transaction as a distributed transaction.
GetSchema()	It returns schema information for the data source of this SqlConnection.
Open()	It is used to open a database connection.
ResetStatistics()	It resets all values if statistics gathering is enabled.

COMMAND

6.1 INTRODUCTION

- This class is used to store and execute SQL statement for database.
- It is a sealed class so that cannot be inherited.

6.2 CONSTRUCTORS

Constructor	Description
SqlCommand()	It is used to initialize a new instance of the SqlCommand class.
SqlCommand(String)	It is used to initialize a new instance of the SqlCommand class with a string parameter.
SqlCommand(String, SqlConnection)	It is used to initialize a new instance of the SqlCommand class. It takes two parameters, first is query string and second is connection string.
SqlCommand(String, SqlConnection, SqlTransaction)	It is used to initialize a new instance of the SqlCommand class. It takes three parameters query, connection and transaction string respectively.
SqlCommand(String, SqlConnection, SqlTransaction, SqlCommandColumnEncryptionSetting)	It Initializes a new instance of the SqlCommand class with specified command text, connection, transaction, and encryption setting.

6.3 METHODS

Method	Description
BeginExecuteNonQuery()	It is used to Initiate the asynchronous execution of the SQL statement described by this SqlCommand.
Cancel()	It tries to cancel the execution of a SqlCommand.
Clone()	It creates a new SqlCommand object that is a copy of the current instance.
CreateParameter()	It creates a new instance of a SqlParameter object.
ExecuteReader()	It is used to send the CommandText to the Connection and builds a SqlDataReader.
ExecuteXmlReader()	It is used to send the CommandText to the Connection and builds an XmlReader object.

ExecuteScalar()	It executes the query and returns the first column of the first row in the result set. Additional columns or rows are ignored.
Prepare()	It is used to create a prepared version of the command by using the instance of SQL Server.
ResetCommandTimeout()	It is used to reset the CommandTimeout property to its default value.
ExecuteNonQuery()	It is used to execute update, insert and delete.