

Basics		
Terms & Phrases	"test" or "hello dolly"	<code>c.Where("test") or c.Where("hello dolly") or c.Where("test" "hello dolly")</code>
Fields	title:"The Right way" and text:"go"	<code>c.Title == "The Right way" or c.Text == "go" or c.Equals("go")</code>
Wildcard	amber	<code>c.ContactName.Contains("amber")</code>
Prefix	amber amber	<code>c.ContactName.StartsWith("amber") or c.ContactName.EndsWith("amber")</code>
Fuzzy	roam~ or roam~0.8	<code>c.ContactName.Like("roam") or c.ContactName.Like("roam", 0.8)</code>
Proximity	"jakarta apache"~10	<code>c.ContactName.Like("jakarta apache", 10)</code>
Inclusive Range	mod_date:[20020101 TO 20030101]	<code>c.ModifiedDate.Between("20020101", "20030101", Inclusion.Both)</code>

Basics (cont)		
Exclusive Range	title:- {Aida TO Carmen}	<code>c.Title.Between("Aida", "Carmen", Inclusion.None)</code>
Boosting	jakarta^4 apache	<code>c.Title.Equals("jakarta").Boost(4) c.Title.Equals("apache")</code>
Boolean Or	"jakarta apache" OR jakarta	<code>where c.Title.Equals("jakarta apache") c.Equals("jakarta")</code>
Boolean And	"jakarta apache" AND "Apache Lucene"	<code>where c.Equals("jakarta apache") && c.Equals("Apache Lucene")</code>
Boolean Not	"jakarta apache" NOT "Apache Lucene"	<code>where c.Equals("jakarta apache") && !c.Equals("Apache Lucene")</code>
Grouping	(jakarta OR apache) AND website	<code>where (c.Title == "jakarta" c.Title == "apache") && (c.Title == "website")</code>

Supported IQueryable methods		
Restriction Operators	Where	<pre>var results = from d in queryable where d.Name == "Sitecore" select d; or var results = queryable.Where(d => d.Name == "Sitecore");</pre>
Projection Operators	Select	<pre>var results = from d in queryable select d.Name; or var results = queryable.Select(d => d.Name);</pre>
Anonymous types		<pre>results = queryable.Select(d => new { d.Name, d.Id });</pre>
Unsupported	SelectMany	
Partitioning Operators	Take	<pre>results = queryable.Take(10);</pre>
Partitioning Operators	Skip	<pre>results = queryable.Skip(10);</pre>
Partitioning Operators	Page	<pre>results = queryable.Page(2, 100);</pre>



By **Kiran Patil** (kpatil)
cheatography.com/kpatil/

sitecorebasics.wordpress.com/

Published 29th November, 2014.

Last updated 11th May, 2016.

Page 1 of 6.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish

Yours!

<https://apollopad.com>

Supported IQueryable methods (cont)

Ordering Operators OrderBy results = queryable.OrderBy(d => d.Name);

Ordering Operators OrderBy Descending results = queryable.OrderByDescending(d => d.Name);

Ordering Operators ThenBy results = queryable.OrderBy(d => d.Name).ThenBy(d => d.Id);

Ordering Operators ThenBy Descending results = queryable.OrderBy(d => d.Name).ThenByDescending(d => d.Id);

Unsupported Reverse

Grouping Operators Unsupported GroupBy - Simple 1
GroupBy - Simple 2
GroupBy - Simple 3
GroupBy - Nested
GroupBy - Comparer
GroupBy - Comparer, Mapped

Set Operators Unsupported Distinct Union
Intersect Except

Element Operators First - Simple results = queryable.First();

Supported IQueryable methods (cont)

Element Operators First - Condition results = queryable.First(d => d.Name == "Sitecore");

Element Operators FirstOrDefault - Simple results = queryable.FirstOrDefault();

Element Operators FirstOrDefault - Condition results = queryable.FirstOrDefault(d => d.Name == "Sitecore");

Element Operators ElementAt results = queryable.ElementAt(10);

Element Operators Last 'result = queryable.Last();' **or**
result = queryable.Last(d => d.Id > 10);

Element Operators LastOrDefault result = queryable.LastOrDefault(); **or**
result = queryable.LastOrDefault(d => d.Id > 10);

Element Operators Single result = queryable.Single(); **or**
result = queryable.Single(d => d.Id > 10);

Supported IQueryable methods (cont)

Element Operators Single-OrDefault result = queryable.SingleOrDefault(); **or**
result = queryable.SingleOrDefault(d => d.Id > 10);

Quantifiers Any - Simple results = queryable.Any();

Quantifiers Any - Grouped results = queryable.Any(d => d.Name == "Sitecore");

Quantifiers Unsupported All

Aggregate Operators Count - Simple results = queryable.Count();

Aggregate Operators Count - Conditional results = queryable.Count(d => d.Id < 10);

Aggregate Operators Unsupported Sum Min Max
Average
Aggregate

Join Operators Unsupported Cross Join Group
Join Cross Join
with Group Join
Left Outer Join



By **Kiran Patil** (kpatil)
cheatography.com/kpatil/

Published 29th November, 2014.
Last updated 11th May, 2016.
Page 2 of 6.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
<https://apollopadd.com>

IQueryable Extensions

Filtering

```
results =
queryable.Fi-
lter(d => d.Id >
4 && d.Id < 8);
```

Facets **Simple Faceting**

```
var results =
queryable.Fa-
cetOn(d =>
d.Name); var
facets = result-
s.GetFacets();
foreach (var
category in
facets.Categ-
ories) { Consol-
e.WriteLine(-
category .Name);
foreach (var
facetValue in
category.Values)
{ Console.Writ-
eLine("{0}:
{1}", facetValu-
e.Name, facetV-
alue.Aggregate);
} }
```

IQueryable Extensions (cont)

Pivot Faceting

```
var results =
queryable.Fa-
cetPivotOn(p =>
p.FacetOn(d =>
d.Name).Face-
tOn(d =>
d.Year)); var
facets =
results.GetF-
acets();
foreach (var
category in
facets.Categ-
ories) {
Console.Writ-
eLine(category
.Name); foreach
(var facetValue
in category.V-
alues) {
Console.Writ-
eLine("{0}:
{1}", facetV-
alue.Name,
facetValue.A-
ggregate); } }
```

Boosting

```
results =
queryable.Wh-
ere(d => d.Id
== 7.Boost(2-
f)).Where(d =>
d.Template.C-
ontains("o"));
```

IQueryable Extensions (cont)

Other **Between**

```
results = querya-
ble.Where(item =>
item.Price.Betw-
een(50.0f,
400.0f, Includ-
ion.Both)); or
results = querya-
ble.Where(item =>
item.Price.Betw-
een(2.0f, 12.0f,
Inclusion.Both)
|| item.Price.B-
etween(80.0f,
400.0f, Includ-
ion.Both)); or
results = querya-
ble.Where(d =>
d.Date.Betwe-
en(new DateTime(-
2004, 12, 31),
DateTime.Now,
Inclusion.Both));
or results =
queryable.Where(d
=> d.Id.Betwe-
en(1, 4, Includ-
ion.Both)); or
results = querya-
ble.Where(d =>
d.Id.Between(1,
4, Inclusion.Lo-
wer)); or results =
queryable.Where(d
=> d.Id.Betwe-
en(1, 4, Includ-
ion.Upper)); or
results = querya-
ble.Where(d =>
d.Id.Between(1,
4, Inclusion.No-
ne));
```



By **Kiran Patil** (kpatil)
cheatography.com/kpatil/

sitecorebasics.wordpress.com/

Published 29th November, 2014.
Last updated 11th May, 2016.
Page 3 of 6.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish
Yours!
<https://apollopad.com>

IQueryable Extensions (cont)

Other string.Contains results = queryable.Where(d => !d.Template.Contains("Hello"));

Other string.CompareTo results = queryable.Where(d => !d.Name.CompareTo("Hello") == 1);

Other Equal results = queryable.Where(d => d.Id.Equal(4));

Other Matches results = queryable.Where(i => i.Template.Matches("^.*\$"));

Other MatchWildcard results = queryable.Where(i => i.Template.Where(i => i.Template.MatchWildcard("H?li*m"));

Other Like results = queryable.Where(i => i.Template.Like("Citecoar"));

Other string.StartsWith results = queryable.Where(d => !d.Name.StartsWith("Hello"));

Other string.EndsWith results = queryable.Where(d => !d.Name.EndsWith("Hello"));

IQueryable Extensions (cont)

GetResults results = queryable.GetResults().Hits.Where(i => i.Document.Name.Contains("o")).Where(hit => hit.Score > 0.6);

GetFacets results = queryable.Where(d => d.Id > 0).FacetOn(d => d.Template, 0).GetFacets();

Daily Life usages

This returns the results of a query on your search index and returns it as a SearchResultItem type.

```
using (var context = ContentSearchManager.GetIndex(item).CreateSearchContext()) {
    IQueryable<SearchResultItem> searchQuery = context.GetQueryable<SearchResultItem>().Where(item => item.Name == "Sitecore") }
```

Daily Life usages (cont)

This converts the query to something your provider understands. For example, for Lucene it is converted to: `_name:sitecore`



By **Kiran Patil** (kpatil)
cheatography.com/kpatil/

sitecorebasics.wordpress.com/

Published 29th November, 2014.
Last updated 11th May, 2016.
Page 4 of 6.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
<https://apollopod.com>

Daily Life usages (cont)

```
+ using (var context =
(+_na- ContentSearchManager.Get-
me:sit- Index(item).CreateSearch-
ecore) Context()) { IQueryable<-
+(titl- e<SearchResultItem>
e:test searchQuery = context.G-
body:- etQueryable<SearchResult-
cms) Item>().Where(item =>
or (+ item["_name"] == "Sitec-
(+_n- ore").Where(item =>
ame:sit- item.Title == "Test ||
ecore) item.Body.Contains("C-
+(titl- MS")) } OR using (var
e:test context = ContentSearc-
body:- hManager.GetIndex
cms)) (item).CreateSearchConte-
xt()) { IQueryable<Sear-
chResultItem> searchQuery
= context.GetQueryable<-
SearchResultItem>().Whe-
re(item => item.Name ==
"Sitecore").Where(item =>
item.Title == "Test ||
item.Body.Contains("CMS")
```

Daily Life usages (cont)

```
Find all var index = ContentSe-
the items archManager.GetIndex("-
that have sitecore_master_in-
"s" in dex"); using (var
their context = index.Create-
names SearchContext()) {
Console.WriteLine(
context.GetQueryable<-
SearchResultItem>().W-
here(resultItem=>resu-
ltItem.Name.Contains("-
s")));
```

```
Finding context.GetQueryable<-
the first <SearchResultItem>().F-
item irstOrDefault(result-
named Item=>resultItem.Name-
Sitecore == "sitecore")
```

```
Finding context.GetQueryable<-
the count <SearchResultItem>().C-
of items ount()
in a
database
```

```
Finding context.GetQueryable<Search-
all the ResultItem>().Where(resultI-
items tem=>resultItem.Template-
that use Name=="Sample Item")
a certain
template
```

```
same context.GetQueryable<-
thing as <SearchResultItem>().W-
above here(resultItem=>res-
but this ultItem.TemplateId==I-
time D.Parse("{76036F5E-CBC-
using the E-46D1 -AF0A-4143F9B55-
template 7AA} ")
id
```

Daily Life usages (cont)

```
find all items context.GetQueryable<-
that have <SearchResultItem>-
some ().Where(resultIte-
content in m=>resultItem["page-
the "Page _title"]!="")
Title" field
```

```
find all items context.GetQueryable<-
that have <SearchResultItem>-
lorem in the ().Where(resultIte-
body m=>resultItem["b-
body"].Contains("lore-
m"))
```

```
finding an context.GetQueryable<-
item where <SearchResultItem>()
item.Fiel- .Where(resultItem=-
ds["Tax- >resultItem["taxono-
onomy my_items"].Contains-
Items"] // is (Sitecore.ContentS-
a multi list earch.Utilities.Id-
field and Helper.NormalizeGuid
contains "- (" {016424DF-865C-49-
{01642- DC-9D7A-40EB8101C7-
4DF-865C-- 17} ", true))
49DC-
9D7A-4-
0EB810-
1C717"
```



By **Kiran Patil** (kpatil)
cheatography.com/kpatil/

sitecorebasics.wordpress.com/

Published 29th November, 2014.

Last updated 11th May, 2016.

Page 5 of 6.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish

Yours!

<https://apollopad.com>

Daily Life usages (cont)

```
Find items based on Data Source string
var query = LinqHelper.CreateQuery(context,
UIFilterHelpers.ParseDataSourceString(txtQuery.Text.Trim())).Select(x => x.GetItem()).AsEnumerable().Where(x => x.Language == Sitecore.Context.Language)
```

Advanced Usages

```
Get Particular template's item for current website and latest version of an item only
var searchResultItem = context.GetQueryable<SearchResultItem>().Where(x => x.Path.Contains(sitecoreItem.Paths.FullPath)).Where(x => x.TemplateId == Sitecore.Data.ID.Parse("YOURTEMPLATEID")).AsEnumerable().Where(x => x != null && x.GetItem() != null && x.Version == x.GetItem().Versions.GetLatestVersion().Version.ToString());
```

Advanced Usages (cont)

```
To get all details related to search index like IsClean, LastUpdated, NumberOfDocuments, NumberOfFields
ISearchIndex
isearchIndex = ContentSearchManager.GetIndex("YOURINDEXNAME");
```

```
The _content field aggregates all fields from the item to allow full text searching
[IndexField(Sitecore.ContentSearch.BuiltInFields.Content)] public string ItemContent { get; set; }
```

```
To do your index field configuration
<fieldMap type="Sitecore.ContentSearch.FieldMap, Sitecore.ContentSearch"> <fieldNames hint="raw:-AddFieldByFieldName">
```



By **Kiran Patil** (kpatil)
cheatography.com/kpatil/

sitecorebasics.wordpress.com/

Published 29th November, 2014.

Last updated 11th May, 2016.

Page 6 of 6.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish

Yours!

<https://apollopad.com>