# END USER GUIDE

## 1. INTRODUCTION

### 1.1. Purpose

The purpose of this  document is to describe all the features, functionalities and capabilities provided by a properly installed, configured and operated instance of the qto application for the perspective of an application end-user. This document is the CONTRACT between you as the potential, former or active user of an qto instance and the product owner of that instance. Thus, should something here be described to work in a particular way, contrarian of your user experience, you should create an issue and assign it to your instance owner.

### 1.2. Document status

This document is updated constantly in every release of the qto. Each version however is considered to be a complete version regarding the qto version it situated in.

### 1.3. Audience

This document could be of interest for any potential and actual end-users of the qto application.

### 1.4. Master storage and storage format

The master storage of this document is the _doc table of the production database of the instance you are using.

### 1.5. Switch projects by the :to operator in the search-box

If you type the ":to <<database-name>>" you will get a drop down which will list the projects databases , to which your instance has access to, by choosing the database from the list and hitting enter you will be redirected to the same url by on the different database.

### 1.6. Version control

The contents of this document MUST be updated according to the EXISTING features, functionalities and capabilities of the application version, in which this document residues.
The version of this document is the same as the version of the qto application. Place your mouse over the upper left corner displaying the database name this document is served from.

### 1.7. Way of working

The qto provides a mean for tracking of this documentation contents to the source code per feature/functionality, thus should you find inconsistencies in the behaviour of the application and the content of this document you should create a bug issue by clicking on the plus button of the enduser_issues and assign it to the owner of your product instance.

## 2. INITIAL CONCEPTS

A single qto web application might have access to one or many databases ( called also projects ).

### 2.1. Accessing multiple applications from the same UI

If you type the ":to <<database-name>>" you will get a drop down which will list the projects databases , to which your instance has access to, by choosing the database from the list and hitting enter you will be redirected to the same url by on the different database.

### 2.2. Switch projects by the :to operator in the search-box

If you type the ":to <<database-name>>" you will get a drop down which will list the projects databases , to which your instance has access to, by choosing the database from the list and hitting enter you will be redirected to the same url by on the different database.

## 3. LOGIN

You need to sent e-mail to the administrator of the qto application ( just try to login with your e-mail and you will get his/her e-mail in the error msg ).
This section provides initial concepts for new users to quickly grasp the basics of the qto application.

### 3.1. Accessing multiple applications from the same UI

If you type the ":to <<database-name>>" you will get a drop down which will list the projects databases , to which your instance has access to, by choosing the database from the list and hitting enter you will be redirected to the same url by on the different database.

### 3.2. Own password change

After you have the password from the administrator change it immidiately after login from the <<app>>/list/users page. You can see only your own password - once you have updated it it gets stored encrypted in the database and even the user administrator cannot read it in clear text.

### 3.3. Switch items by using the :for operator in the search-box

If you type the ":to <<database-name>>" you will get a drop down which will list the projects databases , to which your instance has access to, by choosing the database from the list and hitting enter you will be redirected to the same url by on the different database.

### 3.4. Important peace of information about YOUR Security

Every user except the administrator can see ONLY his/her own user details. Even the instance administrator cannot see your password, but of course he/she can reset it for you.
The qto by itself DOES NOT store any personal details for tracking you on the web, nor will it sell your personal data. Like NEVER! In fact we recommend providing as little personal data as possible - even by not using real personal names.
The qto application users http session to identify you and technically user tracking could have been implemented to gain some user data, yet from the source code of the application AND from the application documentation is evident that any such attempts, have not been planned, done, nor there are any attempts for the foreseeable future.

### 3.5. Logout from the application

You can logout from the application from the left menu. Click the upper right button to open the left menu and click the bottom "logout" link. This will clear your session for ONLY this project as well.

## 4. THE LIST PAGE

The list page presents part or the whole content of a database by the means of UI controls - dynamic html table, forms etc.
In the context of the qto's parley the "listing" is the ui list of control/(s) you get by using the following URL format:

### 4.1. Viewing tables from different projects ( databases )

Each project in qto is actually stored in it's own database, to the access for example the dev_qto, tst_qto and prd_qto projects( which could be any names, but in this example just happen to be the dev, tst and prd databases for qto ) you should simply add the db name as the first url part:
https://qto.fi:441/dev_qto/list/release_issues
https://qto.fi:442/tst_qto/list/release_issues
https://qto.fi:443/prd_qto/list/release_issues

#### 4.1.1. Viewing the full content of the items

You can quickly view the full content of each cell of the listing table by hovering with the mouse on top of it. Note that all the links in the content are replaced with clickable links in the tooltip as well as the internal links such as the following one : enduser_guide_doc-190214224315 ( which just refer to next item in this document).

#### 4.1.2. Viewing the list page

You can use the pick=col1, col2, col3 url parameter to select for only desired attributes.
You could filter the result the same way the filters for the select page work ( see bellow ).
Should there be errors in the loading of the page, they will be displayed in a msg at the top of the page.

#### 4.1.3. Listing url syntax

The listing url syntax mimics the sql select clause syntax, yet in much more simplified form.

##### 4.1.3.1. The "pick" url param

You can use the pick=col1,col2,col3 url parameter to select for only desired attributes to be show in the ui control used for listing.
The following url demonstrates this syntax:
https://qto.fi/qto/list/yearly_issues?as=grid&pick=id,status,name,description&page-size=5&page-num=1&where=status-eq-09-done

##### 4.1.3.2. The "hide" url param

If you do not specify any attribute to pick, you could hide specific attributes by using the "hide=col1,col2,col3" syntax.
https://qto.fi/qto/list/yearly_issues?hide=description

##### 4.1.3.3. The "with=col-operator-value" filter

You can filter the result of the query by using the "with=col-operator-value".
The following examples demonstrates, which operators are supported.
An error message is shown if you do not use existing operator.
The following url demonstrates this syntax:
http://qto.fi/qto/list/yearly_issues?as=grid&pick=id,status,prio,name,weight,start_time,stop_time&page-size=5&page-num=1&where=status-eq-09-done
Note when going to the next page that all the rows' status is 09-done

```
with=status-eq-09-done
list all the items having the attribute "status" equal to
the "09-done" string

with=prio-lt-7
list all the items having the attribute prio smaller than
the number 7

this is the list of all the operators supported
       'eq' => '='
     , 'ne' => '<>'
     , 'gt' => '>'
     , 'lt' => '<'
     , 'ge' => '>='
     , 'le' => '<='
     , 'like' => 'like'
```

##### 4.1.3.4. The "where=col-operator-value" filter

You can filter the result of the query by using the "where=col-operator-value", which works exactly as the with operator, thus the following examples demonstrates, which operators are supported.
An error message is shown if you do not use existing operator.

```
with=status-eq-09-done
list all the items having the attribute "status" equal to
the "09-done" string

with=prio-lt-7
list all the items having the attribute prio smaller than
the number 7

this is the list of all the operators supported
       'eq' => '='
     , 'ne' => '<>'
     , 'gt' => '>'
     , 'lt' => '<'
     , 'ge' => '>='
     , 'le' => '<='
     , 'like' => 'like'
```

#### 4.1.3.5. Filtering with "like"

The filtering with the like operator translates to the SQL "like" operator- the "like-by=<<attr>>&like-val=<<val>> filtering, where <<attr>> stands for the name of the attribute to use the like operator.
Example:
https://qto.fi/qto/list/yearly_issues?as=grid&oa=prio&pick=id,status,prio,name&page-size=5&page-num=1&where=status-eq-09-done

```
# this example url will list all the monthly_issues items
having the "bug" string in their "name" attribute:
http://host-name:3000/qto/list/monthly_issues?as=grid&like-
by=name&like-val=bug
```

### 4.2. Sorting an item table

The listed table is sortable by clicking on the columns OR by navigating with the tab key on the keyboard on a column and hitting Enter.
The sorted column is visually shown as the active one on page load:
https://qto.fi/qto/list/yearly_issues?as=grid&oa=prio&pick=id,status,prio,name&page-size=5&page-num=1&where=status-eq-09-done

#### 4.2.1. The 'as' url syntax for printing the listing page

By default the url syntax of the list page has the "as=grid" default listing format, if you replace it with the "as=print-table" url parameter you will get a bare listing of the data ( all other sorting and paging parameters work as well ) , which you could use for printing as well.

### 4.3. Quick filtering an item table

You can filter the already presented part of the result set in the page by using the search textbox. This is only an ui type of filtering for the already loaded data. This type of filtering is different compared to the url parameters filtering by using the with url param syntax and it filters the already fetched from the db data-set, whereas the with=<<attribute>><<operator>><<value>> filtering does filter on the database side.
You could focus the quick search textbox by pressing the forward slash on your keyboard.
The quick search box works instantaneously, thus hitting enter is not needed.

### 4.4. Setting the item table paging size

You can set the page size of the result set to be fetched from the database by using the "&page-size=<<page-size>>" url parameter or by clicking on the page sizes links bellow the table.
The default and most convenient table paging size is 7, because it allows quick paging of a small result-set without scrolling on the screen ...

### 4.5. Paging - setting the item table's page number

If the result-set requested is larger than the page size you can go to the next page number by using the "&pg-num=<<page-num>>" url parameter.
You could go to the next page number by clicking on the links just bellow the quick search textbox. The table control has UI for setting the table page number.
The pager shows 10 pages at a "pager page" so getting to the end of hundreds of pages ( depending of course on your page size ) is comparably easy.
You could quickly use the / char shortcut to focus to the quick search box and from there use the tab to quickly navigate to the desired page number.

### 4.6. Item table paging

The table paging is decided by the pg-num=<<page-number>> and the pg-size=<<page-size>> url operators.

### 4.7. Keyboard usability in the list page

The order of all the ui elements of the list-as-grid page has been arranged so that the user could cycle trough the whole page by accessing all the elements quickly.
Power-users will find it extremely convenient to cycle and edit small tables.

#### 4.7.1. Navigability of the list page with the keyboard

You can quickly traverse the cells of the table via the tab key, which does go over the non-editable items too ( the id's ) , so that you could quickly scroll the table as scrolling when the editable is in focus does not work.
The whole table is easily scrollable whenever the cursor is on non-editable part of the table ( the id's column ) and whenever the last rows must be edited the page is scrolled so that the rows are situated in the middle and not the bottom of the screen.

#### 4.7.2. Focus the quick search box

You could focus the quick search by typing / IF you are not editing a cell.
Thus the paging on the next cell is quite handy - as you could easily jump onto the quick search and with couple of tabs navigate to the next page.

#### 4.7.3. Undo the edit on a cell

If you were on a cell and types some text without leaving it, but you change your mind you could simply press the Esc key, which will restore the original content of the cell and you could proceed by tab to the next cell.

#### 4.7.4. Keyboard navigation on the edit form

You could open the edit form with the keyboard while your cursors is on the id button of the item in the grid ( Note that the colour of the button text has to change also. By hitting enter the edit form will open witch the title id selected, from there you could cycle with the tab on each control of the form, thus each time the focus leaves a control the data updated or not is saved to the database. You could close the form by hitting enter when the close button is selected.

### 4.8. New item creation (CREATE)

A new item could be added to the table in the ui and thus in the db table by clicking the plus button above the table ( which uses the google material design ui ).
The new button has a fixed position, thus available during scrolling as well from the same position.
The new button changes it's appears when focused via the keyboard, and can be pressed when in focus by hitting enter with the keyboard.
To practice new items' creations and deletions to get comfortable on the app's behaviour you could use the following table:
https://qto.fi/qto/list/test_create_table?as=grid&pick=id,name&page-size=5&page-num=1&od=id

#### 4.8.1. Successful execution

After clicking the plus button the System adds the new row into the database table and presents it into the table ui AS THE FIRST ROW to emphasise the created row - that is the existing sort of the table is changed to the id column. Note that if you had a defined sorting order before the addition of the new item, it has been replaced by the "order by the latest created" sorting order.

#### 4.8.2. Error handling on list page click create new item action

If any error occurs while the creation an error msg is presented clearly with fading effect, which returns the error msg from the database.
On invalid input the data is not created to the database and nothing is stored.

### 4.9. Item edit (UPDATE)

There are 2 ways to edit an item of the qto application :
- inline edit
- form edit

#### 4.9.1. Form edit

You open the form to edit an item from the id button on the left. A modal dialog containing the filled in details of the item appears. You could either simply cycle via the keyboard trough the items, or edit some of the item details, as soon as any of the controls in the form is selected, after leaving the control the data is saved straight to the database.

#### 4.9.2. In-line edit ( UPDATE )

The grid can be edited inline so that the data is updated to the database.
White space in the cells is preserved.
To practice new items' creations updates and deletions to get comfortable on the app's behaviour please use first the development instances of the qto project:
https://qto.fi/qto/list/monthly_issues?as=grid&pick=id,status,prio,name,weight,start_time,stop_time&page-size=5&page-num=1

##### 4.9.2.1. Table columns resizing

You can resize the columns of the tables to a max size of 500 pixels by dragging only one text area. Note however that the textarea will NOT be resized bigger than the maximum width of the current column …
For the visual outlook of the table a certain default values for certain columns' contents widths are assumed.

##### 4.9.2.2. Contents for the table's cells

The table's cells should accept any UTF-8 characters including html entities.
The textarea's width should adjust automatically till the width of the widest cell in the table column.

##### 4.9.2.3. Successful execution

If the single cell inline-edit is successful no msg is presented and the data is updated to the database storage.
If the updated cell was part of the currently sorted column the ui is automatically adjusted to the new sort order ( for example if a numeric sort was applied and the cell had value of 9 with 1..9 range and the smallest to greatest was currently active if the new update is 1 the item will appear in the top of the listing.

##### 4.9.2.4. Error handling on db update error

If any error occurs while updating an error msg is presented clearly with fading effect, which returns the error msg from the database.
On invalid input the data is not updated to the database and the old value in the cell is restored.

##### 4.9.2.5. Nulls handling

Nulls handling is somewhat problematic in ui. For now the behaviour by convention is to leave a nullable record in the database as null, whether the cell of the ui table is left empty ( white space chars are also considered empty)

### 4.10. Item deletion ( DELETE )

You could delete items by clicking the delete button with the trash icon in the beginning of every item.
To practice new items' creations and deletions to get comfortable on the app's behaviour please use first the development instances of the qto project:
https://qto.fi/qto/list/monthly_issues?as=grid&pick=id,status,prio,name,weight,start_time,stop_time&page-size=5&page-num=1
If the deletion is successful the item is removed both from the ui and from the database. No msg is presented.
The usual way if an error has occurred during the delete the error msg is displayed in the top centre  area of the list page.

### 4.11. List as print-table page

The list as print-table page is aimed at producing quickly refined result-set from the database for a further copy paste on to another html page ( wiki etc. )  or even print to paper.
It has all the functionalities as the list as "table" page, without the filtering from the quick search box and without the ui for the pager and page-sizer - the url params for paging and page-sizing work, however. All the url params work as in the grid listing page.
https://qto.fi/qto/list/yearly_issues?as=print-table&pick=id,status,name,description&page-size=5&page-num=1&like-by=status&like-val=03

## 5. THE VIEW DOC PAGE

The view page presents the data of a database table, having nested-set hierarchical model partially or fully by the means of different controls. The page is shortly called the view-doc page

### 5.1. Navigating in the view doc page

When the page is loaded all the content of the document is presented / according to the urls params / which are applying the same filtering as in the list page , but on the hierarchical data-set …
Should there be an error a dynamic snackbar is presented with the error message. The snackbar hides itself by default after 3.9 seconds ( set by default , but it could be pinned to the page by the user to view it properly / send it further).

### 5.2. Using the TOC menu

The Table of Contents ( TOC ) right menu can be opened and closed and scrolled separately from the scroll of the view doc. Clicking on any of the titles scrolls the document page on the left with the clicked title item scrolled on the top of the browser view port.

### 5.3. Searching the view doc by filtering ( beta )

Once the page is loaded the top search omnibox is selected. As soon as you start typing the document the ui starts shrinking the page by presenting ONLY the items containing the search string ( the search is case-insensitive)

### 5.4. Items interlinking

The items interlinking works both in the list-grid and in the view-doc pages ( the description ).
Any <<item>>-<<id>> where <<id>> is a whole number is converted into the jump to the anchor of that exact item id of that <<item>>.
For example :
enduser_guide_doc-190214224374 will generate a link to this "items interlinking" title ( Hold on the Ctrl key and click on the link in the tooltip on the right of this paragraph.
The interlinking works between the different items as well :
For example:
requirements_doc-5
If the item is not a "doc" item - aka not ending with the _doc the link will redirect the user to a filtered table with only this id:
for example: principles-1805311658

### 5.5. The right table of contents menu

You open and close the right click menu from the upper menu icon. The right click menu presents the structure of the document.You could navigate with the keyboard trough the right menu links too.
Note that when you click on a link on the right menu the title of the item you clicked on is scrolled to the top of the page.
Right click on the title items presents the context menu containing several different options to perform on a branch of the document.
Note, that you can quickly navigate with the tab key on the keyboard once the right menu is open and selected ( aka you have to click on it ), you can hit enter once reaching the desired section of the document.

### 5.6. Managing the view doc data with the right click menu ( beta )

This feature is in beta mode still - expect some quirks along the way ...
If you right click on a number of a title or on the title in the right TOC listing and choosing one of the following options you could:

#### 5.6.1. Open & dive

View the current title and it's tree as a separate document altogether

#### 5.6.2. Open in list

see the current branch of the document as table

#### 5.6.3. Export to md

This option exports this branch as Mark Down document ( GitHub syntax ), Microsoft Azure is supported as well.

#### 5.6.4. Pdf print preview

View the current title's tree as pdf printable document convenient for printing as well ...

#### 5.6.5. Add new parent node

Adding new parent node means simply adding a new item in the document on the upper level in the hierarchy as the one from which you right clicked just after it - thus for example if you right clicked the item with number 4.3. which is the 3rd item on the second level from the 4rth item on the first level you will get the 5.0 item and if the 5.0. item already exist it will be set as the 6.0. and so on ...
This feature does not work for all the cases properly for now ... you might have to reload the page if the ui does not behave according to the description above - this is known bug with WIP status ...

#### 5.6.6. Add new sibling node

Adding new sibling node means simply adding a new item in the document on the same hierarchy level as the one from which you right clicked just after it - thus for example if you right clicked the item with number 4.3. which is the 3rd item on the second level from the 4rth item on the first level you will get the 4.4. item and if the 4.4. item already exist it will be set as the 4.5. and so on ...
This feature does not work for all the cases properly for now ... you might have to reload the page if the ui does not behave according to the description above - this is known bug with WIP status ...

## 6. THE SEARCH PAGE

The search page is presented either after you have search globally from either the view page or the list page by typing your search query and hitting enter. Or by navigating to it from the left menu.

### 6.1. Add new child node

Adding new parent node means simply adding a new item in the document on one level bellow in the hierarchy as the one from which you right clicked just after it - thus for example if you right clicked the item with number 4.3. which is the 3rd item on the second level from the 4rth item on the first level you will get the 4.3.1 item and if the 4.3.1. item already exist it will be set as the 4.3.2. and so on ...
This feature does not work for all the cases properly for now ... you might have to reload the page if the ui does not behave according to the description above - this is known bug with WIP status ...

### 6.1. Cycling though the search results

The search results listing provides a slice of the simple text search from the project database based on the postgres relevance algorithm. If you have more than 7 results you can set the size of the listing, go to the next page of the search result, previous etc.

### 6.1.1. Remove node

Adding new parent node means simply adding a new item in the document on one level bellow in the hierarchy as the one from which you right clicked just after it - thus for example if you right clicked the item with number 4.3. which is the 3rd item on the second level from the 4rth item on the first level you will get the 4.3.1 item and if the 4.3.1. item already exist it will be set as the 4.3.2. and so on ...
This feature does not work for all the cases properly for now ... you might have to reload the page if the ui does not behave according to the description above - this is known bug with WIP status ...

## 6.2. Opening an item from the search result and editing

To open an item from the search results click on the open button on the item row in the search listing - the dialog box contains the details of the item in read-only - to actually edit this item click on the edit button on the top in the dialog box, you will be redirect to either the list item page or the view doc page ui of this item.