

RDBI External API specification 1.0 draft

Erik Hollensbe <erik@hollensbe.org>

Table of Contents

1	All Classes	1
2	module DBI	2
3	class DBH	3
3.1	Query Methods	3
4	class STH	4
5	class Pool	5
6	class Result	6
7	class CursorResult < Result	7
8	class Row	8
9	Schema	9
10	Column	10
	Method Index	11

1 All Classes

Boolean reload

[Method on All Classes]

this method will semantically refresh items, such as Schema objects or rows, depending on the context of the object in question.

2 module DBI

DBH connect (*Class klass, Array *args, Proc &block*) [Method on DBI]

class is a ruby class which corresponds to the database driver. it is no longer a string.

*args is a hash with parameter -> value associations, such as :host or :username.

Optionally yields a block for usage, yields a freshly connected DBH.

Array of Class drivers [Method on DBI]

accessor to get at known classes that can be used as drivers.

DBH connect_cached (*Class klass, Array *args*) [Method on DBI]

connect to a new resource if one is required (or desired, see below) with similar parameters as connect().

additional arguments :pool_name and :pool_size can be used to define a Pool (object, see below) which holds a specific subset of connected database handles. Playing with the size here introduces the ability for connect_cached to maintain a minimum number of connections which can be re-used over the lifetime of a program.

Pool pool (*String pool_name*) [Method on DBI]

a pool as described above is an array of database handles. this returns that data as a "Pool" object, with its own API. See later on in the document.

Pool all_connections [Method on DBI]

similar to pool(), this returns all the connections, but ignores pools.

Integer ping (*Class klass, Array *args*) [Method on DBI]

similar to connect(), this issues a ping to the databases. This may issue a connect() before the ping() to do it properly depending on the database implementation.

Boolean reconnect_all [Method on DBI]

reconnects all the known database handles.

DBH last_dbh [Method on DBI]

returns the last returned dbh from connect() or connect_cached()

this method, by definition, can be unpredictable in threaded environments.

3 class DBH

NilClass <code>transaction</code> (<i>Proc &block</i>)	[Method on DBH]
opens a transaction and executes the statements in the block. Yields self.	
Schema <code>table_schema</code> (<i>Symbol table_name</i>)	[Method on DBH]
returns information about a specific table in a Schema object	
Array of Schema <code>schema</code> (<i>Symbol schema_name</i>)	[Method on DBH]
returns information about a specific schema, the current one if none is specified.	
Boolean <code>reconnect</code>	[Method on DBH]
reconnects to the database	
Integer <code>ping</code>	[Method on DBH]
attempts to contact the database, measuring round-trip.	
Object <code>driver</code>	[Method on DBH]
returns the underlying driver.	
String <code>last_query</code>	[Method on DBH]
returns the last query executed or prepared.	
STH <code>last_sth</code>	[Method on DBH]
returns the last statement handle prepared.	
Mutex <code>mutex</code>	[Method on DBH]
returns the mutex for this database. thread management will be per-dbh.	
String <code>preprocess_query</code> (<i>String query</i>)	[Method on DBH]
preprocesses the query and returns what it would look like right before it gets sent to the database.	
Boolean <code>disconnect</code>	[Method on DBH]
disconnects from the database. returns success.	
Symbol <code>bind_style</code> (<i>Symbol of [native, preprocessed] style</i>)	[Method on DBH]
Accessor. Native style delegates to the underlying database connector. preprocessed means we do it.	

3.1 Query Methods

these methods all optionally use a block and yield a result or sth depending on context:

Result <code>select</code> (<i>String query, Array *binds</i>)	[Method on DBH]
performs a query and returns the result.	
Result <code>select_one</code> (<i>String query, Array *binds</i>)	[Method on DBH]
performs a query and returns the result, only yielding the first set of items.	
STH <code>prepare</code> (<i>String query</i>)	[Method on DBH]
prepares a query for execution and returns a statement handle.	
Boolean <code>execute</code> (<i>String query, Array *binds</i>)	[Method on DBH]
blindly executes a query and returns some idea of success.	

4 class STH

String <code>query</code>	[Method on STH]
accessor for the query that was used to generate this sth.	
Result <code>execute (Array *binds)</code>	[Method on STH]
executes the prepared statement. optionally yielding a result if block given.	
Object <code>driver</code>	[Method on STH]
if any, returns the underlying statement handle from the database object.	
Result <code>last_result</code>	[Method on STH]
Returns the last Result this prepared statement has yielded.	
Boolean <code>finish</code>	[Method on STH]
finishes the statement	
DBH <code>dbh</code>	[Method on STH]
returns the dbh this statement handle was created from.	

5 class Pool

Boolean reconnect [Method on Pool]

attempts to reconnect the entire pool of database connections.

Integer ping [Method on Pool]

attempts to ping and average the response time of all database connections.

Boolean disconnect [Method on Pool]

disconnects all the database connections in the pool.

6 class Result

Array binds [Method on Result]

accessor for the binds that created this method

Object fetch (*Integer row_count*, [*Class kind*, *Array *args*]) [Method on Result]

fetches one row, or given an argument, *row_count* rows. If given a Class and arguments, uses it to interpret the array. The class is constructed with the arguments provided at the end, and then a method also called `fetch()` is attempted with the row count, with the number of arrays fetched from the DB passed to it.

Especially for specific class designations, (XML formatting is a good example) output formats may not necessarily equate to a single row, in that case, one "unit" should be returned, and this entailings of this unit should be specified.

If this data is not provided, `fetch` yields a standard array with type converted items.

If the *row_count* is `":all"`, fetches all outstanding rows.

Boolean finish [Method on Result]

finishes the underlying statement handle and invalidates the data. reloading will no longer be possible once this is called and should raise (or maybe we should reprepare/execute?).

STH sth [Method on Result]

returns the statement handle that yielded this result.

Schema schema [Method on Result]

returns a Schema object that corresponds to the data in this result.

7 `class CursorResult < Result`

This class is jus a cursor-oriented method of transmitting results.

8 class Row

row is just an array, but this needs to be thought out a little more.

9 Schema

Array of Column columns [Method on **Schema**]
returns column information (see Column object below) for all elements of the Schema.

Array of Symbol table_names [Method on **Schema**]
returns table names (there may be more than one in the event of a query Schema)
for all the objects a part of this Schema.

10 Column

String name [Method on `Column`]

String type [Method on `Column`]
this is the type the database yields

Class ruby_type [Method on `Column`]
Accessor. this is what ruby thinks this type should be, or you can set it directly which will be used at type conversion time.

Integer precision [Method on `Column`]
(alias: `length`) precision is the first number in a database type. it is aliased to the method `'length'` because sometimes that's what precision actually is depending on the type.

Integer scale [Method on `Column`]
scale is the second number in a database type. this is often the right side of a decimal value or sometimes a factoring quotient.

Boolean nullable? [Method on `Column`]
can this column be null?

String metadata [Method on `Column`]
metadata is a bucket for things we don't understand; namely things like AUTOINCREMENT.

String default [Method on `Column`]
default is the column default – this is provided for informational aspects only and should not be used for anything sane.

Method Index

A

all_connections on DBI 2

B

bind_style on DBH 3

binds on Result 6

C

columns on Schema 9

connect on DBI 2

connect_cached on DBI 2

D

dbh on STH 4

default on Column 10

disconnect on DBH 3

disconnect on Pool 5

driver on DBH 3

driver on STH 4

drivers on DBI 2

E

execute on DBH 3

execute on STH 4

F

fetch on Result 6

finish on Result 6

finish on STH 4

L

last_dbh on DBI 2

last_query on DBH 3

last_result on STH 4

last_sth on DBH 3

M

metadata on Column 10

mutex on DBH 3

N

name on Column 10

nullable? on Column 10

P

ping on DBH 3

ping on DBI 2

ping on Pool 5

pool on DBI 2

precision on Column 10

prepare on DBH 3

preprocess_query on DBH 3

Q

query on STH 4

R

reconnect on DBH 3

reconnect on Pool 5

reconnect_all on DBI 2

reload on All Classes 1

ruby_type on Column 10

S

scale on Column 10

schema on DBH 3

schema on Result 6

select on DBH 3

select_one on DBH 3

sth on Result 6

T

table_names on Schema 9

table_schema on DBH 3

transaction on DBH 3

type on Column 10