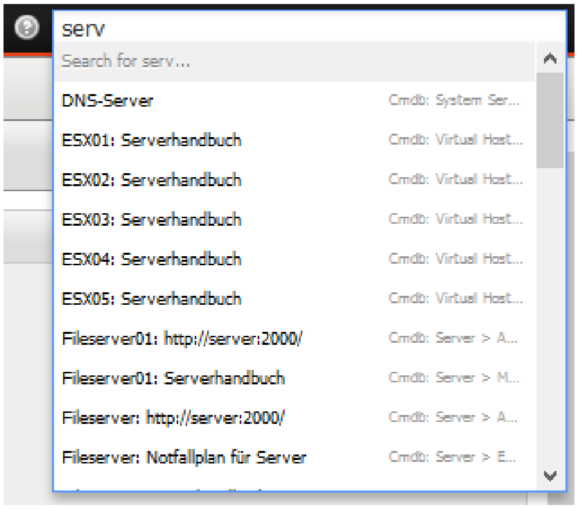
Search

"Seek and you shall find." Specific research in the IT documentation requires a fast and comfortable search. A specific search is often a faster method than clicking through menu structures.

Contents

The Search Field

The search field is located in the web GUI of i-doit in the upper right corner. Here you can search for any term of the IT documentation. While typing, the first results already appear (search suggestions), they can be selected using the mouse or the keyboard (auto-completion) to jump directly to the found data set.



serv

Search for serv...

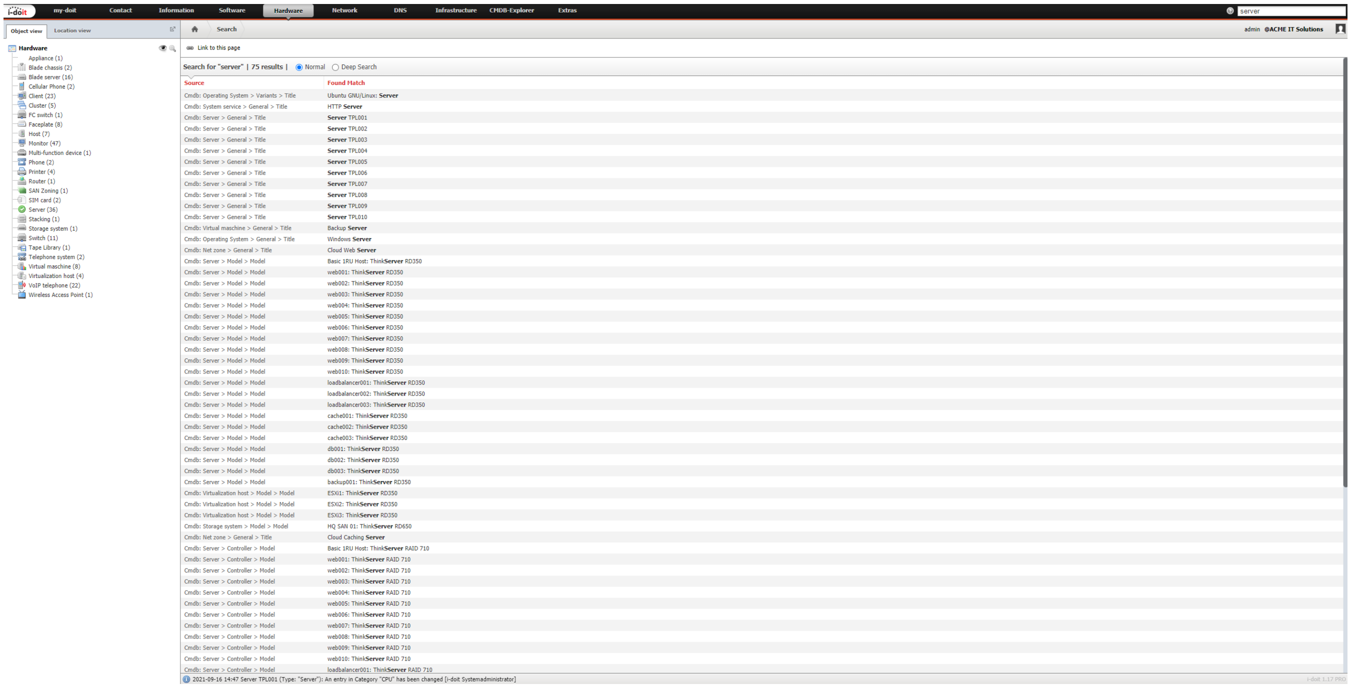
DNS-Server	Cmdb: System Ser...
ESX01: Serverhandbuch	Cmdb: Virtual Host...
ESX02: Serverhandbuch	Cmdb: Virtual Host...
ESX03: Serverhandbuch	Cmdb: Virtual Host...
ESX04: Serverhandbuch	Cmdb: Virtual Host...
ESX05: Serverhandbuch	Cmdb: Virtual Host...
Fileserver01: http://server:2000/	Cmdb: Server > A...
Fileserver01: Serverhandbuch	Cmdb: Server > M...
Fileserver: http://server:2000/	Cmdb: Server > A...
Fileserver: Notfallplan für Server	Cmdb: Server > E...

Apart from the plain text, the search can be restricted to specific object attributes:

- If the search begins with a hash followed by the object ID (`#123`), the overview page of the object associated to this ID will be opened upon pressing the return key.
- If the search begins with the keyword `title` followed by a colon and the object title, the corresponding object will be displayed (`title:acme`).

The Search Results

If the search suggestions mentioned above do not lead to the desired result, then pressing return will present a page with search results. Next to each piece of found data sets also the source is stated, for example, the attribute of a category of an object type in the CMDB module.



Hardware
- Appliance (1)
- Blade chassis (2)
- Blade server (16)
- Cellular Phone (2)
- Client (23)
- Cluster (5)
- FC switch (1)
- Faceplate (8)
- Host (7)
- Monitor (47)
- Multi-function device (1)
- Phone (2)
- Printer (4)
- Router (1)
- SAN Zoning (1)
- SIM card (2)
- Server (36)
- Stacking (1)
- Storage system (1)
- Switch (11)
- Tape Library (1)
- Telephone system (2)
- Virtual maschine (8)
- Virtualization host (4)
- VoIP telephone (22)
- Wireless Access Point (1)

Search for "server" | 75 results | ● Normal ○ Deep Search

Source	Found Match
Cmdb: Operating System > Variants > Title	Ubuntu GNU/Linux: **Server**
Cmdb: System service > General > Title	HTTP **Server**
Cmdb: Server > General > Title	**Server** TPL001
Cmdb: Server > General > Title	**Server** TPL002
Cmdb: Server > General > Title	**Server** TPL003
Cmdb: Server > General > Title	**Server** TPL004
Cmdb: Server > General > Title	**Server** TPL005
Cmdb: Server > General > Title	**Server** TPL006
Cmdb: Server > General > Title	**Server** TPL007
Cmdb: Server > General > Title	**Server** TPL008
Cmdb: Server > General > Title	**Server** TPL009
Cmdb: Server > General > Title	**Server** TPL010
Cmdb: Virtual maschine > General > Title	Backup **Server**
Cmdb: Operating System > General > Title	Windows **Server**
Cmdb: Net zone > General > Title	Cloud Web **Server**
Cmdb: Server > Model > Model	Basic 1RU Host: Think**Server** RD350
Cmdb: Server > Model > Model	web001: Think**Server** RD350
Cmdb: Server > Model > Model	web002: Think**Server** RD350
Cmdb: Server > Model > Model	web003: Think**Server** RD350
Cmdb: Server > Model > Model	web004: Think**Server** RD350
Cmdb: Server > Model > Model	web005: Think**Server** RD350
Cmdb: Server > Model > Model	web006: Think**Server** RD350
Cmdb: Server > Model > Model	web007: Think**Server** RD350
Cmdb: Server > Model > Model	web008: Think**Server** RD350
Cmdb: Server > Model > Model	web009: Think**Server** RD350
Cmdb: Server > Model > Model	web010: Think**Server** RD350
Cmdb: Server > Model > Model	loadbalancer001: Think**Server** RD350
Cmdb: Server > Model > Model	loadbalancer002: Think**Server** RD350
Cmdb: Server > Model > Model	loadbalancer003: Think**Server** RD350
Cmdb: Server > Model > Model	cache001: Think**Server** RD350
Cmdb: Server > Model > Model	cache002: Think**Server** RD350
Cmdb: Server > Model > Model	cache003: Think**Server** RD350
Cmdb: Server > Model > Model	db001: Think**Server** RD350
Cmdb: Server > Model > Model	db002: Think**Server** RD350
Cmdb: Server > Model > Model	db003: Think**Server** RD350
Cmdb: Server > Model > Model	backup001: Think**Server** RD350
Cmdb: Virtualization host > Model > Model	ESXi1: Think**Server** RD350
Cmdb: Virtualization host > Model > Model	ESXi2: Think**Server** RD350
Cmdb: Virtualization host > Model > Model	ESXi3: Think**Server** RD350
Cmdb: Storage system > Model > Model	HQ SAN 01: Think**Server** RD650
Cmdb: Net zone > General > Title	Cloud Caching **Server**
Cmdb: Server > Controller > Model	Basic 1RU Host: Think**Server** RAID 710
Cmdb: Server > Controller > Model	web001: Think**Server** RAID 710
Cmdb: Server > Controller > Model	web002: Think**Server** RAID 710
Cmdb: Server > Controller > Model	web003: Think**Server** RAID 710
Cmdb: Server > Controller > Model	web004: Think**Server** RAID 710
Cmdb: Server > Controller > Model	web005: Think**Server** RAID 710
Cmdb: Server > Controller > Model	web006: Think**Server** RAID 710
Cmdb: Server > Controller > Model	web007: Think**Server** RAID 710
Cmdb: Server > Controller > Model	web008: Think**Server** RAID 710
Cmdb: Server > Controller > Model	web009: Think**Server** RAID 710
Cmdb: Server > Controller > Model	web010: Think**Server** RAID 710
Cmdb: Server > Controller > Model	loadbalancer001: Think**Server** RAID 710

2021-09-16 14:47 Server TPL001 (Type: "Server"): An entry in Category "CPU" has been changed [i-doit Systemadministrator]

Set bookmark/favorite

Each search can be referenced via the URL. Searching the term "acme" results in the URL `http://i-doit/i-doit/search?q=acme`. Search results can be saved as bookmarks/favorites in the web browser in order to be able to access them quicker.

Search Mode

There are two selectable modes for the search which all may lead to different results:

- **`Normal:`** The indexing (see below) is used for the search. This is the default setting.
- **`Deep Search:`** Attributes are scanned one after another. The index is ignored. This search mode takes more time than the others.

The search mode can be selected at **`Extras`** **`Search`** after the first search. At **`Administration`** **`System settings`** **`Tenant settings`** **`Search`** **`Default search mode`** you can select the specific mode you want to use automatically in the future.

Automatic Deep Search

Should a search with the predefined mode (see above) provide an unsatisfactory result or even no result, you can carry out a **`Deep Search`** automatically. You can configure the settings for **`Deep Search`** under **`Administration`** **`System settings`** **`Tenant settings`** **`Search`** **`Automatic DeepSearch`**:

Option	Description
`Active`	The search results are supplemented by an additional **`Deep Search`**.
`Active, when no results are found`	The additional **`Deep Search`** is started when there are no results.
`Deactivated`	The additional **`Deep Search`** is omitted.

Indexing

Initially, the search index is generated automatically when updating or installing. Only for very big databases with more than 500,000 objects it has to be generated manually to ensure that the search is fast and delivers good results. The index can be created or updated in the background regardless of user interactions. This action is triggered via the i-doit Console. An example call and explanations of the possible parameters can be found in the corresponding article.

```
$ php console.php search-index -uadmin -padmin -i 1
```

Alternatively, the Reindex can also be executed in the i-doit administration under **System Tools -> Cache / Database** via the **re-new Search Index** button.

Memory requirement

On a Unix-like operating system the index is 500 MB per 1 million indexed data records. Usually, the duration of a search query is not increased by bigger indexes.

Search via the Console

The CLI tool i-doit controller allows you to search via the command line. The required handler is `search_index` with the `search` parameter. An example:

```
$ sudo -u www-data php console.php search -uadmin -padmin --searchString=acme
+------+-------------------------------------------------------
+------------------------------------------------------------------------+-------+
| ID   | Key                                                   | Found
Match                                                            | Score |
+------+-------------------------------------------------------
+------------------------------------------------------------------------+-------+
| 538  | Database scheme > Global > Title                      | i-doit Tenant ACME IT
Solutions                                        | 4.04  |
| 25   | Organisation > Global > Title                         | ACME IT Solutions
GmbH                                                 | 4.04  |
| 157  | Person groups > Mail Addresses > Title                | Team: intern@acme-it.
example                                           | 4.04  |
| 158  | Persons > Mail Addresses > Title                      | Adam Riese: a.riese@acme-it.
example                                    | 4.04  |
| 160  | Persons > Mail Addresses > Title                      | Mara Thon: m.thon@acme-it.
example                                      | 4.04  |
| 164  | Persons > Mail Addresses > Title                      | Falk Narei: f.narei@acme-it.
example                                    | 4.04  |
| 25   | Organisation > Mail Addresses > Title                 | ACME IT Solutions GmbH: info@acme-it.
example                           | 4.04  |
[...]
+------+-------------------------------------------------------
+------------------------------------------------------------------------+-------+
```

Search via the API

It is also possible to search the IT documentation via the programmable interface (API) of *i-doit*. The required method is `idoit.search`:

```
{
    "version": "2.0",
    "method": "idoit.search",
    "params": {
        "q": "acme",
        "apikey": "c1ia5q",
        "language": "en"
    },
    "id": 1
}
```

The server replies as follows:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "documentId": "1412",
            "key": "Client > Ip > Dns domain",
            "value": "Laptop 001: intern.acme-it.example",
            "type": "cmdb",
            "link": "\/?objID=1412&catgID=47&cateID=47&highlight=acme",
            "score": 0
        },
        // […]
    ],
    "id": 1
}
```

Adjust the Indexing

The indexing of i-doit utilizes many features that are provided by MySQL/MariaDB. These can be customized. For some settings the configuration file of MySQL/MariaDB is adjusted (for example at `/etc/mysql/conf.d/i-doit.cnf`). For other settings it is required to run SQL statements for which the command line client is suitable. Example:

```
mysql -uroot -p -hlocalhost
```

Word Length

An important question for indexing is how long the minimal length of a word has to be. Often this value is set to 3 characters. Terms like "PC 01" will not be found with this value. A suitable specification would be a value of 2 or even 1 character(s).

```
innodb_ft_min_token_size = 2 # minimal number of characters for a searched term
```

This setting could lead to an index which is many times greater than before.

Word Separators

In order to differentiate between words, various characters are used to serve as separators (e.g. space, dot, dash). To find the term "PC-01", the term is separated into "PC" and "01". The word length is again the decisive factor whether "PC" and "01" are indexed or not.

Stop Words

Stop words are the terms which are ignored when searching. Thus the index should not take these words into account. Examples for such words are "at", "that" and "with". MySQL already provides a list of stop words which is relatively small, however, and only contains English terms. This list can be replaced by your own list. You can use the following SQL statements:

```
# Enter the system database of i-doit:
USE idoit_system;

## If the table does not exist yet, it should be created:
CREATE TABLE IF NOT EXISTS isys_search_stopwords (value VARCHAR(18) NOT NULL DEFAULT '') ENGINE=INNODB DEFAULT
CHARSET=latin1;

## Empty the table:
TRUNCATE TABLE isys_search_stopwords;

## Liste von deutschen Stoppwörtern einfügen Insert list of stop words:
INSERT INTO isys_search_stopwords (value)
VALUES ('but'), ('as'), ('also'), ('at'), ('so'), ('on'), ('from'), ('am'), ('until'), ('are'), ('there'),
('by'), ('thus'), ('because'), ('that'), ('the'), ('your'), ('whose'), ('this'),('yet'), ('there'), ('you'),
('through'), ('a'), ('an'), ('he'), ('it'), ('yours'), ('for'), ('had'), ('here'), ('behind'), ('i'),
('their'), ('in'), ('is'), ('yes'), ('each'), ('everyone'), ('everything'), ('can'), ('do'), ('my'), ('mine'),
('with'), ('must'), ('after'), ('afterwards'), ('no'), ('not'), ('now'), ('or'), ('be'), ('his'), ('she'),
('should'), ('shall'), ('sofar'), ('over'), ('and'), ('our'), ('ours'), ('beneath'), ('of'), ('before'),
('when'), ('why'), ('what'), ('continue'), ('further'), ('if'), ('who'), ('become'), ('becomes'), ('how'),
('again'), ('we'), ('to');
```

Lists of stop words in various languages can be found on the internet.

The following settings are required to replace the MySQL list with your list:

```
innodb_ft_server_stopword_table = 'idoit_system/isys_search_stopwords'
```

Activate Index Changes

Three steps are necessary to make changes to the index behaviour effective. First you should restart the MySQL service. The following command is used for Debian-based operating systems:

```
sudo systemctl restart mysql.service
```

Afterwards, the following SQL statement has to be run for each tenant database:

```
OPTIMIZE TABLE isys_search_idx;
```

Complete SQL statement for the first tenant with the database `idoit_data`:

```
USE idoit_data;
OPTIMIZE TABLE isys_search_idx;
```

Finally, re-indexing via the i-doit controller has to be effected (see above).

Increase the limit of 2500 results

If objects are named consecutively e.g. 123456-00001 to 123456-99999 only objects up to 123456-2500 can be found with normal search settings.
This limit is for performance reasons. To increase this limit, an expert setting must be added.

Here is an example where the limit was set to 5000 results.

Key	Value	Type
search.limit	5000	Tenant-wide