Zaval Database Front-end

Version 1.2

User’s Guide

Zaval Creative Engineering Group

http://www.zaval.org

Contents

Introduction to the Zaval Database Front-end

The Zaval Database Front-end solution is the SQL console query tool for relational databases, for which the JDBC (Java Database Connectivity) drivers exist.

The Zaval Database Front-end tool is a light-weight, web-based client, best used by advanced software engineers and database administrators for casual executing the raw SQL code and SQL scripts against the database without having to launch the database-specific administrative clients or tools.

What Can You Do with Zaval Database Front-end

The Zaval Database Front-end provides all facilities that the target database you have connected to allows to perform with the SQL queries. The exact enumeration, however, is database-specific. For all widely used databases, such as PostgreSQL, DB2, Oracle, Sybase, MySQL, any databases that can be accessed through JDBC-ODBC bridge and many more the tool can perform all spectrum of querying and database administration.

The Zaval Database Front-end gives you the same features like a vendor-supplied query and administrative tool. However, it is much lighter and is not bound to exact database.

When To Use The Zaval Database Front-end

The Zaval Database Front-end is best used for casual, irregular access to various databases. If you can write the SQL query yourself, then perhaps you will enjoy that you don’t need to install several DBMS clients on your workstation and that you execute your query against the database faster than with vendor-supplied client.

Another great area of use is the remote access to database via HTTP or HTTPS protocols. Since this tool is a Web-based client, developers and administrators may connect to the database via the SSL configured for all major Web servers with JSP 1.2 compliant engine installed (Resin, Tomcat, WebSphere, WebLogic).

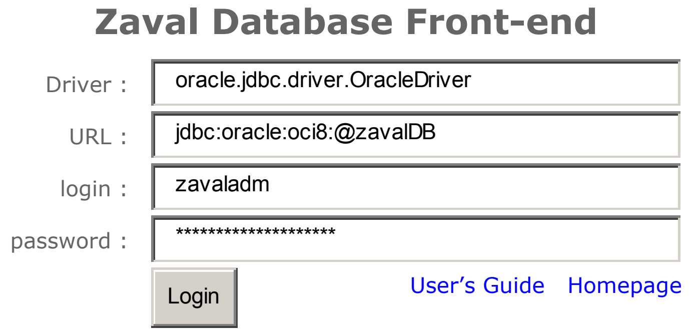
Installing the Zaval Database Front-end

To install the Zaval Database Front-end on your JSP1.2 compliant Web Server, you need to upload the files from the JSP bundle to any configured Web application. We suggest that you create a separate Web application for this tool. When the files upload is complete, the Zaval Database Front-end is ready to use.

In your Web applications classpath settings, you should indicate the paths to libraries of the JDBC drivers that will be available for this application. At present the tool does not perform the search for available JDBC drivers automatically. Therefore, you should check that the database drivers that you plan to query are available in the classpath.

Connecting to the database

In order to log into the database, you have to provide information about database driver, database URL, and your database credentials. The Database Front-end tool receives this information from you using a form dialog, given on the Screenshot 1:



Zaval Database Front-end

Driver :	oracle.jdbc.driver.OracleDriver
URL :	jdbc:oracle:oci8:@zavalDB
login :	zavaladm
password :	********************

Screenshot 1. Database Login Information

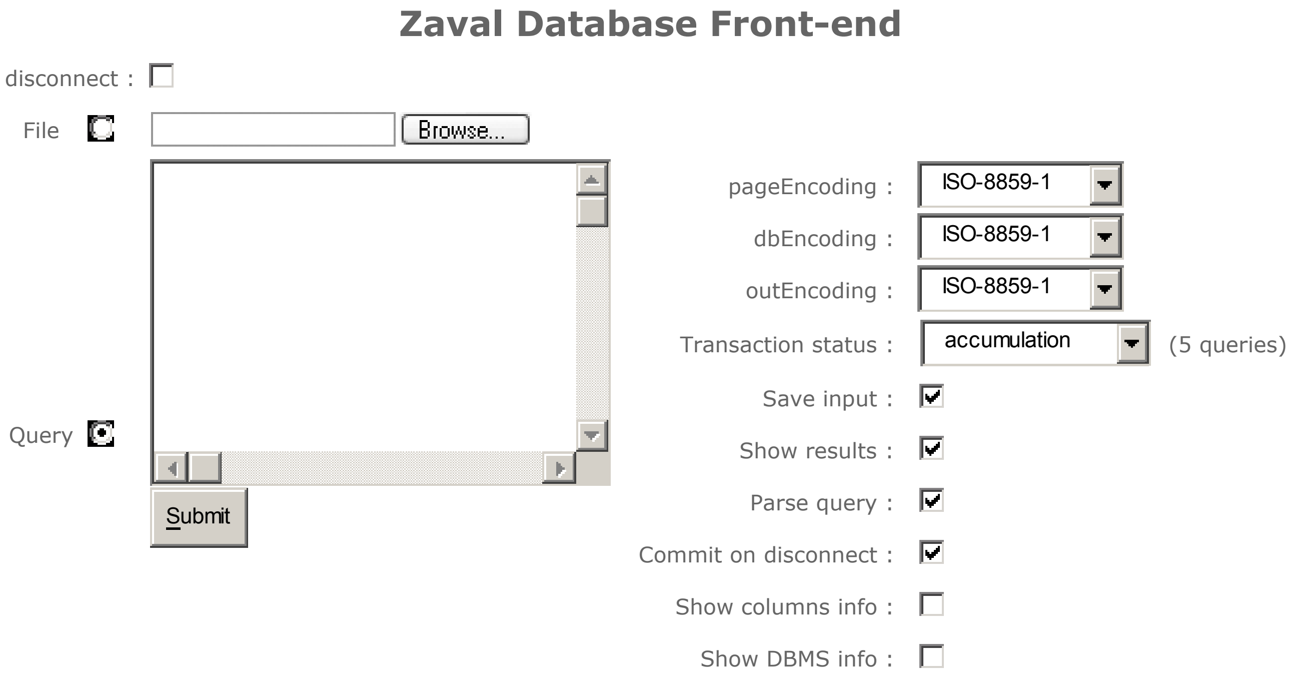
All fields' titles are self-explanatory:

- Driver – specify driver to use when connecting to the database. Note: this driver should be in your Web-application's classpath – see *Installing the Zaval Database Front-end*.
- URL – JDBC URL (see JDBC specification).
- login & password – usage is obvious.

To make life easier, the list of the most popular DBMS drivers and URL formats is provided below this form. When clicking on a driver and a URL link, the corresponding values will be inserted into the appropriate fields above.

Operating the SQL Console

The SQL console offers a simple and intuitively understandable interface. The console that will be available to you when you have logged in is displayed on the Screenshot 2:



Zaval Database Front-end

disconnect : ☐

File ○

Query ●

pageEncoding :	ISO-8859-1
dbEncoding :	ISO-8859-1
outEncoding :	ISO-8859-1
Transaction status :	accumulation (5 queries)
Save input :	☑
Show results :	☑
Parse query :	☑
Commit on disconnect :	☑
Show columns info :	☐
Show DBMS info :	☐

Screenshot 2. SQL Console

There are following features available:

- pageEncoding – this drop-box allows you to set encoding for the html-page you see. This feature is very useful when you are operating with non-English and/or UTF-8 data
- dbEncoding – this is the database internal data representation. For example, when creating database in DB2, you can set to store all your data in UTF/cp1252 or any other encoding different from ISO-8859-1
- outEncoding – this is internal java data representation. Usually you do not need to change this parameter.
- Transaction status – indicates current transaction status if supported.
- Save input – this checkbox allows you to save previously executed query in form (very useful).
- Commit on disconnect – allows to set transaction behavior on disconnect if transaction is not finished yet.
- Show columns info – allows you to see data types and some other info about every field in the table when making select queries.
- Show DBMS info – shows you all available database info.

Query Parse feature

Query parse is the most useful feature available: it allows you to execute several queries delimited by the ';' symbol at once, even if database does not support such functionality. Actually, it parses incoming string and submits each query separately, but for user all this looks like all queries are submitted in one step. By default this feature is enabled. Do not turn it off if you are not sure what you are doing. It is safe to leave this option checked.

Transactions Support

The most valuable feature of the nowadays databases is transactions support. In the current version we make it possible to use transactions – you can start it, than do whenever you want and afterwards save changes or reject them. This is really useful when you are working with the remote database and the connection is really bad. Once it is lost and you were working with one of the City Bank databases (for example) you have no reason to worry about – the changes won't be committed, so data integrity won't be broken.

Once driver you are using for database connection indicates that transactions support is available you'll see **Transaction status** select tab. When it is set to **commit** all queries are committed automatically after the whole batch is executed. If you set it to **accumulate** transaction won't be committed and number of queries in transaction will be displayed before you set it to **commit** manually. When it is set to **rollback** current transaction will be roll backed.

Note: if you set transaction to **commit**/**rollback** and insert any query to the queries area they’ll be executed in the current transaction (before **rollback**/**commit** operation).

When you disconnect from the database front-end’s behavior depends on the **Commit on disconnect** checkbox status. When it is checked all changes you made during the transaction will be committed to the database. Otherwise transaction will be roll backed.

Note: disconnect is not the same as connection timeout – transaction will be committed only if you manually disconnected from the database by using **disconnect** checkbox – see *Disconnecting from the database* section.

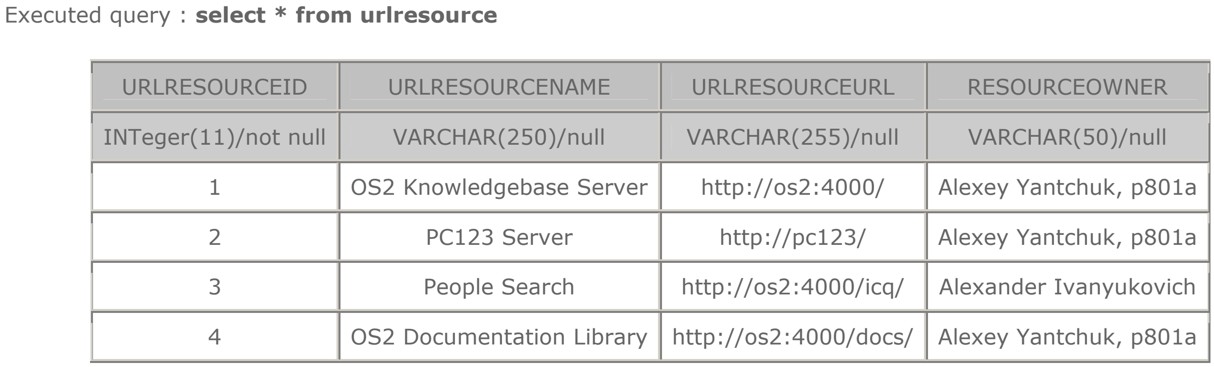
Caution: on any failure during query execution the whole transaction will be roll backed by default. This behavior can be changed – see *Customizing the product* section.

Disconnecting from the database

In order to disconnect from the database, you need to check the disconnect checkbox (see Screenshot 2) and submit a form. If a query is not empty, it will be executed with results will be displayed (if Show Results checkbox is checked) and the database will be disconnected. Submitting an empty query simply disconnects you from the database.

What you receive using the tool

Sample query result set with Show columns info is given on the Screenshot 3:



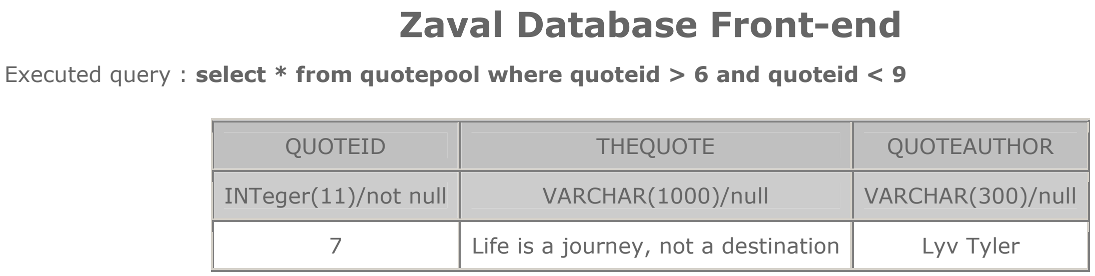
Executed query : **select * from urlresource**

URLRESOURCEID	URLRESOURCENAME	URLRESOURCEURL	RESOURCEOWNER
INTeger(11)/not null	VARCHAR(250)/null	VARCHAR(255)/null	VARCHAR(50)/null
1	OS2 Knowledgebase Server	http://os2:4000/	Alexey Yantchuk, p801a
2	PC123 Server	http://pc123/	Alexey Yantchuk, p801a
3	People Search	http://os2:4000/icq/	Alexander Ivanyukovich
4	OS2 Documentation Library	http://os2:4000/docs/	Alexey Yantchuk, p801a

Screenshot 3. Sample query results

The first row of the table indicates the table column names, the second one gives the column information.

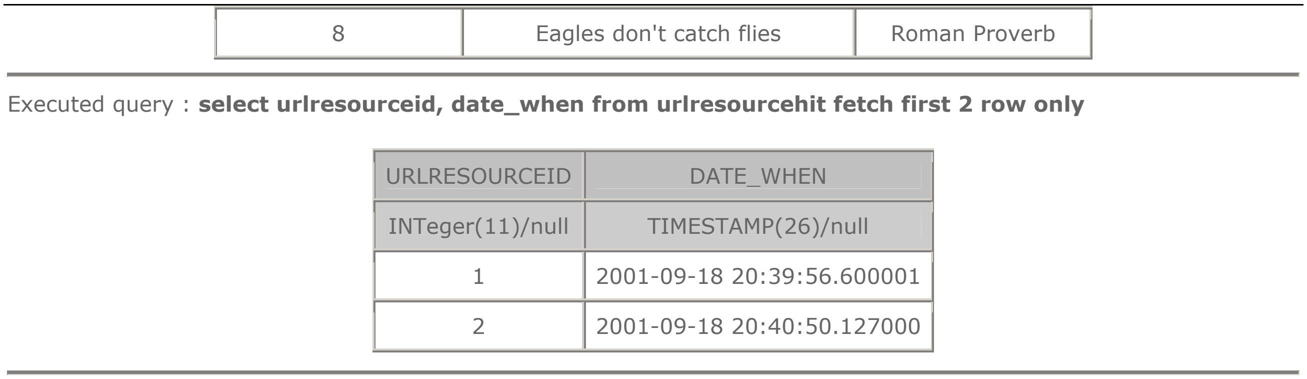
Sample results when you use the parse query feature are given on the Screenshot 4:



Zaval Database Front-end

Executed query : **select * from quotepool where quoteid > 6 and quoteid < 9**

QUOTEID	THEQUOTE	QUOTEAUTHOR
INTeger(11)/not null	VARCHAR(1000)/null	VARCHAR(300)/null
7	Life is a journey, not a destination	Lyv Tyler



8	Eagles don't catch flies	Roman Proverb

Executed query : **select urlresourceid, date_when from urlresourcehit fetch first 2 row only**

URLRESOURCEID	DATE_WHEN
INTeger(11)/null	TIMESTAMP(26)/null
1	2001-09-18 20:39:56.600001
2	2001-09-18 20:40:50.127000

Screenshot 4. Multiple result sets

Usage patterns

There are some situations when this tool will be the best one:

- Remote database administration – you have a database that can’t be updated remotely (probably, for some security reasons - this is a common situation for many hosting providers), but you need to do it fast and have a JSP engine up and running over there. The Zaval Database Front-end is running on the same host as a database - it’s just a local application and there are no limitations for it. For better security you can use it over HTTPS.
- Multiple database administration within one tool – you are database administrator/software developer and occasionally have to deal with several databases. This tool can give you pretty fast access to any of your databases without separate huge GUI installation for each database.

Feel free to invent any other usage patterns and mail them to us.

Customizing the product

There are several features that can be customized:

- pageEncodings list
- dbEncodings list
- outEncodings list
- queryDelimiter symbol
- file (file name – now it’s a single jsp-file)
- rollbackOnFailure option

We tried to make all options self-explanatory, so there are no comments on them here. All these variables are stored as String arrays/char symbol/String respectively at the top of the jsp-file. The source code of the tool is well-commented to make desired changes easy.

Further product plans

Current tool implementation follows the minimalist computing concept. In near future the following features will be added:

- Embedded JNDI-JDBC support;
- Configurable database driver parameters;
- Classpath independent driver loading;
- Driver upload and install.

Support available

All support for software installation and problems should be sent directly to support@zaval.org with “Re: Zaval Database Front-end Support” in subject line and plain text in the message body, describing your request and/or your problem. Since this software is distributed under the General Public License and is maintained by its authors on non-commercial basis, your request will be answered as soon as possible, but no later than 5 business days.

The Zaval Creative Engineering Group carries out its software customization/new software development on the regular basis. For more info contact us at info@zaval.org.