

Ruby Response Format

Using Solr's Ruby output

Solr has an optional Ruby response format that extends its [JSON output](#) in the following ways to allow the response to be safely eval'd by Ruby's interpreter:

- Ruby's single quoted strings are used to prevent possible string exploits
 - \ and ' are the only two characters escaped...
 - unicode escapes not used... data is written as raw UTF-8
- nil used for null
- => used as the key/value separator in maps

Here is a simple example of how one may query Solr using the Ruby response format:

```
require 'net/http'

h = Net::HTTP.new('localhost', 8983)
hresp, data = h.get('/solr/select?q=iPod&wt=ruby')
rsp = eval(data)

puts 'number of matches = ' + rsp['response']['numFound'].to_s
#print out the name field for each returned document
rsp['response']['docs'].each { |doc| puts 'name field = ' + doc['name'] }
```

For simple interactions with Solr, this may be all you need! If you are building complex interactions with Solr, then look at some of the libraries below. Craig L Russell Note: in Ruby 1.9, the encoding of the response may be set to ASCII-8BITS even though Solr responds with UTF-8. See [Ruby issue 2567](#).

Libraries

Active

- [RSolr](#): A lightweight, general purpose client library for Solr.
 - [Sunspot](#): A nice DSL framework for integrating Solr into your models. Built on top of RSolr.
- [DeI Solr](#): A Solr wrapper designed to simplify facet and filter queries.
- [Blacklight](#): A popular and well maintained Ruby on Rails framework for building flexible and attractive front-ends to Solr.

Inactive

- [solr-ruby](#): Low-level access to Solr from Ruby.
- [Flare](#): A plugin adding faceted browsing, AJAX suggest and more to Rails controllers.
- [acts_as_solr](#): A plugin to add full text search capabilities using Solr to [ActiveRecord](#) models. [nice acts_as_solr tutorial](#)
- [acts_as_background_solr](#): An extension to acts_as_solr moving more processing to the background.
- [Searchable](#): A Rails plugin that provides search integration with [ActiveRecord](#). (an alternative to acts_as_solr)

[Category](#)[Query](#)[Response](#)[Writer](#)