Selecting a Single Piece of Data from a Collection



Paul D. Sheriff
Business / IT Consultant

psheriff@pdsa.com www.pdsa.com

Module Goals



Finding a specific item in a collection

Searching for an item using...

- First() / FirstOrDefault()
- Last() / LastOrDefault()
- Single() / SingleOrDefault()

Finding a Single Value

Return Null if Not Found

```
FirstOrDefault(
expression, [default])
```

Search forward in collection

Returns null if no values found or default value

```
LastOrDefault(
expression, [default])
```

Search backward in collection

Returns null if no values found or default value



Exception Thrown if Not Found

First(expression)

Search forward in collection

Throws exception if no values found

Last(expression)

Search backward in collection

Throws exception if no values found



Search for Only a Single Item

Single(expression)

Throws exception if collection is null, or more than one element returned

SingleOrDefault(expressi
 on, [default])

Throws exception if collection is null, or more than one element returned

Otherwise, returns null or default



```
SamplesViewModel vm = new();
try {
  var result = vm.FirstQuery();
  vm.Display(result);
catch (ArgumentNullException ex) {
  vm.Display(ex);
catch (InvalidOperationException ex) {
  vm.Display(ex);
```

◄ Create instance of view model

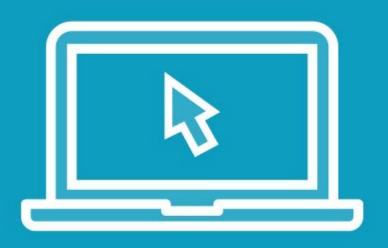
◄ Call the method to try out

◄ The Product collection is null

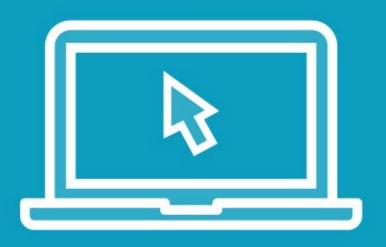
- ◆ First()/Last() = No item found that matches the criteria
- Single*() = Multiple values were found

Demos

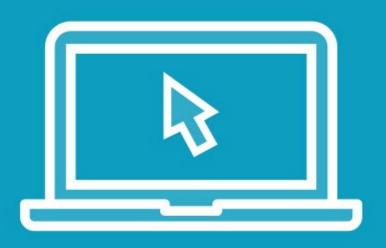




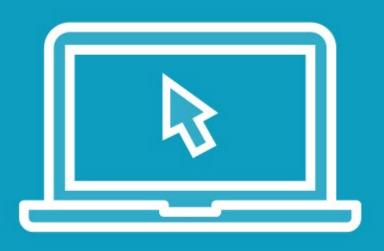
Search forward for an element using First()



Search forward for an element using FirstOrDefault()



Search backwards for an element using Last()



Search backwards for an element using LastOrDefault()

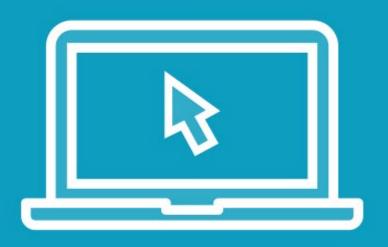
Single*() Methods

Returns a single element

Throws exception if not found

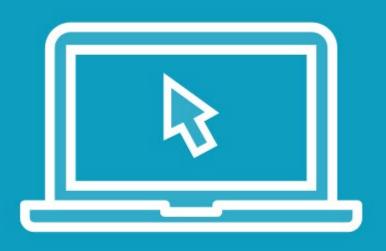
Throws an exception if multiple elements





Search for only one element using Single()





Search for only one element using SingleOrDefault()

When to Use Which Method

First()/Last() vs. FirstOrDefault()/LastOrDefault()

First()/Last()

If you expect the element to be present

Want to handle/throw an exception if

not found

FirstOrDefault()/LastOrDefault()

If you are not sure if element is present

Don't want to handle an exception

Want to get back a null or other default value



First() vs. Single()

First()

If you expect the element to be present

Want to handle/throw an exception if

not found

Only searches until it finds the element Faster than Single()

Single()

If you expect the element to be present

Want to handle/throw an exception if not found

Searches the entire list every time Slower than First()



SingleOrDefault() vs. FirstOrDefault()

SingleOrDefault()

If you expect the element to be present

Want to handle/throw an exception if not found

Must search the entire list every time Slower than FirstOrDefault()

FirstOrDefault()

If you are not sure if element is present

Don't want to handle an exception

Searches only until it finds the element

Faster than SingleOrDefault()



Module Summary



Locate a single item in a collection

- First() / FirstOrDefault()
- Last() / LastOrDefault()
- Single() / SingleOrDefault()

Option to supply your own default value

Catch exceptions where appropriate

Check for null where appropriate

Use *OrDefault() methods to avoid throwing exceptions



Up Next:

Retrieve Specific Items Using Take, Skip, Distinct, and Chunk