Ruby Client for REST API

This article will explain how to use Ruby bindings for REST API.

Prerequisites

- Basic knowledge of Treasure Data, including the Toolbelt.
- A table with some data. An example is provided in the What TD can do for you.
- Basic knowledge of our query language.
- Ruby 1.9 or newer

Installation

The Ruby bindings are included in the `td` (stands for 'T'reasure 'D'ata) gem. Add the following line to your Gemfile.

```
gem 'td', "~> 0.10.22"
```

The source code is available at GitHub.

List Databases and Tables

The following example lists the databases and tables. The API key is your authentication key.

```
require 'td'
require 'td-client'
cln = TreasureData::Client.new(ENV['TREASURE_DATA_API_KEY'])
cln.databases.each { |db|
  db.tables.each { |tbl|
    p tbl.db_name
    p tbl.table_name
    p tbl.count
  }
}
```

Issue Queries

The example below issues a query from a Ruby program. The query API is asynchronous — you can check for query completion by polling the job periodically (e.g. by issuing job.finished? calls).

```
require 'td'
require 'td-client'
cln = TreasureData::Client.new(ENV['TREASURE_DATA_API_KEY'])
job = cln.query('testdb', 'SELECT COUNT(1) FROM www_access')
until job.finished?
  sleep 2
  job.update_progress!
end
job.update_status!  # get latest info
job.result_each { |row| p row }
```

job.result_each(&block) does not put the job result into memory. It iterates through the rows in a streaming fashion.

List and Get the Status of Jobs

The following example lists and gets the status of jobs.

```
require 'td'
require 'td-client'
cln = TreasureData::Client.new(ENV['TREASURE_DATA_API_KEY'])

# recent 20 jobs
cln.jobs.length

# recent 127 jobs of specific status
cln.jobs(0, 127, 'running')
cln.jobs(0, 127, 'success')
cln.jobs(0, 127, 'error')
cln.jobs(0, 127, 'killed')

# get job status
cln.job job_id

# get job result
cln.job_result job_id
```