



co-presented by  & O'REILLY

# Enhancing the Search Box

Greg Gershman

Baltimore, MD

[greg@shelrick.com](mailto:greg@shelrick.com)

<https://github.com/greggersh>

## Enhancing the Search Box

- Search is an integral part of every application
- The purpose of search is to help the user find what they are looking for as quickly as possible
- What can we do to make it more user-friendly, to help the user find what they are looking for quicker
  - Autocomplete
    - **the big lebowski**
  - Autosuggest
    - the big **lebowski**
  - Instant Search

## Autocomplete - Google

- [Google](#)
- [Search.U.S.A.gov](#)

## Autosuggest - Facebook

- [Facebook](#)
- [Quora](#)
- [Bing](#)

## Algorithms

- Autocomplete
  - Prefix search
  - `Movie.where(:conditions => ["title LIKE '?%'", title])`
- Autosuggest
  - More complicated
  - `Movie.where(:conditions => ["title LIKE '?%'" OR title LIKE '% ? %'", title, title])`
  - Edge N-grams
    - requires more advanced processing/architecture
  - Can be more free-text based as well (Facebook/Quora)

## Autocomplete in Rails - Plugins

- Plugins
  - The originals:
    - [https://github.com/david-kerins/auto\\_complete](https://github.com/david-kerins/auto_complete) (Prototype/Rails 2)
    - [https://github.com/chris/auto\\_complete\\_jquery](https://github.com/chris/auto_complete_jquery) (jQuery/Rails 2)
  - Rails 3
    - <https://github.com/crowdint/rails3-jquery-autocomplete>
    - Very simple: [rails3-jquery-autocomplete](#)
    - Examples:
      - <https://github.com/crowdint/rails3-jquery-autocomplete-app>
      - [https://github.com/greggersh/rails3\\_jquery\\_autocomplete\\_example](https://github.com/greggersh/rails3_jquery_autocomplete_example)
    - Has some cool Cucumber integration

## Autocomplete Plugin with jQuery in Rails3

```
# Gemfile
gem 'haml'
gem 'jquery-rails', '>= 1.0.3'
gem 'rails3-jquery-autocomplete'

# run
bundle install
rails generate jquery:install --ui
rails generate autocomplete

# app/views/layouts/application.html.haml
= javascript_include_tag "autocomplete-rails.js"

# app/controllers/home_controller.rb
class HomeController < ApplicationController
  autocomplete :movie, :title #, :full => true

  def index
  end
end

# config/routes.rb
get "home/autocomplete_movie_title" => "home#autocomplete_movie_title"

# app/views/home/index.html.haml
= autocomplete_field_tag 'title', '', home_autocomplete_movie_title_path

Autocomplete
```

## Autocomplete - Roll your own

- jQuery/jQuery UI
- More flexibility in defining the autocomplete/autosuggest method
- More flexibility in tuning performance, caching, display, actions
- JSONP (more on this later)
- <http://jqueryui.com/demos/autocomplete/>
- Highlighting: <http://stackoverflow.com/questions/2435964/jqueryui-how-can-i-custom-format-the-autocomplete-plug-in-results>

## Autocomplete - jQuery UI Example

```
# app/models/movie.rb

def self.autosuggest(query)
  find(:all, :conditions => ['title LIKE ? OR title LIKE ?', "#{query}%", "% #{query}
%"], :limit => 10, :order => 'title ASC, release_year DESC')
end

# app/controllers/movies_controller.rb

def autosuggest
  suggestions = Movie.autosuggest(params[:term])
  render :text => suggestions.collect{|suggestion| { :id => suggestion.slug, :label =>
suggestion.display_title, :value => suggestion.display_title } }.to_json
end

# config/routes.rb

get "autosuggest" => "movies#autosuggest"

# app/views/layouts/application.html.haml

#search-box
= form_tag search_path, :method => :get do
  = text_field_tag 'query', @query, :name => "query", :autocomplete => "off", :class =>
'autosuggest', :size => 50
```

## Autocomplete - jQuery UI Example

```
# public/javascripts/application.js

function monkeyPatchAutocomplete() {
  var oldFn = jQuery.ui.autocomplete.prototype._renderItem;

  jQuery.ui.autocomplete.prototype._renderItem = function( ul, item ) {
    var re = new RegExp(this.term, "i");
    var t = item.label.replace(re, "<span style='font-weight:bold;color:Blue;'>" +
      "$&" +
      "</span>");
    return $( "<li></li>" )
      .data( "item.autocomplete", item )
      .append( "<a>" + t + "</a>" )
      .appendTo( ul );
  };
}

jQuery(document).ready(function() {
  monkeyPatchAutocomplete();
  $( ".autosuggest" ).autocomplete({
    source: "/autosuggest",
    minLength: 2,
    delay: 50,
    select: function( event, ui ) {
      location.href = "/movies/" + ui.item.id;
    }
  });
});
```

## Optimize

- Autocomplete/Autosuggest has to be fast.
- The faster the better
- Use Metal Controller
  - <http://piotrsarnacki.com/2010/12/12/lightweight-controllers-with-rails3/>
- Filter out logging for autocomplete/autosuggest requests
  - <http://stackoverflow.com/questions/5264038/how-do-you-suppress-logging-for-one-particular-action>
  - <http://stackoverflow.com/questions/2196828/how-can-i-disable-logging-in-ruby-on-rails-on-a-per-action-basis>

```
class AutocompleteController < ActionController::Metal
  include ActionController::Rendering

  def index
    suggestions = Movie.autosuggest(params[:q])
    render :text => suggestions.to_json
  end
end
```

## Optimize

- Sunspot/Solr
- Edge N-grams
  - th | the | the\_ | the\_b | the\_bi | the\_big | ...
- Multiple Edge N-grams for phrases
  - th | the | the\_ | the\_b | ... | bi | big | big\_ | big\_l | big\_le | ...
- Can mix in tokenization, parsing of punctuation
- <http://www.lucidimagination.com/blog/2009/09/08/auto-suggest-from-popular-queries-using-edgengrams/>
- <https://github.com/greggersh/sunspot>

## Autosuggest with Solr/Sunspot

```
# Autosuggest
# app/models/movie.rb
# indexes ["th", "the", "the ", "the b", "the bi", "bi", "big", "big "]
searchable do
  autosuggest :title_suggestion, :multiple => true do |movie|
    movie.split_title_on_first_ws
  end
end

def split_title_on_first_ws
  phrases = []
  t = self.title.split
  for i in 0..(t.size - 1)
    phrases << t[i..(t.size - 1)].join(" ")
  end
  phrases
end

def self.autosuggest(query, num_results = 5)
  Movie.search do
    adjust_solr_params do |params|
      params[:q] = "title_suggestion_asm:\#{query}\"
    end
    paginate :page => 1, :per_page => num_results
  end
end
```

## Autosuggest with Solr/Sunspot

```
# Autocomplete
# app/models/movie.rb
# indexes ["th", "the", "the ", "the b", "the bi"]
searchable do
  autocomplete :title_suggestion, :using => :name
end

def self.autocomplete(query, num_results = 5)
  Movie.search do
    adjust_solr_params do |params|
      params[:q] = "title_suggestion_ac:\#{query}\"
    end
    paginate :page => 1, :per_page => num_results
  end
end
```

## Sunspot/Solr

- Obviously, I'd like to clean that up a bit.

## Bells and Whistlers

- Spell checking/autocorrection
- Highlighting
- One click search, custom results

## JSONP

- If you want to do autocomplete/autosuggest across domains, as a service or as a widget, you have to use JSONP.
- JSONP stands for 'loophole.'

```
# JSONP with jQuery UI autocomplete
# public/javascripts/application.js
# within your jQuery autocomplete code
...
url: http://example.com/autocomplete?q= + request.term,
dataType: "jsonp",
data: {
  featureClass: "P",
  style: "full",
  maxRows: 12,
  name_startsWith: request.term
},
...

# app/controllers/autocomplete_controller.rb

def index
  suggestions = Movie.autocomplete(params[:q])
  render :text => "#{params['callback']}({#{suggestions.to_json})}"
end
```

## Testing

- spec/requests
- Sauce Labs
- Check out the cucumber integration from <https://github.com/crowdint/rails3-jquery-autocomplete>

## Questions? Comments? Beer?

- [greg@shelrick.com](mailto:greg@shelrick.com)
- <https://github.com/greggersh>
- <http://search.usa.gov>



co-presented by  & O'REILLY



