AMAZON
ELASTICSEARCH
FOR RUBY ON
RAILS

By Spencer Peloquin
Curtis Senior High School
Engineering Goal. My engineering goal is to create a framework on Ruby for the Amazon ElasticSearch Relational Database. I tested the framework on Windows 10 and Android. My port operates online in a controlled environment using a package manager such as RubyGems, or offline through sideloading. This offers greater security and portability as a piece of software.

In the end, my goal of creating an efficient, cross-platform, database was successful, as it can run on ARM and WebAssembly, as well as traditional x86-64 based processors without problems.
Portage of API Reference Data

Problems:
- Proprietary Software
- High CPU Usage
- Uses assembly code (hinders portability)

Benefits:
- Safer (fewer vulnerabilities)
- Efficient use of CPU
- Supports more Architectures (ARM, webAssembly)

REST API Usage:

Search Query → Database Server (Offline Cache) Server tasks → Reconnect to server
Update cache
Return feedback to server
Open Channels
Features of ElasticSearch.net

- One-to-one mapping with REST API
- Generalized, pluggable architecture
- Configurable, automatic discovery of cluster nodes.
- Persistent, Keep-Alive connections.
- Load balancing across all available nodes.
- Child client support.
- TypeScript support out of the box.

Usage Statistics
Performance

Project Runtime Comparison

ECMA Compliance

Library Bloat
SEARCH DATA (300 SEARCHES)
module Elasticsearch

  module Rails

  # This module adds support for displaying statistics about search duration in the Rails application log
  # by integrating with the `ActiveSupport::Notifications` framework and `ActionController` logger.
  #
  # == Usage
  #
  # # Require the component in your `application.rb` file:
  #
  # # require 'elasticsearch/rails/instrumentation'
  #
  # # You should see a output like this in your application_to_in development environment:
  #
  # # Active Search (321.3 ms) index: "articles", type: "article", body: { query: { } }
  #
  # # Also, the total duration of the request to Elasticsearch is displayed in the Rails request breakdown:
  #
  # # Completed 200 OK in 615ms (Views: 230.9ms | ActiveRecord: 0.0ms | Elasticsearch: 321.3ms)
  #
  # # @note The displayed duration includes the HTTP transfer -- the time it took Elasticsearch
  # # to process your request is available in the `response.took` property.
  #
  # # @see Elasticsearch::Rails::Instrumentation::Publishers
  # # @see Elasticsearch::Rails::Instrumentation::Rallie
  #
  # # @see http://api.rubyonrails.org/classes/ActiveSupport/Notifications.html
  #
  #
  module Instrumentation end
end
LINKED LIBRARIES (VS CODE 2022)
## 7.2.1

* The default git branch `master` has been renamed to `main`.
* Adds compatibility with Ruby 3 [Pull Request](https://github.com/elastic/elasticsearch-rails/pull/992).

## 7.2.0

* Updates specs and dependency to use with `elasticsearch` v7.14.0.
* Update README, remove Virtus (unmaintained).
* Updates `Bundler.with_clean_dev` (deprecated) to `with_unbundled_env` [commit](https://github.com/elastic/elasticsearch-rails/commit/e4545e4fe2a1ce80009205c831d5740365ba8d6c2).
* Deal with `nil` document types in MultiModel [commit](https://github.com/elastic/elasticsearch-rails/commit/cd9c89b78ed443de376099841bba34276d).
* Update dependency to explicitly support version 7 [commit](https://github.com/elastic/elasticsearch-rails/commit/65942e3a392eab0262e665e69c8e6a99494da9408).
* `removed` in `FasE` to open checking existence of indices [commit](https://github.com/elastic/elasticsearch-rails/commit/7ec0ca39f5d77171b89e7c7f38e372).
* Removed `Form page` setup on `especially` [commit](https://github.com/elastic/elasticsearch-rails/commit/cf57cc21e304b0a44f123c1599f37f892a5d93a).
* Removes dependency on extensions [commit](https://github.com/elastic/elasticsearch-rails/commit/e0070b8329ca48b4cb12b513ac81ed78c88bacc61).
* Fixes basic template `elasticsearch` dependency [commit](https://github.com/elastic/elasticsearch-rails/commit/a44c07b2d097545ca41c13686e9cbf9eb9e639).

### ActiveModel

* Fixes indexing to use right logger in client.
* Updates ES client spec for client 7.14.0.
* Updates transport references.

### 7.1.1

* Fix: Ruby 2.7 deprecation warning on `find_in_batches`.
* Updates README for generating app with template. Related: #938.

### ActiveModel
Test and Redesign

- The prototype used a lot of memory and relied heavily on 3rd party software and had a less permissive license. The need for a redesign was evident from the beginning. However, feedback from on AWS allowed me to improve the efficiency of the framework.

- The first thing I did was to create a template on visual studio that would allow me to effectively rewrite the framework. Using original code and help from AWS, I managed to reduce the library size to under a Megabyte. I then compiled the repository for approval from rubygems.

- The result was extremely positive, with a 50% faster compiling time, and a 20% faster interpretation period.

Future Research

- I have published the gem to the RubyGems Repository under the name “ElasticSearch-Rails-Origin.”
- Next, I may try to implement a RESTful API to make networking more efficient.
- Finally, I would spend more time debugging and reducing vulnerabilities.


License

• The MIT License (MIT)

• Copyright © 2022 Spencer Peloquin

• Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

• The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

• THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
JSON Output (Rails)

```ruby
module Search
  class TagSerializer
    include FastJsonapi::ObjectSerializer
    attributes :id, :name, :hotness_score, :supported, short_summary, :rules_html, :time
  end
end

tag = Tag.find(id)
serialized_data = Search::TagSerializer.new(tag).serializable_hash.deepinspect(data, :attributes)
SearchClient.index(id: tag.id, index: "tags_development", body: serialized_data)
SearchClient.get(id: tag.id, index: "tags_development")
```
```ruby
module Search
  class TagSerializer
    include FastJsonapi::ObjectSerializer

    attributes :id, :name, :hotness_score, :supported, short_summary, :rules_html, :time_90
  end
end

tag = Tag.find(id)
serialized_data = Search::TagSerializer.new(tag).serializable_hash.dig(:data, :attributes)
SearchClient.index(id: tag.id, index: "tags_development", body: serialized_data)
SearchClient.get(id: tag.id, index: "tags_development")
```
class TagSerializer
  include FastJsonapi::ObjectSerializer

  attributes :id, :name, :hotness_score, :supported, short_summary, :rules_html, :time_120

end

tag = Tag.find(id)
serialized_data = Search::TagSerializer.new(tag).serializable_hash.dig(:data, :attributes)
SearchClient.index(id: tag.id, index: "tags_development", body: serialized_data)

SearchClient.get(id: tag.id, index: "tags_development")