Pankaj Gaur

Scalable Search 4 Tridion (SS4T) Framework

Contents

[**Preface** 2](#_Toc421808639)

[**Scalable Search 4 Tridion – An Introduction** 3](#_Toc421808640)

[**SS4T Installation, Setup and Configuration** 4](#_Toc421808641)

[**SETUP** 4](#_Toc421808642)

[1. CMS Setup 4](#_Toc421808643)

[2. Solr Setup 4](#_Toc421808644)

[3. Services Setup 4](#_Toc421808645)

[4. Content Delivery Setup 4](#_Toc421808646)

[**CONFIGURATION** 5](#_Toc421808647)

[1. Index Service 5](#_Toc421808648)

[2. Search Service 5](#_Toc421808649)

[**Indexing and Searching Setup Testing** 6](#_Toc421808650)

# **Preface**

This document is intended to provide information on Scalable Search 4 Tridion (SS4T) framework and its feature on a high level.

This document also illustrates the step by step process of setting up and configuring the framework.

After following this document, you should be able to understand the concept of SS4T framework and should be able to successfully setup and utilizes the SS4T framework.

# **Scalable Search 4 Tridion – An Introduction**

Scalable Search 4 Tridion (SS4T) is intended to provide integration of the SDL Tridion with a search engine. Currently, this framework is supporting the SOLR but it can easily be extended to use any other search engine.

The SOLR configured is a multicore configuration and the framework comes with one core in a multi-core setup.

A list of high level features and the capabilities of the framework is as below:

* Indexing of Content from Tridion to SOLR in a configurable way
* Configure components of what all schema can be indexed in SOLR
* Configure what all components of a particular schema can be indexed in SOLR
* Configure what all fields of a component can be indexed in SOLR
* Indexing of Metadata – Not available, but will be included in next release
* Retrieving of the indexed Content from SOLR in the form of JSON
* Retrieval criteria of the component could be – All Content indexed in SOLR, All Content based on a specific schema or multiple schema, All content based on a field criteria
* Faceted searching of the content from SOLR
* Auto-Correct/Suggestion search of content from SOLR
* Support for Pagination and Sorting

# **SS4T Installation, Setup and Configuration**

This document lists the steps to install/setup and configure the SS4T which allows following:

## **SETUP**

Below are various setup steps

### CMS Setup

1. Copy and paste the Deploy 🡪 Templating 🡪 GenericIndexing.Templating.dll which is a templating building block (TBB) to a location on your Tridion CM Server
2. Upload GenericIndexing.Templating.dll TBB to Tridion CMS – Ensure you uses /uploadpdb switch set to false and /folder switch with appropriate folder TCM URI values
3. Create a Component Template with following attributes:
   * Output Format – XML Fragment
   * Component Presentation based on this Component Template will be – Published as Dynamic Component

And in it associate following TBBs: CT\_GetCompopnentAsXML\_CS (refer step a. above), **Publish Binaries in Package**, **Link Resolver** and **Cleanup Template** *(All three are Default Template Building Blocks)*

### Solr Setup

1. Copy and paste the Deploy 🡪solr folder to a location on your machine/server
2. Start the SOLR setup
3. Ensure Solr is running fine by opening the following URL in the browser: http://localhost:8983/solr/

### Services Setup

1. Copy and paste the Deploy 🡪 Services 🡪 SolrIndexSvc on your machine/server and host it in the IIS as Solr Index Service
2. Copy and paste the Deploy 🡪 Services 🡪 SearchSvc on your machine/server and host it in the IIS as Search Service
3. Ensure both the Services hosted above can be browsed
4. Copy and paste the Deploy 🡪 Services 🡪 Configuration folder to a location on your machine/server

### Content Delivery Setup

1. Copy and paste all the JAR files from the Deploy 🡪 Content Delivery 🡪 JAR folder in the Content Delivery **/bin/lib** folder
2. Copy and paste the Deploy 🡪 Content Delivery 🡪 Config 🡪 CustomStorageConfig.xml XML file in the Content Delivery **/bin/config** folder
3. Copy and paste the Deploy 🡪 Content Delivery 🡪 Config 🡪 CustomStorageDAOBundles.xml XML file in the Content Delivery **/bin/config** folder
4. Edit the CustomStorageConfig.xml file to change the value of following nodes:
   1. ServiceEndPoint – URL of Solr Index Service ((***Ref:*** [3 Service Setup – a](#_Services_Setup)))
   2. TemplateIdToIndex – TCM Item ID of Component Template (Ref: [1. CMS Setup – c](#_CMS_Setup)) – This should only be the Item ID and not the fully resolved TCM URI.
5. Open the cd\_storage\_config.xml Storage Configuration file from the **/bin/config** folder and add following node under the Configuration 🡪 Global 🡪 Storages section:

*<StorageBindings>*

*<Bundle src="CustomStorageDAOBundles.xml"/>*

*</StorageBindings>*

## **CONFIGURATION**

Below are various configuration required for setting up the SS4T

### Index Service

* 1. Navigate to the Solr Index Service hosted in the IIS and open the Web.config file (***Ref:*** [3 Service Setup – a](#_Services_Setup))
  2. Ensure appropriate values for following keys:
     1. LoggingConfigPath - Should be set to the absolute path of Configuration\Logging.config file (***Ref:*** [3 Service Setup – d](#_Services_Setup))
     2. CONTENT\_TYPES\_DIR - Should be set to the absolute path of the SolrIndexSvc\Content\_Types folder (***Ref:*** [3 Service Setup – a](#_Services_Setup))
     3. GenericIndexingConfiguration 🡪 rootPath - Should be set to the absolute path of the Configuration folder (***Ref:*** [3 Service Setup – d](#_Services_Setup))
  3. Navigate and open for edit the file - Configuration 🡪 SearchIndexService.config. Edit for appropriate SOLR base URL (This should not be the Admin URL or the URL having core name in it but only the base URL) – For Example: <http://localhost/solr> (***Ref:*** [3 Service Setup – d](#_Services_Setup) and [2 Solr Setup - d](#_Solr_Setup))
  4. Navigate and open for edit the file - Configuration 🡪 logging.config and configure for logging (***Ref:*** [3 Service Setup – d](#_Services_Setup))

### Search Service

1. Navigate to the Search Service hosted in the IIS and open the Web.config file (***Ref:*** [3 Service Setup – b](#_Services_Setup))
2. Ensure appropriate values for following keys/attributes:
   1. The id and url attribute in the Configuration 🡪 solr 🡪 server key should be appropriately set – **id** should be set to the name of the core (which is **en** in current package) and **url** should be set to solr core url similar to [**http://localhost:8983/solr/en**](http://localhost:8983/solr/en)here en is the solr core name**.**
   2. LoggingConfigPath - Should be set to the absolute path of Configuration\Logging.config file (***Ref:*** [3 Service Setup – d](#_Services_Setup))
   3. default\_RecordSize – Set to 10, should be set to default number of record you want to fetch from SOLR. You can also define this value in the HTTP POST request while querying the SOLR, in the absence of which, this default value will be picked
   4. GenericIndexingConfiguration 🡪 rootPath - Should be set to the absolute path of the Configuration folder (***Ref:*** [3 Service Setup – d](#_Services_Setup))

## **Indexing and Searching Setup Testing**

Perform following steps to test and verify the setup:

1. Create a Schema as per below details:

**Root Element Name:** article

**Field Details:**

|  |  |  |
| --- | --- | --- |
| **XML Field Name** | **Field Type** | **Remarks** |
| title | Text | Not Mandatory |
| shortdescription | Text | Not Mandatory |
| longdescription | Text | Not Mandatory |
| smallimage | Multimedia Link | Not Mandatory; Any Multimedia Schema Association |
| largeimage | Multimedia Link | Not Mandatory; Any Multimedia Schema Association |

1. Create sample components based on this schema
2. Link this Schema with the Dynamic Component Template created in the step [1.CMS Setup – c](#_CMS_Setup).
3. Publish sample components
4. Access SOLR Admin panel and execute query - \*:\* - The published components should be available in XML format
5. Use Fidler or a similar tool to Test to test the Search Service. Use below details in Fidler Composer to retrieve for above components:

**Request type:** POST

**URL:** <SEARCH\_SVC\_URL>/GetContentFromSolr where search URL is the URL of the Search Service configured in step [3 Service Setup – b](#_Services_Setup)

**Request Headers:** content-type:application/json

**Request Body:**   
{"ServicePayload":{"ContentType":"article","SolrCore":"en"}}

1. Execute this request in Fidler and verify the results returned