ADO.NET

by Koshin Abdi & Chary Gurney

What is ADO.NET?

- Set of Classes
- Accessed in System.Data namespace
  - System.Data.SqlClient
  - System.Data.Oledb
  - System.Data.ODBC

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ADO.Data.Model;
using ADO.Data.Settings;
```

2 Basic Steps

1. Connect to the database : App Config Setting
    a. <connectionStrings>
    b. <add name ="Northwind"
    c. connectionString = "Server=MS-STDN-012\SQL2014;User=sa;Password=sqlserver;
    Database=Northwind;"/>
    d. </connectionStrings>

2. Access tables to write queries: Data level

Access Tables to Write Queries

System.Data.SqlClient

Classes:

- Connection
- Command
- DataReader
- DataAdapter
- Parameter
- Transaction

Connection

```
public class Settings
    {
        private static string _connectionString;

        public static string ConnectionString
        {
            get
            {
                if (string.IsNullOrEmpty(_connectionString))
                {
                    _connectionString = ConfigurationManager.ConnectionStrings["Northwind"].ConnectionString;
                }

                return _connectionString;
            }
        }
    }
```

Command & DataReader

```
public List<Product> GetAllProducts()
    {
        List<Product> products = new List<Product>();

        using (SqlConnection cn = new SqlConnection(Settings.Settings.ConnectionString))
        {
          //This is Command Syntax
            SqlCommand cmd = new SqlCommand();
            cmd.CommandText = "Select Products.ProductID, Products.productName, Categories.CategoryName, Categories.[Description], Products.UnitPrice, Products.UnitsInStock from Products
inner join Categories on Products.CategoryID = Categories.CategoryID";
            cmd.Connection = cn;

            cn.Open();
              //This is DataReader Syntax
            using (SqlDataReader dr = cmd.ExecuteReader())
            {
                while (dr.Read())
                {
                    products.Add(PopulateFromDataReader(dr));
                }
            }
        }
        return products;
    }
```

BENEFITS OF ADO.NET

1. Interoperability
2. Maintainability
3. Programmability
4. Performance
5. Scalability

Typed Data Sets vs UnTyped Data Sets

untyped dataset: dataset.Table[“Products”].Rows[0][“ProductID”];

typed dataset: NorthwindDataSet.Products[0].ProductID;