Retrieve Data Quickly Using the SqlDataReader

Paul D. Sheriff

BUSINESS SOLUTIONS ARCHITECT, FAIRWAY TECHNOLOGIES, INC.

www.fairwaytech.com   psheriff@fairwaytech.com

Module Goals

Read data using DataReader

Build generic list of entity objects

Use extension methods

Process multiple result sets
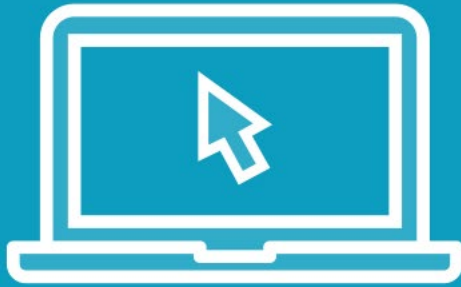
Overview of Data Reader

DataReader

| Forward-only cursor | Fastest method of reading data | Be sure to close (use using block) |

Demo

Simple example of a data reader

DataReader

GetOrdinal(string)
(returns integer)

GetInt32(int)

GetString(int)

GetDateTime(int)

IsDBNull(int)
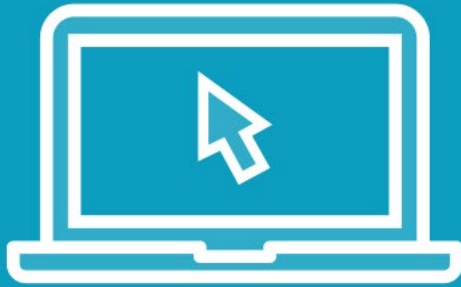
Perform Conversions for Generic List

```csharp
while (dr.Read()) {
  Products.Add(new Product {
      ProductId = dr.GetInt32(dr.GetOrdinal("ProductId")),

      ProductName = dr.GetString(dr.GetOrdinal("ProductName")),

      IntroductionDate = dr.GetDateTime(dr.GetOrdinal("IntroductionDate")),

      Price = dr.GetDecimal(dr.GetOrdinal("Price")),

      RetireDate = dr.IsDBNull(dr.GetOrdinal("RetireDate")) ?

                   (DateTime?)null :

                   Convert.ToDateTime(dr["RetireDate"])
  });
}
```

Demo

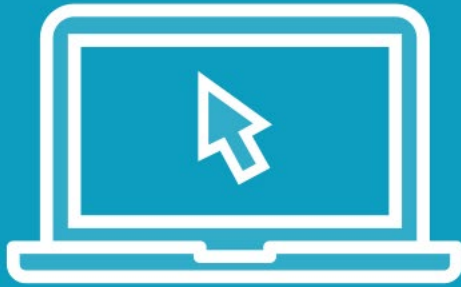**Create List<T> from data reader**

GetFieldValue() Generic Method

GetFieldValue<>() (Automatic conversion)

Still have to call GetOrdinal()

Does not handle nullable fields

Demo

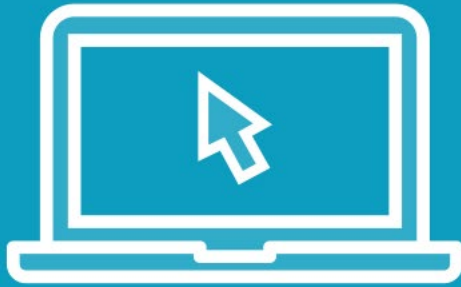**Using GetFieldValue<T>()**

Create Extension Method

GetFieldValue
<T>(string)

Pass in column
name

Handles nullable
fields

Demo

Simplify code using an extension method

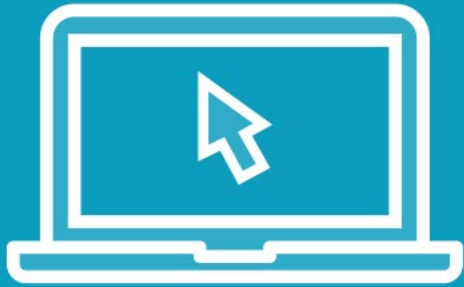# Multiple Result Sets

Pass in two or more SELECT statements

Return two or more SELECT statements from stored procedure

Use NextResult() method

Avoids multiple calls to database

Demo

Multiple result sets

Summary

Several ways to retrieve data

Built-in methods are cumbersome
- Don't handle nullable fields
- Must call GetOrdinal()

Create your own extension method
- Handle nullable fields
- Avoid call to GetOrdinal()

Multiple result sets avoids multiple calls

Coming up in the next module...

Handling exceptions

Gather ADO.NET exception info