

# Lesson: ADO.NET

September 3, 2015  
Andrew & Patty

# Objective

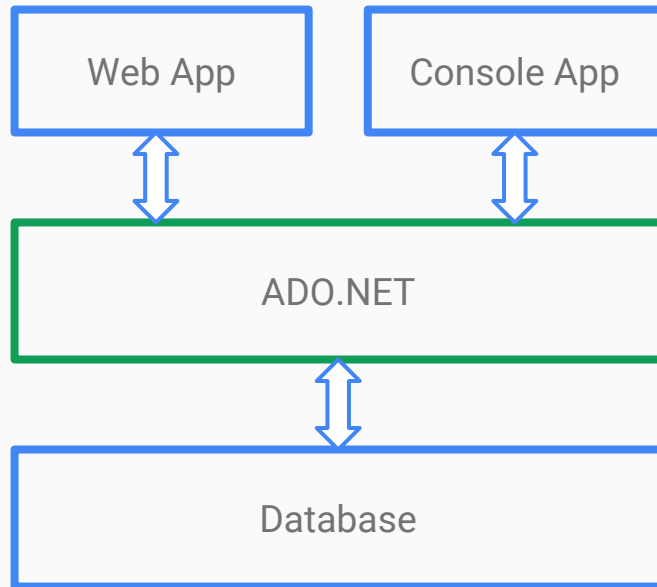
To learn the concept and tools for data access in ADO.NET

What is ADO.NET ?

# Background

ADO.NET is a set of classes that provides data access functionalities. It is a part of the .NET framework.

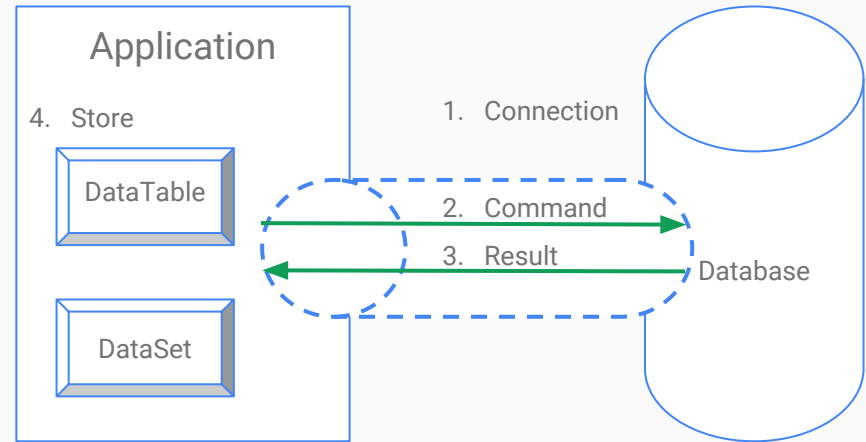
1. Works with relational databases (SQL Server, MySQL, Oracle DB, etc.) and XML databases
2. Serves as “middleware” between application and database.



# Methodology

Common data access methodology:

1. Connect to database
2. Pass request to database (select, insert, update, etc.)
3. Getting back results (queried rows or rows affected)
4. Storing results and displaying to user

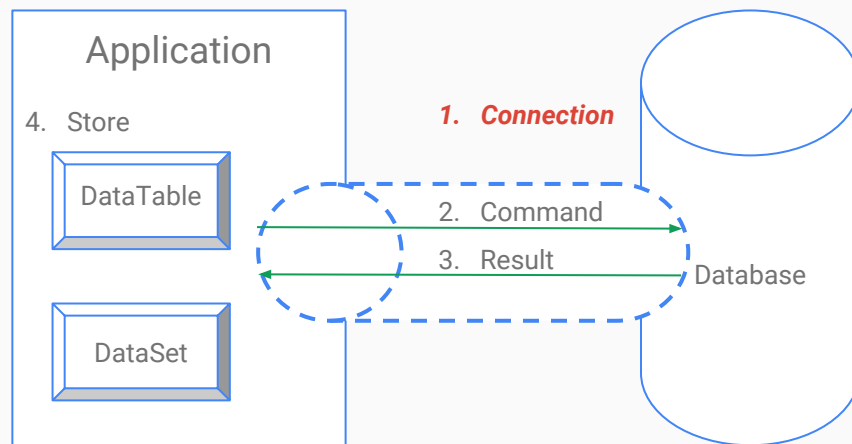


# How to use ADO.NET

# Connection

A Connection object (`SqlConnection`) is used to establish connection to database.

1. `ConnectionString` contains information about data source, server location, and authentication parameters (protocol and user/pwd)
2. `ConnectionString` value set in config file.



# How to create a Connection object

## ConnectionString declaration in app.config

```
<connectionStrings>
  <add name="Northwind"
        connectionString="Server=MS-STDN-001\SQL2014;User Id=sa;Password=sqlserver;
                          Database=Northwind;" />
</connectionStrings>
```

## Creation of SqlConnection object

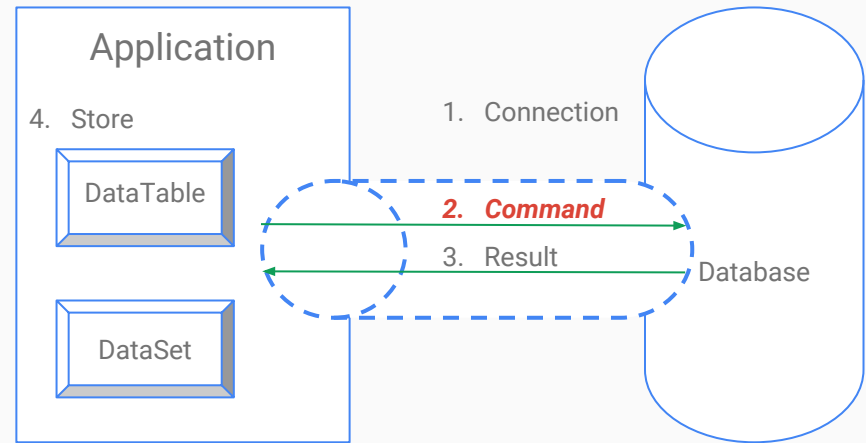
```
SqlConnection cn = null;
cn = new SqlConnection(ConfigurationManager.ConnectionStrings["Northwind"].ConnectionString);
```



# Command

A Command object (`SqlCommand`) is used to tell the database what to do. Common data manipulation commands:

1. Select Command - Returns a set of rows to the application
2. Insert Command - Returns the number of rows inserted.
3. Delete Command - Returns the number of rows deleted.
4. Update Command - Returns the number of rows updated.



# Command: Stored Procedure

## Creation of SqlCommand object

```
SqlCommand cmd = cn.CreateCommand();
```

--OR--

```
SqlCommand cmd = new SqlCommand();
```

## Using a Stored Procedure (with and without parameters)

```
cmd.CommandType = CommandType.StoredProcedure;  
cmd.CommandText = "GetEmployeeByID";
```

```
cmd.CommandType = CommandType.StoredProcedure;  
cmd.CommandText = "GetEmployeeByID";  
SqlParameter param = new SqlParameter("@EmployeeID", employeeID);  
cmd.Parameters.Add(param);
```

# Command: Passing SQL query from application

## Passing query without parameters

```
SqlCommand cmd = cn.CreateCommand();  
cmd.CommandType = CommandType.Text;  
string query = "SELECT * FROM Employees";  
cmd.CommandText = query;
```

## Passing query with parameters by concatenation (un-safe)

```
SqlCommand cmd = cn.CreateCommand();  
cmd.CommandType = CommandType.Text;  
string query = "SELECT * FROM Employees WHERE EmployeeID = '" + userInput + "'";  
cmd.CommandText = query;
```

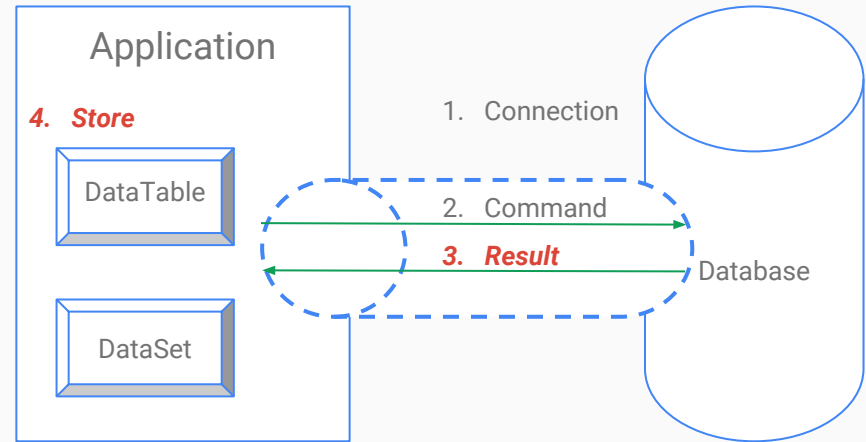
## Passing query with parameters by parameterization (safe)

```
SqlCommand cmd = cn.CreateCommand();  
cmd.CommandType = CommandType.Text;  
string query = "SELECT * FROM Employees WHERE EmployeeID = @EmployeeID";  
cmd.CommandText = query;  
SqlParameter param = new SqlParameter("EmployeeID", userInput);  
cmd.Parameters.Add(param);
```

# Retrieve and Store Result

Objects for retrieving and storing results:

1. **DataReader** - Object used to sequentially read results from a database
2. **DataAdapter** - Object used to fill a DataTable/DataSet with query results
3. **DataSet** - Object which contains DataTables. DataTables are objects that contain results from the database. DataTables consist of DataRow and DataColumn.



# Using DataReader

```
using (SqlDataReader dr = cmd.ExecuteReader())
{
    while (dr.Read())
    {
        int height = dr.GetInt32(0);
        string name = dr.GetString(1);
        DateTime birth = dr.GetDateTime(2);

        Console.WriteLine("Name: {0}", name);
        Console.WriteLine("Height: {0}cm", height);
        Console.WriteLine("Birth Date: {0:d}", birth);
    }
}

//Name: Andrew
//Height: 185 cm
//Birth Date: 11/16/1987
```

# Using DataAdapter/DataTable

```
DataTable table = new DataTable();

SqlDataAdapter da = null;
using (da = new SqlDataAdapter(cmd))
{
    da.Fill(table);
}

foreach(DataRow row in table.Rows)
{
    Console.WriteLine("-----")
    foreach(var item in row.ItemArray)
    {
        Console.WriteLine(Item: {0}, item);
    }
}
```

```
//-----
//Item: Andrew
//Item: Male
//Item: Texas
//-----
//Item: Jacob
//Item: Male
//Item: Missouri
//-----
//Item: Jennifer
//Item: Female
//Item: Texas
```