LINQ Extension Methods

Last Modified on 01/17/2018 3:03 am EST

Cora SeQuence supports several LINQ extension methods, which perform transformations on *IEnumberable* types (standard .NET functionality). The standard query operators allow queries to be applied to any *IEnumerable*-based information source.

In the table below, *seq* is an *IEnumberable* instance, predicate is a boolean expression, and selector is an expression of any type.

LINQ Extension Methods

Method Type	Purpose	Example	Return Value Type
Where(predicate)	Gets the items that meet the condition.	*{Task1}.Tasks.Where(IsCompleted)*	IEnumerable
Last()	Gets the last element.	*Wf.ActivityScope("taskb").Last()*	Element
seq.LastOrDefault()	Gets the last element or returns the default type in case there are no activities. In case of class, it is null. In case of int, it is zero.	*Wf.ActivityScope("taskb").LastOrDefault()*	Element or Null
Last(predicate)	Gets the last element that meets the condition.	*Wf.ActivityScope("taskb").Last(UpdatedAt < now())*	Element
seq.LastOrDefault(predicate)	Gets the last element that meets the condition, or the default type. In case of class it is null. In case of int, it is zero.	*Wf.ActivityScope("taskb").LastOrDefault(UpdatedAt < now())*	Element or Null
First()	Gets the first element.	*{Task1}.Tasks.Where(IsCompleted).First()*	Element
FirstOrDefault()	Gets the first element or the default type in case there are no activities. In case of class, it is null. In case of int, it is zero.	*{Task1}.Tasks.Where(IsCompleted).FirstOrDefault()*	Element or Null
At(int index)	Gets the element at a specific index.	*{Task1}.Tasks.At(1)*	Element
ElementAt(int index)	Same as At().	*{Task1}.Tasks.ElementAt(1)*	Element

Method Type	Purpose	Example	Return Value Type
Count(predicate)	Counts the elements that meet the conditions.	*{Form1}.Query("DataTable1").Count(Field("Price")>100)*	Integer
Count()	Counts the elements.	*{Task1}.Tasks.Count()*	Integer
All(predicate)	Returns true if all elements meet the condition, otherwise it returns false.	*{Form1}.Query("DataTable1").All(Field("Checked")==True)*	Boolean
Any()	Returns true if there are any elements, otherwise it returns false.	*{Form1}.Query("DataTable1").Any()*	Boolean
Any(predicate)	Returns true if there are any elements that meet the condition, otherwise it returns false.	*{Form1}.Query("DataTable1").Any(Field("Checked")==True)*	Boolean
OrderBy(selector)	Sorts elements in ascending order.	*{Form1}.Query("DataTable1").OrderBy(Field("txtName"))*	Elements
seq.OrderByDescending (selector)	Sorts elements in descending order.	*{Form1}.Query("DataTable1").OrderByDescending(Field("txt1"))*	Elements
Reverse()	Inverts the order of the elements in a sequence.	*{Form1}.Query("DataTable1").Select(Field("txtName")).Reverse()*	Elements
Distinct()	Returns distinct elements from a sequence.	*{Form1}.Query("DataTable1").Select(Field("txtName")).Distinct()*	Distinct elements from a sequence
Take(int count)	Returns a specified number of contiguous elements from the start of a sequence.	*{Form1}.Query("DataTable1").Take(1).Select(Field("txt1"))*	Elements

Method Type	Purpose	Example	Return Value Type
seq.TakeWhile(predicate)	Returns elements from a sequence as long as a specified condition is true, and then skips the remaining elements.	*{Form1}.Query("DataTable1").TakeWhile(Field("Status"==True).Select(Field("txt1"))*	Elements
Select(selector)	Selects an element.	*{Form1}.Query("DefaultView").Select(Field("txt1"))*	Elements
seq.SelectMany(selector)	Selects many elements.	Gets all tasks from all instances. *{Task1}.Scope().SelectMany(tasks)* Gets all user IDs that have not taken action on the tasks. *{Task1}.Scope().SelectMany(Tasks).where(not isCompleted).Select(ToID)*	Elements
Skip(int count)	Bypasses the first specified number (in the example, two elements) of elements in a sequence, and then returns the remaining elements	*{Form1}.Query("DataTable1").Skip(2).Select(Field("txt1"))*	Elements
seq.SkipWhile(predicate)	Bypasses elements in a sequence as long as they meet the specified condition and they are located at the beginning of the items list. After this, the remaining elements are returned.	*{Form1}.Query("DataTable1").SkipWhile(Field("cmb1.value" == 0).Select(Field("txt1"))*	Elements
Min(selector)	Returns the element with the minimum value.	*{form1}.Query("DataTable1").Min(field("FieldNum"))*	Element

Method Type	Purpose	Example	Return Value Type
Max(selector)	Returns the maximum value from all values of a specified field.	*{form1}.Query("DataTable1").Max(field("FieldNum"))* To return the whole row: *{form1}.Query("DataTable1").OrderByDescending(Field("FieldNum")).First()*	Element
ToArray()	Converts the elements to an array.	*{form01}.Query("DataTable1").Row().field("Items").ToArray()*	Array