

# Textual Index

- [Textual Index](#)
- [Operation](#)
- [Lucene - Query Parser Syntax](#)

# Textual Index

## Introduction

A textual index is a data structure used in database systems to facilitate efficient search and retrieval of text-based information. It is designed to handle large volumes of textual data and provide quick access to relevant documents or records based on specified search criteria.

When a search query is performed on a database with a textual index, the index is queried to identify relevant documents or records that match the search terms. The index provides information about the location and relevance of the documents, which enables the database system to retrieve and present the results in a timely manner.

Textual indexes play a crucial role in enabling efficient search and retrieval of textual information in databases, making them an essential component in applications that handle large volumes of textual data, such as search engines, content management systems, and document repositories.

Soffid incorporates the textual index from version 3.5 using [the Apache Lucene library](#).

## Index configuration

Soffid allows you to configure the objects you want to use in the textual index. To do this, you must select the proper object from the metadata page and enable the option "Use textual index". Once you enable this option, the textual index will be applied to the attributes of this object that have been included in the quick search.

Notice, from the user interface, it is not interpreted as a Lucene expression.

## Example

1. Enable the "Use textual index" on the User object and save the changes.

**soffid** Search

Main Menu > Administration > Configuration > Global Settings > Metadata 9 / 12

Object type : com.soffid.iam.api.User

Description : Builtin user object

Use textual index : ☒ Yes ☐ No ☐ III

Code	Label	Data type	Built-in type
Filter	Filter	Filter	Filter
<input type="checkbox"/> __COMMONATTRIBUTES__	Common attributes	Separator	Yes
<input type="checkbox"/> userName	User name	String	Yes
<input type="checkbox"/> firstName	First name	String	Yes
<input type="checkbox"/> lastName	Last Name	String	Yes
<input type="checkbox"/> middleName	Middle name	String	Yes

## 2. Check the attributes included in the quick search.

**soffid** Attribute metadata

Main Menu > Administration > Configuration > Global Settings > Metadata 2 / 31

Object type : com.soffid.iam.api.User

Description : Builtin user object

Use textual index : ☒ Yes ☐ No ☐ III

Code : userName

Label : User name

Data type : String

Letter case : Keep as entered by the user

User hint : User hint

Description : Description

Required : ☒ Yes ☐ No

Include in quick search : ☒ Yes ☐ No

Prevent duplicated values : ☐ Yes ☒ No

Multiple values : ☐ Yes ☒ No

Maximum number of rows to display : Maximum number of rows to display

# How does the user interface search work?

Once you have configured the textual index for a specific object, Soffid will apply it when you use Quick Search on this object.

## Example 1

**1.** If you search for users using the text "frankin", then Soffid will display all the users whose userName, firstName, lastName, or middleName match, to some degree, with the typed text following the textual index rules.

frankin

Quick [Basic](#) [Advanced](#)

<input type="checkbox"/>	▼ User name	↕ First name	↕ Last Name	↕ Primary group	↕ Enabled	↕ Manager
<input type="checkbox"/>	aretha	Aretha	Franklin	world	Yes	
<input type="checkbox"/>	frank	Frank	Sinatra	Music	Yes	
<input type="checkbox"/>	frankrudolph	Frank Rudolph	Olson	scientist	Yes	
<input type="checkbox"/>	franks	Frank	Sherwood	scientist	Yes	
<input type="checkbox"/>	rfranklin	Rosalind	Franklin	scientist	Yes	

Displayed rows: 5

2. If you include the attribute manager in the quick search:

soffid

Main Menu > Administration > Resources > Users

Attribute metadata

Attribute metadata 15 / 32

Object type :

Description :

Use textual index :

Code

Filter

☐ rid

☐ fullName

☐ birthDate

☐ picture

☐ \_\_ORGANIZATION

☐ userType

Code :

Label :

Data type :

User hint :

Description :

Required :

Include in quick search :

Prevent duplicated values :

Multiple values :

Maximum number of rows to display :

manager

Manager

User

User hint

Description

☐ ☐ No

☒ ☐

☐ ☐ No

☐ ☐ No

3. And search for "frankin", then Soffid will display all the users whose userName, firstName, lastName, middleName, or manager match with the typed text following the textual index rules.

soffid

Main Menu > Administration > Resources > Users

frankin

Quick [Basic](#) [Advanced](#)

<input type="checkbox"/>	▼ User name	↕ First name	↕ Last Name	↕ Primary group	↕ Enabled	↕ Manager
<input type="checkbox"/>	aretha	Aretha	Franklin	world	Yes	
<input type="checkbox"/>	frank	Frank	Sinatra	Music	Yes	
<input type="checkbox"/>	frankrudolph	Frank Rudolph	Olson	scientist	Yes	
<input type="checkbox"/>	franks	Frank	Sherwood	scientist	Yes	
<input type="checkbox"/>	hlamarr	Hedy	Lamarr	scientist	Yes	franks
<input type="checkbox"/>	inewton	Isaac	Newton	scientist	Yes	frank
<input type="checkbox"/>	janis	Janis	Joplin	Music	Yes	frank
<input type="checkbox"/>	rfranklin	Rosalind	Franklin	scientist	Yes	

Displayed rows: 8

## Example 2

1. If you search for users using the text "manager:frank" Soffid will display all users whose manager matches the text "frank".

soffid

Search

?

⚙

[Main Menu](#) > [Administration](#) > [Resources](#) > [Users](#)

≡

🔍

[Quick](#)
[Basic](#)
[Advanced](#)

<input type="checkbox"/>	▼ User name	↕ Full name	↕ Primary group	↕ Enabled	↕ First name	↕ Last Name	↕ Manager
<input type="checkbox"/>	hlamarr	Hedy Lamarr	scientist	Yes	Hedy	Lamarr	franks
<input type="checkbox"/>	inewton	Isaac Newton	scientist	Yes	Isaac	Newton	frank
<input type="checkbox"/>	janis	Janis Joplin	Music	Yes	Janis	Joplin	frank
<input type="checkbox"/>	ntesla	Nikola Tesla	Support IT	Yes	Nikola	Tesla	frankrudolph

Displayed rows: 4

Notice the difference by searching "manager:frank?":

soffid

Search

?

⚙

[Main Menu](#) > [Administration](#) > [Resources](#) > [Users](#)

≡

🔍

[Quick](#)
[Basic](#)
[Advanced](#)

<input type="checkbox"/>	▼ User name	↕ Full name	↕ Primary group	↕ Enabled	↕ First name	↕ Last Name	↕ Manager
<input type="checkbox"/>	hlamarr	Hedy Lamarr	scientist	Yes	Hedy	Lamarr	franks
<input type="checkbox"/>	ntesla	Nikola Tesla	Support IT	Yes	Nikola	Tesla	frankrudolph

Displayed rows: 2

And by searching "manager:frank\*":

soffid

Search

?

⚙

[Main Menu](#) > [Administration](#) > [Resources](#) > [Users](#)

≡

🔍

[Quick](#)
[Basic](#)
[Advanced](#)

<input type="checkbox"/>	▼ User name	↕ Full name	↕ Primary group	↕ Enabled	↕ First name	↕ Last Name	↕ Manager
<input type="checkbox"/>	hlamarr	Hedy Lamarr	scientist	Yes	Hedy	Lamarr	franks
<input type="checkbox"/>	inewton	Isaac Newton	scientist	Yes	Isaac	Newton	frank
<input type="checkbox"/>	janis	Janis Joplin	Music	Yes	Janis	Joplin	frank
<input type="checkbox"/>	ntesla	Nikola Tesla	Support IT	Yes	Nikola	Tesla	frankrudolph

Displayed rows: 4

And also by searching "manager:fr\*"

soffid

Search

?

⚙

[Main Menu](#) > [Administration](#) > [Resources](#) > [Users](#)

≡

🔍

[Quick](#)
[Basic](#)
[Advanced](#)

<input type="checkbox"/>	▼ User name	↕ Full name	↕ Primary group	↕ Enabled	↕ First name	↕ Last Name	↕ Manager
<input type="checkbox"/>	etaylor	Elizabeth Taylor	Movie Stars	Yes	Elizabeth	Taylor	freddie
<input type="checkbox"/>	hlamarr	Hedy Lamarr	scientist	Yes	Hedy	Lamarr	franks
<input type="checkbox"/>	inewton	Isaac Newton	scientist	Yes	Isaac	Newton	frank
<input type="checkbox"/>	janis	Janis Joplin	Music	Yes	Janis	Joplin	frank
<input type="checkbox"/>	ntesla	Nikola Tesla	Support IT	Yes	Nikola	Tesla	frankrudolph

Displayed rows: 5

### Example 3

**1.** If you search for users using the text "userName:frank\*" Soffid will display all users whose user name matches the text "frank" followed by any other text.

soffid

Search

?

⚙

[Main Menu](#) > [Administration](#) > [Resources](#) > [Users](#)

Quick Basic Advanced

<input type="checkbox"/>	▼ User name	↕ Full name	↕ Primary group	↕ Enabled	↕ First name	↕ Last Name	↕ Manager
<input type="checkbox"/>	frank	Frank Sinatra	Music	Yes	Frank	Sinatra	
<input type="checkbox"/>	frankrudolph	Frank Rudolph Olson	scientist	Yes	Frank Rudolph	Olson	
<input type="checkbox"/>	franks	Frank Sherwood	scientist	Yes	Frank	Sherwood	

Displayed rows: 3

Notice the difference by searching the text "userName:frank?":

soffid

Search

?

⚙

[Main Menu](#) > [Administration](#) > [Resources](#) > [Users](#)

Quick Basic Advanced

<input type="checkbox"/>	▼ User name	↕ Full name	↕ Primary group	↕ Enabled	↕ First name	↕ Last Name	↕ Manager
<input type="checkbox"/>	frankrudolph	Frank Rudolph Olson	scientist	Yes	Frank Rudolph	Olson	
<input type="checkbox"/>	franks	Frank Sherwood	scientist	Yes	Frank	Sherwood	

Displayed rows: 2

## Example 4

- If you search for users using the text "frank" plus the wildcard "?", Soffid will display all users whose userName, firstName, lastName, middleName, or manager match the typed text as long as it has variation in the characters where the wildcard has been used.

soffid

Search

?

⚙

[Main Menu](#) > [Administration](#) > [Resources](#) > [Users](#)

Quick Basic Advanced

<input type="checkbox"/>	▼ User name	↕ First name	↕ Last Name	↕ Primary group	↕ Enabled	↕ Manager
<input type="checkbox"/>	aretha	Aretha	Franklin	world	Yes	
<input type="checkbox"/>	frankrudolph	Frank Rudolph	Olson	scientist	Yes	
<input type="checkbox"/>	franks	Frank	Sherwood	scientist	Yes	
<input type="checkbox"/>	hlamarr	Hedy	Lamarr	scientist	Yes	franks
<input type="checkbox"/>	ntesla	Nikola	Tesla	Support IT	Yes	frankrudolph
<input type="checkbox"/>	rfranklin	Rosalind	Franklin	scientist	Yes	

Displayed rows: 6

Notice the difference by searching "fran?"

soffid

Search

?

⚙

[Main Menu](#) > [Administration](#) > [Resources](#) > [Users](#)

Quick Basic Advanced

<input type="checkbox"/>	▼ User name	↕ First name	↕ Last Name	↕ Primary group	↕ Enabled	↕ Manager
<input type="checkbox"/>	aretha	Aretha	Franklin	world	Yes	
<input type="checkbox"/>	frank	Frank	Sinatra	Music	Yes	
<input type="checkbox"/>	frankrudolph	Frank Rudolph	Olson	scientist	Yes	
<input type="checkbox"/>	franks	Frank	Sherwood	scientist	Yes	
<input type="checkbox"/>	hlamarr	Hedy	Lamarr	scientist	Yes	franks
<input type="checkbox"/>	inewton	Isaac	Newton	scientist	Yes	frank
<input type="checkbox"/>	janis	Janis	Joplin	Music	Yes	frank
<input type="checkbox"/>	ntesla	Nikola	Tesla	Support IT	Yes	frankrudolph
<input type="checkbox"/>	rfranklin	Rosalind	Franklin	scientist	Yes	

Displayed rows: 9

# How does the SCIM interface search work?

1. First of all, you must install the SCIM addon in Soffid.

For more information, you can visit [the How to install SCIM in Soffid? page](#).

2. Then, you can use any REST client to test and consume our SCIM REST web service.

For more information, you can visit [the Testing tool page](#).

3. Finally, you can start to use the SCIM interface search by using Lucene syntaxis

## Lucene syntaxis

Please browse the standard specifications in this link:

<https://bookstack.soffid.com/books/soffid-3-reference-guide/page/lucene-query-parser-syntax>

## Term Modifiers

Lucene supports modifying query terms to provide a wide range of search options. Here are the most common ones:

<b>Wildcard Searches</b>	To perform a single character wildcard search use the "?" symbol. To perform a multiple character wildcard search use the "*" symbol.
<b>Regular Expression Searches</b>	Lucene supports regular expression searches matching a pattern between forward slashes "/"

<b>Fuzzy Searches</b>	<p>To do a fuzzy search use the tilde, "~", symbol at the end of a Single word Term</p> <p><u>Soffid Console &lt;= 3.4 version</u></p> <p>~0.8: stricter search</p> <p>~0.1: more lax search</p> <p><u>Soffid Console &gt; 3.4 version</u></p> <p>An additional (optional) parameter can specify the maximum number of edits allowed. The value is between 0 and 2.</p>
<b>Range Searches</b>	<p>Range Queries allow one to match documents whose field(s) values are between the lower and upper bound specified by the Range Query</p>
<b>Boosting a Term</b>	<p>To boost a term use the caret, "^", symbol with a boost factor (a number) at the end of the term you are searching. The higher the boost factor, the more relevant the term will be.</p>

## Boolean Operators

<b>OR</b>	The OR operator links two terms and finds a matching document if either of the terms exist in a document. This is equivalent to a union using sets
<b>AND</b>	The AND operator matches documents where both terms exist anywhere in the text of a single document. This is equivalent to an intersection using sets.
<b>+</b>	The "+" or required operator requires that the term after the "+" symbol exist somewhere in a the field of a single document.
<b>NOT</b>	The NOT operator excludes documents that contain the term after NOT. This is equivalent to a difference using sets.
<b>-</b>	The "-" or prohibit operator excludes documents that contain the term after the "-" symbol.

## Escaping Special Characters

Lucene supports escaping special characters that are part of the query syntax.

The current list of special characters are + - & || ! ( ) { } [ ] ^ " ~ \* ? : \ /

## Examples

### Example 1

**1.** Use the wildcard search

## 1.1.\*

### Request

```
GET http://<domain>/webservice/scim2/v1/User?textFilter=fran*
```

### Response 200 OK

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 4,
  "startIndex": 1,
  "Resources": [
    {
      "lastName": "Franklin",
      "createdByUser": "ActiveDirectory",
      "fullName": "Rosalind Franklin",
      "active": true,
      "userName": "rfranklin",
      "mailAlias": "",
      "firstName": "Rosalind",
      "createdDate": "2023-08-08 14:26:14",
      "multiSession": true,
      "meta": {
        "location": "http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/2862",
        "links": {
          "roleAccounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+'rfranklin'+and+enabled+eq+true",
          "groupUsers":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+'rfranklin'+and+disabled+eq+false",
          "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.userName+eq+'rfranklin'",
          "issues":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+'rfranklin'"
        }
      }
    }
  ]
}
```

```

        "effectiveGrants":
"http: //soffid. 35x. lab: 8089/soffid/webservice/scim2/v1/User /2862/effectiveGrants"
        },
        "resourceType": "User"
    },
    "modifiedByUser": "ActiveDirectory",
    "schemas": [
        "urn: soffid: com. soffid. iam. api. User"
    ],
    "modifiedDate": "2023-08-08 14:26:14",
    "attributes": {},
    "id": 2862,
    "userType": "I",
    "primaryGroupDescription": "scientist",
    "primaryGroup": "scientist"
},
{
    "lastName": "Franklin",
    "createdByUser": "ActiveDirectory",
    "fullName": "Aretha Franklin",
    "active": true,
    "userName": "aretha",
    "mailAlias": "",
    "firstName": "Aretha",
    "createdDate": "2023-09-06 13:12:54",
    "multiSession": true,
    "meta": {
        "location":
"http: //soffid. 35x. lab: 8089/soffid/webservice/scim2/v1/User /276397",
        "links": {
            "roleAccounts":
"http: //soffid. 35x. lab: 8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+' aretha'
+and+enabled+eq+true",
            "groupUsers":
"http: //soffid. 35x. lab: 8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+' aretha' +and+d
isabled+eq+false",
            "accounts":
"http: //soffid. 35x. lab: 8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users. user
. userName+eq+' aretha' ",
            "issues":

```

```

"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+' aretha' "
,
    "effectiveGrants":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/276397/effectiveGrants"
    },
    "resourceType": "User"
},
"modifiedByUser": "ActiveDirectory",
"schemas": [
    "urn:soffid:com.soffid.iam.api.User"
],
"modifiedDate": "2023-09-06 13:12:54",
"attributes": {},
"id": 276397,
"userType": "I",
"primaryGroupDescription": "World",
"primaryGroup": "world"
},
{
    "lastName": "Sinatra",
    "createdByUser": "ActiveDirectory",
    "fullName": "Frank Sinatra",
    "active": true,
    "userName": "frank",
    "mailAlias": "",
    "firstName": "Frank",
    "createdDate": "2023-09-06 13:12:54",
    "multiSession": true,
    "meta": {
        "location":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/276435",
        "links": {
            "roleAccounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+' frank' +
and+enabled+eq+true",
            "groupUsers":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+' frank' +and+di
sabled+eq+false",
            "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user

```

```

    .userName+eq+' frank' ",
        "issues":
"http: //soffid. 35x. lab: 8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+' frank' ",
        "effectiveGrants":
"http: //soffid. 35x. lab: 8089/soffid/webservice/scim2/v1/User /276435/effectiveGrants"
    },
    "resourceType": "User"
},
"modifiedByUser": "ActiveDirectory",
"schemas": [
    "urn: soffid: com. soffid. iam. api. User"
],
"modifiedDate": "2023-09-06 13:12:55",
"attributes": {},
"id": 276435,
"userType": "I",
"primaryGroupDescription": "Music",
"primaryGroup": "Music"
},
{
    "lastName": "Sherwood",
    "createdByUser": "pgarcia",
    "fullName": "Frank Sherwood",
    "active": true,
    "userName": "franks",
    "mailAlias": "",
    "firstName": "Frank",
    "createdDate": "2023-10-05 15:32:40",
    "multiSession": false,
    "meta": {
        "location":
"http: //soffid. 35x. lab: 8089/soffid/webservice/scim2/v1/User /432644",
        "links": {
            "roleAccounts":
"http: //soffid. 35x. lab: 8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+' franks'
+and+enabled+eq+true",
            "groupUsers":
"http: //soffid. 35x. lab: 8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+' franks' +and+d
isabled+eq+false",
            "accounts":

```

```

"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user
.userName+eq+'franks' ",
    "issues":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+'franks' "
,
    "effectiveGrants":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/432644/effectiveGrants"
    },
    "resourceType": "User"
  },
  "modifiedByUser": "pgarcia",
  "schemas": [
    "urn:soffid:com.soffid.iam.api.User"
  ],
  "modifiedDate": "2023-10-05 15:32:40",
  "attributes": {},
  "id": 432644,
  "userType": "I",
  "primaryGroupDescription": "scientist",
  "primaryGroup": "scientist"
}
]
}

```

## 1.2. ?

### Request

```
http://<domain>/webservice/scim2/v1/User?textFilter=fran?
```

### Response 200 OK

```

{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 2,
  "startIndex": 1,
  "Resources": [
    {
      "lastName": "Sinatra",

```

```
    "createdByUser": "ActiveDirectory",
    "fullName": "Frank Sinatra",
    "active": true,
    "userName": "frank",
    "mailAlias": "",
    "firstName": "Frank",
    "createdDate": "2023-09-06 13:12:54",
    "multiSession": true,
    "meta": {
      "location":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/276435",
      "links": {
        "roleAccounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+'frank'+and+enabled+eq+true",
        "groupUsers":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+'frank'+and+disabled+eq+false",
        "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+eq+'frank'",
        "issues":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+'frank'",
        "effectiveGrants":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/276435/effectiveGrants"
      },
      "resourceType": "User"
    },
    "modifiedByUser": "ActiveDirectory",
    "schemas": [
      "urn:soffid:com.soffid.iam.api.User"
    ],
    "modifiedDate": "2023-09-06 13:12:55",
    "attributes": {},
    "id": 276435,
    "userType": "I",
    "primaryGroupDescription": "Music",
    "primaryGroup": "Music"
  },
  {
```

```
    "lastName": "Sherwood",
    "createdByUser": "pgarcia",
    "fullName": "Frank Sherwood",
    "active": true,
    "userName": "franks",
    "mailAlias": "",
    "firstName": "Frank",
    "createdDate": "2023-10-05 15:32:40",
    "multiSession": false,
    "meta": {
      "location":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/432644",
      "links": {
        "roleAccounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+'franks'+and+enabled+eq+true",
        "groupUsers":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+'franks'+and+disabled+eq+false",
        "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+eq+'franks'",
        "issues":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+'franks'",
        "effectiveGrants":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/432644/effectiveGrants"
      },
      "resourceType": "User"
    },
    "modifiedByUser": "pgarcia",
    "schemas": [
      "urn:soffid:com.soffid.iam.api.User"
    ],
    "modifiedDate": "2023-10-05 15:32:40",
    "attributes": {},
    "id": 432644,
    "userType": "I",
    "primaryGroupDescription": "scientist",
    "primaryGroup": "scientist"
```

```
}
]
}
```

## Example 2

### 1. Use the wildcard search in a specific attribute

#### Request

```
GET http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User?textFilter=userName: frank
```

#### Response 200 OK

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 1,
  "startIndex": 1,
  "Resources": [
    {
      "lastName": "Sinatra",
      "profileServer": "Void host",
      "createdByUser": "admin",
      "fullName": "Frankaaa Sinatra",
      "active": true,
      "userName": "frank",
      "mailAlias": "",
      "mailServer": "Void host",
      "firstName": "Frankaaa",
      "emailAddress": "pgarcia@soffid.com",
      "mailDomain": "soffid.com",
      "createdDate": "2023-06-02 07:41:47",
      "multiSession": false,
      "meta": {
        "location": "http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/3910",
        "links": {
          "roleAccounts":
            "http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+' frank' +
            and+enabled+eq+true",
```

```
"groupUsers":
"http://soffid.35x.lab:8089/soffid/web-service/scim2/v1/GroupUser?filter=user+eq+'frank'+and+dissabled+eq+false",
    "accounts":
"http://soffid.35x.lab:8089/soffid/web-service/scim2/v1/Account?filter=type+eq+U+and+users.userName+eq+'frank'",
    "effectiveGrants":
"http://soffid.35x.lab:8089/soffid/web-service/scim2/v1/User/3910/effectiveGrants"
},
    "resourceType": "User"
},
    "modifiedByUser": "admin",
    "schemas": [
        "urn:soffid:com.soffid.iam.api.User"
    ],
    "modifiedDate": "2023-06-02 07:41:47",
    "attributes": {
        "picture":
"/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAAoHCBYWFRgVFRYYGRgaGhkYGhgaGBoYGHoYHBkaGhgZGhgcIS4lHB4rIRgYJjgmKy8xNTU1GiQ7QDs0Py40NTEBDAwMBGYGEAYGEDEdFh0xMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMf/AABEIAKgBKwMBIgaCEQEDEQH/xAAcAAABBQEBAQAAAAAAAAAADAQIEBQYABWj/xABFEAACAQIDBAECAQFAgcbAAABAgADEQQSIQUxQVEGEyJhcYGRmkKhsQcjUnKSwdHWfGKCoIQzY7LhU3NDZIOT08Lxf/EABQBAQA
AAAAAAAAAAAAAAAAAAD/xAAUEQEAAAAAAAAAAAAAAAAAA/9oADAMBAAIRAxEAPwDeqIVZWecBAQQiCIBHqIDgItogjwICWiqiTo4CAqiZn6ScNnwDn7DI/wAcP/3TUKJX9JcnNwld0dnJ5qMw+UD5zcQYh6qG+gPpGrQblAdThBHUsK3MfEw4wR5n5QG4dtDt8t8JU3DT0/wCZBPYVRvI9ZMpqqg97ytXgTi44n8zz3D9+k4G+4ee79iR/4Pf3C/iY2rtIAakD0gT6WH0lz4/v97oZEPL7RzHu5/nM/U2wg3Fm+Eh1NsN7oA+MDWfxir7Kj975Fx00re04Xuvb+0a+kyVTadtvc+ANh6CANQWNFW2wg3ZnP4R6nX4SFV2y59kKv9x9Tp8JUXMab3gS62MZvaYnx0npI7Vo0oeAJiVKLAXINocl4lnjAjzixsdPnAUNEZpY7H2FiMSbUabM0LnsoPFzp6XM9G60/R/SpEPiCKrjULb6tT906ue86d0DLdEuhz4krVqhkob+Iao0ScLP2vTmPWqNFUIihVUBVUCwAG4AQw3RpEBheYVhjGkQA0sCZJYQbLAj00ZeFqLBQNAqwSKphFGMRwSEAhVWAFUSCTobqSDwHEmUmI6YYJCQapJH2Uc/G0r/pH2y1FEPLCbSw5gaBfz9J4/isWzHfaB61ivPKwqexTrP32RB8Wv8ACU2J+lVrkU8MgHAUVLf2qo+c8xZoL4G1xn0kY5/ZdKY/00F/V80PMX0jxNX/ADMTVYHeM5VfwYfCUmaITAmCoI7r5BzTrmBYDFGDav3yJaKtNjwgG0JPjGHFNFXckYTR2cDqxPgIEAI2PExERmPZBPgCZD0ciG0iE+MtKeznbn2QQDwtv/esDLjBPxGX/S0TB8zfwm2w/RoseleXOH6PUltm3wMBhtiM5si+usu8N0KdvanvKwm/wAFhUQjIl7d36y1C0fcC+JEDzE9Csu8mfodF1G+3pN9XRuQ+cjFLDUQMvh+j6BtRbyhdq7DTJu56j9JattvvFxL9gg3/AHXgeYYjZ6qWPLlu8ZsuhFrZDPsGIemHcsW7faUZTYWTdfxvKNkzswtcA30F766D4fGekbCw4TD01At2Qbd51J8TeBNRQAABYDCBuHlHho2daFeJmgYy0matNGs0YzRmeA8tBu0VngWeAjtBzo4vB5oGlQwimaVo4PAkq0KryIHjg8CJOkt2DSxtLq6hZWGqOp7SHjpuIPEGeN9IuhWKwplMhqU+FRAWfv5l3p56d89yDxy1IHZNEN0BtTozg8QSalBMx3ul0bzKW5zM4z6L8M2t0tVp9zZXHyB+MDyOJPRqv0VVPCxKH71NL+TGQNofRzWoI9Z61Iog1Az5iScqgXHMiBjUp3498lrhlG+DanlpWlmav0Q/y/L/APIEXIL6CEwnynDfJ2BVScpNoDMNgYiwB85ZJsuoTIU08QZabMwig38949RNKmFDLRyMxgKS3CurKeF++c02GwAt2XBHDSSKOFKQLi45b/Cd1AGqi3h+kB38Jbe/oJJpBFgM+N0y8z6RmGIBvx5wDde5PZB8bwkoIIu03kJyMOJnVsQAL6Ad/OAKpSHf6yvXNhuvFq7U1IWl++RhXz3vpAdm3yJjn7DW0IBsYdjy0m6qsShyt90/KBUb0w/Yvt9xmJ14k6fAes3
```

+DZcqoHViqqDYi+gHDhMLhELhETQMoGhsQNM5PLfaW+MxaU6iJTUllIuw9lRbUM3EnlA1BE6DLxM8B5Ea0b1k7NAY4g4Ux  
pEATGBYw7iDKwAxmWHIEbpAv1hAIJYQGA4R4jBHKYBBFiAziYCGKI0mKIDxKHp4p/gnt9unfwzj87S+WVnSxL4Kv3Jn/Aw  
b8oHi+LpAtpwCmTqVH6pL8j/uMFTp7yfd0AEYpCiI0BRfjrAqkoXa1o9sMyG/wCWhpCqLax+P2oEAAAzHXUc0EBuG2w9M  
3yg25f8S2wnTncwDIV4XHw0mTxWiE/bsugPAWwz7+6Ds9sXW4G/mLgEEjkQQbwPYdnbdp0tlYa+nfLBmVhpuni2G2nk5ib  
jox0jD9g+1A090oBmEitj0Q9owi4RzrKDaxdWsQFi0NtLwJe1Nt1stqKgXF8zfvtjKHE9o3r4nX71hfmBI2JNUkK1xfQKD  
qSeQ4eJg2Z8Ndy6U70EuEzszWubs29R690C1wGKw6nLnDhd2jL1KiMNDfvvYD0kKjgKuJp02alXZGKuLSmACAAVNNxYrCh  
iN8yiYoo/1QqBNzI3a6sg20f3l+UDb1TcaayG99YLDVibZuUk110vAhbBpXJHHMU8r3NvIXLxWpA0VAFrgj9+szmA26uHx  
LoVuxF8xPshgCQB4W9Zf5y75h7639bpf+4QLmn7I8B8opESLAQRSZxERYAW84xVE5lgBYxpjmSMcQBuY08c0HaBqFEdaIo  
jhA60UCdaKIC3nTohMDoojLxYBVMbjSP1lGpT+2jp+JSB0WSKZgeB1Kp00ul7jkbk/vwlli6oegjjgSvkDp8CI/pbs/qcX  
Wp7lc51+6/at4A3HlImF X6t6ZG7tr8j+UALLC040F4zEbH7S52sx0vb2eXjNL0VIZATb/n9/KWu19KbXmXQj98IG0TYFU5  
lennDEMHVwpzDQW0+1uGnHnNbsvolSSiy0LS5zWXUILAKoYgXsANe+Q8Hh6yEAG/jrL6i1S3bbTugYXaXQ1kVnDLZbm2rE  
6nyGhHpM9sZiuJS2mtp6ltrFKtJgd1tTPLqFT65SN94HsuGfsAnu+Ufhgt75QfLjIuz+1RvxtHYapAj4/C02a5WxBBvoDo  
d17eMqcZsNajEq+W9iylQwJGg0p0PfNTXoh1uN8rv4fnp++cBcFhkW6ZFym9yxvqzHS5t4W8pG/gUE5YAchbcJMTBrvzEw  
opiBURg8h/4lZtB7dkbyQB56TQ4phaZjFDNUQd9/Td8bQI229jI9RKiXznsuP5V7Kn0EvcEgzKo1ChU89Xb/avrCVqJVLK  
bE72tckHSwHG5Mk70weRdd9txNyL6knmx4+AECT0vHGNIGcTEvEvHgQ0DTs04iNJgKRB0sfnjWeBHCrt052gs8DUARwjVM  
dA4CLEvEJgLmiXnRsB0VYwQl0DhDI0AIRTAXp0o7PNqWJUaLenUI4Am6E+eYeYmDw9UBh37/0e5VqSuji6hkYZWU6gg8Jg  
drdBEO5qwxGSiNSGQu4/lQgj0eAvr4wM1sXH9VVKndfT4zd0cfmE80xrIxFSncKb2DEFhY21K6E7t30Xuwt07lJ8PLA2Ar  
d3n06y5sYGLVBj0S8Cg6Z4oCnkHEzE7Io56qjvvL/ppibsEXgLmM6G4G5LnU7v1gej7Jp2pnw/f78JGxeRLHZOnZ5iQ8ah  
DHxgHwb8L/v8AZkhWdvEqkdll3jjJLDFBhA51I3QLsRvk1jIWJ4wKjG4nWU+GrMa6sqlr amys1hY6kKCQ0EtMYmhJ/fK06  
LJ2qr fcf/xE/lAs8Iju2dxlA0VbWpiv90b9Dr rraTTFt0gDjSsIREgCyx6xwEEzQFcbGNZ4wtAdeNaNBjjAG8FDNBWgaV  
IVYNBCCAtowwgEaywEnRLR0DgI4CII4QHARwWcsKggKizyb6SuknWv1CEdWhsbnsS/E2Gr24Dc0Pdrun3SL+HpGjTI611N  
ze2SnuJJ4E7v+bTxqtUN81wt9c7DtH7icB+7wJWz3GQ7rF2IsLC4VLi3C418pJw1TIwPCQcC+dKiBizLaqtxY2HZf4FT5Q  
9B8ynmNfKBv8AAV8wUjWW00topRpFjvI0HmZ0bBq2S54CU02tqmql+A3DugQ8XiC7Mx1JMteiW0CVCjGw0ovzmezzkuzAL  
cnhaB7LhMcpbRh6ynx/SJDV6tFeq17HIhZVPe268pNh7EZSHrs1j7oNvWbZB00RQEUK0AAA0w1AhbuLE7xy7ryn2jQek3  
WJdk3uo3gfaHdLt60jjFKDraBFwe0VdbqZ1epbUyi21hjSb+IoeyT9YnAX94Dh3jvkLMVnTN3XgLjDmU21PLvgMBtE4cMh  
pliTc62bcBfLa5XvGmshPisis53LZv7h+sVMLthHUHQ8bw0o9JUb3G13ZSGv929s3leSqe26De/bhqDv46jSGQYeatmRCD  
vuo1MhbU6NKwLJdWt97dqAQfbHc3kRAnjF0yLiolueYTqeJRjZXRjyDAn0BmAqsY0UcEMBmsLtdQdXS+rQ0K+2ttcgXRD  
e9gGs5INldRufMpFlud+4ccu6B6SwgnWYXCbfq4dsr13UvlyN1aJTW1xZyBw58LTZb02lTrpnpsDb2gGBKncpIgPYRhEK  
yxloAwJ0JlnZYAmg80M6wVoGLURyiDVo9WgEUzmiRYDZ1o4LHBYAwIRRHBi9afpA5FlD0g22mFplmILkHiv8A9jyUSDtXp  
fQpBshFRl35TdR4kanyE8l25t56z13bUm4JF2HIIl7I0V9YAdtY96jmo7asbl21uearvbKcQA0H0VYJ0qIW/nqaj46R5ze  
0EAv79Q3Y9/a/SMrFTrUqM55LrbuudPSBKwG0enVR3emQDZl7PsEWYaC24mW01MAcPUsuqPc025rxUnmNPKxmedJuWmW8S  
T8BN30bdMVhuoqqQUsNfaAt2HUnlqPLvgR9m1s1F1XeF0g4ixmeQLlAvLV6L4StkfVT7LCGXn+o4SHj6gZrj0gRKKK28i3  
jNVsJc0hzM6X7yJjqqyEbxLXZWyKdXXPc8V9lh+vH1gennFUKL06upAtexBt6RKg0aIATRvu0bCZzBdEcPa+errycDcByGs  
sE6MYNd6FiL+3UbXfwB8IFliNoU1FzUQD74/WQ6rmqv10p0IY6KdeHP5QB2bhUYFKNPnws0NvM6k+Mt6LhBfjAr8ArglKo  
9rnuInbSpBFsDpu/W02lWLLpvGt5SYmpUqsiXsSQo8WNh5m8CJJaud0QA5ew5W2johAbXkCJXdSp60K/VFKpSmST1be2VV  
rjsaL7V7a6jjLPpIow2ITLotHEVKZ/7bpTqW/C7yp2jh7L1V7t1qlYdNVZqRPhYo3/qXgEo4qrTdHd2K0zldCmri/D00IP  
BhoZ6XsnaiuoN73nmW0axavinDG1IBrfVWBdaZps0KntaeYtJ0ysU9FkKezUynIWGZMwzDxUjc3rY7w3nSHYi10upKPCmj  
rvVx7LDv+YnnT0X0jMjjqyHGew0pVW0TEU/9J9zLuveeobLrFls2t5l0m2FAIRBc2VWwon/Uot/mKOZA7Y+6YGF7dWUANY  
gQ3KrUHabDk7zTcdpD7p3RNLY16Lq9JaRQDsGkU6zpfTIdb0ym6kHiNN8FVXQqXuAEps/NG7WEXpPshPIyLinS4esjFHZ  
g6Icpp11stS3DWwax/KB6xhMStRFqJqrC4594I4EboRhMj0Hx62egHzrq9Jjo2XQ0jD7amxP3r8Zq2eAhMS8S8SBzmBzwj

iDtA0Aj1M4C0tAVWhVMCBCKIBRHARiRNq7XTDr du05BKotyzd9hqB3wD7R2hToJnqNbkPeY8gPz3Cefbb6Q1MV2EJCfYp  
1aYc/eBBznuB8oLaWJq1nNR1qDk3ULUVV5atoB4CQEgVGPZGGxAPuZFV/JLIw8r wKDGk7hUL2YDIyZKwN9ymx1vyPlK50Z  
fdWn3sbufz+E021aYrLlCZKo0F0r 7R5CnVNmBvuV90R4Sr 0AFuxg3J+1UZiPSyiBRsyX1Lufwj8zHpm9ykB3kFj6tJGIp1  
U0uSn3Bk09CZFc39qr fuAY/OwgPdXPt0q92YD4LJmwdonD1lfr MynsuNTdTvI7xv8u+VL6Y40fML+s4svBP7mMD1fauAXE  
0SgIv7dN+Aa2mv2Tx9eE87VGRzTcFWU5WB3gzRdCtsAr /AA7XBXVNd68VvzHyPdLHpPsXr l62mPrUH/uKPDp8w4engFDiM  
ASugvpKhsM6nshr 915ouju0Vaytv3WPxmjQICCANf3+/CBisM20NsgqW3bv1Es8NsrHsczs4B3gkCbZKigX+EmJXUiBQbP  
2a6bxr6n1lg1Mr v3yfUr ASqx+LAHfuA4k8LCBwba2itJc7nTlxJ5DmZA6C4tsTjKebgxcjkEBK/HLKfpc5Y6nQbhy5zQ/Q  
3h03WrHggovmczH4CAP6SFHW4pbah8NVH9aGkf9qiE2bsF6/wBY3YLOmZSAe2qKGPgSin+kS8x2CTEYypI ASUKU0A91shJ  
zeFybeF5Z136pVYaWdAfBj l+ZEDzTpXQqUi60EyuVLMihCxW5UtbedTIH8cGxKuoIRVRFB3gKoFj33BM3PTvCipTzDhr+v  
5TzzDui7qote/tWlSPnA9d2HUBUGQemasPrE1ZCHA+0FN3XzXMP0Sdh0cqAdwndKCwo06i7IpcDmV7Vv00Dzg0kVsL70x  
ZM3PC4jWm39Dnfwgu0wdH9tLI3X/x0GNhod5en63hCiEZfcQ5Cf8AyuJ7VNV6HIgqrsCKje2pV3H+rQcUq/4kZWgM2btIJ  
UTEU1C9pespjQB7WLIPsuuYW4G3dPUetDAMPupAIPMEXE8uxRp1M9RRlKXSqqjUoHHV107xZcwm56J4o1MMoNsYMyG27TV  
S04gi0C2Dzs84rGEQH5oy8WmtA0itHgWsiFUQHrDoINU4zGdIek3Wxo4c3QGzuC/a5gdWpIXzF/DeFztjpmL06Uu2/wBoK  
7IvgUUhjMJjHaqxLujltSr1a1In8YRYF6iil+qB/mOLuPMYRQqZtFBb/s1s/wD8Fa5bwgREwrUzmFDEp/NSc007ULb4yQt  
Ra182WsRvV16rEabyrrc0d+hJPdEZFQK9xLL5M6B8PURt5zpqmnd6wjJUC5Hb/E02z0yQt6iWuAHHTEAzhqQdYA8QV6tTm  
aphn0Vs4+sovpoDztq0BA3XLXicJTRnTE1atwQaZUB1dCLhrsRv/MS4p4hAGqZPqqnYxFib0YnR0HAX1Xkbj lB1kdCEV1z  
01vSqEArUwzm7C5BtYdsb9zCBncSuEX2RWfVzKXX+mV75T7FEkcNwf8psHbEnQYnDaaAhqY90xf4SsxeHrn28ZT8qxPwUA  
QKEdYNYzf6Lf7o1jV4sB/Uo+XhCYvCoNTXvj3Bm+N5DyJ9o+SwdU69VHV1cZlII7SnX9P1nqGx9pJWpo67y00L+y3vCeT0  
E4FvSXHRnaYo1At+w5AJ3Wf3Tv47vTlAlu2+jpZzXw1g+903BjxZTuDcXu0/TjQrtipSbI6spG9WFj6H5zbUsRcSPjXUiz  
qGHJgGHxgRMN0ipMmrWPIx9Hb6D3x6yH/8AzqbGwo0/wLNBszohhaq2enlb7SGx9LW9RARK/SBdyn0eAGvryjMCjM2dzdz  
cADco7uZ75dYno3h6JP1yIBpeoMgNt4DAENbQHdaRa+TIWoulWxCK03R+0QSQ2U3zEA2HceUCsw1Km1Zmex6u1glrDjmn+  
QmgwLtuZCklkIKs9ygdTvAG8i3Ezz7Z2MV8Qrkgrmtl4HS4J5ier7LqLYeEBlnCgHYJUB8upHlxHhrIfSHFKcJWdWBC0y6  
nvWzD4iX7rxEwfTqkiISzMQVDCqpsrOpv6nTTja8Cg2p0g69FppfddzyEi7FT64abpV0WBHYBuxsANSeQ750thbJc102xV  
tDlWxIHedwgegbLY2F5Kx9MmpHAjXzkbD0coGpPiZIfVbwPLsVhVpv1TaJrh3v/ANGsSabX5JUUVK0hKC5AbewGcfz64as  
fU03mk6V0Q532zjqSeRftU2Pg6IP6zM2WLHNuNTK/eDUU0qg7stVEPNafg665ErFRmpDqcQm7rKT9gP4i+U94Uy96A1sLS  
pSvcG4/qT2GH3llfglfTHDhL0vsYijUvbXtg3YfiVJJ2VieorLWJy5KjUqWIJ3liEDXUZlvwtxgemMIwiSaRV1V0IZWAI  
I1BB3WIiMkCPadTJBgv0BoEWFd1RS7sFUbyTYTp0DC9IEkj1WUpgr5nKvWcll3ITL/KA/eAZQI/86eeKcHw7ChR6Tp0Ag  
ruuod7aC9PGo1rm2qtra54+sI6Z2am7KtVWqfWU6ahxbRS677k6Xup01liToBqdRsz5Uic/wCdhXvapzemTx423jeMwjK  
ioEUhya0b6up/4mHqb8j21tfXTxHI90gdUZyz0FXrkU9fT9yvSPt0oGhuLE27mG6MwmjotLP2HzNh6jWBR/foudwBNvPKd  
xnToFfhKN1KnC07U2NNyr1PbXmoBANiDbviVqCa5sFWhi9QfNPGLOgVmKRPdwr373qH4BRKmpo4Ubd1nPznToAWY/YH4W/  
WNKnflsNxGtjOnQNv0Y2pnQKx7SaG/Ecd+veJpKeEaoQB0nQNJs3YoWxi1/f6SZtnGU8Jh3rsAci3VLhc77kQHvNok6B5J  
T2saH+ISun8TUqVC60jMtMPmcvncZb3IAABOp7xm2+LZ83ZA7ClTWIZlv3Gt30Y2JtlvpaLOgdw+Is2lgBusLcbi/E8tZ  
6p0a2mHRTfXd5zp0DV0sRKjpdX+ModWrhGDBwxGYXAIscEXvvE6dAotl9HqEDQsVNSqB7ZsLE7gg90X85Z7D2aEGdtXY  
3J8Yk6BdVGAtIuKJtp8506BjNvIztKU+32A2mj5WZCP60UT0583aC2LXcDgpqA51/pr0x+KL0gBwo6xBTJ0FSoi62s1VCa  
fkXQjzkps9UU61NVZ3BpVUYLlNSmAwJDGxLIN3MGd0gazoNtEWfDFSjKSwpNe6X9oITqUvrY6i54Wmqczp0ATiAnToH//2  
Q=="

```
},  
"id": 3910,  
"userType": "I",  
"homeServer": "Void host",
```

```
        "shortName": "pgarcia",
        "primaryGroupDescription": "Music",
        "primaryGroup": "Music"
    }
]
}
```

## Example 3

### 1. Use the Fuzzy Searches

#### Request

```
GET http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User?textFilter=fran~
```

#### Response 200 OK

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 2,
  "startIndex": 1,
  "Resources": [
    {
      "lastName": "Sinatra",
      "createdByUser": "ActiveDirectory",
      "fullName": "Frank Sinatra",
      "active": true,
      "userName": "frank",
      "mailAlias": "",
      "firstName": "Frank",
      "createdDate": "2023-09-06 13:12:54",
      "multiSession": true,
      "meta": {
        "location":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/276435",
        "links": {
          "roleAccounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+' frank' +
and+enabled+eq+true",
```

```

        "groupUsers":
        "http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+' frank' +and+disabled+eq+false",
        "accounts":
        "http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.userName+eq+' frank' ",
        "issues":
        "http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+' frank' ",
        "effectiveGrants":
        "http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/276435/effectiveGrants"
    },
    "resourceType": "User"
},
"modifiedByUser": "ActiveDirectory",
"schemas": [
    "urn:soffid:com.soffid.iam.api.User"
],
"modifiedDate": "2023-09-06 13:12:55",
"attributes": {},
"id": 276435,
"userType": "I",
"primaryGroupDescription": "Music",
"primaryGroup": "Music"
},
{
    "lastName": "Sherwood",
    "createdByUser": "pgarcia",
    "fullName": "Frank Sherwood",
    "active": true,
    "userName": "franks",
    "mailAlias": "",
    "firstName": "Frank",
    "createdDate": "2023-10-05 15:32:40",
    "multiSession": false,
    "meta": {
        "location":
        "http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/432644",
        "links": {
            "roleAccounts":
            "http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+' franks'

```

```

+and+enabled+eq+true",
    "groupUsers":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+'franks'+and+disabled+eq+false",
    "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.userName+eq+'franks'",
    "issues":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+'franks'",
    "effectiveGrants":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/432644/effectiveGrants"
  },
  "resourceType": "User"
},
"modifiedByUser": "pgarcia",
"schemas": [
  "urn:soffid:com:soffid:iam:api:User"
],
"modifiedDate": "2023-10-05 15:32:40",
"attributes": {},
"id": 432644,
"userType": "I",
"primaryGroupDescription": "scientist",
"primaryGroup": "scientist"
}
]
}

```

## 2. Use the Fuzzy Searches: specify the maximum number of edits allowed

### Request

```
GET http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User?textFilter=frankl~2
```

### Response 200 OK

```

{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],

```

```
"totalResults": 4,
"startIndex": 1,
"Resources": [
  {
    "lastName": "Franklin",
    "createdByUser": "ActiveDirectory",
    "fullName": "Rosalind Franklin",
    "active": true,
    "userName": "rfranklin",
    "mailAlias": "",
    "firstName": "Rosalind",
    "createdDate": "2023-08-08 14:26:14",
    "multiSession": true,
    "meta": {
      "location": "http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/2862",
      "links": {
        "roleAccounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+'rfranklin'+and+enabled+eq+true",
        "groupUsers":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+'rfranklin'+and+disabled+eq+false",
        "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.userName+eq+'rfranklin'",
        "issues":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+'rfranklin'",
        "effectiveGrants":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/2862/effectiveGrants"
      },
      "resourceType": "User"
    },
    "modifiedByUser": "ActiveDirectory",
    "schemas": [
      "urn:soffid:com.soffid.iam.api.User"
    ],
    "modifiedDate": "2023-08-08 14:26:14",
    "attributes": {},
    "id": 2862,
```

```
    "userType": "I",
    "primaryGroupDescription": "scientist",
    "primaryGroup": "scientist"
  },
  {
    "lastName": "Franklin",
    "createdByUser": "ActiveDirectory",
    "fullName": "Aretha Franklin",
    "active": true,
    "userName": "aretha",
    "mailAlias": "",
    "firstName": "Aretha",
    "createdDate": "2023-09-06 13:12:54",
    "multiSession": true,
    "meta": {
      "location":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/276397",
      "links": {
        "roleAccounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+'aretha'+and+enabled+eq+true",
        "groupUsers":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+'aretha'+and+disabled+eq+false",
        "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+eq+'aretha'",
        "issues":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+'aretha'",
        "effectiveGrants":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/276397/effectiveGrants"
      },
      "resourceType": "User"
    },
    "modifiedByUser": "ActiveDirectory",
    "schemas": [
      "urn:soffid:com.soffid.iam.api.User"
    ],
    "modifiedDate": "2023-09-06 13:12:54",
```

```
    "attributes": {},
    "id": 276397,
    "userType": "I",
    "primaryGroupDescription": "World",
    "primaryGroup": "world"
  },
  {
    "lastName": "Sinatra",
    "createdByUser": "ActiveDirectory",
    "fullName": "Frank Sinatra",
    "active": true,
    "userName": "frank",
    "mailAlias": "",
    "firstName": "Frank",
    "createdDate": "2023-09-06 13:12:54",
    "multiSession": true,
    "meta": {
      "location":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/276435",
      "links": {
        "roleAccounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+' frank' +
and+enabled+eq+true",
        "groupUsers":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+' frank' +and+di
sabled+eq+false",
        "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user
.userName+eq+' frank' ",
        "issues":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+' frank' ",
        "effectiveGrants":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/276435/effectiveGrants"
      },
      "resourceType": "User"
    },
    "modifiedByUser": "ActiveDirectory",
    "schemas": [
      "urn:soffid:com.soffid.iam.api.User"
    ],
  },
}
```

```

    "modifiedDate": "2023-09-06 13:12:55",
    "attributes": {},
    "id": 276435,
    "userType": "I",
    "primaryGroupDescription": "Music",
    "primaryGroup": "Music"
  },
  {
    "lastName": "Sherwood",
    "createdByUser": "pgarcia",
    "fullName": "Frank Sherwood",
    "active": true,
    "userName": "franks",
    "mailAlias": "",
    "firstName": "Frank",
    "createdDate": "2023-10-05 15:32:40",
    "multiSession": false,
    "meta": {
      "location":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/432644",
      "links": {
        "roleAccounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+'franks'+and+enabled+eq+true",
        "groupUsers":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+'franks'+and+disabled+eq+false",
        "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+eq+'franks'",
        "issues":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+'franks'",
        "effectiveGrants":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/432644/effectiveGrants"
      },
      "resourceType": "User"
    },
    "modifiedByUser": "pgarcia",
    "schemas": [

```

```

        "urn:soffid:com.soffid.iam.api.User"
    ],
    "modifiedDate": "2023-10-05 15:32:40",
    "attributes": {},
    "id": 432644,
    "userType": "I",
    "primaryGroupDescription": "scientist",
    "primaryGroup": "scientist"
}
]
}

```

## Example 4

### 1. Use the boolean operator AND

#### Request

```
GET http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User?textFilter=fran~ AND Sinatra
```

#### Response 200 OK

```

{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 1,
  "startIndex": 1,
  "Resources": [
    {
      "lastName": "Sinatra",
      "profileServer": "Void host",
      "createdByUser": "admin",
      "fullName": "Frankaaa Sinatra",
      "active": true,
      "userName": "frank",
      "mailAlias": "",
      "mailServer": "Void host",
      "firstName": "Frankaaa",
      "emailAddress": "pgarcia@soffid.com",
      "mailDomain": "soffid.com",
    }
  ]
}

```

```
"createdDate": "2023-06-02 07:41:47",
"multiSession": false,
"meta": {
  "location": "http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/3910",
  "links": {
    "roleAccounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+' frank' +
and+enabled+eq+true",
    "groupUsers":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+' frank' +and+di
sabled+eq+false",
    "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user
.userName+eq+' frank' ",
    "effectiveGrants":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/3910/effectiveGrants"
  },
  "resourceType": "User"
},
"modifiedByUser": "admin",
"schemas": [
  "urn:soffid:com.soffid.iam.api.User"
],
"modifiedDate": "2023-06-02 07:41:47",
"attributes": {
  "picture":
"/9j/4AAQSkZJRgABAQAAQABAAQ/2wCEAAoHCBYWFRgVFRYYGRgaGhkYGHgaGBoYGHoYHBkaGhgZGhgCIS4lHB4rIRgYJ
jgmKy8xNTU1GiQ7QDs0Py40NTEBDaWMBgYGEAYGEDEdFh0xMTExMTExMTExMTExMTExMTExMTExMTExMTExMTExMTE
xMTExMTExMTExMTExMf/AABEIAKgBKwMBIgACEQEDEQH/xAAcAAABBQEBAQAAAAAAAAAADAQIEBQYABwj/xABFEAACA
QIDBAcECAQFAGcBAAABAgADEQSSIQUxQVGEyJhcYGRMkKhsQcjUnKSwdHwFGKCoIQzY7LhU3NDZIOTo8LxPf/EABQBAQA
AAAAAAAAAAAAAAAAAAD/xAAUEQEAAAAAAAAAAAAAAAAAA/9oADAMBAAIRAxEAPwDeqIVZwEcBAQQiCIBHqIDgItogj
wICWiqIto4CAqiZn6ScNnWdN7DI/wAcp/3TUKJX9JcNnWld0dNj5qMw+UD5zcQYh6qG+gPpGrQbAdThBHUsK3MfEw4wR5
n5QG4dtd8t8JU3DT0/wCZBpYVRvI9ZMpqq97ytxgTi44n8zz3D9+k4G+4ee79iR/4pF3C/iY2rtIAakD0gT6WH0lz4/v97
oZEPL7RzHu5/nM/U2wg3Fm+Eh1NsN7oA+MDWfxir7Kj975Fx00re04Xuvb+0a+kyVTaDtvC+ANh6CANQwNFw2wg3ZnP4R6
nX4SFV2y59kKv9x9Tp8JUXMab3gS62MZvaYnx0npI7Vo0oeAJiVKLAXINoCl41njAJzixsdPnAUNEZpY7H2FiMSbUabMOL
nsOPFzp6XM9G60/R/SpEPiCKrjULb6tT906ue86d0DLdEuhz4krVqhkob+Iao0ScLP2vTmPwqNFUUIihVUBVUCwAG4AQw3
RpEBhEYVhJgkQA0sCZJYQbLAj00ZeFqLBQNAqwgSKphFGMCRwSEAhVWAFUsCToBqSdwHEmUmI6YYJCQapJH2Uc/G0r/pH2
y1FEpLcBwSx5gaBfz9J4/isWzHfab61ivpKwqexTrP32RB8Ww8ACU2J+lVrkU8MgHAvULf2qo+c8xZo14G1xn0kY5/ZdKY
/00F/V80pMX0jxNX/ADMTVYHeM5VfwrYfCUmaITAmCoI7r5BzTrmBYDFGDav3yJaKtNjwgG0JPjGHFNFXCkyTR2cDqXPgI
EA12PExERmPZBPgCZd0cIg0iE+MtKeznb2QQDwtv/esDLjBPxGXx/S0TB8zfwm2w/Rose1eX0H6PULtm3wMBhtiM5si+us
```

u8N0KdvaNvKwm/wAFhUQjIl7d36y1C0fcC+JEDzE9Csu8mFodF1G+3pN9XRuQ+cjFLDUQMvh+j6BtRbyhdq7DTJu56j9Ja  
tvvFxL9gg3/AHXgeYYjZ6qWPLlu8ZsuhfRzDPSGIemHcsW7faUZTYWTdfxvKNkzswtcA30F766D4fGekbCw4TD01At2Qbd  
51J8TeBNRQAABYDcBuHlHho2dAfeJmgY0mATNGs0YzRmeA8tBu0VngWeAjtBZo4vB5oGlQwimAVo4PAkq0KryIHjg8CJ0  
k2DSxtLq6hZWGq0p7SHjpuIPEGeN9IuhWKwPLMhqU+FRAWFv5l3p56d89yDxy1IHZNEn0BtTozg8QSa1BMx3ul0bzKWv5z  
M4z6L8M2t0tVp9zZXHyB+MDy0JPRqv0VVPcxKH71Nl+TGQNoFzWoI9Z61Iog1Az5iScqgXHMibJUp3498lrhlG+DanlPw  
lmaV0Q/y/L/APIEXIL6CEWnyndfJ2BVScpNoDMNgGYiwB85ZJsuotiU08QZabMwig38949RNKmFDLrYwMxgKS3CurKeFx  
++c02GwAt2XBHDSK0FKqLi45b/Cd1AGqi3h+kB38Jbe/oJJpBFgm+N0y8z6RmGIBvx5wDde5PZB8bWkoI1u03kJyM0JnV  
sQAL6Ad/OAKpSHf6yvXnhuvFq7U1IWl++RhXz3vpAdm3yJjn7DW0IBsYdjy0m6QsShyt90/KBub0w/YVt9xmJ14k6fAes3  
+DZcqoHViqqDYi+gHDhMLhELhETQMoGhsQNM5PLfaw+MxaU6iJTULLIuw9lRbUM3EnlA1BE6DLxM8B5Ea0b1k7NAY4g4Ux  
pEATGBYw7iDKwAxmWHIEbpAv1hAIJYQGA4R4jBHKYBBFiAziYCGKI0mKIDxKHp4p/gnt9unfwzj87S+WVnSxL4Kv3Jn/Aw  
b8oHi+LpAtpwCmTqVH6pL8j/uMFTp7yfD0AEYpCiI0BRfjrAqkoXa1o9sMyG/wCWhpCqLax+P2oEAAAZHXUc0EBuG2w9M  
3yg25f8S2wnTncwDIV4XHw0mTxWiE/bsugPAWwz7+6Ds9sxW4G/mLgEEjkQQbwPYdnbdp0tlYa+nflBmVhpuni2G2nk5ib  
jox0jD9g+1A090oBmEitj0Q9owi4RzrKDaxdWsqFi0NtLwJe1Nt1stqKgXF8zfVtjKHE9o3r4nX71hfmbI2JNUKk1xfQKD  
qSeQ4eJg2Z8Ndy6U70EuEzszWubs29R690C1wGKw6nLnDhD2jL1KiMNDfvvYD0kKjgKuJp02alXZGKuLSmACAAVNNxYrch  
iN8yiYoo/1QqBNzI3a6sg20f3l+UDb1TcaayG99YLDVibZuUk110vAhbBpXJHhMU8r3NvIXLxWpA0VAFrgj9+szmA26uHx  
LoVuxF8xPshgCQB4W9Zf5y75h7639bpf+4QLmn7I8B8opESLAQRSZxERYAW84xVE5lgBYxpjmSMcQBuY08c0HaBqFEdaIo  
jha60UCdaKIC3nTohMDoojLxYBVMbjSP1lGpT+2jp+JSB0WSKZgeB1Kp00ul7jkbk/vwlli6oegjjgSvkDp8CI/pbs/qcX  
Wp7lc51+6/at4A3HlImF6Xt6ZG7tr8j+UALLC040F4zEbH7S52sx0vb2eXjNL0VIZATb/n9/KWu19kBXmXQj98IG0TYFU5  
lennDEMhVwpzDQW0+1uGnHnNbsvolSSiy0Ls5zWXUILAKoYgXsAne+Q8Hh6yEAG/jrL6i1S3bbTugYXaXQ1kVnDLZbm2rE  
6nyGhHpM9sZiuJS2mtp6ltrFKtJgd1tTPLqFT65SN94HsuGfsAnu+Ufhgt75QfLjIuz+1RvxtHYapAj4/C02a5WxBBvoDo  
d17eMqcZsNajEq+W9iylQwJGg0p0PfntXoh1uN8rv4fnp++cBcFhkW6ZFym9yxvqzHS5t4W8pG/gUe5YAchbcJMTBrvzEw  
opiBURg8h/4LztB7dkbyQB56TQ4phaZjFDNUQd9/Td8bQI229jI9RKiXznsuP5V7Kn0EvcEgzKo1ChU89Xb/avrCVqJVLK  
bE72tckHSWGH5Mk70werdd9txNyL6knmx4+AECT0vHGNIgcTEvEvHgQODTs04iNJgKRB0sfnjWebHCRto52gs8DUARwjVM  
dA4CLEvEJgLmiXnRsB0VYwQl0DhDI0AIRTAxP0o7PNqWJUaLenUI4Am6E+eYeYmDw9UBh37/0e5VqSuji6hkYZWU6gg8Jg  
drdBEO5qwxGSiNSGQu4/lQgj0eAvr4wM1sXH9VVKndfT4zd0cfmE80xrIxFsncKb2DEFhY21K6E7t30Xuwt07LJ8PlA2Ar  
d3n06y5sYGLVBj0S8Cg6Z4oCnkHEzE7Io56qjvvL/ppibsEXgLM6G4G5LnU7v1gej7Jp2pnw/f78JGxerLHZ0nZ5iQ8ah  
DHxgHwb8L/v8AZkhWdvEqkd1l3jjJLDFBhA5I3QLsRvk1jIWJ4wKjG4nWU+GrMa6sqrAmys1hY6kKCQ0EtMYmhJ/fK06  
LJ2qrfcF/xE/lAs8Iju2dxlA0VbWpiv90b9DrraTTft0gDjSsIREgCy6xWEEzQFcbGNZ4wtAdeNaNBjjAG8FDNBWgaV  
IVYNBCCAtowwgEaywEnRLR0DgI4CII4QHARwWcsKggKizyb6SuknWv1CEdWbsbnss/E2Gr24Dc0Pdrun3SL+HpGjTI611N  
ze2SnuJJ4E7v+bTxqtUN81wt9c7Dth7icB+7wJWz3GQ7rF2IsLC4VLI3C418pJw1TIwPCQcC+dKiBizLaqtxY2HZf4FT5Q  
9B8ynmNfKBv8AAV8WujWW00topRpFjvI0HMz0bBq2S54CU02tqmql+A3DugQ8XiC7Mx1JMteiW0CVCjGw0ovzmezzkuzAL  
cnhaB7LhMcpbRh6ynx/SJDV6tFeq17HIhZVPe268pNh7EZShrs1j7oNvWbZB00RQEUK0AAA0w1AhbuLE7xy7ryn2jQek3  
WJdk3uo3gfaHdlt60jjFKDrabFwe0VdbqZ1epbUyi21hjSb+IoeyT9YnAX94Dh3jvkLMVnTN3XgLjDmU21PLvgMBtE4cMh  
pliTc62bcBfLa5XvGmshPisis53LZv7h+sVMLthHUHQ8bw0o9JUB3G13ZSGv929s3leSqe26De/bhqDv46jSGQYeotmRCD  
vuo1MhbU6NKwLJdWt97dqAQfbHc3kRAnjF0yLiolveYTqeJRjZXRjyDAn0BmAqsY0UcEMBmsLtdQdXS+rQ0K+2ttcwgXRD  
e9gGs5INldRufMpF1ud+4ccu6B6SwgnWYXcbfq4dsr13UvlyN1aJTW1xZyBw58LTzb02lTrpnpsDb2gGBKnkcpIgPYRhEK  
yxloAwJ0JlnZYAmg80M6wVoGLURyidVo9WgEUzmiRYDZ1o4LHBYAwIRRHBI9afpA5FlD0g22mFplmILkHiv8A9jyUSDtXp  
fQpBshFRl35TdR4kanyE8l25t56z13bUm4JF2HIIl7IOV9YAdtY96jmo7asbl21uearvbkCQA0H0VYJ0qIW/nqaj46R5ze  
0EA79Q3Y9/a/SMrFTrUqM55LrbuudPSBKwG0enVR3emQDZl7PsEWYaC24mW01MACPUsuqPc025rxUnmNPKxmedJuwWmW8S  
T8BN30bdMVhuoqqQUsNfaAt2HUnlqPLvgR9m1s1F1XeF0g4ixmeqLlAvLV6L4StkfVT7LcGXn+o4SHj6gZrj0gRKKK28i3

jNVsJc0hzM6X7yJjqqyEbXLXZWyKdXXPc8V9lh+vH1gennFUKl06upAtexBt6RKg0aIATrVu0bCZzBdEcPa+errycDcByGs  
sE6MYNd6FiL+3UbXfwB8IFliNoU1FzUQD74/WQ6rmqv10p0IY6KdeHP5QB2bhUYFKNPNwsoNvM6k+Mt6LhBfjAr8ArglKo  
9rnuInbSpBFsDpu/W02lWLLpvGt5SYmpUqsiXsSQo8WNh5m8CJj aud0QA5ew5W2j oHAbXkCJXdSp60K/VFKpSmSTlbe2VV  
rjsaL7V7a6jjLPpIow2ITLotHEVKZ/7bpTqW/C7yp2jh7L1V7t1qlydNVZqRPhYo3/qXgEo4qrTdHd2K0zldCmRI/D00IP  
BhoZ6XsnaiuoN73nmW0axavinDG1IBRfVWBdaZps0KntaeYtJ0ysU9FkKezUynIWGZMwzDxUjc3rY7w3nSHYi10upKPCmj  
rvVx7LDv+YnnT0XjMjJqqyHGew0pVW0TEU/9J9zLuveeobLrFls2t5l0m2FAIRBc2VWwon/Uot/mKOZA7Y+6YGF7dWUANY  
gQ3KrUHABdK7zTcdpD7p3RNLy16Lq9JaRQDsGkU6zpfTIdb0ym6kHiNN8FVXQqXuAEps/NG7WEXpishPIyLinS4esjFHZ  
g6Icpp11stS3DWwax/KB6xhMStRFqJqrC4594I4EboRhMj0Hx62egHzr9JjJo2XQ0jD7amxP3r8Zq2eAhMS8S8SBzmBzwj  
iDtA0Aj1M4C0tAVWhVMCBCKIBRHARiIRNq7XTDrdu05BKotyzd9hqB3wD7R2hToJnqNbkPeY8gPz3Cefbb6Q1MV2EJCfYp  
1aYc/eBBznuB8oLaWJq1nNR1qDk3ULUVV5atoB4CQEgVGPZGGxAPuZfV/JLIw8rWKGK7hUL2YDIyZKwN9ymx1vyPlK50Z  
fdWn3sbufz+E021aYrLLCZKo0F0r7R5CnVnMBvuV90R4Sr0AFuxg3J+1UZIpsYiBRsyX1Lufwj8zHpm9ykB3kFj6tJGIp1  
U0uSn3Bk09CZFc39qrfuAY/OwgPdXPt0q92YD4LJmwdonD1lfrMynsuNTdTvI7xv8u+VL6Y40fML+s4svBP7mMD1fauAXE  
0SgIv7dN+AA2mv2Tx9eE87VGRzTcFWU5WB3gzRdCtsAr/AA7XBxVNd68VvzHyPdLHpPsXrL62mPrUH/uKPDp8w4engFDiM  
ASugvpKhsM6nshr915ouju0Vaytv3WPxmJQICCANf3+/CBisM20NsgqW3bv1Es8NsrHsczs4B3gkCbZKigX+EmJXUiBQbP  
2a6bxr6n1lg1Mrv3yfUrASqx+LAHfuA4k8LCBwba2itJc7nTlxJ5DmZA6C4tsTjKebgxcjkEBK/HLKfpc5Y6nQbhy5zQ/Q  
3h03WrHgqovmczH4CAP6SFHW4pbah8NVH9aGkf9qiE2bsF6/wBY3YLOmZSAe2qKGPgSin+kS8x2CTEYypIASUKU0A91shJ  
zeFybeF5Z136pVYAWdAfBjl+ZEDzTpXQqUi60EyuVLMihCxW5UtbedTIH8cGxKuoIRVRFB3gKoFj33BM3PTvCipTzDhr+v  
5TzzDui7qote/tW1sPnA9d2HUBUGQemasPrE1ZCHA+0FN3XzXMP0Sdh0cqAdwndKCwo06i7IpcDmV7Vv00Dzg0kVsL70x  
ZM3PC4jWm39Dnfwgu0wdH9tli3X/x0GNhod5en63hCiEZfcQ5Cf8AyuJ7VNV6HIgqrsCKje2pV3H+rQcUq/4kZWgM2btIJ  
UTEU1C9pespjQB7WLIPsuuYW4G3dPUetDAMPupAIPMEXE8uxRp1M9RR1KXSqqjUoHHV107xZcwm56J4o1MMoNsYMyG27TV  
S04gi0C2Dzs84rGEQH5oy8WmtA0itHgWsiFUQHRDoINU4zGdIEk3Wxo4c3QGzuC/a5gdWpIXZF/DeFztjpmL06Uu2/wBoK  
7IvgUUhjMJjHaqxLujltSr1alIn8YRYF6iil+qB/m0LuPMYRQqZtFbb/s1s/wD8Fa5bwgREwrUzmFDEp/NSc007ULb4yQt  
Ra182WsRvV16rEabyrrcOd+hJPdEZFQK9xLL5M6B8PURt5zpqmnd6wjJuc5Hb/E02z0yQt6iWuAHHtEAzhQdYA8QV6tTm  
aphn0Vs4+sovpoDztq0BA3XLXicJTRnTE1atwQaZUB1dCLhrsRv/MS4p4hAGqZPqqnYxFIb0YnR0HAX1XkbjLB1kdCEV1z  
01vSqEARUwzm7C5BTYdsb9zCBncSuEX2RWfvZkXX+mV75T7FEKcNWF8psHbEnQYnDaaAhqY90xf4SsxeHrn28ZT8qxPwUA  
QKEdYNYzf6Lf7o1jV4sB/Uo+XhCYvCoNTXVj3Bm+N5DyJ9o+SwdU69VHV1cZLI17SnX9P1nqGx9pJWpo67y00L+y3vCeT0  
E4FvSXHRnaYo1At+w5AJ3Wf3Tv47vTLA1u2+jpZzXw1g+903BjxZTuDcxu0/TjQrtipSbI6spG9WFj6H5zbUsRcSPjXUiz  
qGHJgGHxgRMN0ipMmrWPIx9Hb6D3x6yH/8AzqbGwo0/wLNBszohhaq2enlb7SGx9LW9RARK/SBdyn0eAGvryjMCjM2dzdz  
cADco7uZ75dYno3h6JP1yIBpeoMgNt4DAENbQHdaRa+TIWoulWxCk03R+0QSq2U3zEA2HceUCsw1Km1Zmex6u1grDjmn+  
QmgwLtUzCklkIKs9ygdTvAG8i3Ezz7Z2MV8Qrkgrmtl4HS4J5ier7LqLYeEBlnCgHYJUB8upHlxHhrIfSHFKcJWdWBC0y6  
nvWzD4iX7rxEwfTqkiISzMqVDCqpsrOpv6nTTja8Cg2p0g69FppfddzyEi7FT64abpV0WBHYBuxsANSeQ75othbJc102xV  
tDlWxIHedwgegbLY2F5Kx9MMpHAjXzkbD0coGpPiZiFVbwPLsVhVpv1TaJrh3v/ANGsSabX5JUUVk0hKC5AbewGcfz64as  
fU03mk6V0Q532zjqSeRftU2Pg6IP6zM2WLHNuNTK/eDUU0qg7stVEPnAFg665ErFRmpDqcQm7rKT9gP4i+U94Uy96A1sLS  
pSvcG4/qT2GH3llfglFtHDHl0vsYijUvbXtg3YfiVJJ2VieorLWJy5KjUqwIJ3liEDXUZlvwtxgemMIwiSaRV1V0IZWAI  
I1BB3WIiMkCPAdtJBgv0BoEWfd1RS7sFUbyTYTp0DC9Iekj1wUpgr5nKvWcLl3ITL/KA/eAZQI/86eeKcHw7ChR6Tp0Ag  
ruuod7aC9PGo1rm2qtra54+sI6Z2am7KtVWvqfWU6ahxbRS677k6Xup01liToBqdRsZ5Uic/wCdhXvapzemTx423jeMwjK  
ioEUhya0b6up/4mHqb8j21tfXTxHI90gdUZyz0FXrkU9fT9yvSPt0oGhuLE27mG6MwmjotLP2HzNh6jWBR/foudwBNvPKd  
xnToFfhKN1KnC07U2NNyr1PbXmoBANiDbviVqCa5sFWhi9QfNPGL0gVmKRPdwr373qH4BRkmop4Ubd1nPznToAWY/YH4W/  
WNKnflsNxGtjOnQNv0Y2pnQKx7SaG/Ecd+veJpKeEaoQB0nQNJs3YoWxI1/f6SZtnGU8Jh3rsAcI3VLhc77kQHvNok6B5J  
T2saH+ISun8TUqVC60jMtMPmcvncZb3IAABOp7xM2+LZ83ZA7ClTWIZlv23Gt30Y2Jtlvpal0gDw+Is2lgBusLcbi/E8tZ

```

6p0a2mHRTfXd5zp0DV0sRKj pdsX+ModWrhGDBwxGYXAI sRcEXvvE6dAotl9HqeDQsVNSqB7ZsLE7gg90X85Z7D2aEGdtXY
3J8Yk6BdVGAtIuKJtp8506BjNvIztkU+32A2mj5WZCP60UT0583aC2LXcDgpqA51/pr0x+KL0gBwo6xBTJ0FSoi62s1VCa
fkXQjzkps9UU61NVZ3BpVUYLlNSmAwJDGxLIN3MGd0gazoNtEwfDFSjKSwpNe6X9oITqUvrY6i54Wmqczp0ATiAnToH//2
Q=="

    },
    "id": 3910,
    "userType": "I",
    "homeServer": "Void host",
    "shortName": "pgarcia",
    "primaryGroupDescription": "Music",
    "primaryGroup": "Music"
  }
]
}

```

## 2. Use the boolean operator +

### Request

```
GET http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User?textFilter=fran~ +bacall
```

### Response 200 OK

```

{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 3,
  "startIndex": 1,
  "Resources": [
    {
      "lastName": "Bacall",
      "createdByUser": "ActiveDirectory",
      "fullName": "Lauren Bacall",
      "active": true,
      "userName": "lbacall",
      "mailAlias": "",
      "firstName": "Lauren",
      "createdDate": "2023-08-08 14:26:14",
      "multiSession": true,
      "meta": {

```

```
    "location": "http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/2844",
    "links": {
      "roleAccounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+'lbacall'+and+enabled+eq+true",
      "groupUsers":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+'lbacall'+and+disabled+eq+false",
      "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.userName+eq+'lbacall'",
      "issues":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+'lbacall'",
      "effectiveGrants":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/2844/effectiveGrants"
    },
    "resourceType": "User"
  },
  "modifiedByUser": "pgarcia",
  "schemas": [
    "urn:soffid:com.soffid.iam.api.User"
  ],
  "modifiedDate": "2023-08-22 17:34:07",
  "attributes": {},
  "id": 2844,
  "userType": "I",
  "primaryGroupDescription": "Music",
  "primaryGroup": "Music"
},
{
  "lastName": "Sinatra",
  "createdByUser": "ActiveDirectory",
  "fullName": "Frank Sinatra",
  "active": true,
  "userName": "frank",
  "mailAlias": "",
  "firstName": "Frank",
  "createdDate": "2023-09-06 13:12:54",
  "multiSession": true,
```

```
    "meta": {
      "location":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/276435",
      "links": {
        "roleAccounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+' frank' +
and+enabled+eq+true",
        "groupUsers":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+' frank' +and+di
sabled+eq+false",
        "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user
.userName+eq+' frank' ",
        "issues":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+' frank' ",
        "effectiveGrants":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/276435/effectiveGrants"
      },
      "resourceType": "User"
    },
    "modifiedByUser": "ActiveDirectory",
    "schemas": [
      "urn:soffid:com.soffid.iam.api.User"
    ],
    "modifiedDate": "2023-09-06 13:12:55",
    "attributes": {},
    "id": 276435,
    "userType": "I",
    "primaryGroupDescription": "Music",
    "primaryGroup": "Music"
  },
  {
    "lastName": "Sherwood",
    "createdByUser": "pgarcia",
    "fullName": "Frank Sherwood",
    "active": true,
    "userName": "franks",
    "mailAlias": "",
    "firstName": "Frank",
    "createdDate": "2023-10-05 15:32:40",
```

```

    "multiSession": false,
    "meta": {
      "location":
"http: //soffid. 35x. lab: 8089/soffid/webservice/scim2/v1/User /432644",
      "links": {
        "roleAccounts":
"http: //soffid. 35x. lab: 8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+' franks'
+and+enabled+eq+true",
        "groupUsers":
"http: //soffid. 35x. lab: 8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+' franks' +and+d
isabled+eq+false",
        "accounts":
"http: //soffid. 35x. lab: 8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users. user
. userName+eq+' franks' ",
        "issues":
"http: //soffid. 35x. lab: 8089/soffid/webservice/scim2/v1/Issue?filter=user. userName+eq+' franks' "
,
        "effectiveGrants":
"http: //soffid. 35x. lab: 8089/soffid/webservice/scim2/v1/User /432644/effectiveGrants"
      },
      "resourceType": "User"
    },
    "modifiedByUser": "pgarcia",
    "schemas": [
      "urn: soffid: com. soffid. iam. api. User"
    ],
    "modifiedDate": "2023-10-05 15: 32: 40",
    "attributes": {},
    "id": 432644,
    "userType": "I",
    "primaryGroupDescription": "scientist",
    "primaryGroup": "scientist"
  }
]
}

```

### 3. Use the boolean operator -

#### Request

GET http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User?textFilter=fran~ - Sherwood

## Response 200 OK

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 1,
  "startIndex": 1,
  "Resources": [
    {
      "lastName": "Sinatra",
      "createdByUser": "ActiveDirectory",
      "fullName": "Frank Sinatra",
      "active": true,
      "userName": "frank",
      "mailAlias": "",
      "firstName": "Frank",
      "createdDate": "2023-09-06 13:12:54",
      "multiSession": true,
      "meta": {
        "location":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/276435",
        "links": {
          "roleAccounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+' frank' +
and+enabled+eq+true",
          "groupUsers":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+' frank' +and+di
sabled+eq+false",
          "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user
.userName+eq+' frank' ",
          "issues":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+' frank' ",
          "effectiveGrants":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/276435/effectiveGrants"
        },
        "resourceType": "User"
      }
    }
  ]
}
```

```

    },
    "modifiedByUser": "ActiveDirectory",
    "schemas": [
        "urn:soffid:com.soffid.iam.api.User"
    ],
    "modifiedDate": "2023-09-06 13:12:55",
    "attributes": {},
    "id": 276435,
    "userType": "I",
    "primaryGroupDescription": "Music",
    "primaryGroup": "Music"
}
]
}

```

## Example 5

### 1. U

#### Request

```

GET
http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User?textFilter=(firstName:aretha OR
firstName:Rosalind)
AND lastName:Franklin AND birthDate:1979-01-01

```

#### Response 200 OK

```

{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 2,
  "startIndex": 1,
  "Resources": [
    {
      "lastName": "Franklin",
      "createdByUser": "ActiveDirectory",
      "fullName": "Aretha Franklin",
      "active": true,
      "userName": "aretha",

```

```
    "mailAlias": "",
    "firstName": "Aretha",
    "createdDate": "2023-09-06 13:12:54",
    "multiSession": true,
    "meta": {
      "location":
"http: //soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/276397",
      "links": {
        "roleAccounts":
"http: //soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+'aretha'+and+enabled+eq+true",
        "groupUsers":
"http: //soffid.35x.lab:8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+'aretha'+and+disabled+eq+false",
        "accounts":
"http: //soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.userName+eq+'aretha' ",
        "issues":
"http: //soffid.35x.lab:8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+'aretha' "
      },
      "effectiveGrants":
"http: //soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/276397/effectiveGrants"
    },
    "resourceType": "User"
  },
  "modifiedByUser": "pgarcia",
  "schemas": [
    "urn:soffid:com.soffid.iam.api.User"
  ],
  "modifiedDate": "2023-10-05 16:02:40",
  "attributes": {
    "birthDate": "1979-01-01 00:00:00"
  },
  "id": 276397,
  "userType": "I",
  "primaryGroupDescription": "World",
  "primaryGroup": "world"
},
{
  "lastName": "Franklin",
```

```
"createdByUser": "ActiveDirectory",
"fullName": "Rosalind Franklin",
"active": true,
"userName": "rfranklin",
"mailAlias": "",
"firstName": "Rosalind",
"createdDate": "2023-08-08 14:26:14",
"multiSession": true,
"meta": {
  "location": "http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/2862",
  "links": {
    "roleAccounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/RoleAccount?filter=userCode+eq+'rfranklin'+and+enabled+eq+true",
    "groupUsers":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/GroupUser?filter=user+eq+'rfranklin'+and+disabled+eq+false",
    "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.userName+eq+'rfranklin'",
    "issues":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Issue?filter=user.userName+eq+'rfranklin'",
    "effectiveGrants":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/2862/effectiveGrants"
  },
  "resourceType": "User"
},
"modifiedByUser": "pgarcia",
"schemas": [
  "urn:soffid:com.soffid.iam.api.User"
],
"modifiedDate": "2023-10-05 16:03:02",
"attributes": {
  "birthDate": "1979-01-01 00:00:00"
},
"id": 2862,
"userType": "I",
"primaryGroupDescription": "scientist",
"primaryGroup": "scientist"
```

}

]

}

# Operation

## Operation

The Lucene index information is stored in files arranged in a folder structure. This folder structure is replicated in every Soffid Console and every Sync Server and also is saved in the database.

In case an instance (Docker, Kubernetes, or stand-alone) detects an inconsistency, the information will be overwritten with the database data.

When you update an object, marked as the textual index, a task will be created. The **soffid agent** will execute this task and the Sync Server will update the database tables related to the textual index.

## Folder structure

The folder structure is the following:

- `../index/<TENANT>/<SOFFID_OBJECT>`

## Example

1. Here you are the folder structure for the Soffid Console

```
root@da8136799e55:/opt/soffid/iam-console-3/index/master# ll
total 16
drwxr-x--- 4 root root 4096 Jun  2 09:58 ./
drwxr-x--- 3 root root 4096 Jun  2 08:53 ../
drwxr-x--- 2 root root 4096 Jun  2 09:58 com.soffid.iam.api.Group/
drwxr-x--- 2 root root 4096 Jun  2 10:45 com.soffid.iam.api.User/
root@da8136799e55:/opt/soffid/iam-console-3/index/master#
```

```
root@da8136799e55:/opt/soffid/iam-console-3/index/master# cd com.soffid.iam.api.Group/
root@da8136799e55:/opt/soffid/iam-console-3/index/master/com.soffid.iam.api.Group# ls
_1f.fdm  _1f.kdi      _1f_Lucene90_0.tim  _1p.fdt  _1p.kdm      _1p_Lucene90_0.tip  _1q.si  _1s.cfs
_1f.fdt  _1f.kdm      _1f_Lucene90_0.tip  _1p.fdx  _1p_Lucene90_0.doc  _1p_Lucene90_0.tmd  _1r.cfe  _1s.si
_1f.fdx  _1f_Lucene90_0.doc  _1f_Lucene90_0.tmd  _1p.fnm  _1p_Lucene90_0.dvd  _1p.si              _1r.cfs  segments_1n
_1f.fnm  _1f_Lucene90_0.dvd  _1f.si              _1p.kdd  _1p_Lucene90_0.dvm  _1q.cfe             _1r.si  write.lock
_1f.kdd  _1f_Lucene90_0.dvm  _1p.fdm             _1p.kdi  _1p_Lucene90_0.tim  _1q.cfs             _1s.cfe
root@da8136799e55:/opt/soffid/iam-console-3/index/master/com.soffid.iam.api.Group#
```

2. And the folder structure for the Sync Server

```

root@17beaa887a1e:/opt/soffid/iam-sync/index/master# ll
total 16
drwxr-xr-x 4 root root 4096 Jun  2 11:00 ./
drwxr-xr-x 3 root root 4096 Jun  2 09:39 ../
drwxr-xr-x 2 root root 4096 Jun  2 11:00 com.soffid.iam.api.Group/
drwxr-xr-x 2 root root 4096 Jun  2 10:45 com.soffid.iam.api.User/
root@17beaa887a1e:/opt/soffid/iam-sync/index/master#

```

```

root@17beaa887a1e:/opt/soffid/iam-sync/index/master/com.soffid.iam.api.Group# ls
_1f_i.liv  _1f.kdi  _1f_Lucene90_0.tip  _1p.fnm  _1p_Lucene90_0.dvm  _1q.cfs  _1s.cfs  write.lock
_1f.fdm  _1f.kdm  _1f_Lucene90_0.tmd  _1p.kdd  _1p_Lucene90_0.tim  _1q.si  _1s.si
_1f.fdt  _1f_Lucene90_0.doc  _1f.si  _1p.kdi  _1p_Lucene90_0.tip  _1r.cfe  _1t.cfe
_1f.fdx  _1f_Lucene90_0.dvd  _1p.fdm  _1p.kdm  _1p_Lucene90_0.tmd  _1r.cfs  _1t.cfs
_1f.fnm  _1f_Lucene90_0.dvm  _1p.fdt  _1p_Lucene90_0.doc  _1p.si  _1r.si  _1t.si
_1f.kdd  _1f_Lucene90_0.tim  _1p.fdx  _1p_Lucene90_0.dvd  _1q.cfe  _1s.cfe  segments_1o
root@17beaa887a1e:/opt/soffid/iam-sync/index/master/com.soffid.iam.api.Group#

```

## Database

The database tables involved:

- **SC\_LUINPA**
- **SC\_LUNIND**

## Example

### 1. The database structure

```

MariaDB [soffid_35x]> select * from SC_LUNIND;
+-----+-----+-----+-----+
| LIP_ID | LIP_NAME | LIP_TEN_ID | LIP_TIMSTA |
+-----+-----+-----+-----+
| 5378 | master/com.soffid.iam.api.User | 1 | 1685702740517 |
| 5774 | master/com.soffid.iam.api.Group | 1 | 1685703228965 |
+-----+-----+-----+-----+

```

```

MariaDB [soffid_35x]> select * from SC_LUINPA \G;
***** 1. row *****
LIP_ID: 5379
LIP_LIN_ID: 5378
LIP_NAME: _v_Lucene90_0.dvd
LIP_ORDER: 0
LIP_TEN_ID: 1
LIP_DATA: ?\Lucene90DocValuesData  Mi!k
Lucene90_0 e(Y;RooGaoYl(  m
LIP_TIMSTA: 1685697735029
***** 2. row *****
LIP_ID: 5380
LIP_LIN_ID: 5378
LIP_NAME: _v.fdm
LIP_ORDER: 0
LIP_TEN_ID: 1
LIP_DATA: ?\Lucene90FieldsIndexMeta  Mi!k
0  ?  0  6  A  0  $  (  )
LIP_TIMSTA: 1685697735025
***** 3. row *****
LIP_ID: 5381

```

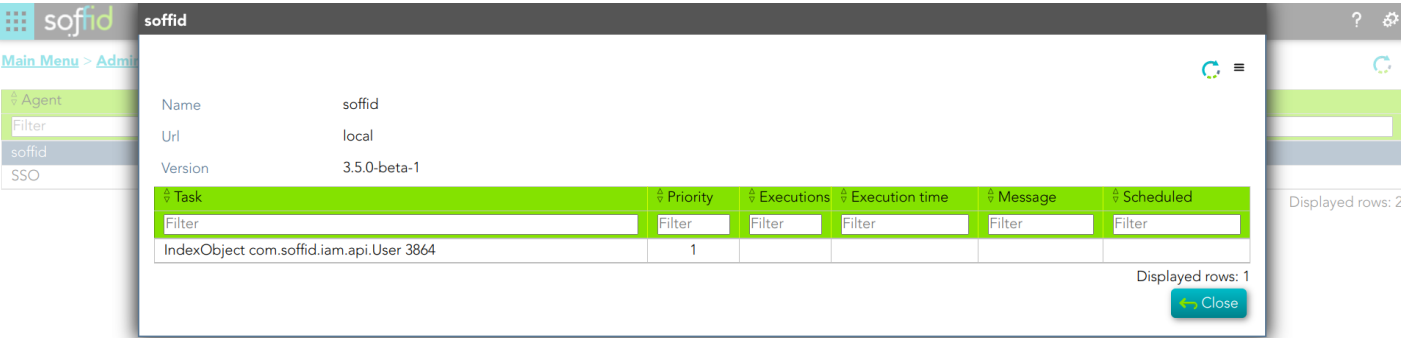
## soffid agent

You can check the soffid agent status by visiting the Sync Server monitoring page:

Main Menu > Administration > Monitoring and reporting > Sync server monitoring

# Example

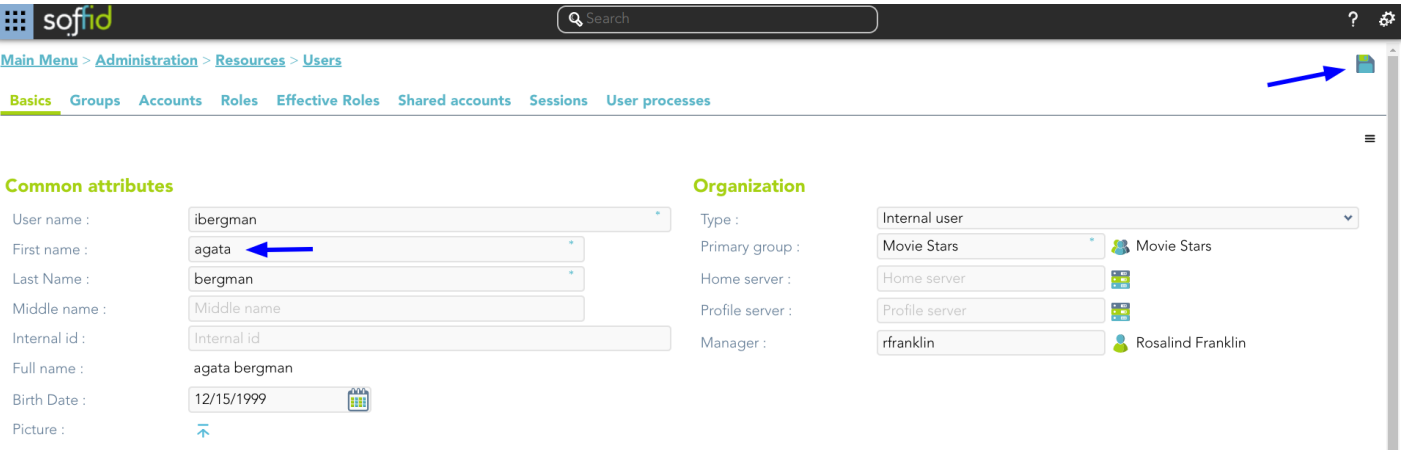
1. A soffid agent pending task:



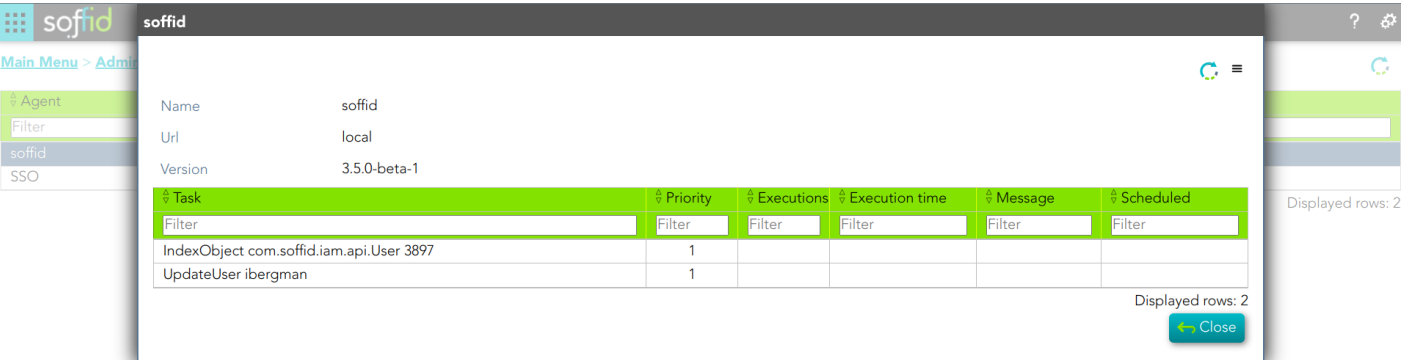
# Step-by-step

## Example 1

1. You update one user's data and save the changes.



2. New tasks are created and executed.



3. Then Sync Server indexes the updated text and places the index file.

```

root@da8136799e55:/opt/soffid/iam-console-3/index/master/com.soffid.iam.api.User# ls -al
total 108
drwxr-x--- 2 root root 4096 Jun  2 13:23 .
drwxr-x--- 4 root root 4096 Jun  2 09:58 ..
-rw-r----- 1 root root 405 Jun  2 10:42 _12.cfe
-rw-r----- 1 root root 2437 Jun  2 10:42 _12.cfs
-rw-r----- 1 root root 350 Jun  2 10:42 _12.si
-rw-r----- 1 root root 405 Jun  2 13:16 _15.cfe
-rw-r----- 1 root root 2608 Jun  2 13:16 _15.cfs
-rw-r----- 1 root root 350 Jun  2 13:16 _15.si
-rw-r----- 1 root root 67 Jun  2 09:40 _b_2.liv
-rw-r----- 1 root root 405 Jun  2 09:22 _b.cfe
-rw-r----- 1 root root 3463 Jun  2 09:22 _b.cfs
-rw-r----- 1 root root 385 Jun  2 09:22 _b.si

```

4. Then Sync Server and updates the database table SC\_LUNIND by upgrading the LIP\_TIMSTA field of the User object or by creating a new record if it did not previously exist.

```

MariaDB [soffid_35x]> select * from SC_LUNIND;
+-----+-----+-----+-----+
| LIP_ID | LIP_NAME                                | LIP_TEN_ID | LIP_TIMSTA |
+-----+-----+-----+-----+
| 5378   | master/com.soffid.iam.api.User         | 1          | 1685711801976 |
| 5784   | master/com.soffid.iam.api.Group        | 1          | 1685703660618 |
+-----+-----+-----+-----+

```

5. When the following search will be performed, the very first thing to do is check the database file. If it is necessary update the file system and finally perform the search.

## Example 2

1. The task engine mode is Read only

soffid

Search

Main Menu > Administration > Configuration > Integration engine > Smart engine settings

Task engine mode: Read only (blue arrow)

Tasks limit per transaction: 15 (If a single transaction creates more than this number of tasks, tasks will be held until Soffid administrator releases them.)

Scripting language: Javascript

Undo Confirm changes

2. You update one user's data and save the changes.

soffid

Search

Main Menu > Administration > Resources > Users

Basics Groups Accounts Roles Effective Roles Shared accounts Sessions User processes

Common attributes

User name: ibergman

First name: agata (blue arrow)

Last Name: bergman

Middle name: Middle name

Internal id: Internal id

Full name: agata bergman

Birth Date: 12/15/1999

Picture: [icon]

Organization

Type: Internal user

Primary group: Movie Stars (Movie Stars icon)

Home server: Home server (server icon)

Profile server: Profile server (server icon)

Manager: rfranklin (Rosalind Franklin icon)

### 3. A new task is created and executed

The screenshot shows the soffid web interface with a modal window for configuring a task. The modal has a header bar with the soffid logo and a 'soffid' title. Below the header, there are fields for 'Name' (soffid), 'Url' (local), and 'Version' (3.5.0-beta-1). A table below these fields lists tasks with columns for Task, Priority, Executions, Execution time, Message, and Scheduled. The table has a filter bar and one row of data. A 'Close' button is at the bottom right of the modal.

Task	Priority	Executions	Execution time	Message	Scheduled
IndexObject com soffid iam api User 3897	1				

### 4. Then Sync Server indexes the updated text and places the index file.

### 5. Then Sync Server and updates the database table SC\_LUNIND by upgrading the LIP\_TIMSTA field of the User object or by creating a new record if it did not previously exist.

### 6. When the following search will be performed, the very first thing to do is check the database file. If it is necessary update the file system and finally perform the search.

# Lucene - Query Parser Syntax

- Overview
- Terms
- Fields
- Term Modifiers
  - Wildcard Searches
  - Regular expression Searches
  - Fuzzy Searches
  - Proximity Searches
  - Range Searches
  - Boosting a Term
- Boolean Operators
  - OR
  - AND
  - +
  - NOT
  - -
- Grouping
- Field Grouping
- Escaping Special Characters

## Overview

Although Lucene provides the ability to create your own queries through its API, it also provides a rich query language through the Query Parser, a lexer which interprets a string into a Lucene Query using JavaCC.

Generally, the query parser syntax may change from release to release. This page describes the syntax as of the current release. If you are using a different version of Lucene, please consult the copy of docs/queryparsersyntax.html that was distributed with the version you are using.

Before choosing to use the provided Query Parser, please consider the following:

1. If you are programmatically generating a query string and then parsing it with the query parser then you should seriously consider building your queries directly with the query API. In other words, the query parser is designed for human-entered text, not for program-generated text.
2. Untokenized fields are best added directly to queries, and not through the query parser. If a field's values are generated programmatically by the application, then so should query clauses for this field. An analyzer, which the query parser uses, is designed to convert human-entered text to terms. Program-generated values, like dates, keywords, etc., should be consistently program-generated.
3. In a query form, fields which are general text should use the query parser. All others, such as date ranges, keywords, etc. are better added directly through the query API. A field with a limit set of values, that can be specified with a pull-down menu should not be added to a query string which is subsequently parsed, but rather added as a TermQuery clause.

[https://lucene.apache.org/core/9\\_6\\_0/queryparser/org/apache/lucene/queryparser/classic/package-summary.html#Overview](https://lucene.apache.org/core/9_6_0/queryparser/org/apache/lucene/queryparser/classic/package-summary.html#Overview)

# Terms

A query is broken up into terms and operators. There are two types of terms: Single Terms and Phrases.

A Single Term is a single word such as "test" or "hello".

A Phrase is a group of words surrounded by double quotes such as "hello dolly".

Multiple terms can be combined together with Boolean operators to form a more complex query (see below).

Note: The analyzer used to create the index will be used on the terms and phrases in the query string. So it is important to choose an analyzer that will not interfere with the terms used in the query string.

# Fields

Lucene supports fielded data. When performing a search you can either specify a field, or use the default field. The field names and default field is implementation specific.

You can search any field by typing the field name followed by a colon ":" and then the term you are looking for.

As an example, let's assume a Lucene index contains two fields, title and text and text is the default field. If you want to find the document entitled "The Right Way" which contains the text "don't go this way", you can enter:

```
title: "The Right Way" AND text: go
```

or

```
title: "The Right Way" AND go
```

Since text is the default field, the field indicator is not required.

Note: The field is only valid for the term that it directly precedes, so the query

```
title: The Right Way
```

Will only find "The" in the title field. It will find "Right" and "Way" in the default field (in this case the text field).

# Term Modifiers

Lucene supports modifying query terms to provide a wide range of searching options.

## Wildcard Searches

Lucene supports single and multiple character wildcard searches within single terms (not within phrase queries).

To perform a single character wildcard search use the "?" symbol.

To perform a multiple character wildcard search use the "\*" symbol.

The single character wildcard search looks for terms that match that with the single character replaced. For example, to search for "text" or "test" you can use the search:

```
te?t
```

Multiple character wildcard searches looks for 0 or more characters. For example, to search for test, tests or tester, you can use the search:

```
test*
```

You can also use the wildcard searches in the middle of a term.

```
te*t
```

Note: You cannot use a \* or ? symbol as the first character of a search.

## Regular Expression Searches

Lucene supports regular expression searches matching a pattern between forward slashes "/". The syntax may change across releases, but the current supported syntax is documented in the [RegExp](#) class. For example to find documents containing "moat" or "boat":

```
/[ mb] oat/
```

## Fuzzy Searches

Lucene supports fuzzy searches based on Damerau-Levenshtein Distance. To do a fuzzy search use the tilde, "~", symbol at the end of a Single word Term. For example to search for a term similar in spelling to "roam" use the fuzzy search:

```
roam~
```

This search will find terms like foam and roams.

An additional (optional) parameter can specify the maximum number of edits allowed. The value is between 0 and 2, For example:

```
roam~1
```

The default that is used if the parameter is not given is 2 edit distances.

Previously, a floating point value was allowed here. This syntax is considered deprecated and will be removed in Lucene 5.0

# Proximity Searches

Lucene supports finding words are a within a specific distance away. To do a proximity search use the tilde, "~", symbol at the end of a Phrase. For example to search for a "apache" and "jakarta" within 10 words of each other in a document use the search:

```
"jakarta apache"~10
```

## Range Searches

Range Queries allow one to match documents whose field(s) values are between the lower and upper bound specified by the Range Query. Range Queries can be inclusive or exclusive of the upper and lower bounds. Sorting is done lexicographically.

```
mod_date: [ 20020101 TO 20030101]
```

This will find documents whose mod\_date fields have values between 20020101 and 20030101, inclusive. Note that Range Queries are not reserved for date fields. You could also use range queries with non-date fields:

```
title:{Aida TO Carmen}
```

This will find all documents whose titles are between Aida and Carmen, but not including Aida and Carmen.

Inclusive range queries are denoted by square brackets. Exclusive range queries are denoted by curly brackets.

## Boosting a Term

Lucene provides the relevance level of matching documents based on the terms found. To boost a term use the caret, "^", symbol with a boost factor (a number) at the end of the term you are searching. The higher the boost factor, the more relevant the term will be.

Boosting allows you to control the relevance of a document by boosting its term. For example, if you are searching for

```
jakarta apache
```

and you want the term "jakarta" to be more relevant boost it using the ^ symbol along with the boost factor next to the term. You would type:

```
jakarta^4 apache
```

This will make documents with the term jakarta appear more relevant. You can also boost Phrase Terms as in the example:

```
"jakarta apache" ^4 "Apache Lucene"
```

By default, the boost factor is 1. Although the boost factor must be positive, it can be less than 1 (e.g. 0.2)

# Boolean Operators

Boolean operators allow terms to be combined through logic operators. Lucene supports AND, "+", OR, NOT and "-" as Boolean operators(Note: Boolean operators must be ALL CAPS).

## OR

The OR operator is the default conjunction operator. This means that if there is no Boolean operator between two terms, the OR operator is used. The OR operator links two terms and finds a matching document if either of the terms exist in a document. This is equivalent to a union using sets. The symbol || can be used in place of the word OR.

To search for documents that contain either "jakarta apache" or just "jakarta" use the query:

```
"jakarta apache" jakarta
```

or

```
"jakarta apache" OR jakarta
```

## AND

The AND operator matches documents where both terms exist anywhere in the text of a single document. This is equivalent to an intersection using sets. The symbol && can be used in place of the word AND.

To search for documents that contain "jakarta apache" and "Apache Lucene" use the query:

```
"jakarta apache" AND "Apache Lucene"
```

## +

The "+" or required operator requires that the term after the "+" symbol exist somewhere in a the field of a single document.

To search for documents that must contain "jakarta" and may contain "lucene" use the query:

```
+jakarta lucene
```

## NOT

The NOT operator excludes documents that contain the term after NOT. This is equivalent to a difference using sets. The symbol ! can be used in place of the word NOT.

To search for documents that contain "jakarta apache" but not "Apache Lucene" use the query:

```
"jakarta apache" NOT "Apache Lucene"
```

Note: The NOT operator cannot be used with just one term. For example, the following search will return no results:

```
NOT "jakarta apache"
```

## -

The "-" or prohibit operator excludes documents that contain the term after the "-" symbol.

To search for documents that contain "jakarta apache" but not "Apache Lucene" use the query:

```
"jakarta apache" -"Apache Lucene"
```

# Grouping

Lucene supports using parentheses to group clauses to form sub queries. This can be very useful if you want to control the boolean logic for a query.

To search for either "jakarta" or "apache" and "website" use the query:

```
(jakarta OR apache) AND website
```

This eliminates any confusion and makes sure you that website must exist and either term jakarta or apache may exist.

## Field Grouping

Lucene supports using parentheses to group multiple clauses to a single field.

To search for a title that contains both the word "return" and the phrase "pink panther" use the query:

```
title:(+return +"pink panther")
```

## Escaping Special Characters

Lucene supports escaping special characters that are part of the query syntax. The current list special characters are

```
+ - && || ! ( ) { } [ ] ^ " ~ * ? : \ /
```

To escape these character use the \ before the character. For example to search for (1+1):2 use the query:

```
\(1\+1\)\:2
```

### Interface Summary

Interface	Description
<a href="#">QueryParserConstants</a>	Token literal values and constants.

### Class Summary

Class	Description
<a href="#">MultiFieldQueryParser</a>	A QueryParser which constructs queries to search multiple fields.

<b>QueryParser</b>	This class is generated by JavaCC.
<b>QueryParserBase</b>	This class is overridden by QueryParser in QueryParser.jj and acts to separate the majority of the Java code from the .jj grammar file.
<b>QueryParserTokenManager</b>	Token Manager.
<b>Token</b>	Describes the input token stream.

Enum Summary

<b>Enum</b>	<b>Description</b>
<b>QueryParser.Operator</b>	The default operator for parsing queries.

Exception Summary

<b>Exception</b>	<b>Description</b>
<b>ParseException</b>	This exception is thrown when parse errors are encountered.

Error Summary

<b>Error</b>	<b>Description</b>
<b>TokenMgrError</b>	Token Manager Error.