Umbraco Examine Search

1. What is Examine?

This is a provider based Indexer/Searcher API and wraps the Lucene.Net indexing/searching engine. Working with Examine is very easy and allows you to query or index almost any content in web site.

Umbraco Examine is the Umbraco implementation of Examine. It is not exclusive to Umbraco and can be used as a completely stand alone component on any project that needs a fast Index.

License:

Examine is under Microsoft Public License (Ms-PL), this is mean, we are free of charge to use it, but can not use it examine name, logo or trademark. More info - <https://examine.codeplex.com/license?LicenseHistoryId=37500>

1. Implementation

* Basic configuration

All configuration comes whit main Umbraco installation.

You can start using examine search immediately.

**Examine Terminology**

* + Indexer - This is the object that performs the storing data into the index.
  + Searcher – The searcher is the object that performs the searching of data that is stored in the index.
  + Index Set - An index set is what defines an index, where the index is saved and how the information is stored in the index.

Naming conventions - Our Indexer, Searcher and associated Index Set must all be named according to convention so that they match.

Conventions:

* + {name}Indexer
  + {name}Searcher
  + {name}IndexSet

Examples:

* + ExternalIndexer
  + ExternalSearcher
  + ExternalIndexSet

After installing Umbraco site look in the config folder. There is all config files that umbraco needs.

* + **/Config/ExamineIndex.config**

By default is configured 3 IndexSets – InternalIndexSet, InternalMemberIndexSet and ExternalIndexSet.

All index sets that starts with ‘Internal’ prefix are internal umbraco and we don not have work there. The same is valid also for searchers and indexers.

Umbraco provides for us ExternalIndexSet:

<IndexSet SetName="ExternalIndexSet" IndexPath="~/App\_Data/TEMP/ExamineIndexes/External/" />

We can extend this configuration, but this will be seen in next sliders.

For now we can search in all site content(fields and document types).

* + **/Config/ExamineSettings.config**

There is ExamineIndexProviders and ExamineSearchProviders. Also by three of kind – internal and external.

* + - ExternalIndexer

<add name="ExternalIndexer" type="UmbracoExamine.UmbracoContentIndexer, UmbracoExamine"/>

We can extend it with this additional properties:

* + - * **dataService** - the type that this provider will instantiate in order to query Umbraco for the data that it requires. Generally this shouldn't need to change unless you want to use test data from a non-umbraco source or you have very custom requirements.
      * **indexSet** - explicitly specifies the index set to use. Generally this wired up based on naming convensions.
      * **supportUnpublished** - if you want the indexer to index content that is not published.
      * **supportProtected** - if you want the indexer to index content that is protected.
      * **runAsync** = will process the queue files into the index asynchronously, unless you are testing, this should always be true.
      * **interval** = how often the async service will process the file queue in seconds.
      * **analyzer** = the Lucene.Net analyzer to use when storing data. See: <http://www.aaron-powell.com/lucene-analyzer>
      * **enableDefaultEventHandler** = will automatically listen for Umbraco events and index when required.
      * **logLevel**="Info" or "Verbose". Info is the default, Verbose will show more detailed logs
    - ExternalSearcher

<add name="ExternalSearcher" type="UmbracoExamine.UmbracoExamineSearcher, UmbracoExamine" />

We can extend it with this additional property:

* + - * **indexSet** - explicitly specifies the index set to use. Generally this wired up based on naming convensions.

Note: In ExamineSearchProviders section, defaultProvider must be the name of our search provider - e.g. – ExternalSearcher. This is by default.

* Custom configuration

<IndexSet SetName="ExternalIndexSet" IndexPath="~/App\_Data/TEMP/ExamineIndexes/External/">

<IndexAttributeFields>

<!-- Set here all page properties that we want to be indexed. -->

<add Name="id" />

<add Name="version" />

<add Name="parentID" />

<add Name="writerID" />

<add Name="creatorID" />

</IndexAttributeFields>

<IndexUserFields>

<!-- Set here all site custom properties that we want to be indexed. -->

<add Name="testTitle" EnableSorting="true" />

<add Name="testDescription" EnableSorting="true" />

</IndexUserFields>

<IncludeNodeTypes>

<!-- Set here all site document types that we want to be indexed. -->

<add Name="Test"/>

</IncludeNodeTypes>

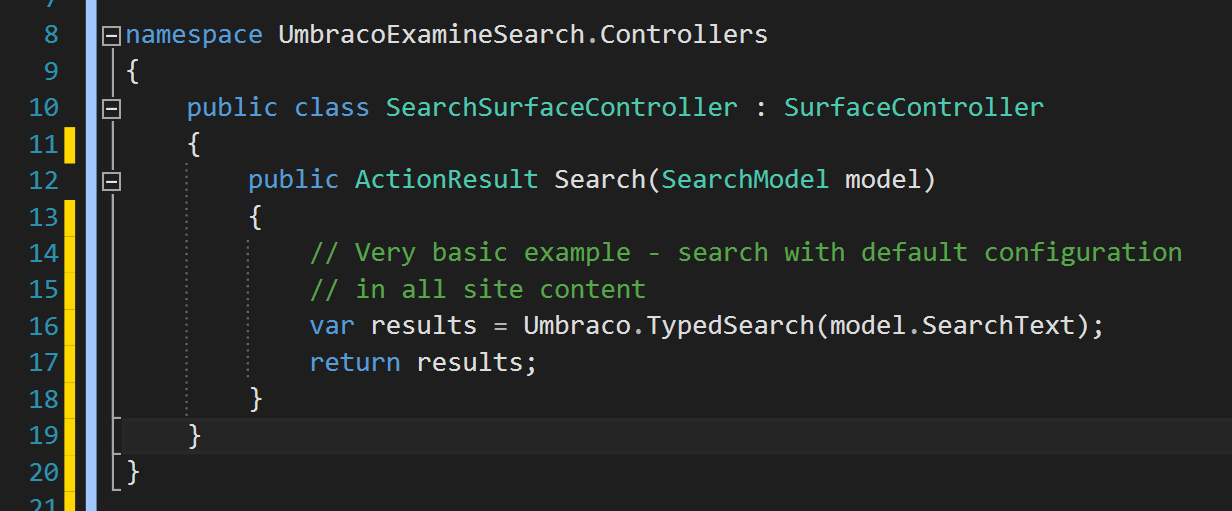
<ExcludeNodeTypes>

<!-- Set here all site document types that we want to NOT be indexed. -->

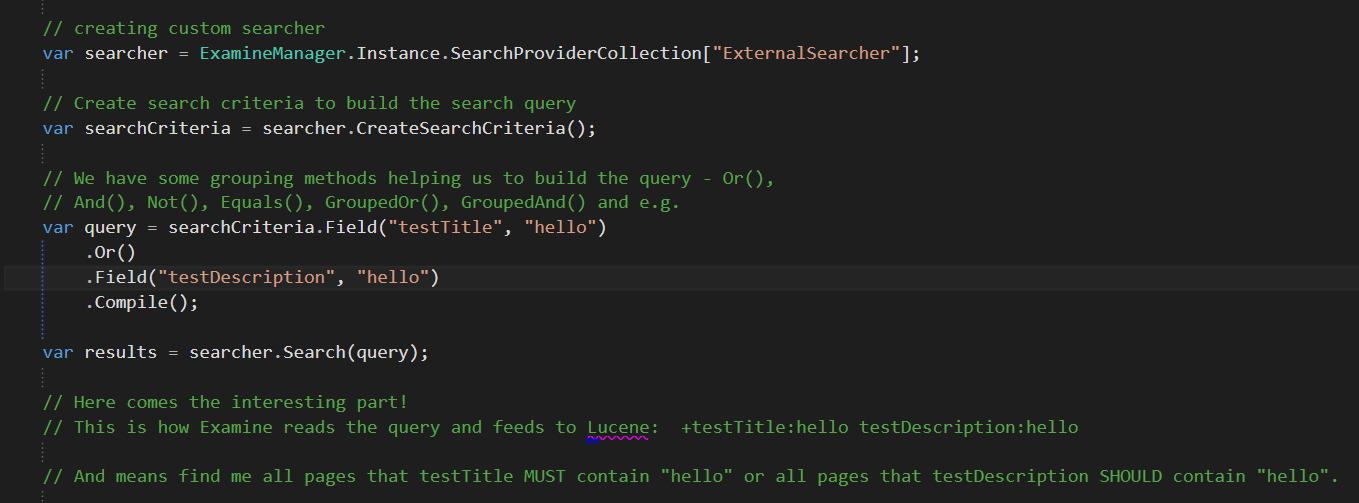
</ExcludeNodeTypes>

</IndexSet>

* Basic code examples



Creating custom searcher:



**Difference between Lucene boolean clauses – MUST and SHOULD**

Assume that there are two clauses: Clause A and Clause B.

* + Clause A have SHOULD, Clause B – SHOULD

This will imply that even if one of the clause is satisfied (A or B), then the document will be a hit.

* + Clause A have MUST, Clause B – SHOULD

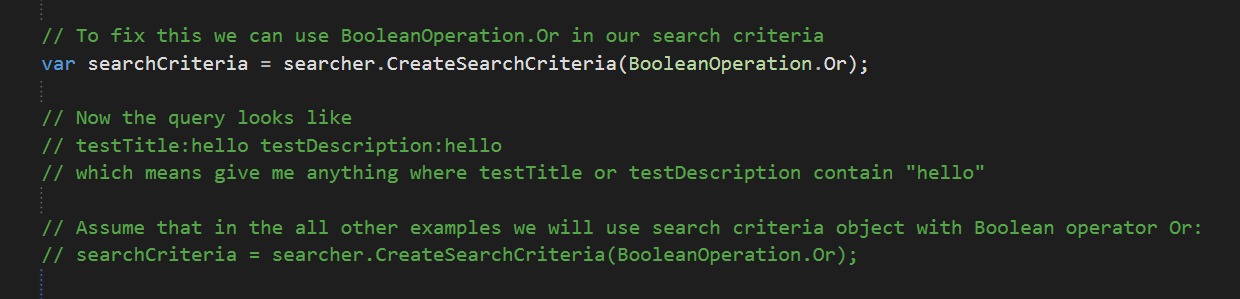
In this case, a document will be a hit when it "will" satisfy clause A whether this document satisfies clause B or not.

But if the document does not satisfies clause A, then no matter whether it satisfies clause B or not, it will not be a hit.

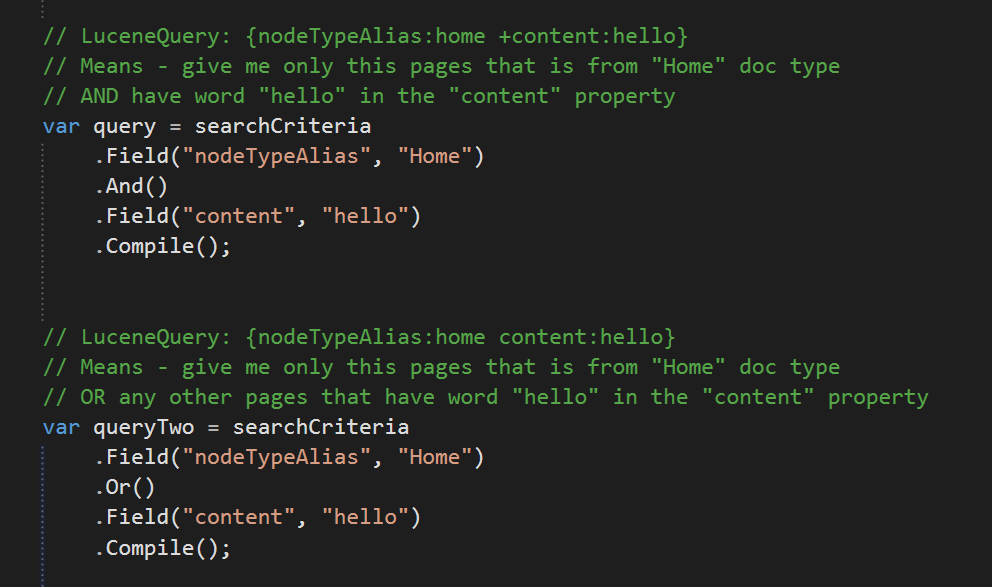
* + Clause A have MUST, Clause B – MUST

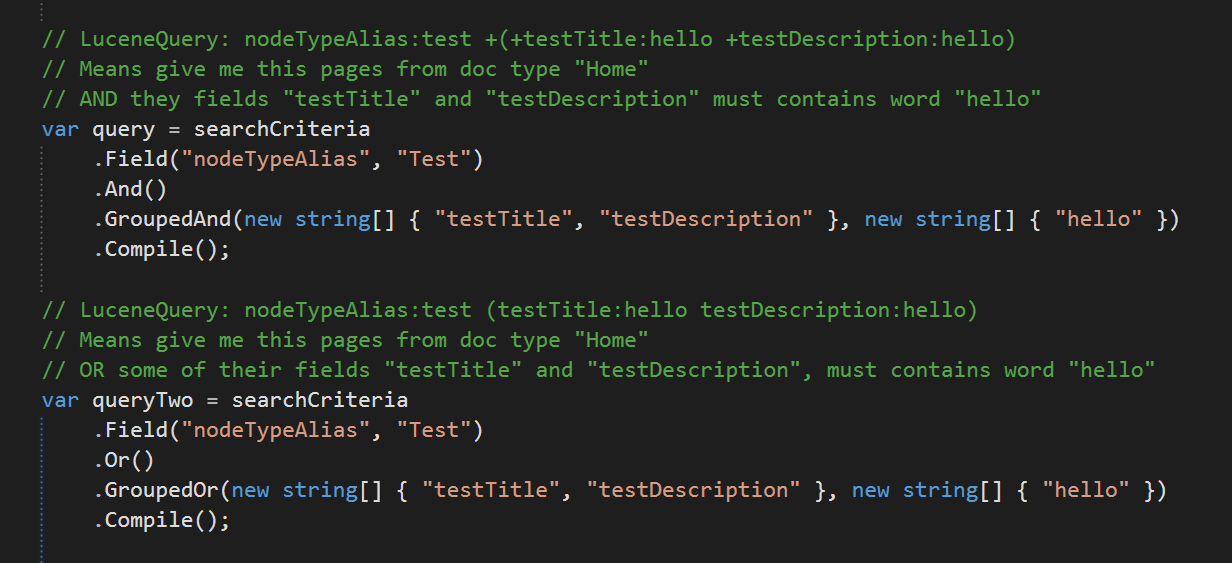
In this case, a document will be a hit, only when it will satisfy "both" the clauses. If it will fail to satisfy even one of the clause, then it will not be a hit.

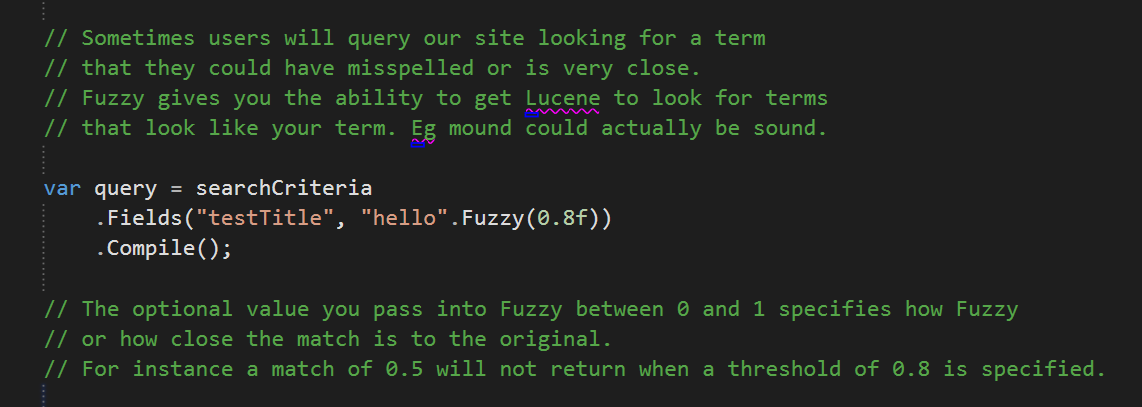
There is third clause - MUST\_NOT. Use this operator for clauses that must not appear in the searching documents.



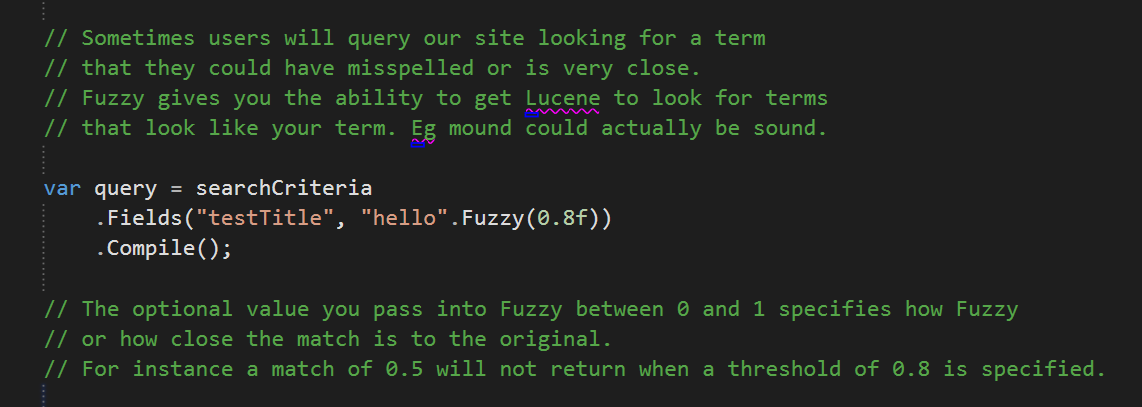
Another examples:



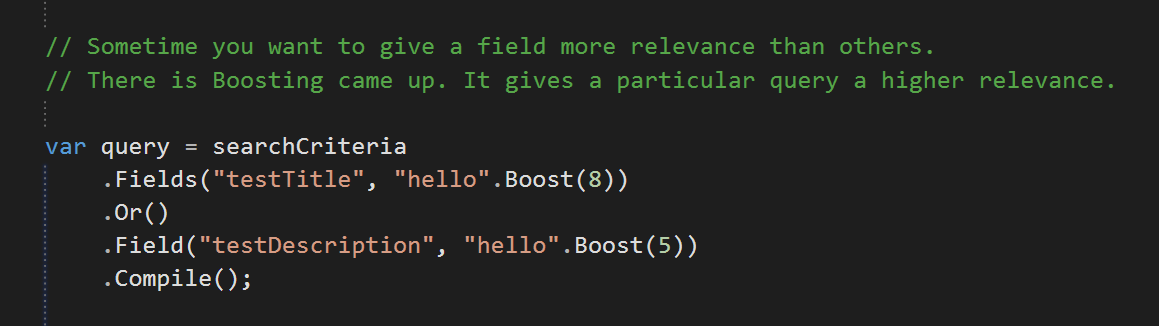




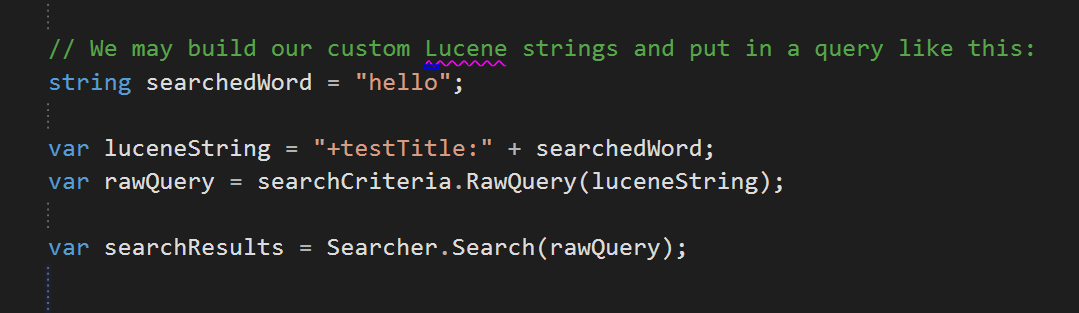
Fuzzy:



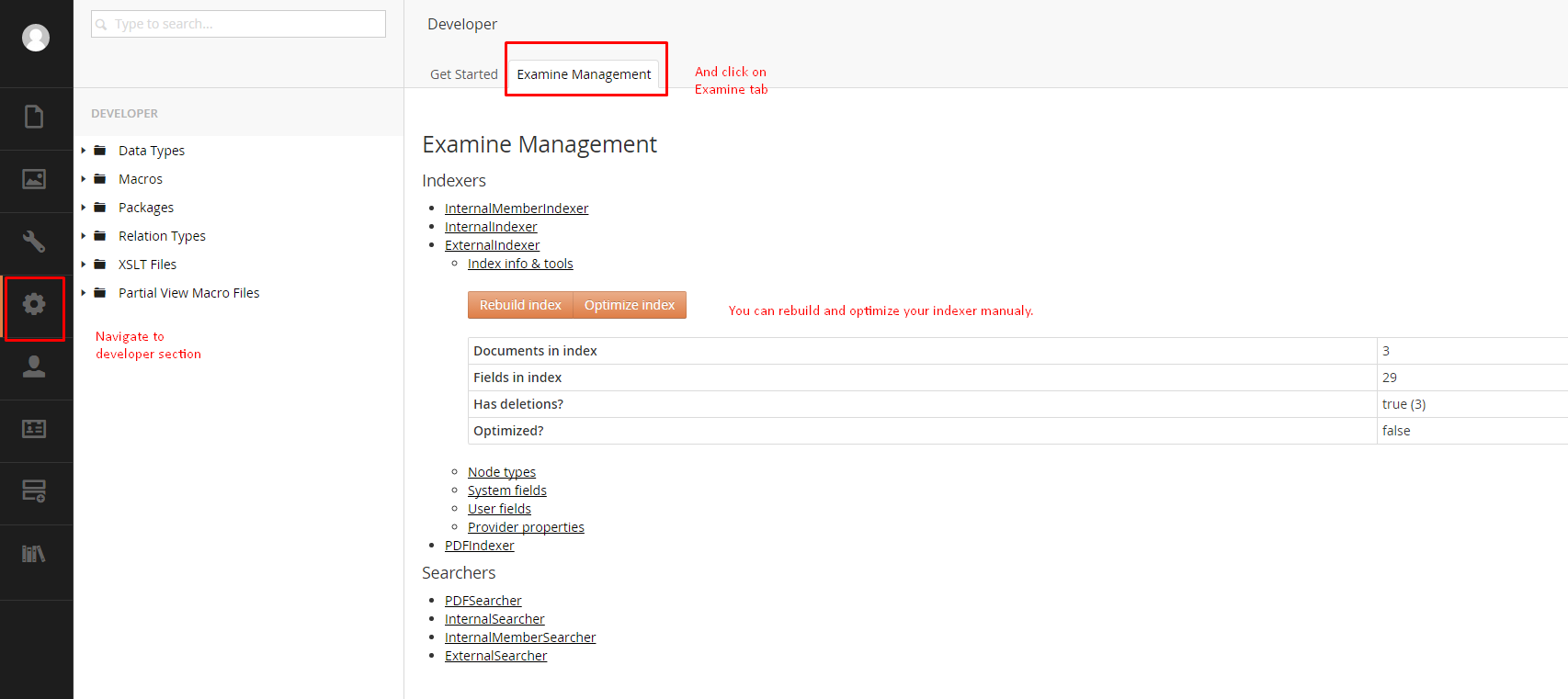
Boosting:

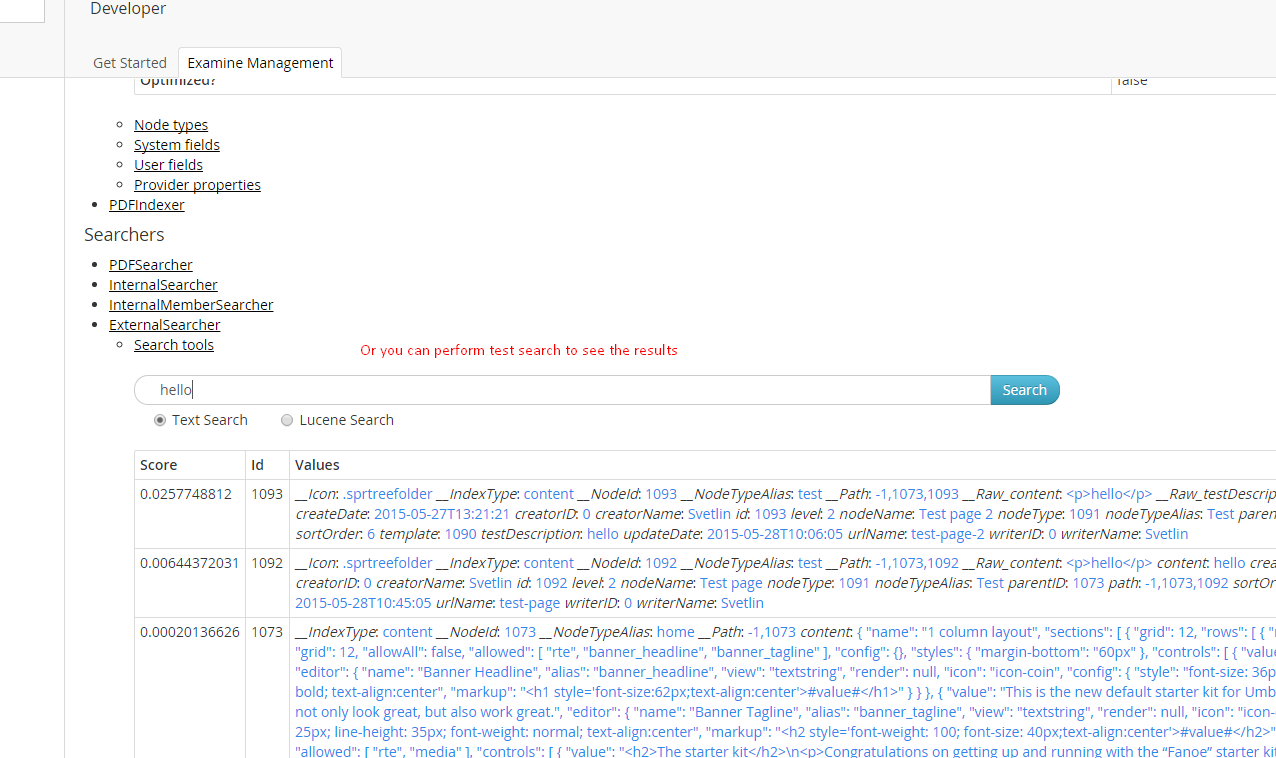


Building raw Lucene query:



* Examine interface in Umbraco back office





1. Additional modules

* Umbraco indexing for PDF files

This module will add new Indexer and Searcher to ~/Config/ExamineSettings.config called: "PDFIndexer" and "PDFSearcher"

and new Index Set to ~/Config/ExamineIndex.config called "PDFIndexSet"

After that you can search in selected pdf files, using the Examine Search API like examples below. But you need to specify your searcher:

var searcher = ExamineManager.Instance.SearchProviderCollection["PDFSearcher"];

You can install this module via NuGet – use user interface or type this command in Package manager console in Visual studio:

Install-Package UmbracoCms.UmbracoExamine.PDF

1. Conclusion and negatives

Lucene is super fast search engine and after the Umbraco Examine steps of this platform, we have in our hands very powerful tool.

As you see - working and managing all this resources is very easy, so I can not say any negative comments about Umbraco Examine.