

CS 683 Emerging Technologies
Fall Semester, 2006
Doc 24 Rails 8 More AJAX
Nov 21, 2006

Copyright ©, All rights reserved. 2006 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

References

script.aculo.us, Common Ajax Javascript library, <http://script.aculo.us/>

Seaside Sriptaculous Implementation, <http://scriptaculous.seasidehosting.st/>

Why Ajax Sucks (Most of the Time) Jacob Nielsen <http://www.usabilityviews.com/ajaxsucks.html>

Agile Web Development with Rails 2nd Ed Bl.16 October 25, Thomas & Hanson, The Pragmatic Bookshelf, PDF

Why aren't people using Migrations?

It does not work

It works but it is not useful

It is not complete enough to be useful

It is not documented enough to be useful

It is too buggy to be used

It works but not as useful as my existing tools

It is easier to use my existing tools than to learn it

Its main benefit occurs in long term projects, we have short term projects

I think it will be easier to use my existing tools than learn it

I am too busy to learn something new even if it saves me time

As Designers

How do we product useful tools?

As Developers

How do we know when a new tool/language will be better?

Script.aculo.us

Effects

Appear

Fade

Puff

Blind down/up

Slide down/up

Shake

Pulsate

etc

Drag & Drop

Async Forms

Modifying Page source

Controls

Autocompleter

In place editing

Sliders

<http://scriptaculous.seasidehosting.st> demo

Jakob Nielsen on Ajax

New or inexperienced Web designers
Just say no to Ajax

Web page

User's view of information

Unit of navigation

Textual address used to retrieve information

Unit of information on server

Ajax

Breaks this model of a web page

Back button does not work

78% of browsers Ajax compatible

Printing problems

Authoring problems

Search Problems

Edit in Place

http://127.0.0.1:3000/ajax/editInPlaceDemo

Name: Roger

Langaue: Smalltalk

Name: Roger
Langaue: Click to edit

Name:

Pete

ok

cancel

Langaue: Smalltalk

Name: Pete

Langaue: Smalltalk

This example is from the text

A Model to Edit

```
class CreateUsers < ActiveRecord::Migration
  def self.up
    create_table :users do |t|
      t.column :name, :string
      t.column :langauge, :string
    end
  end

  def self.down
    drop_table :users
  end
end
```

Controller

```
class AjaxController < ApplicationController
  in_place_edit_for :user, :name
  in_place_edit_for :user, :language

  def editInPlaceDemo
    puts 'in place demo'
    @user = User.find(1)
  end
end
```

Layout - ajax.rhtml

```
<html>
<head>
  <title>User: <%= controller.action_name %>
  </title>
  <%= stylesheet_link_tag 'scaffold' %>
  <%= javascript_include_tag :defaults %>
</head>
  <%= @content_for_layout %>
</body>
</html>
```

editInPlaceDemo.rhtml

```
<% for column in User.column_names %>
  <% next if column == 'id'%>
  <p>
    <b><%= column.capitalize %>:</b>
    <%= in_place_editor_field "user", column, {}, {
      :load_text_url=> url_for(:action=>"get_user_#{column}",
        :id=>@user)
    } %>
  </p>
<% end %>
```

Html Source

```
<html>
<head>
  <title>User: editInPlaceDemo
  </title>
  <link href="/stylesheets/scaffold.css?1162334913" media="screen"
rel="Stylesheet" type="text/css" />
  <script src="/javascripts/prototype.js?1164073972" type="text/javascript"></
script>
<script src="/javascripts/effects.js?1162250023" type="text/javascript"></script>
<script src="/javascripts/dragdrop.js?1162250023" type="text/javascript"></script>
<script src="/javascripts/controls.js?1162250023" type="text/javascript"></script>
<script src="/javascripts/application.js?1162250023" type="text/javascript"></script>
</head>
</html>
```

Html Source

```
<p> <b>Name:</b>
    <span class="in_place_editor_field" id="user_name_1_in_place_editor">Roger</span><script
type="text/javascript">
//
new Ajax.InPlaceEditor('user_name_1_in_place_editor', '/ajax/set_user_name/1', {loadTextURL:'/ajax/
get_user_name/1'})
//]]&gt;
&lt;/script&gt;
&lt;/p&gt;
&lt;p&gt; &lt;b&gt;Langauge:&lt;/b&gt;
    &lt;span class="in_place_editor_field" id="user_langauge_1_in_place_editor"&gt;Smalltalk&lt;/span&gt;&lt;script
type="text/javascript"&gt;
//<![CDATA[
new Ajax.InPlaceEditor('user_langauge_1_in_place_editor', '/ajax/set_user_langauge/1', {loadTextURL:'/
ajax/get_user_langauge/1'})
//]]&gt;
&lt;/script&gt;
    &lt;/p&gt;
&lt;/body&gt;</pre></div><div data-bbox="487 960 509 981" data-label="Page-Footer"><p>16</p></div>
```

Auto-complete

Documentation

<http://wiki.script.aculo.us/scriptaculous/show/Ajax.AutoCompleteter>

Example - Selecting Languages

Favorite language

Selections for s

Favorite language
Scheme
Smalltalk
Squeak

After selection

Favorite language

type s, increase font

Favorite language
Scheme
Smalltalk
Squeak

increase font, type s

Favorite language
Scheme
Smalltalk
Squeak

This example is modified from the text

Controller

```
class AjaxController < ApplicationController

  include LanguageList

  def autocomplete
    render(:layout => false)
  end

  def autocomplete_language
    re = Regexp.new("^#{params[:user][:language]}", "i")
    @languages= LANGUAGES.find_all do ||
      l.match re
    end
    render :layout=>false
  end
end
```

app/helpers/language_list.rb

```
module LanguageList
  LANGUAGES = %w{ Ada      Basic  C      C++      Delphi Emacs\ Lisp Forth
                 Fortran Haskell Java   JavaScript Lisp  Perl   Python
                 Ruby    Scheme Smalltalk Squeak}
end
```

app/views/ajax/autocomplete.rhtml

```
<html>
<head>
  <title>User: <%= controller.action_name %>
</title>
  <%= javascript_include_tag :defaults %>
  <%= stylesheet_link_tag 'autocomplete' %>

</head>
<body>

  <p><label for="user_language">Favorite language</label>
  <input id="user_language" name="user[language]" size="30" type="text" /></p>
  <div class="autocomplete" id="user_language_autocomplete"></div>

  <script type="text/javascript">
  //<![CDATA[
    var user_language_auto_completer = new Ajax.Autocompleter("user_language",
'user_language_autocomplete', '/ajax/autocomplete_language', {tokens: ['']})
  //]]>
  </script>

</body>
</html>
```

Autocompleter parameters

```
new Ajax.Autocompleter(id_of_text_field, id_of_div_to_populate, url, options);
```

Options	Default value
paramName	name of the element
tokens	[]
frequency	0.4
minChars	1
indicator	null
updateElement	null
afterUpdateElement	null

app/views/ajax/autocomplete.rhtml

Used to format possible choices for Autocompleter

Note choices are a list

```
<ul class="autocomplete_list">  
<% @languages.each do || %>  
<li class="autocomplete_item"><%= | %></li>  
<% end %>  
</ul>
```

public/stylesheets/autocomplete.css

```
input#text {  
  width: 250px;  
}
```

Formats choice list to look & act like
drop down menu

```
div.autocomplete {  
  position: absolute;  
  background-color: white;  
  border: 1px solid #888;  
  margin: 0px;  
  padding: 0px;  
  z-index: 1000;  
}
```

```
div.autocomplete ul, div.autocomplete ul li {  
  list-style-type: none;  
  line-height: 1.7;  
  margin: 0px;  
  padding: 0px;  
}
```

```
div.autocomplete ul li.selected {  
  background-color: #ffb;  
}
```

Warning

Autocompleter hides the text in the autocomplete div
except when showing choices

Error messages are placed in the autocomplete div

You need to view the page source to see error messages

With Rails helpers in a form

```
<html>
<head>
  <title>User: <%= controller.action_name %>
</title>
  <%= javascript_include_tag :defaults %>
  <%= stylesheet_link_tag 'autocomplete' %>
</head>
<body>
  <% form_for :user, @user, :url => {:action => 'update'} do |form| %>
    <p>Username <%= @user.name %></p>

    <p><label for="user_favorite_language">Favorite language</label>
    <%= text_field 'user', 'language' %></p>
    <div class="autocomplete"
      id="user_language_auto_complete"></div>
    <%= auto_complete_field :user_language,
      :url=>{:action=>'autocomplete_language'}, :tokens => ',' %>

    <%= submit_tag 'Edit' %>
  <% end %>
</body>
</html>
```

Modified Controller

```
class AjaxController < ApplicationController
  include LanguageList

  def autocomplete
    @user = User.find(1)
    render(:layout => false)
  end

  def autocomplete_language
    re = Regexp.new("^#{params[:user][:language]}", "i")
    @languages= LANGUAGES.find_all do ||
      l.match re
    end
    render :layout=>false
  end
end
```

Screenshots

Before text entry

Username Roger

Favorite language

Second auto complete

Username Roger

Favorite language

Auto-complete with s

Username Roger

Favorite language

Username Roger

Favorite language

After selection

Username Roger

Favorite language

Helpers verses Direct

Which is better?

```
<%= auto_complete_field :user_language,  
      :url=>{:action=>'autocomplete_language'}, :tokens => ',' %>
```

```
<script type="text/javascript">  
  //<br/>    var user_language_auto_completer = new Ajax.Autocompleter("user_language",<br/>'user_language_autocomplete', '/ajax/autocomplete_language', {tokens: ['']})<br/>  //]]&gt;<br/>&lt;/script&gt;</pre></div><div data-bbox="487 960 509 981" data-label="Page-Footer"><p>30</p></div>
```