NAME − MQdb::Database

DESCRIPTION

Generalized handle on an DBI database handle. Used to provide an instance which holds connection information and allows a higher level get_connection/ disconnect logic that persists above the specific DBI connections. Also provides a real object for use with the rest of the toolkit.

SUMMARY

MQdb::Database provides the foundation of the MappedQuery system. Databases are primarily specified with a URL format. The URL format includes specification of a driver so this single method can select among the supported DBD drivers. Currently the system supports MYSQL, Oracle, and SQLite. The URL also allows the system to provide the foundation for doing federation of persisted objects. Each DBObject contains a pointer to the Database instance where it is stored. With the database URL and internal database ID, each object is defined in a global space.

Attributes of MQdb::Database

driver : mysql, oracle, sqlite (default mysql)
user : username if the database requires
password : password if the database requires
host : hostname of the database server machine
port : IP port of the database if required
(mysql default is 3306)
dbname : database/schema name on the database server
for sqlite, this is the database file

Example URLS

mysql://<user>:<pass>@<host>:<port>/<database_name>
mysql://<host>:<port>/<database_name>
mysql://<user>@<host>:<port>/<database_name>
mysql://<host>/<database_name>
oracle://<user>:<pass>@/<database_name>
oracle://<user>:<pass>@<host>:<port>/<database_name>
sqlite:///<database_file>

CONTACT

Jessica Severin <jessica DOT severin AT gmail DOT com>

LICENSE

```
* Software License Agreement (BSD License)
* MappedQueryDB [MQdb] toolkit
* copyright (c) 2006−2009 Jessica Severin
* All rights reserved.
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
*     * Redistributions of source code must retain the above copyright
*       notice, this list of conditions and the following disclaimer.
*     * Redistributions in binary form must reproduce the above copyright
*       notice, this list of conditions and the following disclaimer in the
*       documentation and/or other materials provided with the distribution.
*     * Neither the name of Jessica Severin nor the
*       names of its contributors may be used to endorse or promote products
*       derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS ''AS IS'' AND ANY
* EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS BE LIABLE FOR ANY
* DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

APPENDIX

The rest of the documentation details each of the object methods. Internal methods are usually preceded with a _

new

```
Description: instance creation method
Parameter  : a hash reference of options same as attribute methods
Returntype : instance of this Class (subclass)
Exceptions : none
```

new

```
Description: instance creation method
Returntype : instance of this Class (subclass)
Exceptions : none
```

init

```
Description: initialization method which subclasses can extend
Returntype : $self
Exceptions : subclass dependent
```

new_from_url

```
Description: primary instance creation method
Parameter  : a string in URL format
Returntype : instance of MQdb::Database
Examples   : my $db = MQdb::Database->new_from_url("mysql://<user>:<pass>@<host>
             e.g. mysql://<host>:<port>/<database_name>
             e.g. mysql://<user>@<host>:<port>/<database_name>
             e.g. mysql://<host>/<database_name>
             e.g. sqlite:///<database_file>
my $class = shift;
Exceptions : none
```

copy

```
Description: makes a copy of the database configuration.
             New instance will have its own database connection
Returntype : instance of MQdb::Database
```

dbc

```
Description: connects to database and returns a DBI connection
Returntype : DBI database handle
Exceptions : none
```

disconnect

```
Description: disconnects handle from database, but retains object and
             all information so that it can be reconnected again
             at a later time.
Returntype : none
Exceptions : none
```

full_url

```
Description: returns the URL of this database with user and password
Returntype : string
Exceptions : none
```

url

```
Description: returns URL of this database but without user:password
             used for global referencing and federation systems
Returntype : string
Exceptions : none
```

xml

```
      Description: returns XML of this database but without user:password
                   used for global referencing and federation systems
      Returntype : string
      Exceptions : none
```

execute_sql

```
      Description    : executes SQL statement with external parameters and placeholde
      Example        : $db->execute_sql("insert into table1(id, value) values(?,?)", $
      Parameter[1]   : sql statement string
      Parameter[2..] : optional parameters for the SQL statement
      Returntype     : none
      Exceptions     : none
```

do_sql

```
      Description    : executes SQL statement with "do" and no external parameters
      Example        : $db->do_sql("insert into table1(id, value) values(null,'hello w
      Parameter      : sql statement string with no external parameters
      Returntype     : none
      Exceptions     : none
```

fetch_col_value

```
      Arg (1)    : $sql (string of SQL statement with place holders)
      Arg (2...) : optional parameters to map to the placehodlers within the SQL
      Example    : $value = $self->fetch_col_value($db, "select some_column from my_ta
      Description: General purpose function to allow fetching of a single column from
      Returntype : scalar value
      Exceptions : none
      Caller     : within subclasses to easy development
```