

Documentation for mysqlservlet

A simple general purpose servlet to access a mysql database from a web site

[Lucio Chiappetti](#) [IASF Milano](#) 12 Jan 06 14:31

Contents

[Introduction](#)
[Features and functionality](#)
[How to build a web page calling the servlet](#)
[S/W requisites](#)
[Getting the s/w](#)

Introduction

Since long I wished to have a spreadsheet-like interface to access a [mysql](#) database, allowing to view the content of a table, and to edit its content. Since I found nothing which satisfied me, I decided to write a little Java servlet of my own, which allows to access any mysql database table from a web page.

Also I needed something like this for internal affairs inside our institute, so I added some specific, though general-purpose, functionalities.

Features and functionality

mysqlservlet allows, once [installed](#) under a running Java servlet engine, to [build web pages](#) which allow to access one or more mysql database tables, to look at them, and to [edit them](#) (one record at a time).

The main features are :

- The mysql host, database, table and user on whose behalf doing the operation are specified in the [calling web page](#)
- The operation can be controlled by a further internal username (editor name) specified in the [calling web page](#), so that more users can be masked under the same mysql user.
- In normal mode (the only mode if the editor has a name of guest) the user can view the content of the mysql table as an HTML table.
- The (guest) user can also restrict the view to a subset of columns.
- The (guest) user can also sort the table in ascending or descending order on any combination of columns (deciding also the sequence of the columns).
- The (guest) user with some mysql experience can also restrict the view to a range of records, or according to a mysql SELECT condition.
- The (guest) user can also restrict the view to the records which satisfy some predefined conditions on a special "status column".
- Non-guest users may also click on a record (provided the table has an auto increment column) to open it for editing, and edit it as a simple HTML form.
- The content of the form is driven by the mysql types of the columns.
- Edited records can be timestamped and signed with the editor names, if appropriate special columns are defined in the table.
- The default editing action modifies the selected record.
- However the (non-guest) user can select that clicking on a record instead either creates a duplicate, or inserts a new empty record.
- Deletion of records is allowed only indirectly, i.e. assigning to the status column (if present) a value of Deleted, which marks the record for deletion, and later physically committing pending deletions at the click of a button.
- The status column has been defined having in mind three possible states of a record (active, retired or deleted), but other usages are not forbidden.
- The logical associations between "special" columns as used by the servlet and actual column names in the table is controlled in the [calling web page](#).

I will now describe the functionality of the servlet, starting from the look and feel of a typical screen. As explained in [the next section](#) typically one will have a frameset with two frames, the upper one used for login, and the lower one for the output. I show only the lower one, since the upper one has a look which depends on the way [you](#) build [your](#) site !

The typical page for a non-guest user will look like this, with a top "keyboard" (an HTML table with

several buttons and control) followed by the display of the database table (the latter can be quite long and require scrolling with the browser scroll bar).

Display: any <input checked="" type="radio"/> ; non-retired <input type="radio"/> ; non-deleted <input type="radio"/> ; active only <input type="radio"/> ; Apply						Column sort & hide menu	
Edit: current <input checked="" type="radio"/> ; duplicate <input type="radio"/> ; new <input type="radio"/> ; Apply						Commit deletions	
Extra condition <input type="text"/> Apply Reset						Limit from <input type="text"/> to <input type="text"/> Apply	
seq	time	editor	anint	astring	adate	flag	unsup
1	2006-01-11 17:05:53.0	lucio	11	paraponzi	2006-01-01	Yes	UNSUPPORTED FLOAT
2	2006-01-11 17:06:33.0	lucio	22	ponziponzipo	2005-12-31	No	UNSUPPORTED FLOAT
3	2006-01-11 17:07:23.0	lucio	300	trallallero	1982-06-15	No	UNSUPPORTED FLOAT
4	2006-01-11 17:08:14.0	lucio	666	trallalla	1992-07-16	Yes	UNSUPPORTED FLOAT

The buttons and links in the above table are disabled, with the exception of the numbered links in the first (green) column labelled "seq". Clicking on them will show you what the record looks like when opened for edit. Click on the "Dont" button to close the record and return.

The above sample demonstrates also the typical [cell colour coding](#).

For a guest user the top "keyboard" will have some functions disabled (additionally the "Display" panel can be disabled if the table does not contain a status column, irrespective of the user) as shown below.

Display: any <input checked="" type="radio"/> ; non-retired <input type="radio"/> ; non-deleted <input type="radio"/> ; active only <input type="radio"/> ; Apply						Column sort & hide menu	
no editing allowed						no deletions allowed	
Extra condition <input type="text"/> Apply Reset						Limit from <input type="text"/> to <input type="text"/> Apply	

Below I will explain :

- [colour codes](#) for table cells
- [how to sort or hide columns](#)
- [other functions of the keyboard](#)

The [colour codes](#) for the table cells depend on the mysql type (as seen by java) plus some additional interpretation specific of the servlet. Each cell has two possible looks, one in display mode, and one in edit mode.

type	support	look in display mode		look in edit mode	
auto increment	special	13	URL on green	Dont Save	couple of buttons, the second can be Save or Add according to editing mode
editor column	special	manuzio	italic red on default	editing... duplicating... adding new...	message depending on editing mode, direct editing not allowed
boolean	planned	T	white on blue	n/a	currently unsupported
byte	planned	64	darker cyan	n/a	currently unsupported
short	planned	256	dark cyan	n/a	currently unsupported
long	yes	1024	cyan		

float	yes	1.0	dark orange	<input type="text" value="1.23"/>	text area
double	planned	2.0	orange	n/a	currently unsupported
string	yes	xyzyy	default	<input type="text" value="xyzyy"/>	text area
date	yes	1492-10-12	italics on light yellow	<input type="text" value="2038-01-19"/>	text area
time	planned	09:30:05	italics on yellow	n/a	currently unsupported
timestamp	yes	2006-01-11 17:05:53.0	italic red on default	editing... duplicating... adding new...	message depending on editing mode, direct editing not allowed
enumeration	yes	forse che si	white	<input type="text" value="forse che si"/>	menu with ENUM allowed values
set	planned	big,bad	gray	TBD	currently unsupported
anything else	no	UNSUPPORTED TYPE	message on red	unchanged	editing not possible

There is no syntactic check on the value input in a text area (besides the implicit ones for menus or for fields like timestamp or editor which are generated internally).

The rare case when mysql error messages are generated, typically because of incorrect editing, they will appear in red, and should be recoverable in a self-understandable way.

Note however that inputting invalid values in most cases does not generate an error, but the insertion of a null or default value in a cell. Thus, you'd better verify the content of an edited record after editing.

The sorting and hiding of specific columns is controlled by the Columns sort and hide menu button, which opens a screenful like this

Column	Show/Hide	Sort options
seq	compulsory	Sort: No <input type="radio"/> ; Asc <input checked="" type="radio"/> ; Desc <input type="radio"/> as <input type="text" value="3"/>
time	compulsory	Sort: No <input checked="" type="radio"/> ; Asc <input type="radio"/> ; Desc <input type="radio"/> as <input type="text" value=""/>
editor	Show <input checked="" type="radio"/> ; Hide <input type="radio"/>	Sort: No <input type="radio"/> ; Asc <input checked="" type="radio"/> ; Desc <input type="radio"/> as <input type="text" value="2"/>
anint	Show <input checked="" type="radio"/> ; Hide <input type="radio"/>	Sort: No <input type="radio"/> ; Asc <input checked="" type="radio"/> ; Desc <input type="radio"/> as <input type="text" value="1"/>
astring	Show <input checked="" type="radio"/> ; Hide <input type="radio"/>	Sort: No <input checked="" type="radio"/> ; Asc <input type="radio"/> ; Desc <input type="radio"/> as <input type="text" value=""/>
adate	Show <input type="radio"/> ; Hide <input checked="" type="radio"/>	Sort: No <input checked="" type="radio"/> ; Asc <input type="radio"/> ; Desc <input type="radio"/> as <input type="text" value=""/>
flag	Show <input type="radio"/> ; Hide <input checked="" type="radio"/>	Sort: No <input type="radio"/> ; Asc <input checked="" type="radio"/> ; Desc <input type="radio"/> as <input type="text" value="4"/>
unsup	Show <input type="radio"/> ; Hide <input checked="" type="radio"/>	Sort: No <input checked="" type="radio"/> ; Asc <input type="radio"/> ; Desc <input type="radio"/> as <input type="text" value=""/>

The example shown indicates that we have requested to hide the last three columns, and to sort in ascending order on columns (in this order) anint,editor,seq,flag.

Note that sorting on an hidden column is allowed.

The request will be processed as soon as the user clicks on one of the "Apply choices ..." buttons.

Note that some special columns cannot be hidden.

The top three buttons (outside of the table with the column list) are used as follows:

- Show all columns is equivalent to clicking all "Show" radio buttons and applying.
- Unsort all is equivalent to clicking all Sort "No" radio buttons
- Cancel returns to the table display without any change

The meaning of the radio buttons in the table with the column list should be self explanatory (sorting can occur in ascending or descending order).

The numbered menu is used instead as a sequence number to select the priority in which the columns selected for sorting are used. It is set to blank for unsorted columns, and remembers the previous value for sorted columns. However if all columns are unsorted, it assumes the default order of the column in the table.

If you click on Asc or Desc to sort an unsorted column and leave the sequence number blank, such columns will be sorted last.

If you assign the same sequence number to more columns, they will be sequenced in alphabetic order of the name.

The other functions of the top keyboard are as follows :

- The "Display" area is enabled if a status column is defined in the [calling web page](#). However if the status column is not present in the file other than with the "Any" radio button will generate an error.
It allows to select for display only the records whose status is not Retired, not Deleted or Active (other values are not considered, use extra condition below).
Click on "Any" to display all records.
After clicking on a radio button do not forget to Apply it.
- The sort and hide button has been explained [above](#).
- The "Edit" area is enabled for non guest users. It allows to specify whether an editing call (clicking on the links in the green column) for a given record will
 - edit the current record in place
 - make a duplicate of the current record, and edit that
 - make a new empty record and edit that

After clicking on a radio button do not forget to Apply it.
... and do not forget in which editing mode you are !

Note that an edited record once closed will appear in the display in the position corresponding to the table sorting criteria (or last if none), not necessarily close to where it was before !

Note also that edited record are always assigned an updated timestamp (if a timestamp column is present), an updated "editor signature" (if an editor column is present) and a new sequence number (if the record has been duplicated or added), without interaction with or control of the user.

- The "Commit deletions" button, enabled when editing is allowed, will physically and irrecoverably remove records marked for deletion (i.e. with the status column previously set to Deleted).
No further confirmation is requested, so use with care !
- The "Extra condition" area allows to enter a mysql condition to further limit the output, e.g. `anint < 100` and `astring like 'p%'`. You need to know mysql to use it. Be prepared to handle syntax error messages.
Such condition is implicitly ANDed with other implicit conditions (e.g. those from the "Display" panel).
After entering a condition do not forget to Apply it.

Note that the extra condition is remembered, but is not displayed !

In case of doubt reset it entering a blank condition and pressing Apply. This will get rid of the condition altogether.

The Reset button will not clear the remembered condition, but just, hopefully, clear the entire form ... hopefully, because with a form using javascripts and dynamic content, one never knows.

- The "Limit" area is used to restrict the number of records shown, and is applied last, as mysql users know. So if you say limit from 5 to 7 and issue a query displaying 10 entries, it will display the 5th, 6th and 7th of them (not an absolute position in the file ! if you change other criteria and the new query produces 6 entries, only the 5th and 6th will appear, if it produces 4 entries, none will appear !).
The only difference with the mysql LIMIT clause is that this one works from and to a (relative) position and not "offset and for" like native mysql (which is less natural, I'm a real (Fortran) programmer and never count from zero to nine fingers).
Both limits shall be entered, if one is left empty, limits are reset.
After entering a condition do not forget to Apply it.

Note that also the limits are remembered, but not displayed !

In case of doubt reset then entering blank values and pressing Apply. This will get rid of the limits altogether.

How to build a web page calling the servlet

You can access any database table, or family of tables (in a mysql database which you are entitled to access) building just two web pages (at least, I do it with two) :

- The first page is just a frameset container

```
<FRAMESET ROWS="10%,*" BORDER=0>
  <FRAME SRC="login.html" NAME=UP MARGINWIDTH=0 MARGINHEIGHT=0>
  <FRAME SRC="blank.html" NAME=DOWN>
</FRAMESET>
```

The aesthetics and the frame names are optional (in [this colour](#)), what matters is that you have two frames and that they are named UP and DOWN (the latter taking most of the screen). The top frame is initialized with the login page described below, while the bottom frame is initialized with anything (I use a [blank.html](#) file containing a single i.e. a non-breaking space ... OK, so I do work with three web pages). The bottom frame is where the servlet will display its output.

- The login page, named [login.html](#) in the example above, shall be arranged like this

```
<FORM method=POST action="http://yourhost:8080/servlet/mysqlservlet" target="DOWN">
<INPUT TYPE=HIDDEN NAME="mysqlhost" VALUE="localhost">
<INPUT TYPE=HIDDEN NAME="mysqladb" VALUE="dbname">
<INPUT TYPE=HIDDEN NAME="mysqluser" VALUE="anusername">
<INPUT TYPE=HIDDEN NAME="mysqlpwd" VALUE="optional">
<INPUT TYPE=HIDDEN NAME="source" >
Table: <SELECT NAME="mysqltable">
  <OPTION VALUE="sample">sample
  <OPTION VALUE="test">test
</SELECT>
<INPUT TYPE=HIDDEN NAME="seqColumn" VALUE="seq">
<INPUT TYPE=HIDDEN NAME="timeColumn" VALUE="modified">
<INPUT TYPE=HIDDEN NAME="statusColumn" VALUE="status">
<INPUT TYPE=HIDDEN NAME="editorColumn" VALUE="editor">
Username:
<INPUT type="text" size=8 maxlength=8 name="editor" value="guest" >
<INPUT TYPE=SUBMIT VALUE="Login" onclick=this.form.source.value="login">
<INPUT TYPE=SUBMIT VALUE="Logout" onclick=this.form.source.value="logout">
(to change table please logout and login again)
</FORM>
```

i.e. as a form whose action is the path to the servlet (you shall insert your hostname here), and whose other controls are explained below.

- The important thing is that you do not change the names of the controls, and do not remove the hidden **source** and the two **Login** and **Logout** buttons.
- The other controls may be at your discretion :
 - hidden input, as exemplified in the majority of cases above when you want to freeze the parameter value to some predefined content
 - a text area, as exemplified for **editor** above, when you allow your user to choose freely the value
 - a menu select, as exemplified for **mysqltable** above, when you restrict the user choice to a few predefined values
 - in some cases missing altogether (a default is supplied)
- An explanation of the various input controls follows:
 - The first 4 parameters are used to establish the JDBC connection with mysql. The security of such connection is established assigning permissions within the mysql server and is none of our business.
 - **mysqlhost** shall contain the hostname fo the mysql server. I have always used only localhost for maximum security (i.e. mysql and servlet run on the same machine). Compulsory, no default, typically hidden.
 - **mysqladb** shall contain the mysql database name. Compulsory, no default, usually hidden unless you give choice to the user.
 - **mysqluser** is the mysql user under which the connection is authenticated. Compulsory, no default. Hidden if you want all mysql operations to occur on behalf of a single username, otherwise you may let each user enter a personal username in an input area or menu.
 - **mysqlpwd** is the mysql password associated to **mysqluser**. It can be omitted altogether if you want to let the servlet derive it under a standard rule (which I won't reveal it, but should be clear if you look in the code ... otherwise ask me). My preference is to use a single mysql user with such a standard password. But you might want to read it from a form.

- **mysqltable** is the mysql table within **mysqlldb** to be accessed. Compulsory, no default. Typically a choice within a menu.
- **editor** is a further user name which can be used to "stamp" each edited record, if this is enabled via **editorColumn**. It can be omitted altogether or set to null, if you want the editor to default to **mysqluser**.
There is a special value of guest : when the editor is guest, editing of the mysql tables is disabled, and the user can only browse existing tables but not edit their records.
- the following 4 controls are relevant to column names in the current mysql table, so I assume that all tables accessed via one form will share the same definition and usage of those columns.
- **editorColumn** gives the name of a column which will contain the stamp (or signature) of the **editor** which modified a record. It can be omitted altogether or set to none if you want to disable signing.
The column in the table should be a CHAR or VARCHAR typically of 8 characters.
- **statusColumn** gives the name of a column which contains the record status. A record status of Deleted is used for deferred deletion of records (explained in [features](#)). It can be omitted altogether or set to none if your table does not contain such a column.
The absence of such a column (or setting **statusColumn** to none) inhibits physical deletions of files from a table
- **seqColumn** gives the name of the auto increment column in the table. If omitted defaults to seq. The presence of such a column in the table is mandatory if you want to be able to edit it.
If your form points (explicitly or implicitly) to a column not existing in the table, editing will be disabled.
- **timeColumn** gives the name of the timestamp column in the table. If omitted defaults to time. If such column is present in the file it will be timestamped automatically at each edit (if it is missing no action)

S/W requisites

I have developed and tested the servlet under Linux SuSE 9.2, using as Java compiler and runtime environment to be specified in the CLASSPATH (j2ee.jar) the version j2ee-1.4.

In addition I use as servlet environment jsdk2.1 (whose server.jar and servlet.jar shall be prepended to the CLASSPATH).

In addition again I use as JDBC mm.mysql.jdbc-1.2c (whose org directory shall also be prepended to the CLASSPATH).

To install the servlet

- [Get](#) the java source,
- set the CLASSPATH,
- compile it,
- install mysqlservlet.class in the webpages/WEB-INF/servlets directory of your JSDK,
- and start the JSDK with its startserver script.
- then [write a web page](#) invoking the servlet and
- test it

Good luck !

If you want to experiment with other java stuff you are welcome to do so, but you will be rolling on your own. Eventually keep me posted.

Getting the s/w

The source code for the servlet is available in file [mysqlservlet.java](#).

The documentation is represented by the current page.

