**Mysql2 - A modern, simple and very fast Mysql library for Ruby - binding to libmysql**

The Mysql2 gem is meant to serve the extremely common use-case of connecting, querying and iterating on results. Some database libraries out there serve as direct 1:1 mappings of the already complex C API's available. This one is not.

It also forces the use of UTF-8 [or binary] for the connection [and all strings in 1.9, unless Encoding.default\_internal is set then it'll convert from UTF-8 to that encoding] and uses encoding-aware MySQL API calls where it can.

The API consists of two classes:

Mysql2::Client - your connection to the database

Mysql2::Result - returned from issuing a #query on the connection. It includes Enumerable.

**Installing**

gem install mysql2

You may have to specify --with-mysql-config=/some/random/path/bin/mysql\_config

**Usage**

Connect to a database:

# this takes a hash of options, almost all of which map directly

# to the familiar database.yml in rails

# See http://api.rubyonrails.org/classes/ActiveRecord/ConnectionAdapters/MysqlAdapter.html

client = [Mysql2](http://www.rubydoc.info/gems/mysql2/0.3.11/Mysql2)::[Client](http://www.rubydoc.info/gems/mysql2/0.3.11/Mysql2/Client).[new](http://www.rubydoc.info/gems/mysql2/0.3.11/Mysql2/Client#initialize-instance_method)(:host => "localhost", :username => "root")

Then query it:

results = client.query("SELECT \* FROM users WHERE group='githubbers'")

Need to escape something first?

escaped = client.escape("gi'thu\"bbe\0r's")

results = client.query("SELECT \* FROM users WHERE group='#{escaped}'")

You can get a count of your results with results.count.

Finally, iterate over the results:

results.each do |row|

# conveniently, row is a hash

# the keys are the fields, as you'd expect

# the values are pre-built ruby primitives mapped from their corresponding field types in MySQL

# Here's an otter: http://farm1.static.flickr.com/130/398077070\_b8795d0ef3\_b.jpg

end

Or, you might just keep it simple:

client.query("SELECT \* FROM users WHERE group='githubbers'").each do |row|

# do something with row, it's ready to rock

end

How about with symbolized keys?

# NOTE: the :symbolize\_keys and future options will likely move to the #query method soon

client.query("SELECT \* FROM users WHERE group='githubbers'").each(:symbolize\_keys => true) do |row|

# do something with row, it's ready to rock

end

You can get the headers and the columns in the order that they were returned by the query like this:

headers = results.fields # <= that's an array of field names, in order

results.each(:as => :array) do |row|

# Each row is an array, ordered the same as the query results

# An otter's den is called a "holt" or "couch"

end

**Cascading config**

The default config hash is at:

[Mysql2](http://www.rubydoc.info/gems/mysql2/0.3.11/Mysql2)::[Client](http://www.rubydoc.info/gems/mysql2/0.3.11/Mysql2/Client).[default\_query\_options](http://www.rubydoc.info/gems/mysql2/0.3.11/Mysql2/Client#default_query_options-class_method)

which defaults to:

{:async => false, :as => :hash, :symbolize\_keys => false}