# 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.

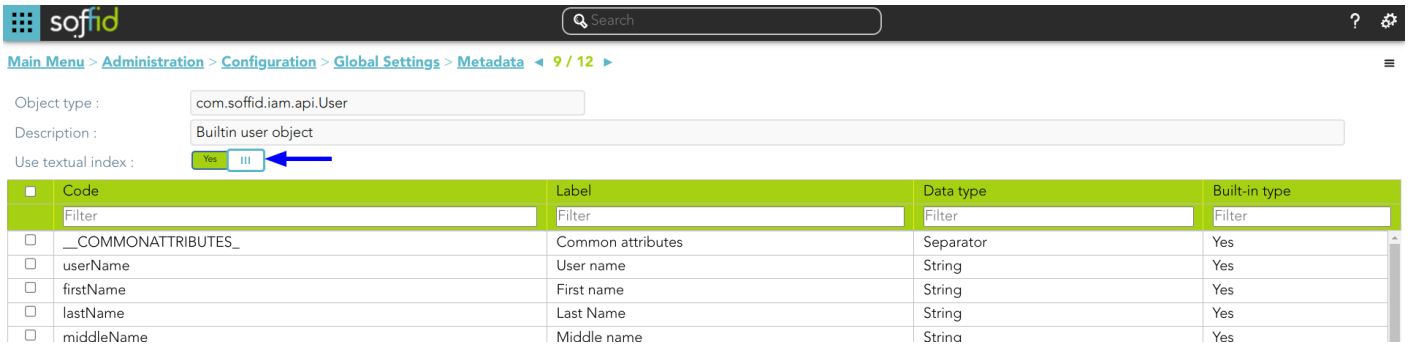
# Index  configuration

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.




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

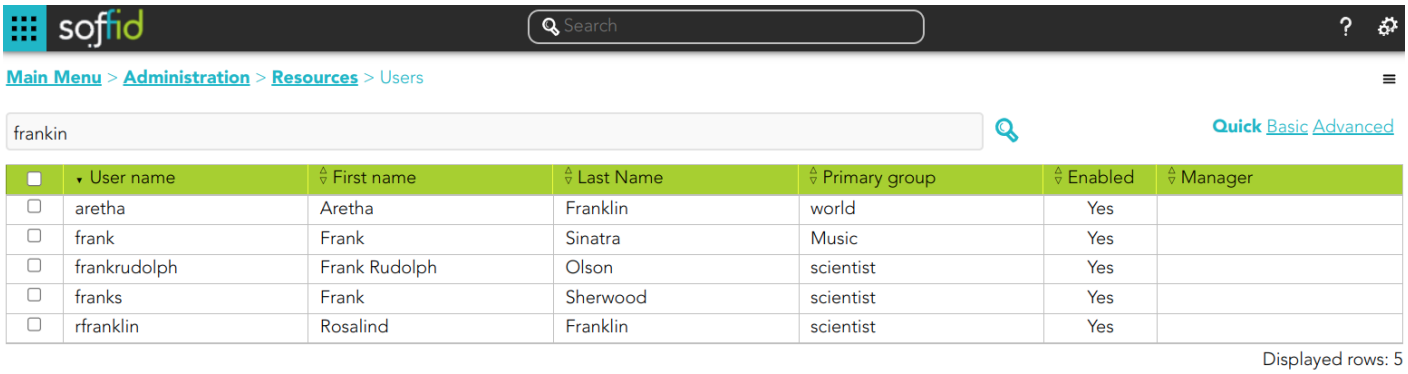



# How does the user interface search work?

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

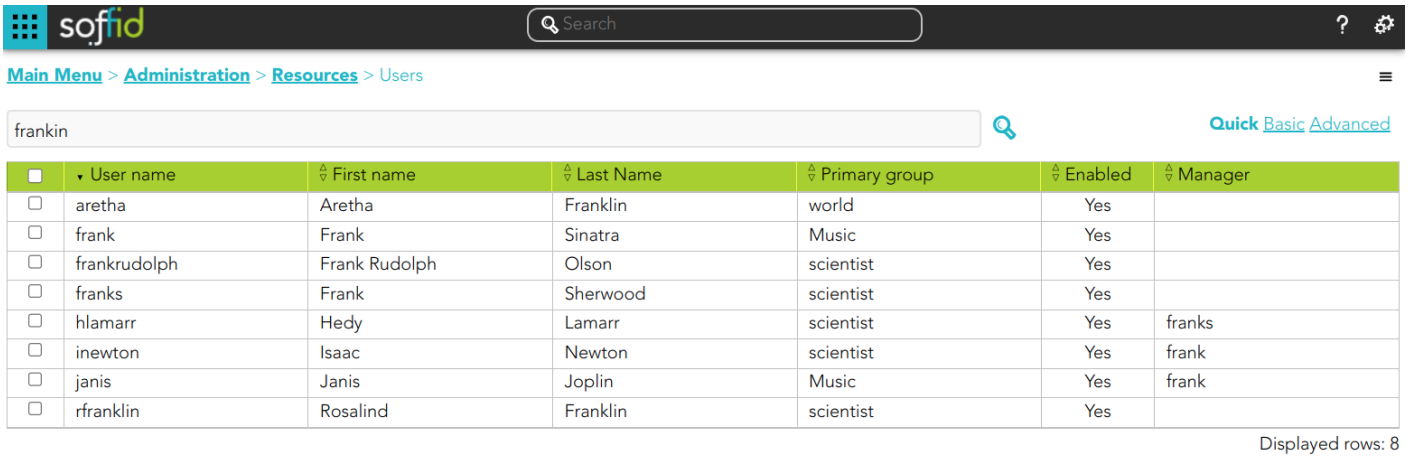
**1.** If you search for users using the text "*frankin*", then Soffid will display all the users whose userName, firstName, lastName, or middleName match, to some degree, with the typed text following the textual index rules.




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

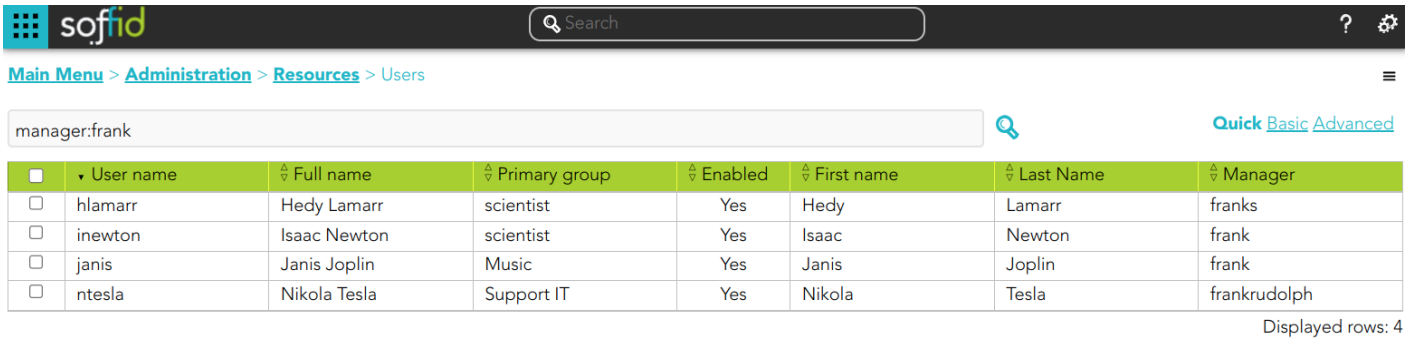



**3.** And search for "*frankin*", then Soffid will display all the users whose userName, firstName, lastName, middleName, or manager match with the typed text following the textual index rules.




# Example 2

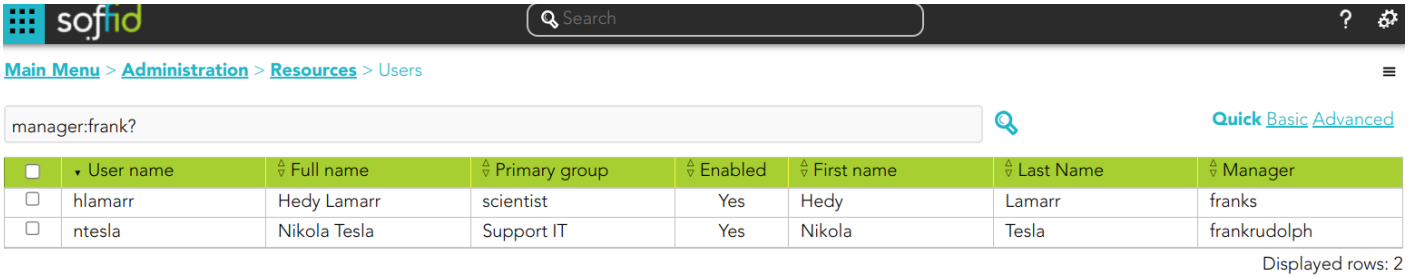
**1.** If you search for users using the text "manager:frank" Soffid will display all users whose manager matches the text "frank".




| User name | Full name | Primary group | Enabled | First name | Last Name | Manager |
|---|---|---|---|---|---|---|
| hlamarr | Hedy Lamarr | scientist | Yes | Hedy | Lamarr | franks |
| inewton | Isaac Newton | scientist | Yes | Isaac | Newton | frank |
| janis | Janis Joplin | Music | Yes | Janis | Joplin | frank |
| ntesla | Nikola Tesla | Support IT | Yes | Nikola | Tesla | frankrudolph |

Displayed rows: 4

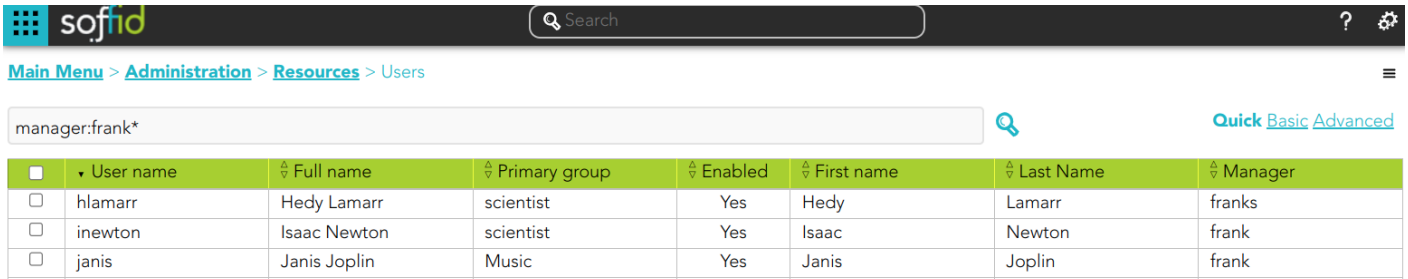
Notice the difference by searching "manager:frank?":




| User name | Full name | Primary group | Enabled | First name | Last Name | Manager |
|---|---|---|---|---|---|---|
| hlamarr | Hedy Lamarr | scientist | Yes | Hedy | Lamarr | franks |
| ntesla | Nikola Tesla | Support IT | Yes | Nikola | Tesla | frankrudolph |

Displayed rows: 2

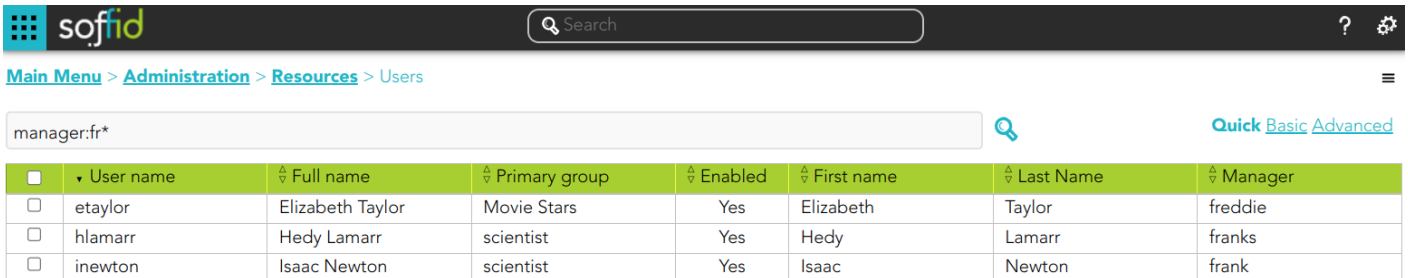
And by searching "manager:frank*":




| User name | Full name | Primary group | Enabled | First name | Last Name | Manager |
|---|---|---|---|---|---|---|
| hlamarr | Hedy Lamarr | scientist | Yes | Hedy | Lamarr | franks |
| inewton | Isaac Newton | scientist | Yes | Isaac | Newton | frank |
| janis | Janis Joplin | Music | Yes | Janis | Joplin | frank |
| ntesla | Nikola Tesla | Support IT | Yes | Nikola | Tesla | frankrudolph |

Displayed rows: 4

And also by searching "manager:fr*"

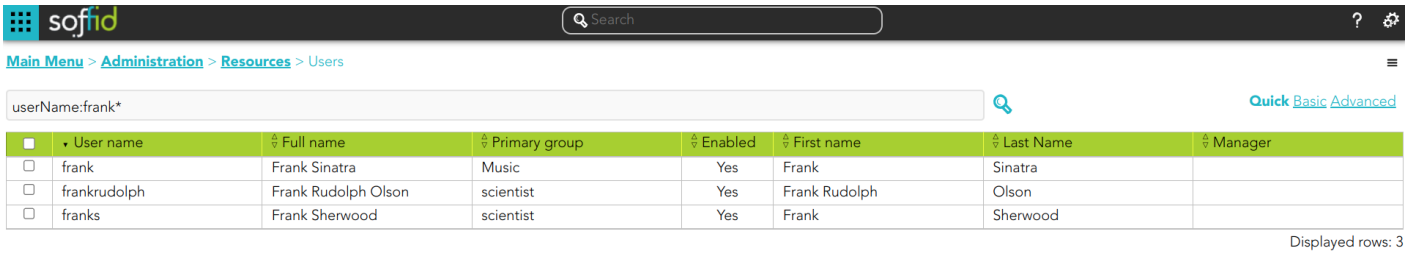



| User name | Full name | Primary group | Enabled | First name | Last Name | Manager |
|---|---|---|---|---|---|---|
| etaylor | Elizabeth Taylor | Movie Stars | Yes | Elizabeth | Taylor | freddie |
| hlamarr | Hedy Lamarr | scientist | Yes | Hedy | Lamarr | franks |
| inewton | Isaac Newton | scientist | Yes | Isaac | Newton | frank |
| janis | Janis Joplin | Music | Yes | Janis | Joplin | frank |
| ntesla | Nikola Tesla | Support IT | Yes | Nikola | Tesla | frankrudolph |

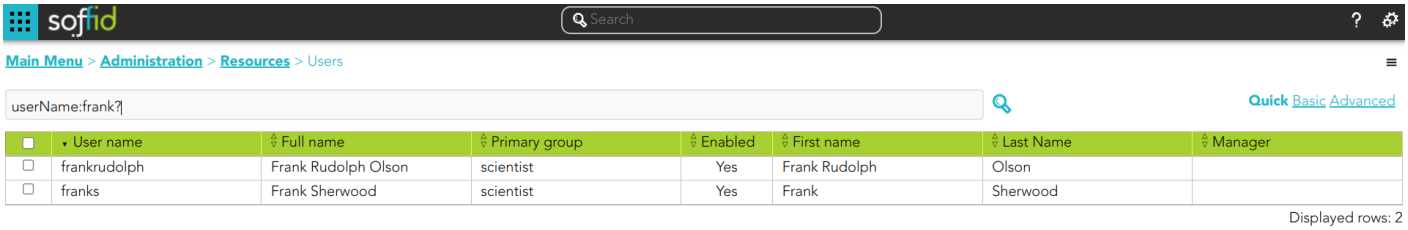
Displayed rows: 5

# 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.

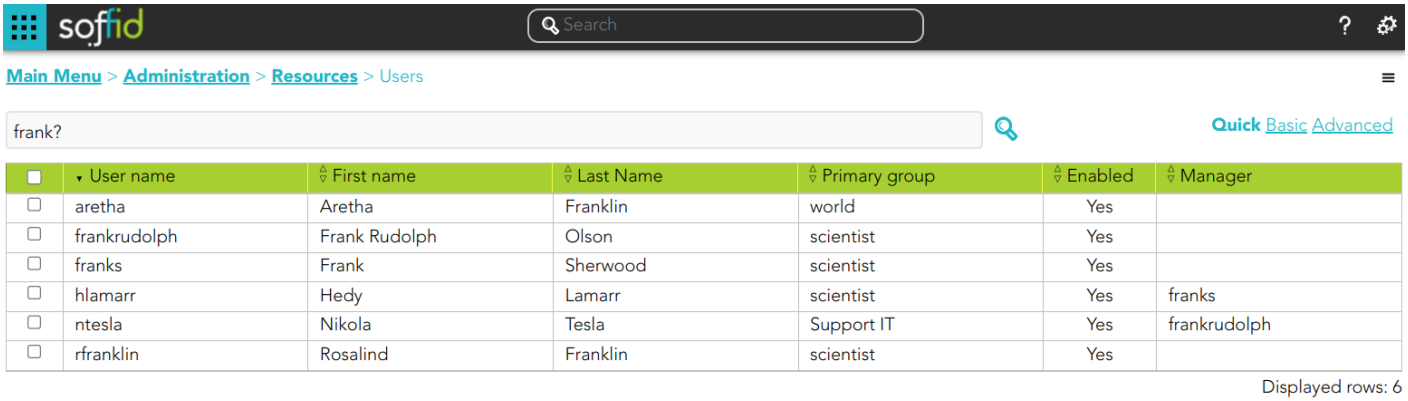



Notice the difference by searching the text "userName:frank?":

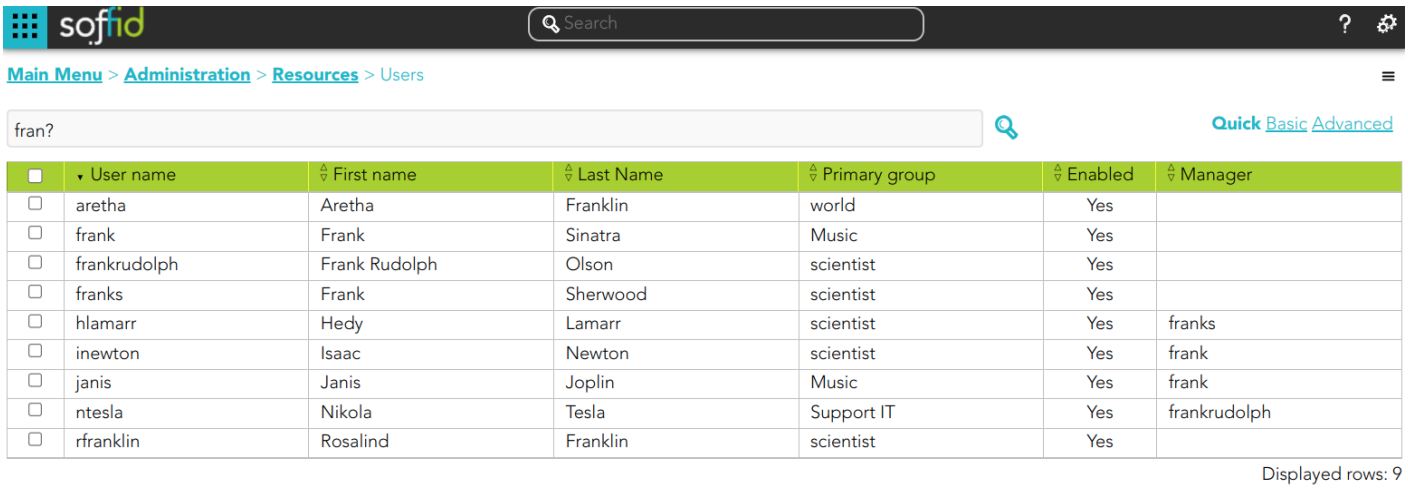



# Example 4

**1.** 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.




Notice the difference by searching "fran?"



# 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:

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

| | |
|---|---|
| **Fuzzy Searches** | To do a fuzzy search use the tilde, "~", symbol at the end of a Single word Term<br>Soffid Console <= 3.4 version<br>~0.8: stricter search<br>~0.1: more lax search<br>Soffid Console > 3.4 version<br>An additional (optional) parameter can specify the maximum number of edits allowed. The value is between 0 and 2. |
| **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 |
| **Boosting a Term** | 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. |

# Boolean Operators

| | |
|---|---|
| **OR** | 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 |
| **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 "+" or required operator requires that the term after the "+" symbol exist somewhere in a the field of a single document. |
| **NOT** | The NOT operator excludes documents that contain the term after NOT. This is equivalent to a difference using sets. |
| **-** | 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.user.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+f
alse",
               "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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+disabled+eq+fal
se",
               "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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+f
alse",
               "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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+fal
se",
                "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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+f
alse",
                "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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/webservice/scim2/v1/GroupUser?filter=user+eq+'frank'+and+disabled+eq+fal
se",
               "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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/4AAQSkZJRgABAQAAAQABAAD/2wCEAAoHCBYWFRgVFRYYGRgaGhkYGhgaGBoYGhoYHBkaGhgZGhgcIS4lHB4
rIRgYJjgmKy8xNTU1GiQ7QDs0Py40NTEBDAwMBgYGEAYGEDEdFh0xMTExMTExMTExMTExMTExMTExMTExM
TExMTExMTExMTExMTExMTExMf/AABEIAKgBKwMBIgACEQEDEQH/xAAcAAABBQEBAQAAAAAAAAAAA
ADAQIEBQYABwj/xABFEAACAQIDBAcECAQFAgcBAAABAgADEQQSIQUxQVEGEyJhcYGRMkKhsQcjUnKSwdHwFGKCo
iQzY7LhU3NDZIOTo8LxFf/EABQBAQAAAAAAAAAAAAAAAAD/xAAUEQEAAAAAAAAAAAAAAAAAAA/9oADAM
BAAIRAxEAPwDeqIVZwEcBAQQiCIBHqIDgItogjwICWiqIto4CAqiZn6ScnNnwDn7yAcp/3TUKJX9cNnWgTi44n8zz3D
+k+4dtd79iR/4dtd79iR/4dtd79iR/
```

Ne+Q8Hh6yEAG/jrL6i1S3bbTugYXaXQ1kVnDLZbm2rE6nyGhHpM9sZiuJS2mtp6ltrFKtJgd1tTPLqFT65SN94HsuGfsA
nu+Ufhgt75QfLjluz+1RvxtHYapAj4/C02a5WxBBvoDod17eMqcZsNajEq+W9iylQwJGgOp0PfNTXoh1uN8rv4fnp++c
BcFhkw6ZFYm9yxvqzHS5t4W8pG/gUe5YAchbcJMTBrvzEwopiBUrg8h/4lZtB7dkbyQB56TQ4phaZjFDNUQd9/Td8bQI
229jI9RKiXznsuP5V7Kn0EvcEgzKo1ChU89Xb/avrCVqJVLKbE72tckHSwHG5Mk7OweRdd9txNyL6knmx4+AECTOvH
GNIgcTEvEvHgQODTs04iNJgKRBOsfnjWeBHcRto52gs8DUARwjVMdA4CLEvEJgLmiXnRsB0VYwQloDhDI0AIRTAxP0o
7PNqWJUaLenUI4Am6E+eYeYmDw9UBh37/Oe5VqSujI6hkYZWU6gg8JgdrdBEo5qwxGSiNSGQu4/lQgjOeAvr4wM1s
XH9VVKndfT4zd0cfmE80xrIxFSncKb2DEFhY21K6E7t3OXuwto7lJ8PlA2Ard3nO6y5sYGlVBj0S8Cg6Z4oCnkHEzE7Io5
6qjvvL/ppibsEXgLmM6G4G5LnU7v1gej7Jp2pnw/f78JGXeRLHZOnZ5iQ8ahDHxgHwb8L/v8AZkhwDvEqkdl13jjJlDFBh
A51I3QLsRvk1jIWJ4wKjG4nWU+GrMa6sqlramys1hY6kKCQOEtMYmhJ/fKO6LJ2qrfcF/xE/lAs8Iju2dxlA0VbWPiV90b9
DrrraTTFtOgDjSsIREgCyx6xWEEzQFcwbGNZ4wtAdeNaNBjjAG8FDNBWgaVlVYNBCCAtowwgEaywEnRLR0DgI4CII4
QHARwWcsKggKizyb6SuknWv1CEdWhsbnss/E2Gr24DcOPdrun3SL+HpGjTI611Nze2SnuJJ4E7v+bTxqtUN81wt9c7
DtH7icB+7wJWz3GQ7rF2IsLC4VLi3C418pJw1TIwPCQcC+dKiBizLaqtxY2HZf4FT5Q9B8ynmNfKBv8AAV8wUjWWO0t
opRpFjvI0HMzObBq2S54CUO2tqmq1+A3DugQ8XiC7Mx1JMteiWOCVCjGwOovzmezzkuzALcnhaB7LhMcpbRh6ynx/
SJDV6tFeq17HIhZVPe268pNh7EZSHrs1j7oNvWbzB00RQEUKOAAAgOw1AhbuLE7xy7ryn2jQek3WJdk3uo3gfaHdLt6
0jjFKDraBFwe0VdbqZ1epbUyi21hjSb+IoeyT9YnAX94Dh3jvklMVnTN3XgLjDmU21PLvgMBtE4cMhpliTc62bcBfLa5Xv
GmshPisis53LZv7h+svMLthHUHQ8bwOo9JUb3G13ZSGv929s3leSqe26De/bhqDv46jSGQYeotmRCDvuo1MhbU6NK
wLJdWt97dqAQfbHc3kRAnjF0yLiolueYTqeJRjZXRjyDAn0BmAqsyOUcEMBmsLtdQdXS+rqOK+2ttcwgXRDe9gGs5INl
dRufMpF1ud+4ccu6B6SwgnWYXCbfq4dsrl3UvlyN1aJTW1xZyBw58LTZbO2lTrpnpsDb2gGBKnkcpIgPYRhEKyxloAwJ
0JlnZYAmg80M6wVoGlURyiDVo9WgEUzmiRYDZ1o4LHBYAwIRRHBI9afpA5Fld0g22mFplmILkHIv8A9jyUSDtXpfQpB
shFRl35TdR4kanyE8l25t56zl3bUm4JF2HIll7IOV9YAdtY96jmo7asbl21uearvbkCQAOHOVYJOqIW/nqaj46R5ze0EAv7
9Q3Y9/a/SMrFTrUqM55LrbuudPSBKwGOenVR3emQDZl7PsEWYaC24mWO1MAcPUsuqPc025rxUnmNPKxmeDJuWm
W8ST8BN30bdMVhuoqqQUsNfaAt2HUnlqPLvgR9m1s1F1XeFOg4ixmeqLlAvLV6L4StkfVT7LcGXn+o4SHj6gZrj0gRK
KK28i3jNVsJcOhzM6X7yJjqyEbxLXZWyKdXXPc8V9lh+vH1gennFUKlO6upAtexBt6RKG0aIATrVuObCZzBdEcPa+erry
cDcByGssE6MYNd6FiL+3UbXfwB8IFliNoU1FzUQD74/WQ6rmqv1Op0IY6KdeHP5QB2bhUYFKNPNwsoNvM6k+Mt6Lh
BfjAr8ArglKo9rnuInbSpBFsDpu/WO2lWLLpvGt5SYmpUqsiXsSQo8WNh5m8CJjaud0QA5ew5W2joHAbXkCJXdSp60K/
VFKpSmST1be2VVrjsaL7V7a6jjLPpIow2ITLotHEVKZ/7bpTqW/C7yp2jh7L1V7t1qlydNVZqRPhYo3/qXgEo4qrTdHd2K
0zldCMrl/DOOIPBhoZ6XsnaiuoN73nmW0axavinDG1IBRfVWBdaZpsOKntaeYtJOysU9FkKezUynIWGZMwzDxUjc3rY7
w3nSHYi10upKPcMjrvVx7LDv+YnntXOjMjjqyHGew0pVW0TEU/9J9zLuveeobLrFls2t5lOm2FAIrBc2VWWon/Uot/mKO
ZA7Y+6YGXF7dWUANygQ3KrUHabDk7zTcdpD7p3RNlY16Lq9JaRQDsgkU6zpftIdbOym6kHiNN8FVXQqXuAEps/NG7
WExPipshPlyLinS4esjFHZg6Icpp11stS3DWwax/KB6xhMStRFqJqrC4594I4EboRhMj0Hx62egHzrq9Jjo2XQOjD7amxP
3r8Zq2eAhMS8S8SBzmBzwjiDtA0Aj1M4COtAVWhVMCBCKIBRHARiiRNq7XTDrdu05BKotyzd9hqB3wD7R2hToJnqNb
kPeY8gPz3Cefbb6Q1MV2EJCfYp1aYc/eBBznuB8oLaWJq1nNR1qDk3ULUVV5atoB4CQEqVGPZGGxAPuZFV/JLIw8rwK
DGk7hUL2YDIyZKwN9ymx1vyPlK50ZfdWn3sbufz+E021aYrLlCZKo0FOr7R5CnVNmBvuV9OR4SrOAFuxg3J+1UZiPS
yiBRsyX1Lufwj8zHpm9ykB3kFj6tJGIp1UOuSn3Bk09CZFc39qrfuAY/OwgPdXPtOq92YD4LJmwdonD1lfrMynsuNTdTvI
7xv8u+Vl6Y4OfML+s4svBP7mMD1fauAXE0SgIv7dN+Aa2mv2Tx9eE87VGRzTcFWU5WB3gzRdCtsAr/AA7XBXVNd6
8VvzHyPdLHpPsXrl62mPrUH/uKPdP8w4engFDiMASugvpKhsM6nshr915ouju0Vaytv3WPxmjQICCANf3+/CBisM2ONs
gqW3bv1Es8NsrHsczs4B3gkCbZKigX+EmJXUiBQbP2a6bxr6n1lg1Mrv3yfUrASqx+LAHfuA4k8LCBWba2itJc7nTlxJ5
DmZA6C4tsTjKebgxcjkEBK/HLKfpc5Y6nQbhy5zQ/Q3hO3WrHgqovmczH4CAP6SFHW4pbah8NVH9aGkf9qiE2bsF6/
wBY3YLomZSAe2qKGPgSin+kS8x2CTEYypiASUKU0A91shJzeFybeF5Z136pVYaWdAfBjl+ZEDzTpXQqUi6OEyuVLMi
hCxW5UtbedTlH8cGxKuoIRVRFB3gKoFj33BM3PTvCipTzDhr+v5TzzDUi7qote/tW1sPnA9d2HUBUGQemastPrE1ZCH

A+0FN3XzXMPOSdh0cqAdwndKCwoO6i7IpcDmV7VvO0Dzg0kVsl70xZM3PC4jWm39Dnfwgu0wdH9tlI3X/xOGNhod
5en63hCiEZfcQ5Cf8AyuJ7VNv6HIgqrsCKje2pV3H+rQcUq/4kZWgM2btIJUTEU1C9pespjQB7WLIPsuuYW4G3dPUetD
AMpupAIPMEXE8uxRp1M9RRlKXSqqjUoHHV107xZcwm56J4o1MMoNsyMyG27TVSO4gi0C2Dzs84rGEQH5oy8WMtA
0itHgwSiFUQHrDoINU4zGdIek3WXo4c3QGzuC/a5gdWpIXzF/DeFztjpMlO6Uu2/wBoK7IvgUUhjMJjHaqxLujltSr1alIn8Y
RYF6ii1+qB/mOLuPMyRQqZtFBb/s1s/wD8Fa5bwgREwrUzmFDEp/NScOO7ULb4yQtRa182WsRvV16rEabyrrcOd+hJ
PdEZFQK9xlL5M6B8PURt5zpqmnd6wjJUc5Hb/E02z0yQt6iWuAHHtEAZhqQdYA8QV6tTmaphnOVs4+sovpoDztqOB
A3XlXicJTRnTE1atwQaZUB1dCLhrsRv/MS4p4hAGqZPqqnYxFIb0YnR0HAX1XkbjlB1kdCEV1z01vSqEArUwzm7C5BtY
dsb9zCBncSuEX2RWfvZkXX+mV75T7FEkcNWf8psHbEnQYnDaaAhqY9Oxf4SsxeHrn28ZT8qxPwUAQKEdYNyZf6Lf7
o1jV4sB/Uo+XhCYvCoNTXVj3Bm+N5DyJ9o+SwDU69VHV1cZlII7SnX9P1nqGx9pJWpo67yO0L+y3vCeTOE4FvSXHR
naYo1At+w5AJ3Wf3Tv47vTlA1u2+jpZzXw1g+903BjxZTuDcxuO/TjQrtipSbI6spG9WFj6H5zbUsRcSPjXUizqGHJgGHx
gRMN0ipMmrWPIx9Hb6D3x6yH/8AzqbGwo0/wLNBszohhaq2enlb7SGx9LW9RArK/SBdynOeAGvryjMCjM2dzdzcADc
o7uZ75dYno3h6JP1yIBpeoMgNt4DAENbQHdaRa+TlWoulWxCk03R+0QSq2U3zEA2HceUCsw1Km1Zmex6u1g1rDj
mN+QmgwLtUzCklkIKs9ygdTvAG8i3Ezz7Z2MV8QrkgrmtI4HS4J5ier7LqLYeEBlNcgHYJUb8upHlxHhrIfSHFKcJWdWB
C0y6nvWzD4iX7rxEwfTqkiISzMqVDcqpsrOpv6nTTja8Cg2p0g69FppfddzyEi7FT64abpV0WBHYBuxsANSeQ75othbJc
1O2xVtDlWxIHedwgegbLY2F5Kx9MMpHAjXzkbD0coGpPiZlfVbwPLsVhVpv1TaJrh3v/ANGsSabX5JUVvK0hKC5Abew
Gcfz64asfU03mk6V0Q532zjqSeRftU2Pg6IP6zM2WLHNuNTK/eDUU0qg7stVEPnAFg665ErFRmpDqcQm7rKT9gP4i+
U94Uy96A1slSpSvcG4/qT2GH3lLfglFtHDhLOvsYijUvbXtg3YfiVJJ2VieorLWJy5KjUqwIJ3lijEDXUZlvwtxgemMIwiSaRV1
V0IZWAII1BB3WIiMkCPaDtJBgvOBoEWFd1RS7sFUbyTYTp0DC9Iekj1wUpgrR5nKvWcLl3ITL/KA/eAZQI/86eeKcHw7C
hR6Tp0Agruuod7aC9PGo1rm2qtra54+sI6Z2am7KtVWVqfWU6ahxbRS677k6Xup011iToBqdRsz5UIc/wCdhXvapze
mTx423jeMwjKioEUhyaOb6up/4mHqb8j21tfXTxHI9OgdUZyzOFXrkU9fT9yvSPtOoGhuLE27mG6MWmjotLP2HzNh6j
WBR/foudwBNvPKdxnToFfhKN1KnCO7U2NNyr1PbXmoBANiDbviVqCa5sFWHi9QfNPGLOgVmKRPdwr373qH4BRKm
op4Ubd1nPznToAWY/YH4W/WNKnflsNxGtjOnQNv0Y2pnQKx7SaG/EcD+veJpKeEaoQBOnQNJs3YoWxI1/f6SZtnGU8J
h3rsAci3VLhc77kQHvNok6B5JT2saH+ISun8TUqVC6OjMtMPmcvncZb3IAABOp7xM2+LZ83ZA7CltWIZlv23Gt3OY2Jt
lvpaLOgDw+Is2lgBusLcbi/E8tZ6p0a2mHRTfXd5zp0DV0sRKjpdsX+ModWrhGDBwxGYXAIsRcEXvvE6dAotl9HqeDQ
sVNSqB7ZsLE7gg90X85Z7D2aEGdtXY3J8Yk6BdVGAtIuKJtp8506BjNvIztkU+32A2mj5WZCP60UTO583aC2LXcDgpq
A51/pr0x+KLOgBwo6xBTJ0FSoi62s1VCafkXQjzkps9UU61NVZ3BpVUYLlNSmAwJDGxLIN3MGdOgazoNtEWfDFSjKSw
pNe6X9olTqUvrY6i54Wmqczp0ATiAnToH//2Q=="
        },
        "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+fal
se",
               "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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+f
alse",
               "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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+enable
d+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.user.userName+e
q+'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+f
alse",
                "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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+disabled+eq+fal
se",
                "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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+f
alse",
                "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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+disabled+eq+fal
se",
               "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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/4AAQSkZJRgABAQAAAQABAAD/2wCEAAoHCBYWFRgVFRYYGRgaGhkYGhgaGBoYGhoYHBkaGhgZGhgcIS4lHB4
rIRgYJjgmKy8xNTU1GiQ7QDs0Py40NTEBDAwMBgYGEAYGEDEdFh0xMTExMTExMTExMTExMTExMTExMTExM
TExMTExMTExMTExMTExMTExMf/AABEIAKgBKwMBIgACEQEDEQH/xAAcAAABBQEBAQAAAAAAAAAAA
ADAQIEBQYABwj/xABFEAACAQIDBAcECAQFAgcBAAABAgADEQQSIQUxQVEGEyJhcYGRMkKhsQcjUnKSwdHwFGKCo
iQzY7LhU3NDZIOTo8LxFf/EABQBAQAAAAAAAAAAAAAAAAD/xAAUEQEAAAAAAAAAAAAAAAAAAA/9oADAM
BAAIRAxEAPwDeqIVZwEcBAQQiCIBHqIDgItogjwICWiqIto4CAqiZn6ScnNnwDn7yAcp/3TUKJX9cNnWgTi44n8zz3D
+k+4dtd79iR/4dtd79iR/4dtd79iR/
```

Ne+Q8Hh6yEAG/jrL6i1S3bbTugYXaXQ1kVnDLZbm2rE6nyGhHpM9sZiuJS2mtp6ltrFKtJgd1tTPLqFT65SN94HsuGfsA
nu+Ufhgt75QfLjluz+1RvxtHYapAj4/C02a5WxBBvoDod17eMqcZsNajEq+W9iylQwJGgOp0PfNTXoh1uN8rv4fnp++c
BcFhkw6ZFYm9yxvqzHS5t4W8pG/gUe5YAchbcJMTBrvzEwopiBUrg8h/4lZtB7dkbyQB56TQ4phaZjFDNUQd9/Td8bQI
229jI9RKiXznsuP5V7Kn0EvcEgzKo1ChU89Xb/avrCVqJVLKbE72tckHSwHG5Mk7OweRdd9txNyL6knmx4+AECTOvH
GNIgcTEvEvHgQODTs04iNJgKRBOsfnjWeBHcRto52gs8DUARwjVMdA4CLEvEJgLmiXnRsB0VYwQloDhDI0AIRTAxP0o
7PNqWJUaLenUI4Am6E+eYeYmDw9UBh37/Oe5VqSujI6hkYZWU6gg8JgdrdBEo5qwxGSiNSGQu4/lQgjOeAvr4wM1s
XH9VVKndfT4zd0cfmE80xrIxFSncKb2DEFhY21K6E7t3OXuwto7lJ8PlA2Ard3nO6y5sYGlVBj0S8Cg6Z4oCnkHEzE7Io5
6qjvvL/ppibsEXgLmM6G4G5LnU7v1gej7Jp2pnw/f78JGXeRLHZOnZ5iQ8ahDHxgHwb8L/v8AZkhwDvEqkdl13jjJlDFBh
A51I3QLsRvk1jIWJ4wKjG4nWU+GrMa6sqlramys1hY6kKCQOEtMYmhJ/fKO6LJ2qrfcF/xE/lAs8Iju2dxlA0VbWPiV90b9
DrrraTTFtOgDjSsIREgCyx6xWEEzQFcwbGNZ4wtAdeNaNBjjAG8FDNBWgaVlVYNBCCAtowwgEaywEnRLR0DgI4CII4
QHARwWcsKggKizyb6SuknWv1CEdWhsbnss/E2Gr24DcOPdrun3SL+HpGjTI611Nze2SnuJJ4E7v+bTxqtUN81wt9c7
DtH7icB+7wJWz3GQ7rF2IsLC4VLi3C418pJw1TIwPCQcC+dKiBizLaqtxY2HZf4FT5Q9B8ynmNfKBv8AAV8wUjWWO0t
opRpFjvI0HMzObBq2S54CUO2tqmq1+A3DugQ8XiC7Mx1JMteiWOCVCjGwOovzmezzkuzALcnhaB7LhMcpbRh6ynx/
SJDV6tFeq17HIhZVPe268pNh7EZSHrs1j7oNvWbzB00RQEUKOAAAgOw1AhbuLE7xy7ryn2jQek3WJdk3uo3gfaHdLt6
0jjFKDraBFwe0VdbqZ1epbUyi21hjSb+IoeyT9YnAX94Dh3jvklMVnTN3XgLjDmU21PLvgMBtE4cMhpliTc62bcBfLa5Xv
GmshPisis53LZv7h+svMLthHUHQ8bwOo9JUb3G13ZSGv929s3leSqe26De/bhqDv46jSGQYeotmRCDvuo1MhbU6NK
wLJdWt97dqAQfbHc3kRAnjF0yLiolueYTqeJRjZXRjyDAn0BmAqsyOUcEMBmsLtdQdXS+rqOK+2ttcwgXRDe9gGs5INl
dRufMpF1ud+4ccu6B6SwgnWYXCbfq4dsrl3UvlyN1aJTW1xZyBw58LTZbO2lTrpnpsDb2gGBKnkcpIgPYRhEKyxloAwJ
0JlnZYAmg80M6wVoGlURyiDVo9WgEUzmiRYDZ1o4LHBYAwIRRHBI9afpA5Fld0g22mFplmILkHIv8A9jyUSDtXpfQpB
shFRl35TdR4kanyE8l25t56zl3bUm4JF2HIll7IOV9YAdtY96jmo7asbl21uearvbkCQAOHOVYJOqIW/nqaj46R5ze0EAv7
9Q3Y9/a/SMrFTrUqM55LrbuudPSBKwGOenVR3emQDZl7PsEWYaC24mWO1MAcPUsuqPc025rxUnmNPKxmeDJuWm
W8ST8BN30bdMVhuoqqQUsNfaAt2HUnlqPLvgR9m1s1F1XeFOg4ixmeqLlAvLV6L4StkfVT7LcGXn+o4SHj6gZrj0gRK
KK28i3jNVsJcOhzM6X7yJjqyEbxLXZWyKdXXPc8V9lh+vH1gennFUKlO6upAtexBt6RKG0aIATrVuObCZzBdEcPa+erry
cDcByGssE6MYNd6FiL+3UbXfwB8IFliNoU1FzUQD74/WQ6rmqv1Op0IY6KdeHP5QB2bhUYFKNPNwsoNvM6k+Mt6Lh
BfjAr8ArglKo9rnuInbSpBFsDpu/WO2lWLLpvGt5SYmpUqsiXsSQo8WNh5m8CJjaud0QA5ew5W2joHAbXkCJXdSp60K/
VFKpSmST1be2VVrjsaL7V7a6jjLPpIow2ITLotHEVKZ/7bpTqW/C7yp2jh7L1V7t1qlydNVZqRPhYo3/qXgEo4qrTdHd2K
0zldCMrl/DOOIPBhoZ6XsnaiuoN73nmW0axavinDG1IBRfVWBdaZpsOKntaeYtJOysU9FkKezUynIWGZMwzDxUjc3rY7
w3nSHYi10upKPcMjrvVx7LDv+YnntXOjMjjqyHGew0pVW0TEU/9J9zLuveeobLrFls2t5lOm2FAIrBc2VWWon/Uot/mKO
ZA7Y+6YGXF7dWUANygQ3KrUHabDk7zTcdpD7p3RNlY16Lq9JaRQDsgkU6zpftIdbOym6kHiNN8FVXQqXuAEps/NG7
WExPipshPlyLinS4esjFHZg6Icpp11stS3DWwax/KB6xhMStRFqJqrC4594I4EboRhMj0Hx62egHzrq9Jjo2XQOjD7amxP
3r8Zq2eAhMS8S8SBzmBzwjiDtA0Aj1M4COtAVWhVMCBCKIBRHARiiRNq7XTDrdu05BKotyzd9hqB3wD7R2hToJnqNb
kPeY8gPz3Cefbb6Q1MV2EJCfYp1aYc/eBBznuB8oLaWJq1nNR1qDk3ULUVV5atoB4CQEqVGPZGGxAPuZFV/JLIw8rwK
DGk7hUL2YDIyZKwN9ymx1vyPlK50ZfdWn3sbufz+E021aYrLlCZKo0FOr7R5CnVNmBvuV9OR4SrOAFuxg3J+1UZiPS
yiBRsyX1Lufwj8zHpm9ykB3kFj6tJGIp1UOuSn3Bk09CZFc39qrfuAY/OwgPdXPtOq92YD4LJmwdonD1lfrMynsuNTdTvI
7xv8u+Vl6Y4OfML+s4svBP7mMD1fauAXE0SgIv7dN+Aa2mv2Tx9eE87VGRzTcFWU5WB3gzRdCtsAr/AA7XBXVNd6
8VvzHyPdLHpPsXrl62mPrUH/uKPdP8w4engFDiMASugvpKhsM6nshr915ouju0Vaytv3WPxmjQICCANf3+/CBisM2ONs
gqW3bv1Es8NsrHsczs4B3gkCbZKigX+EmJXUiBQbP2a6bxr6n1lg1Mrv3yfUrASqx+LAHfuA4k8LCBWba2itJc7nTlxJ5
DmZA6C4tsTjKebgxcjkEBK/HLKfpc5Y6nQbhy5zQ/Q3hO3WrHgqovmczH4CAP6SFHW4pbah8NVH9aGkf9qiE2bsF6/
wBY3YLomZSAe2qKGPgSin+kS8x2CTEYypiASUKU0A91shJzeFybeF5Z136pVYaWdAfBjl+ZEDzTpXQqUi6OEyuVLMi
hCxW5UtbedTlH8cGxKuoIRVRFB3gKoFj33BM3PTvCipTzDhr+v5TzzDUi7qote/tW1sPnA9d2HUBUGQemastPrE1ZCH

A+0FN3XzXMPOSdh0cqAdwndKCwoO6i7IpcDmV7VvO0Dzg0kVsl70xZM3PC4jWm39Dnfwgu0wdH9tlI3X/xOGNhod
5en63hCiEZfcQ5Cf8AyuJ7VNv6HIgqrsCKje2pV3H+rQcUq/4kZWgM2btIJUTEU1C9pespjQB7WLIPsuuYW4G3dPUetD
AMpupAIPMEXE8uxRp1M9RRlKXSqqjUoHHV107xZcwm56J4o1MMoNsyMyG27TVSO4gi0C2Dzs84rGEQH5oy8WMtA
0itHgwSiFUQHrDoINU4zGdIek3WXo4c3QGzuC/a5gdWpIXzF/DeFztjpMlO6Uu2/wBoK7IvgUUhjMJjHaqxLujltSr1alIn8Y
RYF6ii1+qB/mOLuPMyRQqZtFBb/s1s/wD8Fa5bwgREwrUzmFDEp/NScOO7ULb4yQtRa182WsRvV16rEabyrrcOd+hJ
PdEZFQK9xlL5M6B8PURt5zpqmnd6wjJUc5Hb/E02z0yQt6iWuAHHtEAZhqQdYA8QV6tTmaphnOVs4+sovpoDztqOB
A3XlXicJTRnTE1atwQaZUB1dCLhrsRv/MS4p4hAGqZPqqnYxFIb0YnR0HAX1XkbjlB1kdCEV1z01vSqEArUwzm7C5BtY
dsb9zCBncSuEX2RWfvZkXX+mV75T7FEkcNWf8psHbEnQYnDaaAhqY9Oxf4SsxeHrn28ZT8qxPwUAQKEdYNyZf6Lf7
o1jV4sB/Uo+XhCYvCoNTXVj3Bm+N5DyJ9o+SwDU69VHV1cZlII7SnX9P1nqGx9pJWpo67yO0L+y3vCeTOE4FvSXHR
naYo1At+w5AJ3Wf3Tv47vTlA1u2+jpZzXw1g+903BjxZTuDcxuO/TjQrtipSbI6spG9WFj6H5zbUsRcSPjXUizqGHJgGHx
gRMN0ipMmrWPIx9Hb6D3x6yH/8AzqbGwo0/wLNBszohhaq2enlb7SGx9LW9RArK/SBdynOeAGvryjMCjM2dzdzcADc
o7uZ75dYno3h6JP1yIBpeoMgNt4DAENbQHdaRa+TIWoulWxCk03R+0QSq2U3zEA2HceUCsw1Km1Zmex6u1g1rDj
mN+QmgwLtUzCklkIKs9ygdTvAG8i3Ezz7Z2MV8QrkgrmtI4HS4J5ier7LqLYeEBlNcgHYJUb8upHlxHhrIfSHFKcJWdWB
C0y6nvWzD4iX7rxEwfTqkiISzMqVDcqpsrOpv6nTTja8Cg2p0g69FppfddzyEi7FT64abpV0WBHYBuxsANSeQ75othbJc
1O2xVtDlWxIHedwgegbLY2F5Kx9MMpHAjXzkbD0coGpPiZIfVbwPLsVhVpv1TaJrh3v/ANGsSabX5JUVvK0hKC5Abew
Gcfz64asfU03mk6V0Q532zjqSeRftU2Pg6IP6zM2WLHNuNTK/eDUU0qg7stVEPnAFg665ErFRmpDqcQm7rKT9gP4i+
U94Uy96A1slSpSvcG4/qT2GH3lLfglFtHDhLOvsYijUvbXtg3YfiVJJ2VieorLWJy5KjUqwIJ3lijEDXUZlvwtxgemMIwiSaRV1
V0IZWAII1BB3WIiMkCPaDtJBgvOBoEWFd1RS7sFUbyTYTp0DC9Iekj1wUpgrR5nKvWcLl3ITL/KA/eAZQI/86eeKcHw7C
hR6Tp0Agruuod7aC9PGo1rm2qtra54+sI6Z2am7KtVWVqfWU6ahxbRS677k6Xup011iToBqdRsz5UIc/wCdhXvapze
mTx423jeMwjKioEUhyaOb6up/4mHqb8j21tfXTxHI9OgdUZyzOFXrkU9fT9yvSPtOoGhuLE27mG6MWmjotLP2HzNh6j
WBR/foudwBNvPKdxnToFfhKN1KnCO7U2NNyr1PbXmoBANiDbviVqCa5sFWHi9QfNPGLOgVmKRPdwr373qH4BRKm
op4Ubd1nPznToAWY/YH4W/WNKnflsNxGtjOnQNv0Y2pnQKx7SaG/EcD+veJpKeEaoQBOnQNJs3YoWxI1/f6SZtnGU8J
h3rsAci3VLhc77kQHvNok6B5JT2saH+ISun8TUqVC6OjMtMPmcvncZb3IAABOp7xM2+LZ83ZA7CltWIZlv23Gt3OY2Jt
lvpaLOgDw+Is2lgBusLcbi/E8tZ6p0a2mHRTfXd5zp0DV0sRKjpdsX+ModWrhGDBwxGYXAIsRcEXvvE6dAotl9HqeDQ
sVNSqB7ZsLE7gg90X85Z7D2aEGdtXY3J8Yk6BdVGAtIuKJtp8506BjNvIztkU+32A2mj5WZCP60UTO583aC2LXcDgpq
A51/pr0x+KLOgBwo6xBTJ0FSoi62s1VCafkXQjzkps9UU61NVZ3BpVUYLlNSmAwJDGxLIN3MGdOgazoNtEWfDFSjKSw
pNe6X9olTqUvrY6i54Wmqczp0ATiAnToH//2Q=="
        },
        "id": 3910,
        "userType": "l",
        "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+f
alse",
               "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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+disabled+eq+fal
se",
                "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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+f
alse",
                "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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+disabled+eq+fal
se",
                "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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+f
alse",
               "accounts":
"http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/Account?filter=type+eq+U+and+users.user.userName+e
q+'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+enable
d+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.user.userName+e
q+'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_1.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: ?◆lLucene90DocValuesData    ◆Mi◆◆!k
                                        mr◆
Lucene90_0 e(Y;R◆◆oGa◆◆Йr◆◆(◆◆        ◆◆◆
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: ?◆lLucene90FieldsIndexMeta   ◆Mi◆◆!k
                                         mr◆ ◆◆
      0                ◆?         0       6          ◆A         0       $      ◆(◆◆        ◆◆
LIP_TIMSTA: 1685697735025
*************************** 3. row ***************************
    LIP_ID: 5381
```

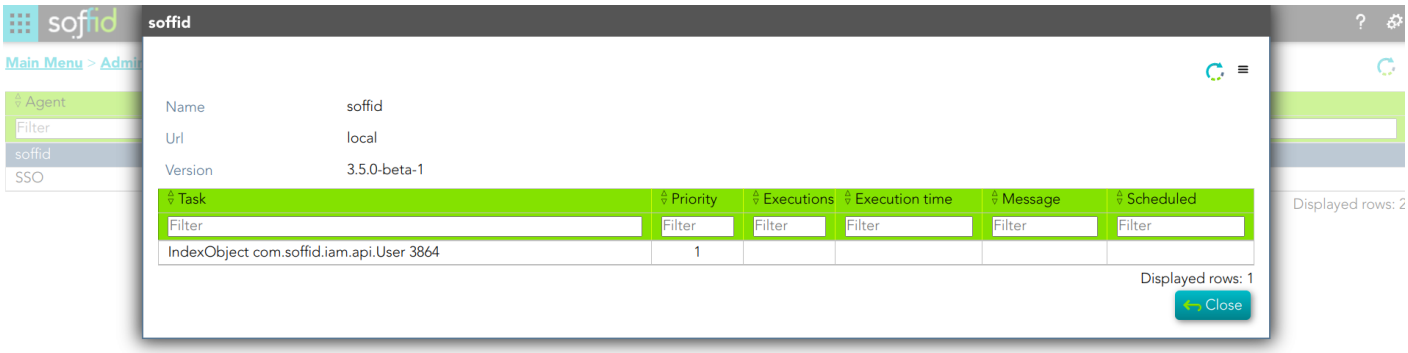
# soffid agent

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

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

# Example

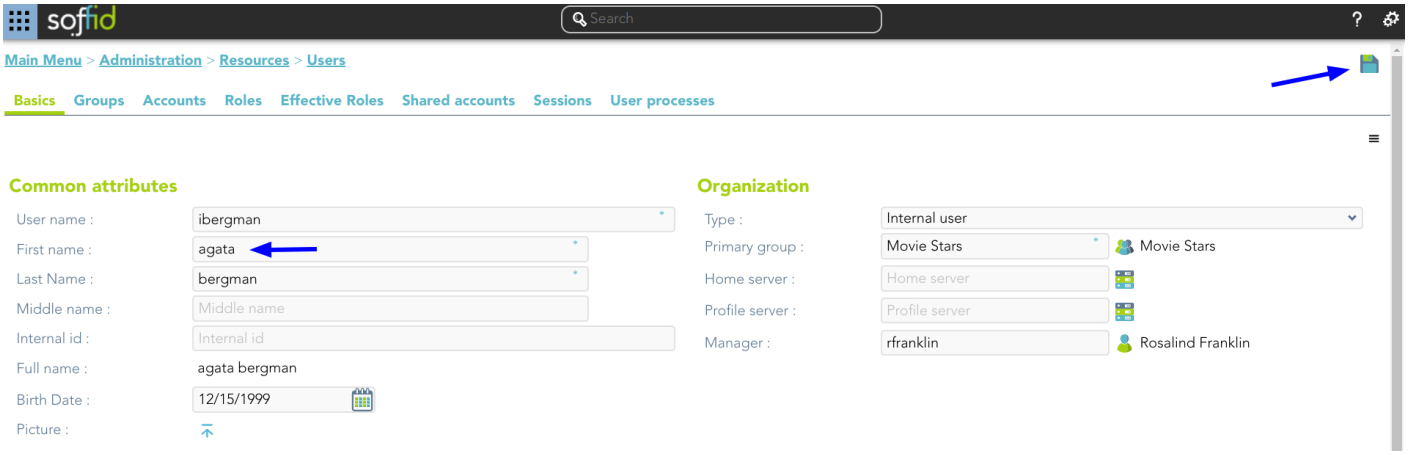
**1.** A soffid agent pending task:



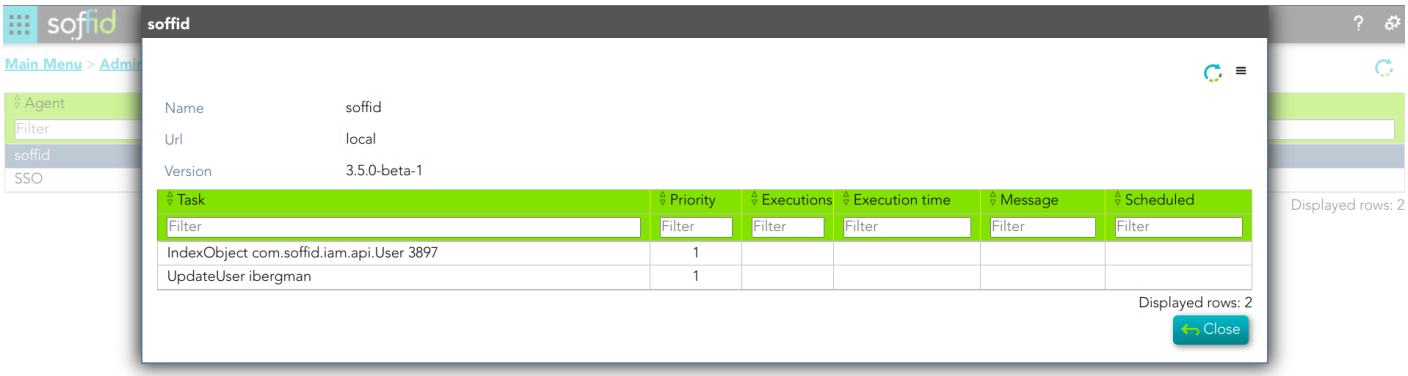

# Step-by-step

## 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
```

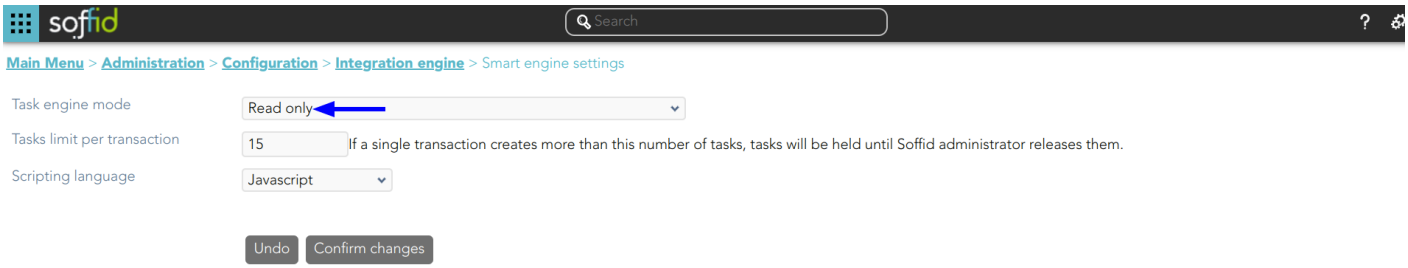
**4.** Then Sync Server and updates the database table SC_LUNIND by upgrading the LIP_TIMSTA field of the User object or by creating a new record if it did not previously exist.

```
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 |
+--------+--------------------------------+------------+---------------+
```

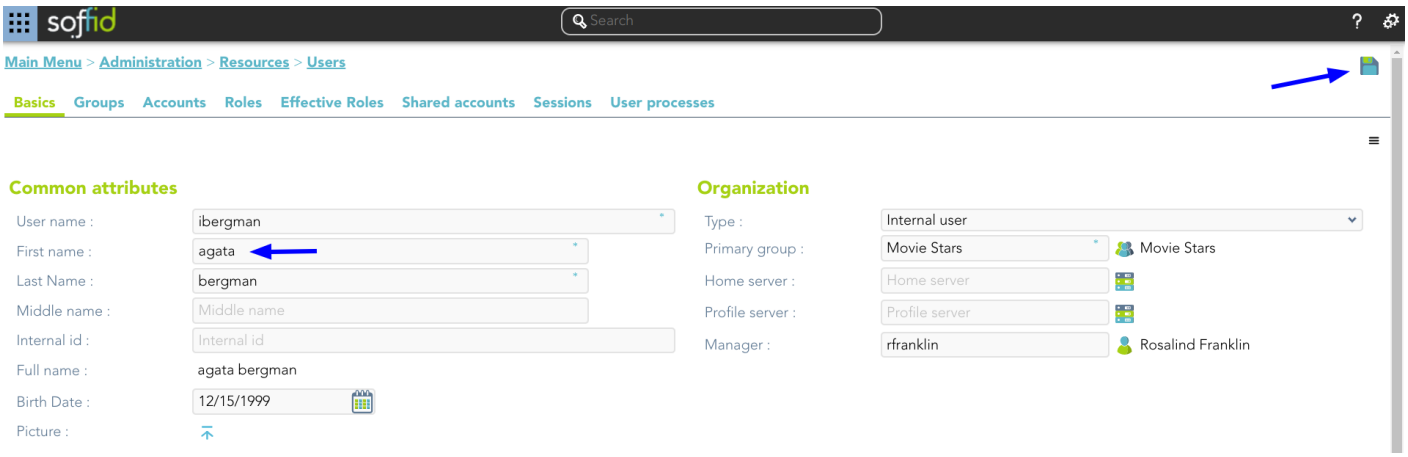
**5.** When the following search will be performed, the very first thing to do is check the database file. If it is necessary update the file system and finally perform the search.

# Example 2

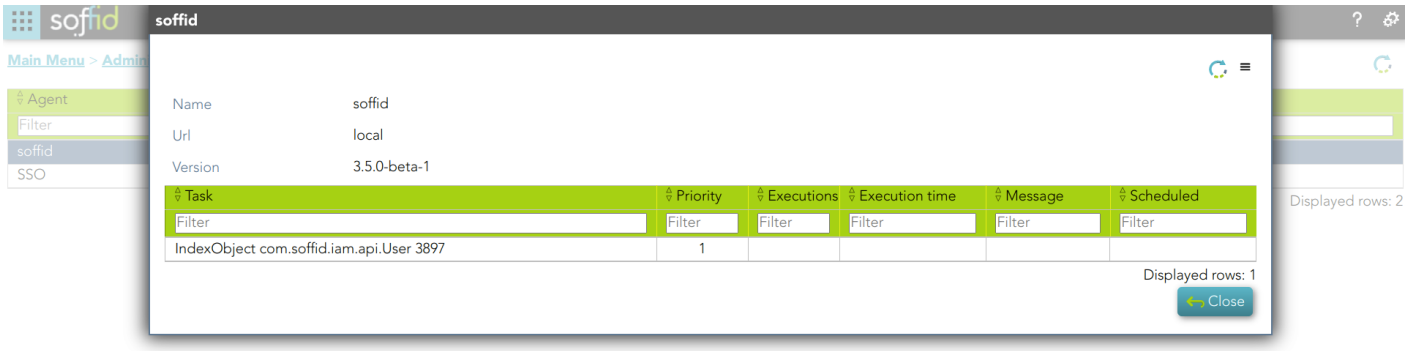
**1.** The task engine mode is Read only




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



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




**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

# 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 |
|---|---|
| **QueryParserConstants** | Token literal values and constants. |

Class Summary

| Class | Description |
|---|---|
| **MultiFieldQueryParser** | A QueryParser which constructs queries to search multiple fields. |

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

## Enum Summary

| Enum | Description |
|---|---|
| **QueryParser.Operator** | The default operator for parsing queries. |

## Exception Summary

| Exception | Description |
|---|---|
| **ParseException** | This exception is thrown when parse errors are encountered. |

## Error Summary

| Error | Description |
|---|---|
| **TokenMgrError** | Token Manager Error. |