Sitecore Search Developers Cheat Sheet

by Kiran Patil (kpatil) via cheatography.com/20093/cs/2946/

Basics

Terms & Phrases	"test" or "-hello dolly"	`c.Where("test")` [illegible] `dolly")`
Fields	title:"The Right way" and text:"go"	`c.Title == "The Right way" or c.Text == " go"` **or** `c.Equals(" go")`
WildCard	*amber*	`c.ContactName.Contains("amber")`
Prefix	amber amber	c.ContactName.StartsWith("amber") or c.ContactName.EndsWith("amber")
Fuzzy	roam or roam^0.8	`c.ContactName.Like("roam") or c.ContactName.Like("roam", 0.8)`
Proximity	"jakarta apache"~10	`c.ContactName.Like("jakarta apache", 10)`
Inclusive Range	mod_date:[20020101 TO 20030101]	`c.ModifiedDate.Between("20020101", "20030101", Inclusion.Both)`
Exclusive Range	title:{Aida TO Carmen}	`c.Title.Between("Aida", "Carmen", Inclusion.None)`
Boosting	jakarta^4 apache	`c.Title.Equals("jakarta".Boost(4)) || c.Title.Equals("apache”)`
Boolean Or	"jakarta apache" OR jakarta	`where c.Title.Equals("jakarta apache") || c.Equals("jakarta")`

Basics (cont)

Boolean And	"jakarta apache" AND "-Apache Lucene"	`where c.Equals("jakarta apache") && c.Equals("A`
Boolean Not	"jakarta apache" NOT "-Apache Lucene"	`where c.Equals("jakarta apache") && !c.Equals("A`
Grouping	(jakarta OR apache) AND website	`where (c.Title == "jakarta" || c.Title == "apache` [illegible] `site")`

Supported IQueryable methods

Restriction Operators	Where	`var results = from d in queryable where d.Name == " S` [illegible] `d;` **or** `var results = queryable.Where(d => d.Name ==`
Projection Operators	Select	`var results = from d in queryable select d.Name;` **or** `var` [illegible] `queryable.Select(d => d.Name);`
Anonymous types		`results = queryable.Select(d => new { d.Name, d.Id`
Unsupported	SelectMany	
Partitioning Operators	Take	results = `queryable.Take(10);`
Partitioning Operators	Skip	`results = queryable.Skip(10);`

Supported IQueryable methods (cont)

Partitioning Operators	Page	[illegible]
Ordering Operators	OrderBy	results = queryable
Ordering Operators	OrderByDescending	results = queryable
Ordering Operators	ThenBy	results = queryable
Ordering Operators	ThenByDescending	results = queryable
Unsupported	Reverse	
Grouping Operators	Unsupported	GroupBy -Simple 1 GroupBy -Compar
Set Operators	Unsupported	Distinct Union Inte
Element Operators	First - Simple	[illegible]
Element Operators	First - Condition	`results = quer`
Element Operators	FirstOrDefault - Simple	`results = quer`
Element Operators	FirstOrDefault - Condition	`results = quer` [illegible] `core");`

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

sitecorebasics.wordpress.com/

Published 29th November, 2014.
Last updated 11th May, 2016.

Supported IQueryable methods (cont)

Element Operators	ElementAt	`results = querya ble.El eme ntA t(10);`
Element Operators	Last	'result = queryable.Last();' **or** `result = querya ble.Last(d => d.Id > 10);`
Element Operators	LastOr-Default	`result = querya ble.La stO rDe fault();` **or** `result = querya ble.La stO rDe fault(d => d.Id > 10);`
Element Operators	Single	`result = querya ble.Si ngle();` **or** `result = querya ble .Si ngle(d => d.Id > 10);`
Element Operators	Single-OrDefault	`result = querya ble.Si ngl eOr Def ault();` **or** `result = querya ble.Si ngl eOr Def ault(d => d.Id > 10);`
Quanti-fiers	Any - Simple	`results = querya ble.Any();`
Quanti-fiers	Any - Grouped	`results = querya ble.Any(d => d.Name == " Sit eco r e");`
Quanti-fiers	Unsupp-orted	All
Aggregate Operators	Count - Simple	`results = querya ble.Co unt();`
Aggregate Operators	Count - Condi-tional	`results = querya ble.Co unt(d => d.Id < 10);`
Aggregate Operators	Unsupp-orted	Sum Min Max Average Aggregate
Join Operators	Unsupp-orted	Cross Join Group Join Cross Join with Group Join Left Outer Join

IQueryable Extensions

Filtering		`results = querya ble.Fi lte r(d => d.Id > 4 && d.Id <`
Facets	Simple Faceting	`var results = querya ble.Fa cetOn(d => d.Name); var fa`
	Pivot Faceting	`var results = querya ble.Fa cet Piv otOn(p => p.Face t` … `A ggr egate); } }`
Boosting		`results = querya ble.Wh ere(d => d.Id == 7.Boos t(2 f)`

IQueryable Extensions (cont)

Other	Between	`results = querya bl`
		`m.P ric e.B etw een`
		`h ere(d => d.Date.B`
		`w een(1, 4, Inclus`
		`(d => d.Id.B etw ee`

Other	string.Co ntains	`results = querya bl`
Other	string.Co mpareTo	`results = querya bl`
Other	Equal	`results = querya bl`
Other	Matches	`results = querya bl`

IQueryable Extensions (cont)

Other	MatchWildcard	`results = queryable.Where(i => i.Template.W`
Other	Like	`results = queryable.Where(i => i.Template.Like ("Cl tec oar "));`
Other	string.StartsWith	`results = queryable.Where(d => !d.Name.StartsWith ("Hello "));`
Other	string.EndsWith	`results = queryable.Where(d => !d.Name.EndsWith("Hello"));`
GetResults		`results = queryable.GetResults().Hits.Where(i => i.Document.Name.Contains ("c` `t.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()) { IQu` `e(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	`using (var context = Conten`
+(+_name:sitecore) + (title:test body:*cms*) or (+(+_name:sitecore) + (title:test body:*cms*))	`using (var context = ContentSearchManager.GetIndex(` `= "Sitecore").Where(item => item.Title == "Test || item.`

Daily Life usages (cont)

Find all the items that have "s" in their names	`tem =>resultItem.Nam`
Finding the first item named Sitecore	`context.GetQueryable`
Finding the count of items in a database	`context.GetQueryable`
Finding all the items that use a certain template	context.GetQueryable<SearchRes
same thing as above but this time using the template id	`context.GetQueryable`
find all items that have some content in the "Page Title" field	`context.GetQueryable`
find all items that have lorem in the body	`context.GetQueryable`

Daily Life usages (cont)

finding an item where item.Fields["TaxonomyItems"] // is a multi list field and contains "{01-642-4DF-86-5C--49DC-9D7A-4-0EB-810-1C7-17}"

```
context.GetQueryable<SearchResultItem>().W[illegible]tItem=>resultItem[" taxono
```

Find items based on Data Source string

```
var query = LinqHelper.C[illegible]uery(context,                    UIFilterHelpers.ParseDa
o re.Context.Language)
```

Advanced Usages

Get Particular template's item for current website and latest version of an item only

```
var searchResultItem = context.GetQueryable<SearchResultItem>()
Version().Version.ToString());
```

Advanced Usages (cont)

To get all details related to search index like IsClean, LastUpdated, NumberOfDocuments, NumberOfFields

```
[illegible]
```

The _content field aggregates all fields from the item to allow full text searching

```
[IndexField(Sitecore.ContentSearch.BuiltinFie
    set;     }
```

To do your index field configuration

```
<fieldMap type="Sitecore.ContentSearch.FieldMap, Si
 yFieldName">
```