

Textual index

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 a textual index 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.

Image

ISS

IGA

AM

PAM

IRC

Search in Soffid...

?

SA

Main Menu > Configuration > Global Settings > Metadata

4 out of 11

Builtin user object

com.soffid.iam.base.api.User

Download CSV File

Import

Expand all

Collapse all

Basics :

Object type * :

com.soffid.iam.base.api.User

Description * :

Builtin user object

Use textual index :

Yes

Add new

	Code	Label	Data type	Built-in type
	__COMMONATTRIBUTES__	Common attributes	Separator	Yes
	userName	User name	String	Yes
	firstName	First name	String	Yes
	lastName	Last Name	String	Yes
	middleName	Middle name	String	Yes
	fullName	Full name	String	Yes
	__ORGANIZATION__	Organization	Separator	Yes
	userType	Type	User type	Yes

Displayed rows: 26

Undo

Apply changes

ADV

2. Check the attributes if the option "Included in the quick search" is enabled.

Image

The screenshot shows the 'Attribute metadata' configuration page for a 'Builtin user object' (com.soffid.iam.base.api.User). The interface includes a left sidebar with navigation buttons (ISS, IGA, AM, PAM, IRC) and a right sidebar with a search bar and filters. The main configuration area is divided into two sections: 'Attribute type' and 'User interface'. In the 'Attribute type' section, the 'Code' is 'userName', the 'Label' is 'User name', and the 'Data type' is 'String'. In the 'User interface' section, the 'Letter case' is 'Keep as entered by the user', the 'User hint' is 'User hint', and the 'Description' is 'Description'. At the bottom, there are two toggle switches: 'Required' (set to 'Yes') and 'Include in quick search' (set to 'Yes'). The 'Include in quick search' toggle is highlighted with a red rectangle.

Attribute metadata

Builtin user object
com.soffid.iam.base.api.User

Delete 2 out of 26

Expand all Collapse all

▼ Attribute type :

Code * : userName

Label : User name

Data type : String

▼ User interface :

Letter case * : Keep as entered by the user

User hint : User hint

Description : Description

Required : ☒ Yes

Include in quick search : ☒ Yes

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.

Image

Main Menu > Resources > Users

QuickBasicAdvanced

Add newDownload CSV FileImport...

frankin

Q

View

☐	User name ^	Primary group	Enabled	Last Name	First name	Full name	EMail
☐	aretha	world	Yes	Franklin	Aretha	Aretha Franklin	
☐	frank	world	Yes	Sinatra	Frank	Frank Sinatra	
☐	frankrudolph	world	Yes	Olson	Frank Rudolph	Frank Rudolp...	
☐	franks	world	Yes	Sherwood	Frank	Frank Sherwood	
☐	rfranklin	world	Yes	Franklin	Rosalind	Rosalind Fran...	

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

Image

ISS

IGA

AM

PAM

IRC

Attribute metadata

 **Builtin user object**
com.soffid.iam.base.api.User

 Delete 27 out of 27

<

>

Expand all

Collapse all

08

Attribute type :

Code * :

manager

Label :

Manager



Data type: :

User



User interface :

User hint :

User hint

Description :

Description



Required :

☐

No

Include in quick search :

☒

Yes

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.

Image

Quick Basic Advanced

Add new

 Download CSV File Import

frankin

 View

☐	User name ^ 	Last Name 	First name 	Enabled 	Primary group 	Manager 
☐	aeinstein	Einstein	Albert	Yes	franchises	franks
☐	aretha	Franklin	Aretha	Yes	world	
☐	frank	Sinatra	Frank	Yes	world	
☐	frankrudolph	Olson	Frank Rudolph	Yes	world	
☐	franks	Sherwood	Frank	Yes	world	
☐	inewton	Newton	Isaac	Yes	consultants	frank
☐	rfranklin	Franklin	Rosalind	Yes	world	

Example 2

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

Image

Quick Basic Advanced

Add new

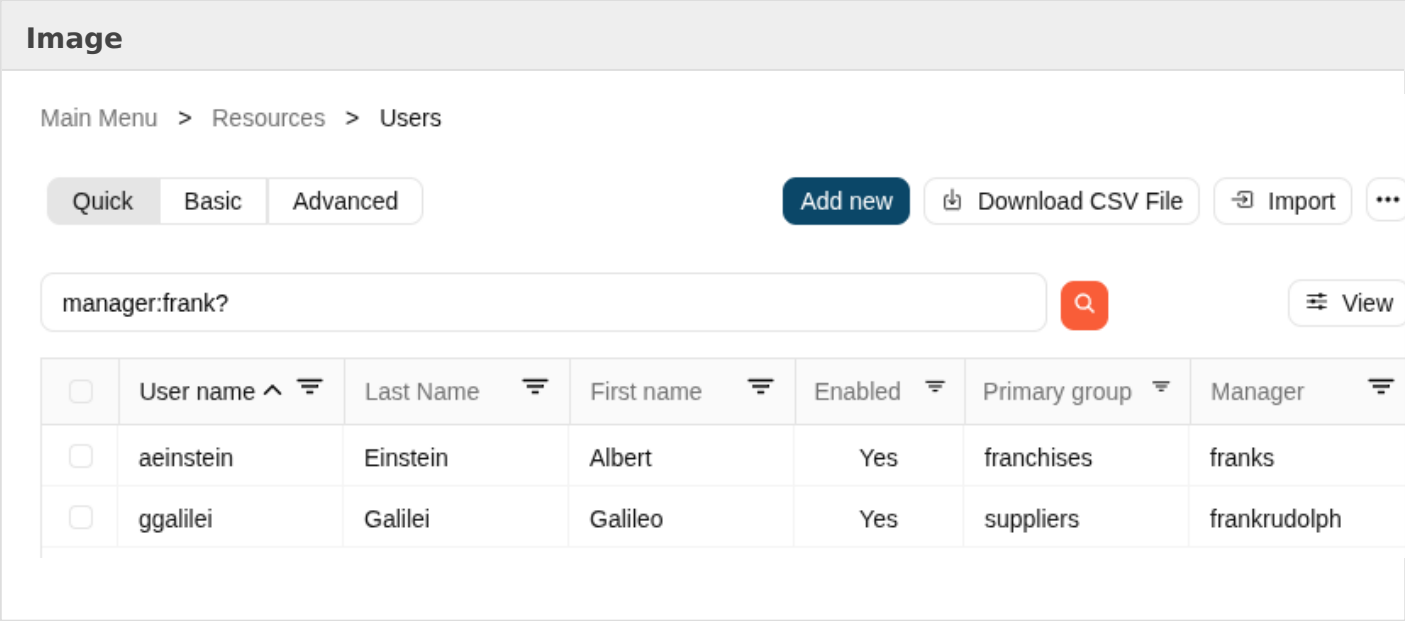
 Download CSV File Import

manager:frank

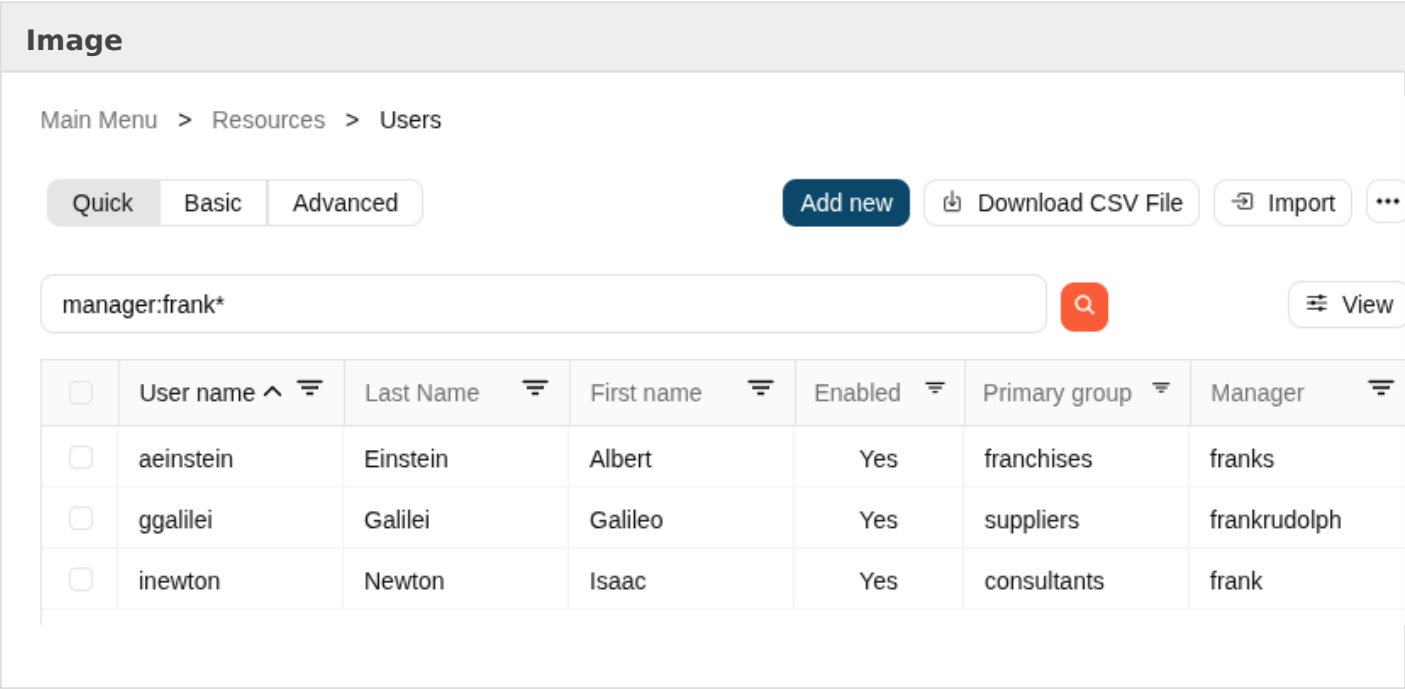
 View

☐	User name ^ 	Last Name 	First name 	Enabled 	Primary group 	Manager 
☐	aeinstein	Einstein	Albert	Yes	franchises	franks
☐	ggalilei	Galilei	Galileo	Yes	suppliers	frankrudolph
☐	inewton	Newton	Isaac	Yes	consultants	frank

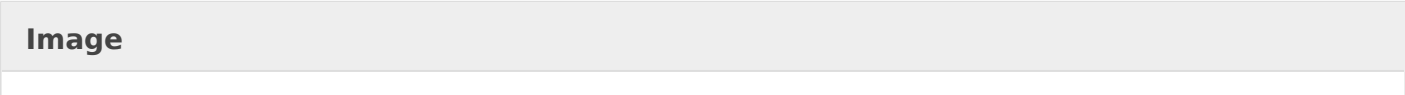
Notice the difference by searching "manager:frank?":



And by searching "manager:frank*":



And also by searching "manager:fr*"



Main Menu > Resources > Users

Quick Basic Advanced Add new Download CSV File Import ...

manager:fr* View

☐	User name ^	Last Name	First name	Enabled	Primary group	Manager
☐	aeinstein	Einstein	Albert	Yes	franchises	franks
☐	ggalilei	Galilei	Galileo	Yes	suppliers	frankrudolph
☐	inewton	Newton	Isaac	Yes	consultants	frank

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.

Image

Main Menu > Resources > Users

Quick Basic Advanced Add new Download CSV File Import ...

userName:frank* View

☐	User name ^	Last Name	First name	Enabled	Primary group	Manager
☐	frank	Sinatra	Frank	Yes	world	
☐	frankrudolph	Olson	Frank Rudolph	Yes	world	
☐	franks	Sherwood	Frank	Yes	world	

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

Image

Quick

Basic

Advanced

Add new

Download CSV File

Import

...

userName:frank?



View

☐	User name ^	Last Name	First name	Enabled	Primary group	Manager
☐	frankrudolph	Olson	Frank Rudolph	Yes	world	
☐	franks	Sherwood	Frank	Yes	world	

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.

Image

Quick

Basic

Advanced

Add new

Download CSV File

Import

...

frank?



View

☐	User name ^	Last Name	First name	Enabled	Primary group	Manager
☐	aeinstein	Einstein	Albert	Yes	franchises	franks
☐	aretha	Franklin	Aretha	Yes	world	
☐	frankrudolph	Olson	Frank Rudolph	Yes	world	
☐	franks	Sherwood	Frank	Yes	world	
☐	ggalilei	Galilei	Galileo	Yes	suppliers	frankrudolph
☐	rfranklin	Franklin	Rosalind	Yes	world	

Notice the difference by searching "fran?"

Image

Quick Basic Advanced

Add new

Download CSV File

Import



fran?



View

☐	User name ^	Last Name	First name	Enabled	Primary group	Manager
☐	aeinstein	Einstein	Albert	Yes	franchises	franks
☐	aretha	Franklin	Aretha	Yes	world	
☐	frank	Sinatra	Frank	Yes	world	
☐	frankrudolph	Olson	Frank Rudolph	Yes	world	
☐	franks	Sherwood	Frank	Yes	world	
☐	ggalilei	Galilei	Galileo	Yes	suppliers	frankrudolph
☐	inewton	Newton	Isaac	Yes	consultants	frank
☐	rfranklin	Franklin	Rosalind	Yes	world	

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

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. 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 <u>Soffid Console <= 3.4 version</u> ~0.8: stricter search ~0.1: more lax search <u>Soffid Console > 3.4 version</u> 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+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+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+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"
}
]
}

```

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",

```

[illegible]

14k6fAes3+DZcqoHViqqDYi+gHDhMLhELhETQMoGhsQNM5PLfaW+MxaU6ijTUlluw9IRbUM3EnIA1BE6DLxM8B5Ea
0b1k7NAY4g4UxpEATGBYw7iDKwAxmWHIEbpAv1hAlJYQGA4R4jBHKYBBFiAziYCGKI0mKIDxKHp4p/gnt9unfwzj875
+WVnSxL4Kv3Jn/Awb8oHi+LpAtpwCmTqVH6pL8j/uMFTp7yfD0AEnYpciOBRfjrAqkoXa1o9sMyG/wCWnpCqLax+P2
oEAAAZHXUcOEBuG2w9M3yg25f8S2wnTNcwDIV4XHw0mTxWle/bsugPAWvz7+6Ds9sxW4G/mLgEEjkQQbwPYdnb
dpOtIYa+nfLBmVhpuni2G2nk5ibjox0jD9g+1A09OoBmEitj0Q9owi4RzrKDaxdWsqFiONtLwje1Nt1stqKgXF8zfvTjKHE
9o3r4nX71hfmBI2JNUK1xfQKDqSeQ4ejg2Z8Ndy6U7OEUeEszWubs29R690C1wGKw6nLnDhd2jL1KiMNDfVVYD0kK
jgKujpO2aIXZGKuISmACAAVNNxYrcHiN8yiYoo/1QqBNzI3a6sg20f3l+Udb1TcaayG99YLDVibZuUk110vAhbBpXJHH
MU8r3NvIXIxWpAOVAFrgj9+szmA26uHxLoVuxF8xPshgCQB4W9Zf5y75h7639bpf+4QLmn7I8B8opESLAQRSZxYER
AW84xVE5IgBYxpjmSMcQBuYO8c0HaBqFEdalohA60UCdaKIC3nTohMDoojLxYBVMbjsP1IGpT+2jp+JSBOWSKZgeB
1Kp0Oul7jkbk/vwlli6oegjgSvkDp8Cl/pbs/qcXWp7lc51+6/at4A3HllmFX6t6ZG7tr8j+UAILCO40F4zEbH7S52sx0vb2
eXjNL0VIZATb/n9/KWu19kBxmXQj98IGOTYFU5lennDEMhVwpzDQWO+1uGnHnNbsvolSSiyOLs5zWXUILAKoYgXsA
Ne+Q8Hh6yEAG/jrL6i1S3bbTugYXaXQ1kVnDLZbm2rE6nyGhHpM9sZiuJ52mtp6lfrFKtJgd1tTPLqFT65SN94HsuGfsA
nu+Ufhgt75QfLjuz+1RvxtHYapAj4/C02a5WxBBVoDod17eMqcZsNajEq+W9iyIQWjGgOp0PfNTXoh1uN8rv4fnp++c
BcFhkW6ZFYM9yxvqzHS5t4W8pG/gUe5YAchbcJMTBrvzEwopiBUrg8h/4IZtB7dkbyQB56TQ4phaZjFDNUQd9/Td8bQI
229jI9RKIXznsuP5V7Kn0EvcEgzKo1ChU89Xb/avrCVqjVLKbE72tckHSwHG5Mk7OweRdd9txNyL6knmx4+AECTOVH
GNlGcTEvEvHgQODTs04iNjgKRBOsfnjWeBHcRto52gs8DUARwjVMdA4CLEvEjgLmiXnRsB0VYwQloDhDI0AIRTAxP0o
7PNqWJUaLenUI4Am6E+eYeYmDw9UBh37/Oe5VqSujl6hkYZWU6gg8JgdrdBEO5qwxGSiNSGQu4/IQgjOeAvr4wM1s
XH9VVKndfT4zd0cfmE80xrlxF5nckB2DEFhY21K6E7t3OXuwt07Ij8PIA2Ard3nO6y5sYGIvBj0S8Cg6Z4oCnkHEzE7Io5
6qjvvL/ppibSEXgLM6G4G5LnU7v1gej7Jp2pnw/f78JGxeRLHZOnZ5iQ8ahDHxgHwb8L/v8AZkhWdVEqkd13jjlIDFBh
A51I3QLsRvk1jIwJ4wKjG4nWU+GrMa6sqlramys1hY6kKCQOEtMYmhj/fKO6LJ2qrFcF/xE/IAS8lju2dxIA0VbWPiV90b9
DrrraTTFtOgdJjSsIREgCyx6xWEEZqFcbwGNZ4wtAdeNaNBjjAG8FDNBWgaVIVYNBCCAtowwgEaywEnRLR0DgI4CIH4
QHARwWcsKggKizyb6SuknWv1CEdWhsbns/E2Gr24DcOPdrun3SL+HpGjTI611Nze2SnuJJ4E7v+bTxqtUN81wt9c7
DtH7icB+7wJWz3GQ7rF2IsLC4VLi3C418pJw1TIwPCQcC+dKiBizLaqtxY2HZf4FT5Q9B8ynmNfKBv8AAV8wUjWWO0t
opRpFjvI0HMzObBq2S54CUO2tqmql+A3DugQ8XiC7Mx1JMteiWOCVCjGwOovzmezzkuzALcnhaB7LhMcpbRh6ynx/
SjDV6tFeq17HihZVPe268pNh7EZSHrs1j7oNvWbzb00RQEUKOAAAgOw1AhbuLE7xy7ryn2jQek3WjDk3uo3gfaHdLt6
0jjFKDraBFwe0VdbqZ1epbUyi21hjSb+loeyT9YnAX94Dh3jvklMVnTN3XgLjDmU21PLvgMBtE4cMhpliTc62bcBfLa5Xv
GmshPisis53LzV7h+svMLthHUHQ8bwOo9JUb3G13ZSGv929s3leSqe26De/bhqDv46jSGQYeotmRCDvuo1MhbU6NK
wLjdWt97dqAQfbHc3kRAnjF0yLiolueYTqeJRjZXRjyDAn0BmAqsYUOcEMBmsLtdQdXS+rqOK+2ttcwgXRDe9gGs5INI
dRufMpF1ud+4ccu6B6SwgnWYXCbfq4dsrl3UvlyN1ajTW1xZyBw58LTZbO2ITrpnpsDb2gGBKncplgPYRhEKyxloAwJ
0JlnZYAmg80M6wVoGIURyiDVo9WgEUzmiRYDZ1o4LHBYAwIRRHBI9afpA5FId0g22mFplmILkHlv8A9jyUSDtXpfQpB
shFRI35TdR4kanyE8I25t56zl3bUm4JF2HIII7IOV9YAdtY96jmo7asbl21uearvbkCQAOHOVYJQoIw/nqaj46R5ze0EA7
9Q3Y9/a/SMrFTrUqM55LrbuudPSBKwGOenVR3emQDZl7PsEWYaC24mWO1MAcPUsuqPc025rxUnmNPKxmeDjuWm
W8ST8BN30bdMVhuoqqQUsnfaAt2HUnlqPLvgR9m1s1F1XeFOg4ixmeqLIAvLV6L4StkFVT7LcGXn+o4SHj6gZrj0gRK
KK28i3jNVsjcOhzM6X7yjjqyEbxLXZWYkdXXPc8V9lh+vH1gennFUKIO6upAtexBt6RKG0aIATrVuObCZzBdEcPa+erry
cDcByGssE6MYNd6FiL+3UbXfwB8IFliNoU1FzUQD74/WQ6rmqv1Op0IY6KdeHP5QB2bhUYFKNPnwsoNvM6k+Mt6Lh
BfjAr8ArgIKo9rnunlbSpBFsDpu/WO2IWLLpvGt5SYmpUqsiXsSQo8WNh5m8CJjaud0QA5ew5W2joHAbXkCjXdSp60K/
VFKpSmST1be2VVrjsaL7V7a6jjLPplow2ITLotHEVKZ/7bpTqW/C7yp2jh7L1V7t1qlYdNVZqRPhYo3/qXgEo4qrTdHd2K
0zldCmrl/DOOIPBhoZ6XsnaiuoN73nmW0axavinDG1IBrfVWBdaZpsOKntaeYtJOysU9FkKezUynIWGZMwzDxUjc3rY7
w3nSHYi10upKPcMjrvVx7LDv+YnntXOjMjjqyHGew0pVW0TEU/9J9zLuveeobLrFls2t5IOm2FAIrBc2VWWon/Uot/mKO
ZA7Y+6YGF7dWUANYgQ3KrUHabDk7zTcdpD7p3RNIY16Lq9JaRQDsgkU6zpfldbOym6kHiNN8FVXQqXuAEps/NG7

WExPipshPlyLinS4esjFHZg6lcpp11stS3DWwax/KB6xhMStRFqJqrC4594I4EboRhMj0Hx62egHzrq9Jjo2XQOjD7amxP
3r8Zq2eAhMS8S8SBzmBzwwiDtA0Aj1M4CotAVWhVMCBCKIBRHARiIRNq7XTDrdu05BKotydzd9hqbB3wD7R2hToJnqNb
kPeY8gPz3Cefbb6Q1MV2EJCfYp1aYc/eBBznuB8oLaWJq1nNR1qDk3ULUVV5atoB4CQEqVGPZGGxAPuZfVJ/Llw8rwK
DGk7hUL2YDlyZKwN9ymx1vyPIK50ZfdWn3sbufz+E021aYrLICZKo0FOr7R5CnVNmBvuV9OR4SrOAFuxg3J+1UZiPS
yiBRsyX1Lufwj8zHpm9ykB3kFj6tJGlp1UOuSn3Bk09CZFc39qrFuAY/OwgPdXPtOq92YD4LJmwdonD1lfrMynsuNTdTvl
7xv8u+VI6Y4OfML+s4svBP7mMD1fauAXE0Sglv7dN+Aa2mv2Tx9eE87VGRzTcFWU5WB3gzRdCtsAr/AA7XBxVNd6
8VvzHyPdLHpPsXrl62mPrUH/uKpDP8w4engFDiMASugvpKhsM6nshr915ouju0Vaytv3WPxmjQICCANf3+/CBisM2ONs
ggW3bv1Es8NsrHsczs4B3gkCbZKigX+EmjXUiBQBp2a6bxr6n1lg1Mrv3yfUrASqx+LAHfuA4k8LCBWba2itj7nTlxJ5
DmZA6C4tsTjKebgxcjkEBK/HLKfpc5Y6nQbhy5zQ/Q3hO3WrHgqovmczH4CAP6SFHW4pbah8NVH9aGkf9qiE2bsF6/
wBY3YLomZSAe2qKGPgSin+kS8x2CTEYypIASUKU0A91shJzeFybeF5Z136pVYaWdAfBjI+ZEDzTpXQqUi6OEyuVLMi
hCxW5UtbedTIH8cGxKuolRVRFB3gKoFj33BM3PTvCipTzDhr+v5TzzDUI7qote/tW1sPnA9d2HUBUGQemastPrE1ZCH
A+0FN3XzXMPOSdh0cqAdwndKCwo06i7lpcDmV7VvO0Dzg0kVsl70xZM3PC4jWm39Dnfwgu0wdH9tli3X/xOGNhod
5en63hCiEZfcQ5Cf8Ayuj7VNV6HlgqrsCKje2pV3H+rQcUq/4kZWgM2btIJUTEU1C9pespJB7WLIPsuuYW4G3dPUetD
AMPupAIPMEXE8uxRp1M9RRIKXSqqjUoHHV107xZcwm56J4o1MMoNsyMyG27TVSO4gi0C2Dzs84rGEQH5oy8WMtA
OitHgwSiFUQHrDoINU4zGdleK3WXo4c3QGzuC/a5gdWpIXzF/DeFztjpMIO6Uu2/wBoK7IvgUUhjMjJHaqxLujtSr1alln8Y
RYF6ii1+qB/mOLuPMYRQqZtFBb/s1s/wD8Fa5bwgREwrUzmFDEp/NScOO7ULb4yQtRa182WsRvV16rEabyrrcOd+hj
PdEZFQK9xIL5M6B8PURT5zpqmnd6wjUc5Hb/E02z0yQt6iWuAHtEAzhqQdYA8QV6tTmaphnOVs4+sovpoDztqOB
A3XIXicJTRnTE1atwQaZUB1dCLhrsRv/MS4p4hAGqZPqqnYxFlb0YnR0HAX1XkbjIB1kdCEV1z01vSqEArUwzm7C5BtY
dsb9zCBncSuEX2RWfVzKXX+mV75T7FEkcNWF8psHbEnQYnDaaAhqY9Oxf4SsxeHrn28ZT8qxPwUAQKEdYNYzf6Lf7
o1jV4sB/Uo+XhCYvCoNTXVj3Bm+N5DyJ9o+SvDU69VHV1cZIII7SnX9P1nqGx9pJWpo67yOOL+y3vCeTOE4FvSXHR
naYo1At+w5AJ3Wf3Tv47vTIA1u2+jpZzXw1g+903BjxZTuDcxuO/TjQrtipSbl6spG9WFj6H5zbUsRcSPjXUizqGHJgGHx
gRMN0ipMmrWPlx9Hb6D3x6yH/8AzqbGwo0/wLNBszohhaq2enlb7SGx9LW9RArk/SBdynOeAGvryjMCjM2dzdzcADc
o7uZ75dYno3h6JP1ylBpeoMgNt4DAENbQHdaRa+TIWoulWxck03R+0QSq2U3zEA2HceUCsw1Km1Zmex6u1g1rDj
mN+QmgwLtUzCklkIKs9ygdTvAG8i3Ezz7Z2MV8QrkgRmtl4HS4J5ier7LqLYeEBINcgHYJUb8upHlxHhrlfSHfKcJWdWB
C0y6nvWzD4iX7rxEwfTqkilSzMqVDCqpsrOpv6nTTja8Cg2p0g69FppfddzyEi7FT64abpV0WBHYBuxsANSeQ75othbJc
1O2xVtDIWxlHedwgegbLY2F5Kx9MMpHAjXzkbD0coGpPiZlfbwPLsVhVpv1Tajrh3v/ANGsSabX5JUVvK0hKC5Abew
Gcfz64asfU03mk6V0Q532zjqSeRftU2Pg6IP6zM2WLHNuNTK/eDUU0qg7stVEPnAFg665ErFRmpDqcQm7rKT9gP4i+
U94Uy96A1sISpSvcG4/qT2GH3ILfglFtHDhLOvsYijUvbXtg3YfiVJJ2VieorLWJy5KjUqwlJ3lijEDXUZlvtxgemMIwiSaRV1
V0iZWAlI1BB3WliMkCPaDtJBgvOBoEWFd1RS7sFUbyTYTp0DC9Iekj1wUpgrR5nKvWclI3ITL/KA/eAZQI/86eeKcHw7C
hR6Tp0Agruod7aC9PGolrm2qtra54+sl6Z2am7KtVWVqfWU6ahxbRS677k6Xup011iToBqdRsz5Ulc/wCdhXvapze
mTx423jeMwjKioEUhyaOb6up/4mHqb8j21tfXTxHI9OgdUZyzOFXrkU9ft9yvSPtOoGhuLE27mG6MWmjotLP2HzNh6j
WBR/foudwBNvPKdxnToFfhKN1KnCO7U2NNyr1PbXmoBANiDbviVqCa5sFWHi9QfNPGLogVmKRPdwr373qH4BRK
mop4Ubd1nPznToAWY/YH4W/WNKnfIsNxGtjOnQNv0Y2pnQKx7SaG/EcD+veJpKeEaoQBOOnQNjs3YoWxl1/f6SZtnGU8j
h3rsAci3VLhc77kQHvNok6B5JT2saH+ISun8TUqVC6OjMtMPmCvncZb3IAABOp7xM2+LZ83ZA7ClTWIZlv23Gt3OY2Jt
lvpaLOgDw+Is2lgBusLcbi/E8tZ6p0a2mHRTfXd5zp0DV0sRKjpdSx+ModWrhGDBwxGYXAlsRcEXvvE6dAotl9HqeDQ
sVNSqB7ZsLE7gg90X85Z7D2aEGdtXY3J8Yk6BdVGAtluKJtp8506BjNvlztkU+32A2mj5WZCP60UTO583aC2LXcDgpp
A51/pr0x+KLOgBwo6xBTJ0FSoi62s1VCafkXQjzkps9UU61NVZ3BpVUYLINSmAwJDGxLIN3MGdOgazoNtEWfDFSjKSw
pNe6X9oITqUvrY6i54Wmqczp0ATiAnToH//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+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+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+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+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+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 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",  
          "self": "http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User/3910",  
          "list": "http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User?textFilter=fran~ AND Sinatra",  
          "listAll": "http://soffid.35x.lab:8089/soffid/webservice/scim2/v1/User" }  
      }  
    }  
  ]  
}
```

[illegible]

14k6fAes3+DZcqoHViqqDYi+gHDhMLhELhETQMoGhsQNM5PLfaW+MxaU6ijTUlluw9IRbUM3EnIA1BE6DLxM8B5Ea
0b1k7NAY4g4UxpEATGBYw7iDKwAxmWHIEbpAv1hAlJYQGA4R4jBHKYBBFiAziYCGKI0mKIDxKHp4p/gnt9unfwzj875
+WVnSxL4Kv3Jn/Awb8oHi+LpAtpwCmTqVH6pL8j/uMFTp7yfD0AEnYpciOBRfjrAqkoXa1o9sMyG/wCWnpCqLax+P2
oEAAAZHXUcOEBuG2w9M3yg25f8S2wnTNcwDIV4XHw0mTxWle/bsugPAWvz7+6Ds9sxW4G/mLgEEjkQQbwPYdnb
dpOtIYa+nfLBmVhpuni2G2nk5ibjox0jD9g+1A09OoBmEitj0Q9owi4RzrKDaxdWsqFiONtLwje1Nt1stqKgXF8zfvTjKHE
9o3r4nX71hfmBI2JNUK1xfQKDqSeQ4ejg2Z8Ndy6U7OEUeZszWubs29R690C1wGKw6nLnDhd2jL1KiMNDfVVYD0kK
jgKujpO2aIXZGKuISmACAAVNNxYrcHiN8yiYoo/1QqBNzI3a6sg20f3l+Udb1TcaayG99YLDVibZuUk110vAhbBpXJHH
MU8r3NvIXIxWpAOVAFrgj9+szmA26uHxLoVuxF8xPshgCQB4W9Zf5y75h7639bpf+4QLmn7I8B8opESLAQRSZxYER
AW84xVE5IgBYxpjmSMcQBuYO8c0HaBqFEdalohA60UCdaKIC3nTohMDoojLxYBVMbjsP1IGpT+2jp+JSBOWSKZgeB
1Kp0Oul7jkbk/vwlli6oegjgSvkDp8Cl/pbs/qcXWp7lc51+6/at4A3HllmFX6t6ZG7tr8j+UAILCO40F4zEbH7S52sx0vb2
eXjNL0VIZATb/n9/KWu19kBxmXQj98IGOTYFU5lennDEMhVwpzDQWO+1uGnHnNbsvolSSiyOLs5zWXUILAKoYgXsA
Ne+Q8Hh6yEAG/jrL6i1S3bbTugYXaXQ1kVnDLZbm2rE6nyGhHpM9sZiuJ52mtp6lfrFKtJgd1tTPLqFT65SN94HsuGfsA
nu+Ufhgt75QfLjuz+1RvxtHYapAj4/C02a5WxBBVoDod17eMqcZsNajEq+W9iyIQWjGgOp0PfNTXoh1uN8rv4fnp++c
BcFhkW6ZFYM9yxvqzHS5t4W8pG/gUe5YAchbcJMTBrvzEwopiBUrg8h/4IZtB7dkbyQB56TQ4phaZjFDNUQd9/Td8bQI
229jI9RKIXznsuP5V7Kn0EvcEgzKo1ChU89Xb/avrCVqjVLKbE72tckHSwHG5Mk7OweRdd9txNyL6knmx4+AECTOVH
GNlGcTEvEvHgQODTs04iNjgKRBOsfnjWeBHcRto52gs8DUARwjVMdA4CLEvEjgLmiXnRsB0VYwQloDhDI0AIRTAxP0o
7PNqWJUaLenUI4Am6E+eYeYmDw9UBh37/Oe5VqSujl6hkYZWU6gg8jgdrdBEO5qwxGSiNSGQu4/IQgjOeAvr4wM1s
XH9VVKndfT4zd0cfmE80xrlxF5nckB2DEFhY21K6E7t3OXuwt07Ij8PIA2Ard3nO6y5sYGIvBj0S8Cg6Z4oCnkHEzE7Io5
6qjvvL/ppibsEXgLM6G4G5LnU7v1gej7Jp2pnw/f78JGxeRLHZOnZ5iQ8ahDHxgHwb8L/v8AZkhWdvEqkd13jjlIDFBh
A51I3QLsRvk1jIwJ4wKjG4nWU+GrMa6sqlramys1hY6kKCQOEtMYmhj/fKO6LJ2qrFcF/xE/IAS8lju2dxIA0VbWPiV90b9
DrrraTTFtOgdJjSsIREgCyx6xWEEzQFcwbGNZ4wtAdeNaNBjjAG8FDNBWgaVIVYNBCCAtowwgEaywEnRLR0DgI4CI4
QHARwWcsKggKizyb6SuknWv1CEdWhsbns/E2Gr24DcOPdrun3SL+HpGjTI611Nze2SnuJJ4E7v+bTxqtUN81wt9c7
DtH7icB+7wJWz3GQ7rF2IsLC4VLi3C418pJw1TIwPCQcC+dKiBizLaqtxY2HZf4FT5Q9B8ynmNfKBv8AAV8wUjWWO0t
opRpFjvI0HMzObBq2S54CUO2tqmql+A3DugQ8XiC7Mx1JMteiWOCVCjGwOovzmezzkuzALcnhaB7LhMcpbRh6ynx/
SjDV6tFeq17HihZVPe268pNh7EZSHrs1j7oNvWbzb00RQEUKOAAAgOw1AhbuLE7xy7ryn2jQek3WjDk3uo3gfaHdLt6
0jjFKDraBFwe0VdbqZ1epbUyi21hjSb+loeyT9YnAX94Dh3jvklMVnTN3XgLjDmU21PLvgMBtE4cMhpliTc62bcBfLa5Xv
GmshPisis53LzV7h+svMLthHUHQ8bwOo9JUb3G13ZSGv929s3leSqe26De/bhqDv46jSGQYeotmRCDvuo1MhbU6NK
wLjdWt97dqAQfbHc3kRAnjF0yLiolueYTqeJRjZXRjyDAn0BmAqsYUOcEMBmsLtdQdXS+rqOK+2ttcwgXRDe9gGs5INI
dRufMpF1ud+4ccu6B6SwgnWYXCbfq4dsrl3UvlyN1ajTW1xZyBw58LTZbO2ITrpnpsDb2gGBKncplgPYRhEKyxloAwJ
0JlnZYAmg80M6wVoGIURyiDVo9WgEUzmiRYDZ1o4LHBYAwIRRHBI9afpA5FId0g22mFplmILkHlv8A9jyUSDtXpfQpB
shFRI35TdR4kanyE8I25t56zl3bUm4JF2HIII7IOV9YAdtY96jmo7asbl21uearvbkCQAOHOVYJQoIw/nqaj46R5ze0EA7
9Q3Y9/a/SMrFTrUqM55LrbuudPSBKwGOenVR3emQDZl7PsEWYaC24mWO1MAcPUsuqPc025rxUnmNPKxmeDjuWm
W8ST8BN30bdMVhuoqqQUsnfaAt2HUnlqPLvgR9m1s1F1XeFOg4ixmeqLIAvLV6L4StkFVT7LcGXn+o4SHj6gZrj0gRK
KK28i3jNVsjcOhzM6X7yjjqyEbxLXZWYkdXXPc8V9lh+vH1gennFUKIO6upAtexBt6RKG0aIATrVuObCZzBdEcPa+erry
cDcByGssE6MYNd6FiL+3UbXfwB8IFliNoU1FzUQD74/WQ6rmqv1Op0IY6KdeHP5QB2bhUYFKNPnwsoNvM6k+Mt6Lh
BfjAr8ArgIKo9rnunlbSpBFsDpu/WO2IWLLpvGt5SYmpUqsiXsSQo8WNh5m8CJjaud0QA5ew5W2joHAbXkCjXdsP60K/
VFKpSmST1be2VVrjsaL7V7a6jjLPplow2ITLotHEVKZ/7bpTqW/C7yp2jh7L1V7t1qlydNVZqRPhYo3/qXgEo4qrTdHd2K
0zldCmrl/DOOIPBhoZ6XsnaiuoN73nmW0axavinDG1IBrfVWBdaZpsOKntaeYtJOysU9FkKezUynIWGZMwzDxUjc3rY7
w3nSHYi10upKPcMjrvVx7LDv+YnntXOjMjjqyHGew0pVW0TEU/9J9zLuveeobLrFls2t5IOm2FAIrBc2VWWon/Uot/mKO
ZA7Y+6YGF7dWUANYgQ3KrUHABdk7zTcdpD7p3RNIY16Lq9JaRQDsgkU6zpftldBOym6kHiNN8FVXQqXuAEps/NG7

WEXpipshPlyLinS4esjFHZg6lcpp11stS3DWwax/KB6xhMStRFqJqrC4594I4EboRhMj0Hx62egHzrq9Jjo2XQOjD7amxP
3r8Zq2eAhMS8S8SBzmBzWjiDtA0Aj1M4CotAVWhVMCBCKIBRHARiIRNq7XTDrdu05BKotyZd9hqbB3wD7R2hToJnqNb
kPeY8gPz3Cefbb6Q1MV2EJCfYp1aYc/eBBznuB8oLaWJq1nNR1qDk3ULUVV5atoB4CQEqVGPZGGxAPuZfVJ/Llw8rwK
DGk7hUL2YDlyZKwN9ymx1vyPIK50ZfdWn3sbufz+E021aYrLICZKo0FOr7R5CnVNmBvuV9OR4SrOAFuxg3J+1UZiPS
yiBRsyX1Lufwj8zHpm9ykB3kFj6tJGlp1UOuSn3Bk09CZFc39qrFuAY/OwgPdXPtOq92YD4LJmwDonD1lfrMynsuNTdTvl
7xv8u+VI6Y4OfML+s4svBP7mMD1fauAXE0Sglv7dN+Aa2mv2Tx9eE87VGRzTcFWU5WB3gzRdCtsAr/AA7XBxVNd6
8VvzHyPdLHpPsXrl62mPrUH/uKpDP8w4engFDiMASugvpKhsM6nshr915ouju0Vaytv3WPxmJQICCANf3+/CBisM2ONs
ggW3bv1Es8NsrHsczs4B3gkCbZKigX+EmjXUiBQbP2a6bxr6n1lg1Mrv3yfUrASqx+LAHfuA4k8LCBWba2itj7nTlxJ5
DmZA6C4tsTjKebgxcjkEBK/HLKfpc5Y6nQbhy5zQ/Q3hO3WrHgqovmczH4CAP6SFHW4pbah8NVH9aGkf9qiE2bsF6/
wBY3YLomZSAe2qKGPgSin+kS8x2CTEYypIASUKU0A91shJzeFybeF5Z136pVYaWdAfBjI+ZEDzTpXQqUi6OEyuVLMi
hCxW5UtbedTIH8cGxKuolRVRFB3gKoFj33BM3PTvCipTzDhr+v5TzzDUI7qote/tW1sPnA9d2HUBUGQemastPrE1ZCH
A+0FN3XzXMPOSdh0cqAdwndKCwo06i7lpcDmV7VvO0Dzg0kVsl70xZM3PC4jWm39Dnfwgu0wdH9tli3X/xOGNhod
5en63hCiEzfcQ5Cf8Ayuj7VNV6HlgqrsCKje2pV3H+rQcUq/4kZWgM2btIJUTEU1C9pespJB7WLIPsuuYW4G3dPUetD
AMPupAIPMEXE8uxRp1M9RRIKXSqqjUoHHV107xZcwm56J4o1MMoNsyMyG27TVSO4gi0C2Dzs84rGEQH5oy8WMtA
OitHgwSiFUQHrDoINU4zGdleK3WXo4c3QGzuC/a5gdWpIXzF/DeFztjpMIO6Uu2/wBoK7IvgUUhjMjJHaqxLujtSr1alln8Y
RYF6ii1+qB/mOLuPMYRQqZtFBb/s1s/wD8Fa5bwgREwrUzmFDEp/NScOO7ULb4yQtRa182WsRvV16rEabyrrcOd+hj
PdEZFQK9xIL5M6B8PURT5zpqmnd6wjjUc5Hb/E02z0yQt6iWuAHtEAzhqQdYA8QV6tTmaphnOVs4+sovpoDztqOB
A3XIXicJTRnTE1atwQaZUB1dCLhrsRv/MS4p4hAGqZPqqnYxFlb0YnR0HAX1XkbjIB1kdCEV1z01vSqEArUwzm7C5BtY
dsb9zCBncSuEX2RWfVzKXX+mV75T7FEkcNWF8psHbEnQYnDaaAhqY9Oxf4SsxeHrn28ZT8qxPwUAQKEdYNYzf6Lf7
o1jV4sB/Uo+XhCYvCoNTXVj3Bm+N5DyJ9o+SvDU69VHV1cZIII7SnX9P1nqGx9pJWpo67yOOL+y3vCeTOE4FvSXHR
naYo1At+w5AJ3Wf3Tv47vTIA1u2+jpZzXw1g+903BjxZTuDcxuO/TjQrtipSbl6spG9Wfj6H5zbUsRcSPjXUizqGHJgGHx
gRMN0ipMmrWPlx9Hb6D3x6yH/8AzqbGwo0/wLNBszohhaq2enlb7SGx9LW9RArk/SBdynOeAGvryjMCjM2dzdzcADc
o7uZ75dYno3h6JP1ylBpeoMgNt4DAENbQHdaRa+TIWoulWxck03R+0QSq2U3zEA2HceUCsw1Km1Zmex6u1g1rDj
mN+QmgwLtUzCklkIKs9ygdTvAG8i3Ezz7Z2MV8QrkgRmtl4HS4J5ier7LqLYeEBINcgHYJUb8upHlxHhrlfSHfKcJWdWB
C0y6nvWzD4iX7rxEwfTqkilSzMqVDCqpsrOpv6nTTja8Cg2p0g69FppfddzyEi7FT64abpV0WBHYBuxsANSeQ75othbJc
1O2xVtDIWxIHedwgegbLY2F5Kx9MMpHAjXzkbD0coGpPiZlfbwPLsVhVpv1Tajrh3v/ANGsSabX5JUVvK0hKC5Abew
Gcfz64asfU03mk6V0Q532zjqSeRftU2Pg6IP6zM2WLHNuNTK/eDUU0qg7stVEPnAFg665ErFRmpDqcQm7rKT9gP4i+
U94Uy96A1sISpSvcG4/qT2GH3ILfglFtHDhLOvsYijUvbXtg3YfiVJJ2VieorLWJy5KjUqwlJ3lijEDXUZlvtwxgemMlwiSaRV1
V0iZWAlI1BB3WliMkCPaDtJBgvOBoEWFd1RS7sFUbyTYTp0DC9Iekj1wUpgrR5nKvWclI3ITL/KA/eAZQI/86eeKcHw7C
hR6Tp0Agruod7aC9PGolrm2qtra54+sl6Z2am7KtVWVqfWU6ahxbRS677k6Xup011iToBqdRsz5Ulc/wCdhXvapze
mTx423jeMwjKioEUhyaOb6up/4mHqb8j21tfXTxHI9OgdUZyzOFXrkU9ft9yvSPtOoGhuLE27mG6MWmjotLP2HzNh6j
WBR/foudwBNvPKdxnToFfhKN1KnCO7U2NNyr1PbXmoBANiDbviVqCa5sFWHi9QfNPGLogVmKRPdwr373qH4BRK
mop4Ubd1nPznToAWY/YH4W/WNKnfIsNxGtjOnQNv0Y2pnQKx7SaG/EcD+veJpKeEaoQBOOnQNjs3YoWxI1/f6SZtnGU8j
h3rsAci3VLhc77kQHvNok6B5JT2saH+ISun8TUqVC6OjMtMPmcvncZb3IAABOp7xM2+LZ83ZA7ClTWIZlv23Gt3OY2Jt
lvpaLOgDw+Is2lgBusLcbi/E8tZ6p0a2mHRTfXd5zp0DV0sRKjpdSx+ModWrhGDBwxGYXAlsRcEXvvE6dAotl9HqeDQ
sVNSqB7ZsLE7gg90X85Z7D2aEGdtXY3J8Yk6BdVGAtluKJtp8506BjNvlztkU+32A2mj5WZCP60UTO583aC2LXcDgpp
A51/pr0x+KLOgBwo6xBTj0FSoi62s1VCafkXQjzkps9UU61NVZ3BpVUYLINSmAwJDGxLIN3MGdOgazoNtEWfDFSjKSw
pNe6X9oITqUvrY6i54Wmqczp0ATiAnToH//2Q=="

},

"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.user.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+disabled+eq+false"
```

```
se",
    "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+f
```

```

else",
    "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+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"
}
]
```

```
}
```

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.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": "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.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": "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

Images

```
root@200d60e4947a:/opt/soffid/iam-console-4/index/master# ll
total 16
drwxr-x--- 4 root root 4096 Sep 27 07:21 ./
drwxr-xr-x 3 root root 4096 Jul 11 16:36 ../
drwxr-x--- 2 root root 4096 Sep  7 15:00 bpm/
drwxr-x--- 2 root root 4096 Sep 27 07:45 com.soffid.iam.base.api.User/
```

```

root@200d60e4947a:/opt/soffid/iam-console-4/index/master/com.soffid.iam.base.api.User# ll
total 88
drwxr-x--- 2 root root 4096 Sep 27 07:45 ./
drwxr-x--- 4 root root 4096 Sep 27 07:21 ../
-rw-r----- 1 root root 447 Sep 27 07:40 _4.cfe
-rw-r----- 1 root root 2519 Sep 27 07:40 _4.cfs
-rw-r----- 1 root root 340 Sep 27 07:40 _4.si
-rw-r----- 1 root root 67 Sep 27 07:41 _b_3.liv
-rw-r----- 1 root root 447 Sep 27 07:40 _b.cfe
-rw-r----- 1 root root 3695 Sep 27 07:40 _b.cfs
-rw-r----- 1 root root 378 Sep 27 07:40 _b.si
-rw-r----- 1 root root 447 Sep 27 07:40 _c.cfe
-rw-r----- 1 root root 2530 Sep 27 07:40 _c.cfs
-rw-r----- 1 root root 340 Sep 27 07:40 _c.si
-rw-r----- 1 root root 447 Sep 27 07:41 _e.cfe
-rw-r----- 1 root root 2662 Sep 27 07:41 _e.cfs
-rw-r----- 1 root root 338 Sep 27 07:41 _e.si
-rw-r----- 1 root root 447 Sep 27 07:41 _i.cfe
-rw-r----- 1 root root 2623 Sep 27 07:41 _i.cfs
-rw-r----- 1 root root 338 Sep 27 07:41 _i.si
-rw-r----- 1 root root 447 Sep 27 07:44 _m.cfe
-rw-r----- 1 root root 2685 Sep 27 07:44 _m.cfs
-rw-r----- 1 root root 338 Sep 27 07:44 _m.si
-rw-r----- 1 root root 564 Sep 27 07:44 segments_m

```

2. And the folder structure for the Soffid Syncserver

Images

```

root@sync-server-version4:/opt/soffid/iam-sync/index/master# ll
total 16
drwxr-xr-x 4 root root 4096 Sep 27 07:32 ./
drwxr-xr-x 3 root root 4096 Sep 16 16:02 ../
drwxr-xr-x 2 root root 4096 Sep 16 16:03 bpm/
drwxr-xr-x 2 root root 4096 Sep 27 08:17 com.soffid.iam.base.api.User/

root@sync-server-version4:/opt/soffid/iam-sync/index/master/com.soffid.iam.base.api.User# ll
total 88
drwxr-xr-x 2 root root 4096 Sep 27 08:17 ./
drwxr-xr-x 4 root root 4096 Sep 27 07:32 ../
-rw-r--r-- 1 root root 447 Sep 27 07:40 _4.cfe
-rw-r--r-- 1 root root 2519 Sep 27 07:40 _4.cfs
-rw-r--r-- 1 root root 340 Sep 27 07:40 _4.si
-rw-r--r-- 1 root root 67 Sep 27 07:41 _b_3.liv
-rw-r--r-- 1 root root 447 Sep 27 07:40 _b.cfe
-rw-r--r-- 1 root root 3695 Sep 27 07:40 _b.cfs
-rw-r--r-- 1 root root 378 Sep 27 07:40 _b.si
-rw-r--r-- 1 root root 447 Sep 27 07:41 _e.cfe
-rw-r--r-- 1 root root 2662 Sep 27 07:41 _e.cfs
-rw-r--r-- 1 root root 338 Sep 27 07:41 _e.si
-rw-r--r-- 1 root root 447 Sep 27 07:41 _i.cfe
-rw-r--r-- 1 root root 2623 Sep 27 07:41 _i.cfs
-rw-r--r-- 1 root root 338 Sep 27 07:41 _i.si
-rw-r--r-- 1 root root 447 Sep 27 07:44 _m.cfe
-rw-r--r-- 1 root root 2685 Sep 27 07:44 _m.cfs
-rw-r--r-- 1 root root 338 Sep 27 07:44 _m.si
-rw-r--r-- 1 root root 447 Sep 27 08:17 _q.cfe
-rw-r--r-- 1 root root 2532 Sep 27 08:17 _q.cfs
-rw-r--r-- 1 root root 338 Sep 27 08:17 _q.si
-rw-r--r-- 1 root root 564 Sep 27 08:17 segments_q
-rw-r--r-- 1 root root 0 Sep 27 07:41 write.lock

```

Database

The database tables involved:

- SC_LUINPA
- SC_LUNIND

Example

1. The database structure

Images

```
MariaDB [soffid042]> select * from SC_LUNIND;
+-----+-----+-----+-----+
| LIP_ID | LIP_NAME                               | LIP_TEN_ID | LIP_TIMSTA |
+-----+-----+-----+-----+
| 101390 | master/bpm                             | 1          | 1758038614646 |
| 197420 | master/com.soffid.iam.base.api.User    | 1          | 1758961050456 |
+-----+-----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [soffid042]> select * from SC_LUINPA \G
***** 1. row *****
      LIP_ID: 101391
    LIP_LIN_ID: 101390
      LIP_NAME: _b.nvm
    LIP_ORDER: 0
    LIP_TEN_ID: 1
    LIP_DATA: ?\lLucene90NormsMetadata    i d\{*V.<A\nc\
      +      \(\<C\
    LIP_TIMSTA: 1757257253850
***** 2. row *****
      LIP_ID: 101392
    LIP_LIN_ID: 101390
      LIP_NAME: _b_Lucene90_0.tip
    LIP_ORDER: 0
    LIP_TEN_ID: 1
    LIP_DATA: ?\lBlockTreeTermsIndex    i d\{*V.<A\nc\
    Lucene90_0      \(\
    LIP_TIMSTA: 1757257253855
```

soffid agent

You can check the soffid agent status by visiting the [Sync Server monitoring](#) page.

Example

1. A soffid agent pending task.

Image

soffid

Url :

local

Task	Priority	Executions	Execution time	Message	Scheduled
index	Search	Search ...	Search ...	Search ...	Search ...
IndexObject com.soffid.iam.base.api.User 197099	1				

Step-by-step

Example 1

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

Image

**Rosalind Franklin**

rfranklin



Basics

Groups

Accounts

Roles

Effective
RolesShared
accounts

Sessions

User
processes

Issues

Tokens

OTP
devices

Backups

Expand all

Collapse all



▼ Common attributes :

User name * :

rfranklin

First name * :

Rosalind

Last Name * :

Franklin

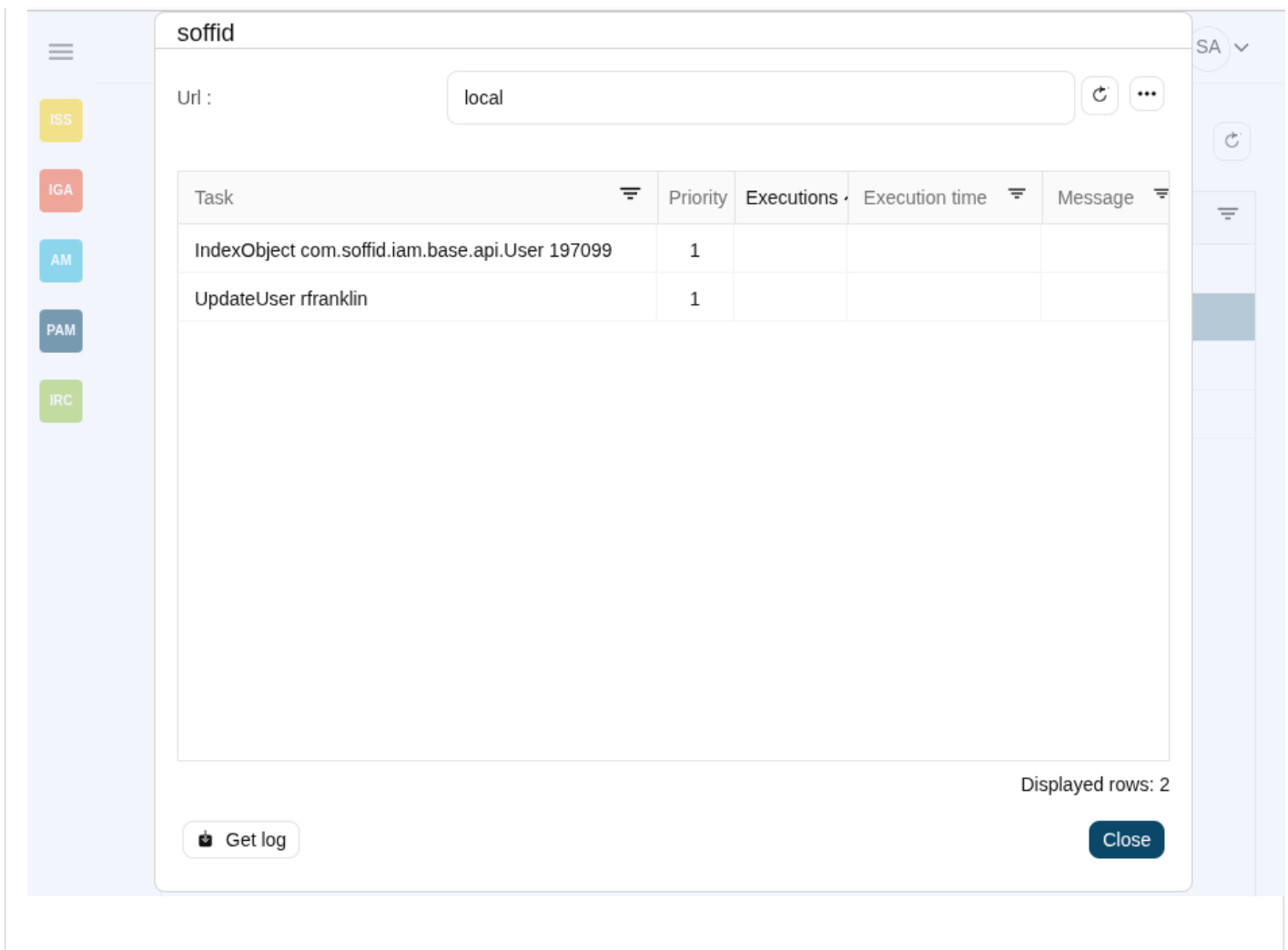
Middle name :

Middle name

Full name :

Rosalind Franklin

2. New tasks are created and executed.**Image**

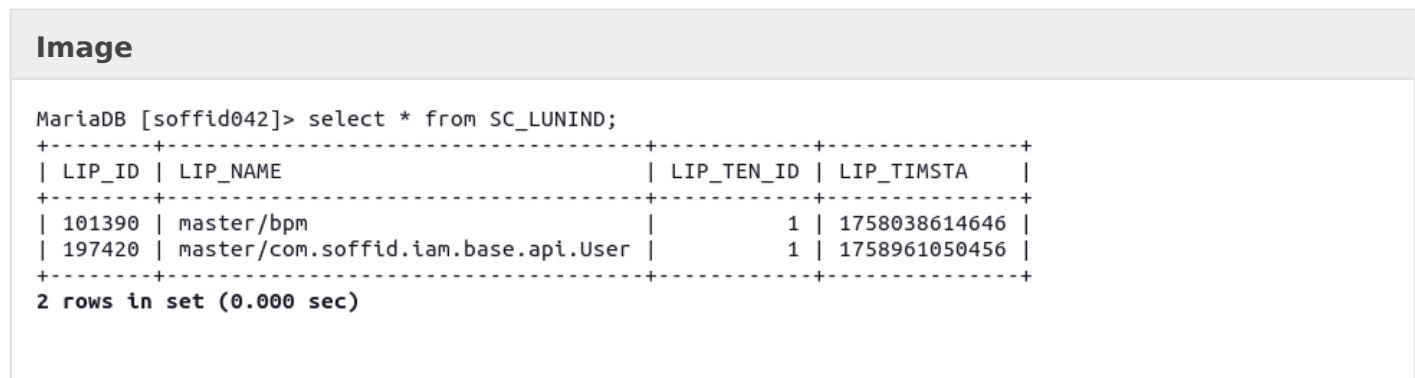


3. Then Sync Server indexes the updated text and places the index file.

Image

```
root@200d60e4947a:/opt/soffid/iam-console-4/index/master/com.soffid.iam.base.api.User# ll
total 88
drwxr-x--- 2 root root 4096 Sep 27 07:45 ./
drwxr-x--- 4 root root 4096 Sep 27 07:21 ../
-rw-r----- 1 root root 447 Sep 27 07:40 _4.cfe
-rw-r----- 1 root root 2519 Sep 27 07:40 _4.cfs
-rw-r----- 1 root root 340 Sep 27 07:40 _4.si
-rw-r----- 1 root root 67 Sep 27 07:41 _b_3.liv
-rw-r----- 1 root root 447 Sep 27 07:40 _b.cfe
-rw-r----- 1 root root 3695 Sep 27 07:40 _b.cfs
-rw-r----- 1 root root 378 Sep 27 07:40 _b.si
-rw-r----- 1 root root 447 Sep 27 07:40 _c.cfe
-rw-r----- 1 root root 2530 Sep 27 07:40 _c.cfs
-rw-r----- 1 root root 340 Sep 27 07:40 _c.si
-rw-r----- 1 root root 447 Sep 27 07:41 _e.cfe
-rw-r----- 1 root root 2662 Sep 27 07:41 _e.cfs
-rw-r----- 1 root root 338 Sep 27 07:41 _e.si
-rw-r----- 1 root root 447 Sep 27 07:41 _i.cfe
-rw-r----- 1 root root 2623 Sep 27 07:41 _i.cfs
-rw-r----- 1 root root 338 Sep 27 07:41 _i.si
-rw-r----- 1 root root 447 Sep 27 07:44 _m.cfe
-rw-r----- 1 root root 2685 Sep 27 07:44 _m.cfs
-rw-r----- 1 root root 338 Sep 27 07:44 _m.si
-rw-r----- 1 root root 564 Sep 27 07:44 segments_m
```

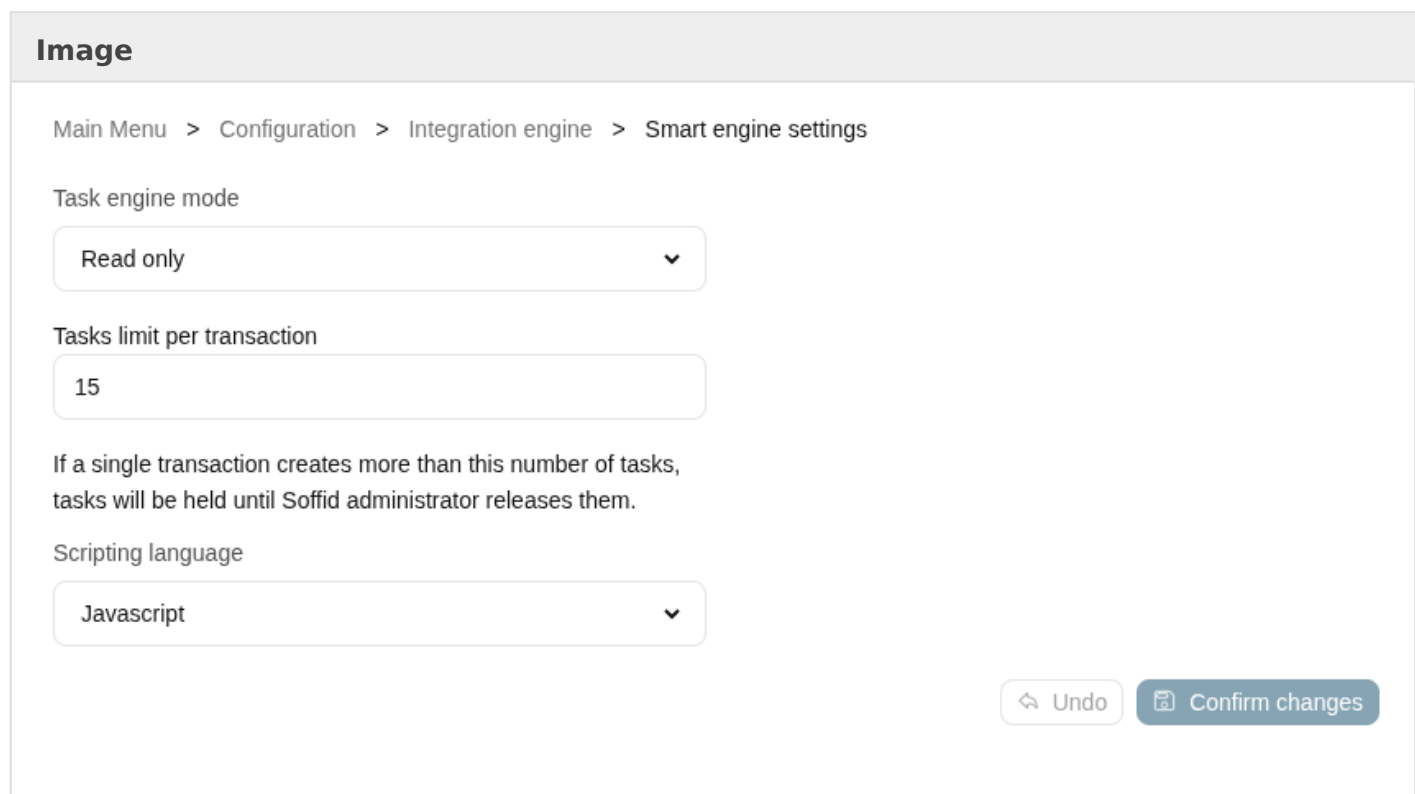
4. Then Sync Server and updates the database table SC_LUNIND by upgrading the LIP_TIMSTA field of the User object or by creating a new record if it did not previously exist.



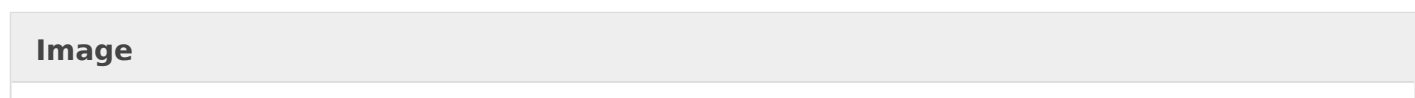
5. When the following search will be performed, the very first thing to do is check the database file. If it is necessary update the file system and finally perform the search.

Example 2

1. The task engine mode is Read only



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



**Rosalind Franklin**

rfranklin



Basics

Groups

Accounts

Roles

Effective
RolesShared
accounts

Sessions

User
processes

Issues

Tokens

OTP
devices

Backups

Expand all

Collapse all



▼ Common attributes :

User name * :

rfranklin

First name * :

Rosalind

Last Name * :

Franklin

Middle name :

Middle name

Full name :

Rosalind Franklin

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

soffid

Url :

local

Task	Priority	Executions	Execution time	Message	Scheduled
index	Search	Search ...	Search ...	Search ...	Search ...
IndexObject com.soffid.iam.base.api.User 197099	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

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
```

Java classes

Interface	Description
<u>QueryParserConstants</u>	Token literal values and constants.

Class Summary

Class	Description
<u>MultiFieldQueryParser</u>	A QueryParser which constructs queries to search multiple fields.
<u>QueryParser</u>	This class is generated by JavaCC.
<u>QueryParserBase</u>	This class is overridden by QueryParser in QueryParser.jj and acts to separate the majority of the Java code from the .jj grammar file.
<u>QueryParserTokenManager</u>	Token Manager.
<u>Token</u>	Describes the input token stream.

Enum Summary

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

Exception Summary

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

Error Summary

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