

1. Stanislav B:	2
1.1 Search	3
1.1.1 Api	4
1.1.1.1 api/v1/search	5
1.1.1.2 api/v1/search/clear	6
1.1.1.3 api/v1/search/create	7
1.1.1.4 api/v1/search/delete	8
1.1.1.5 api/v1/search/drop	9
1.1.1.6 api/v1/search/dropAndCreate	10
1.1.1.7 api/v1/search/getByIds	11
1.1.1.8 api/v1/search/indexing	12
1.1.1.9 api/v1/search/info	13
1.1.1.10 api/v1/search/isExists	14
1.1.1.11 api/v1/search/isReady	15
1.1.1.12 api/v1/search/parseQuery	16
1.1.1.13 api/v1/search/state	18
1.1.1.14 api/v1/search/version	19
1.1.1.15 swagger	20
1.1.2 Configuration	21
1.1.2.1 appsettings.json	22
1.1.2.2 NLog.config	23
1.1.2.3 searchServiceManager.json	24
1.1.3 Docker	25
1.1.4 Downloads	26
1.1.5 Nuget	27
1.1.6 Query language	28

Stanislav B: .

(Stanislav B)

. : sbatyrgareev@gmail.com

Downloads

- 9 . • [Stanislav B • Nuget](#)
- 12 . • [Stanislav B Query language](#)
- 13 . • [Stanislav B Docker](#)
- 14 . • [Stanislav B swagger](#)
- 15 . • [Stanislav B NLog.config](#)
- 17 . • [Stanislav B appsettings.json](#)
- 18 . • [Stanislav B • searchServiceManager.json](#)
- 20 . • [Stanislav B Configuration](#)
- 32 . • [Stanislav B api/v1/search/drop](#)
- 33 . • [Stanislav B • api/v1/search/dropAndCreate](#)
- 35 . • [Stanislav B • api/v1/search/getByIds](#)
- 37 . • [Stanislav B api/v1/search/info](#)
- 46 . • [Stanislav B • api/v1/search/clear](#)
- 46 . • [Stanislav B • api/v1/search/delete](#)
- 46 . • [Stanislav B •](#)

Search

Api

api/v1/search

The method performs a full-text index search

GET `api/v1/search?text=one%20~2%20two&index=test&field=%2A&count=10000&offset=0`

POST `api/v1/search`

```
{
  "query": "one ~2 two",
  "startTimestamp": null,
  "endTimestamp": null,
  "startPublish": "2022-10-04T18:51:28.446Z",
  "endPublish": "2022-11-04T18:51:28.446Z",
  "field": "*",
  "count": 10000,
  "offset": 0,
  "fields": [],
  "filters": [],
  "order": 0,
  "orderField": null,
  "indexName": "test*",
  "isShort": true
}
```

Response

```
{
  "TotalHits": 1,
  "ScoreDocs": 1,
  "IsReady": true,
  "ReadyIndexes": 0,
  "hits": [
    {
      "ID": "ExternalId",
      "TIMESTAMP": 638031841443630000,
      "PUBLISHED": 638031841443630000,
      "Terms": [
        "ONE",
        "TWO",
        "ONE"
      ],
      "content": ""
    }
  ]
}
```

api/v1/search/clear

The method frees up RAM by unloading objects that have been cached

GET /api/v1/search/clear

Response

200 OK

api/v1/search/create

The method creates a new index

GET `api/v1/search/create?index=test`

Response

200 OK

api/v1/search/delete

The method deletes an array of documents by their external identifier

POST /api/v1/search/delete

```
{
  "ids": [
    "ExternalId"
  ],
  "indexName": "test"
}
```

Response

200 OK

api/v1/search/drop

The method deletes the index

DELETE api/v1/search/drop?name=test

Response

200 OK

api/v1/search/dropAndCreate

The method deletes and creates the index again

```
GET api/v1/search/dropAndCreate?index=test  
Response  
200 OK
```

api/v1/search/getByIds

Id

The method returns existing documents by external Id

POST api/v1/search/getByIds

Request

```
{
  "ids": [
    "ExternalId"
  ],
  "indexName": "test"
}
```

Response

```
{
  "ids": [
    "ExternalId"
  ],
  "TotalHits": 1,
  "ScoreDocs": 1
,
  "IsReady": true
}
```

api/v1/search/indexing

The method indexes the text

POST api/v1/search/indexing

Request

```
{
  "items": [
    {
      "id": "ExternalId",
      "fields": [
        {
          "name": "content",
          "value": "one two five seven nine"
        }
      ]
    },
    {
      "publishTime": "2022-11-04T18:42:24.363Z",
      "timestamp": "2022-11-04T18:42:24.363Z",
      "overwrite": true
    }
  ],
  "date": "2022-11-04T18:42:24.363Z",
  "indexName": "test"
}
```

Response

200 OK

api/v1/search/info

The method returns information about all indexes

GET /api/v1/search/info

Response

```
[
  {
    "Name": "TEST",
    "IsReady": true,
    "InMemoryTermCount": 2,
    "TermCount": 0,
    "IndexSize": 1,
    "LoadDocumentCount": 1,
    "UniqueCount": 0,
    "Total": 1,
    "State": 12,
    "LastUsage": "2022-11-04T18:54:39.6107501Z",
    "ErrorMessage": null,
    "ShardReady": 1,
    "TermRepositoryInfo": {
      "IsReady": true,
      "ErrorMessage": "",
      "Count": 4,
      "TreeItemCount": 0,
      "WildcardCount": 0,
      "ShardReady": 1
    }
  }
]
```

api/v1/search/isExists

The method checks whether the index exists

GET api/v1/search/isExists?index=test

Response

```
{  
  "isExists": true  
}
```

api/v1/search/isReady

The method checks whether the index is ready to work

GET `api/v1/search/isReady?index=test`

```
{  
  "isReady": true  
}
```

api/v1/search/parseQuery

The method parses the search query

GET api/v1/search/parseQuery?text=%28one%20~%20two%29%20-%28six%29

Response

```
{
  "query": {
    "Clauses": [
      {
        "Occur": 0,
        "Query": {
          "Clauses": [
            {
              "Occur": 0,
              "Query": {
                "Field": "*",
                "Distance": 2,
                "InOrder": false,
                "Left": {
                  "Field": "*",
                  "Word": "ONE",
                  "Bucket": null,
                  "BucketIds": null,
                  "Freq": 0,
                  "IsFill": false,
                  "IsProcessed": false,
                  "IsFreqCalc": false
                },
                "Right": {
                  "Field": "*",
                  "Word": "TWO",
                  "Bucket": null,
                  "BucketIds": null,
                  "Freq": 0,
                  "IsFill": false,
                  "IsProcessed": false,
                  "IsFreqCalc": false
                },
                "Freq": 0,
                "IsFill": false,
                "IsProcessed": false,
                "IsFreqCalc": false
              }
            },
            {
              "Occur": 0,
              "Query": {
                "Field": "*",
                "Word": "SIX",
                "Bucket": null,
                "BucketIds": null,
                "Freq": 0,
                "IsFill": false,
                "IsProcessed": false,
                "IsFreqCalc": false
              }
            }
          ],
          "Freq": 0,
          "IsFill": false,
          "IsProcessed": false,
          "IsFreqCalc": false
        }
      },
      {
        "Occur": 2,
        "Query": {
          "Clauses": [
            {
              "Occur": 0,
              "Query": {
                "Field": "*",
                "Word": "SIX",
                "Bucket": null,
                "BucketIds": null,
                "Freq": 0,
                "IsFill": false,
                "IsProcessed": false,
                "IsFreqCalc": false
              }
            }
          ],
          "Freq": 0,
          "IsFill": false,
          "IsProcessed": false,
          "IsFreqCalc": false
        }
      }
    ]
  }
}
```



```
        "IsFreqCalc": false
      }
    }
  ],
  "Freq": 0,
  "IsFill": false,
  "IsProcessed": false,
  "IsFreqCalc": false
}
],
"Freq": 0,
"IsFill": false,
"IsProcessed": false,
"IsFreqCalc": false
}
}
Response headers
```

api/v1/search/state

The method returns information about the state of the index

GET `api/v1/search/state?index=test`

Response

```
{
  "name": "TEST",
  "isReady": true,
  "inMemoryTermCount": 0,
  "termCount": 0,
  "indexSize": 0,
  "loadDocumentCount": 0,
  "uniqueCount": 0,
  "total": 0,
  "state": 0,
  "lastUsage": "2022-11-04T18:32:05.3343396Z",
  "errorMessage": null,
  "shardReady": 0,
  "termRepositoryInfo": {
    "isReady": true,
    "errorMessage": null,
    "count": 0,
    "treeItemCount": 0,
    "wildCardCount": 0,
    "shardReady": 0
  }
}
```

api/v1/search/version

Method return information about api version

GET /api/v1/search/version

Response

```
{  
  "version": "2.0.14.0"  
}
```

swagger

<http://localhost:2664/swagger/index.html>

Configuration

appsettings.json

2664

The default port is 2664

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*",
  "Bsa.Search.Api": {
    "Host": "http://+:2664"
  },
  "profiles": {
    "bsa_search_api": {
      "commandName": "BsaSearch",
      "dotnetRunMessages": "true",
      "launchBrowser": true,
      "launchUrl": "swagger",
      "applicationUrl": "http://+:2664",
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    }
  }
}
```

NLog.config

```
<?xml version="1.0" encoding="utf-8" ?>
<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <targets>
    <target xsi:type="ColoredConsole"
            name="ColoredConsole-Trace"
            layout="${longdate} ${uppercase:${level}} ${logger} ${message}"
            useDefaultRowHighlightingRules="true" />

    <target xsi:type="File"
            name="FlatFile-Trace"
            layout="${longdate} ${uppercase:${level}} ${logger} ${message}"
            fileName="/Logs/Bsa.Search.Api.Host/${shortdate}-Trace.log"
            keepFileOpen="false"
            encoding="UTF-8" />

    <target xsi:type="File"
            name="FlatFile-Error"
            layout="${longdate} ${uppercase:${level}} ${logger} ${message} ${exception:
format=tostring}"
            fileName="/Logs/Bsa.Search.Api.Host/${shortdate}-Errors.log"
            keepFileOpen="false"
            encoding="UTF-8" />

  </targets>

  <rules>
    <logger name="*" minlevel="Info" writeTo="ColoredConsole-Trace" />
    <logger name="*" minlevel="Info" writeTo="FlatFile-Trace" />
    <logger name="*" minlevel="Error" writeTo="FlatFile-Error" />
  </rules>
</nlog>
```

searchServiceManager.json

Search engine configuration file

```
{
  "ClearMemoryInterval": "00:01:00",
  "CloseIndexInterval": "00:30:00",
  "IndexLiveInterval": 60,
  "UseLocalTermRepository": "true",
  "PercentMemoryUsage": 80,
  "ShardCount": 16,
  "AvailablePhysicalMemory": 128000,
  "IndexFolder": "Data\\searchindex"
}
```

where:

ClearMemoryInterval -

ClearMemoryInterval - the interval for clearing the cache of search indexes

CloseIndexInterval - IndexLiveInterval

CloseIndexInterval - The index lifetime check interval is associated with IndexLiveInterval

IndexLiveInterval - ,

IndexLiveInterval - the lifetime of the index in minutes, if the index is not used in the specified time interval, it is closed

UseLocalTermRepository - ,

UseLocalTermRepository - use a local repository to store terms, otherwise you need to transfer an external repository

PercentMemoryUsage - AvailablePhysicalMemory

PercentMemoryUsage - percentage of memory used from the maximum possible AvailablePhysicalMemory as a percentage

ShardCount - , 2 1 16

ShardCount - the number of shards in the index, currently there are 2 possible values of 1 or 16

AvailablePhysicalMemory - , ()

AvailablePhysicalMemory - the maximum available amount of RAM on the computer, after which the process of clearing memory (closing indexes) will be initiated

IndexFolder -

IndexFolder - location of indexes

Docker

docker pull bsacore/bsasearchapihost

<https://hub.docker.com/r/bsacore/bsasearchapihost>

Downloads

<https://bsasearch.org/Bsa.Search.2.0.14.zip>

<https://bsasearch.org/Ru-WordForms.zip> - ,

<https://hub.docker.com/r/bsacore/bsasearchapihost>

Nuget

<https://www.nuget.org/packages/Bsa.Search.Core>

Query language

Key words

#	Operator	Name	Description
1	()	Priority of operations	Required for grouping expressions. (<i>Investment* money</i>)
2	' '	Exact phrase (without word forms)	A set of documents containing the specified sequence of words "as is", i.e. without taking into account word forms. ! The use of other operators is not possible in this case. <i>'big house' => big house</i>
3	" "	Phrase	A set of documents containing the specified sequence of words, taking into account word forms. <i>"person account" => person account, people's accounts. If the word forms are in the *.terms file</i>
4	*	Wildcard	<i>invest* => investment, investing, investigation etc.</i> <i>"person* account*" => personal account, person account etc.</i> The operator is applicable to words and phrases, must go at the end of the word without a space.
5	&	AND	<i>'Elon Musk' & CEO</i> <i>Musk* & "Boring Company"</i> <i>"Elon* Musk*" & "Boring Company*"</i>
6		OR	<i>Musk* "Elon* Musk"</i> <i>First* "Second* Other*"</i> <i>"Elon*" 'Boring Company'</i>
7	-	NOT	<i>-spam</i> <i>-(spam* negativ*)</i>
8	~n	Search for words at a distance	Search for words, phrases, or groups at a given distance; n is a positive integer in the range (1;N] within which the search is performed. <i>"Dragon" ~100 "Elon* Musk"</i> <i>(Elon*)Mask CEO CTO) ~3 ("founder* SpaceX" "Boring Company*")</i> <i>'Elon' ~3 'founder SpaceX'</i>
9	/n	Search for words at a distance in the forward direction	Search for words, phrases, or groups at a given distance forward; n is a positive integer in the range (1;N] within which the search is performed. <i>Elon* /5 Musk</i> <i>"founder* SpaceX" /5 Elon*</i>
10	\n	Search for words at a distance in the backward direction	Search for words, phrases, or groups at a given distance forward; n is a positive integer in the range (1;N] within which the search is performed. <i>Elon* \3 Musk</i> <i>Musk* \15 SpaceX</i>
11	?	Any character	<i>"(E ?n*) ~3 (M?sk)"; //=>(E an Elon Elen Elons...) ~3 (Mask Mus Mesk....)</i>

Word forms

if you need to add your own word forms, then add a file with extension '*.terms' Separate each word form with a newline sign

Example:

dry dried dried

draw drew drawn