

CS 683 Emerging Technologies
Fall Semester, 2005
Doc 23 Rails Model Example
Nov 17, 2005

Copyright ©, All rights reserved. 2005 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

Agile Web Development with Rails, Thomas & Hanson, The Pragmatic Bookshelf, 2005

Action Pack Documentation, <http://ap.rubyonrails.com/>

Active Record Documentation, <http://ar.rubyonrails.com/>

Book Example

Run MySql database

Create the databases:

```
cs683BookStore_development
cs683BookStore_test
cs683BookStore_production
```

In cs683BookStore_development create Books table

```
CREATE TABLE `books` (  
  `id` int(11) NOT NULL auto_increment,  
  `title` varchar(100) NOT NULL,  
  `author` varchar(50) NOT NULL,  
  `date` date,  
  PRIMARY KEY (`id`)
```

Create a rails instance using
rails cs683BookStore

Edit config/database.yml (see right)

Generate scaffold

```
ruby script/generate scaffold Book Admin
```

Start server

```
ruby script/server
```

cs683BookStore/config/database.yml

```
development:  
  adapter: mysql  
  database: cs683BookStore_development  
  username: whitney  
  password:  
  
test:  
  adapter: mysql  
  database: cs683BookStore_test  
  username: whitney  
  password:  
  
production:  
  adapter: mysql  
  database: cs683BookStore_production  
  username: whitney  
  password:
```

Controller

```
class ApplicationController < ActionController::Base
end
```

```
class AdminController < ApplicationController
  def index
    list
    render :action => 'list'
  end
end
```

```
  def list
    @book_pages, @books = paginate :books, :per_page => 10
  end
end
```

```
  def show
    @book = Book.find(params[:id])
  end
end
```

```
  def new
    @book = Book.new
  end
end
```

```
  def destroy
    Book.find(params[:id]).destroy
    redirect_to :action => 'list'
  end
end
```

```
  def create
    @book = Book.new(params[:book])
    if @book.save
      flash[:notice] = 'Book was successfully created.'
      redirect_to :action => 'list'
    else
      render :action => 'new'
    end
  end
end
```

```
  def edit
    @book = Book.find(params[:id])
  end
end
```

```
  def update
    @book = Book.find(params[:id])
    if @book.update_attributes(params[:book])
      flash[:notice] = 'Book was successfully updated.'
      redirect_to :action => 'show', :id => @book
    else
      render :action => 'edit'
    end
  end
end

end
```

Finding paginate

```
ActionController::Base.template_class = ActionView::Base
```

```
ActionController::Base.class_eval do
  include ActionController::Filters
  include ActionController::Layout
  include ActionController::Flash
  include ActionController::Benchmarking
  include ActionController::Rescue
  include ActionController::Dependencies
  include ActionController::Pagination
  include ActionController::Scaffolding
  include ActionController::Helpers
  include ActionController::Cookies
  include ActionController::Caching
  include ActionController::Components
  include ActionController::Verification
  include ActionController::Streaming
  include ActionController::SessionManagement
  include ActionController::Macros::AutoComplete
  include ActionController::Macros::InPlaceEditing
end
```

Model

```
class Book < ActiveRecord::Base  
end
```

Views - layout

app/views/layouts/admin.rhtml

```
<html>
<head>
  <title>Admin: <%= controller.action_name %></title>
  <%= stylesheet_link_tag 'scaffold' %>
</head>
<body>

<p style="color: green"><%= flash[:notice] %></p>

<%= @content_for_layout %>

</body>
</html>
```

public/stylesheets/scaffold.css

```
body { background-color: #fff; color: #333; }

body, p, ol, ul, td {
  font-family: verdana, arial, helvetica, sans-serif;
  font-size: 13px;
  line-height: 18px;
}

pre {
  background-color: #eee;
  padding: 10px;
  font-size: 11px;
}

a { color: #000; }
a:visited { color: #666; }
a:hover { color: #fff; background-color:#000; }

etc.
```

<h1>Listing books</h1>

Views - app/views/admin/list.rhtml

<table>

<tr>

<% for column in Book.content_columns %>

<th><%= column.human_name %></th>

<% end %>

</tr>

<% for book in @books %>

<tr>

<% for column in Book.content_columns %>

<td><%=h book.send(column.name) %></td>

<% end %>

<td><%= link_to 'Show', :action => 'show', :id => book %></td>

<td><%= link_to 'Edit', :action => 'edit', :id => book %></td>

<td><%= link_to 'Destroy', { :action => 'destroy', :id => book }, :confirm => 'Are you sure?' %></td>

</tr>

<% end %>

</table>

<%= link_to 'Previous page', { :page => @book_pages.current.previous } if @book_pages.current.previous %>

<%= link_to 'Next page', { :page => @book_pages.current.next } if @book_pages.current.next %>

<%= link_to 'New book', :action => 'new' %>

Views - new

app/views/admin/new.rhtml

```
<h1>New book</h1>

<%= start_form_tag :action => 'create' %>
  <%= render :partial => 'form' %>
  <%= submit_tag "Create" %>
<%= end_form_tag %>

<%= link_to 'Back', :action => 'list' %>
```

app/views/admin/_form.rhtml

```
<%= error_messages_for 'book' %>

<!--[form:book]-->
<p><label for="book_title">Title</label><br/>
<%= text_field 'book', 'title' %></p>

<p><label for="book_author">Author</label><br/>
<%= text_field 'book', 'author' %></p>

<p><label for="book_date">Date</label><br/>
<%= date_select 'book', 'date' %></p>
<!--[eoform:book]-->
```



New book

Title

Author

Date

Create

[Back](#)

Helpers - Keep code out of templates

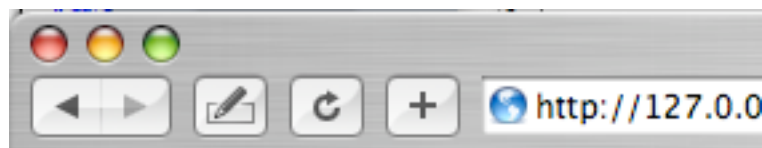
app/views/admin/new.rhtml

```
<h1>New book</h1>
<%= example_helper %>
<%= start_form_tag :action => 'create' %>
  <%= render :partial => 'form' %>
  <%= submit_tag "Create" %>
<%= end_form_tag %>

<%= link_to 'Back', :action => 'list' %>
```

app/helpers/admin_helper.rhtml

```
module AdminHelper
  def example_helper
    Time.now
  end
end
```



New book

Thu Nov 17 15:09:40 PST 2005

Title

Author

Date

2005 November 17

Create

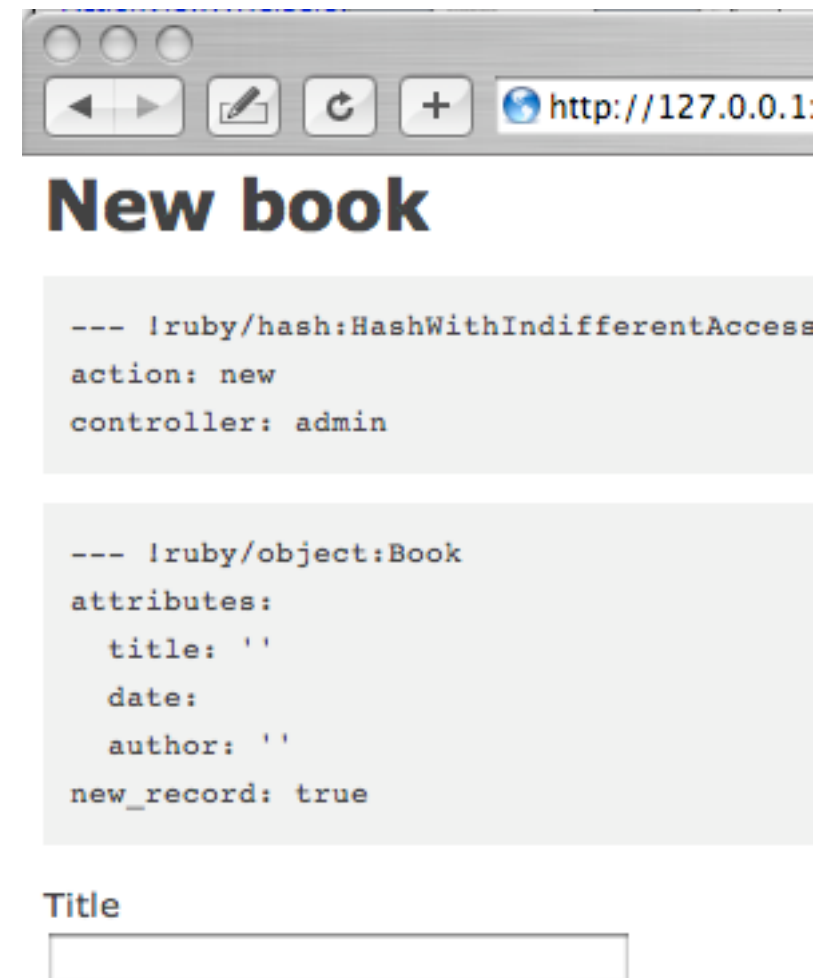
[Back](#)

Built-in Helper - debug

app/views/admin/new.rhtml

```
<h1>New book</h1>
<%= debug params%>
<%= debug @book %>
<%= start_form_tag :action => 'create' %>
  <%= render :partial => 'form' %>
  <%= submit_tag "Create" %>
<%= end_form_tag %>

<%= link_to 'Back', :action => 'list' %>
```



Validation

```
class Book < ActiveRecord::Base
  validates_uniqueness_of :title
  validates_presence_of :title, :author, :date
end
```

Testing Model

test/fixtures/books.yml

```
cat_book:
  id: 1
  title: Cat in the Hat
  author: Dr. Seuss
  date: 1967-12-20
hello_book:
  id: 2
  title: HelloWorld
  author: Me
  date: 1967-12-20
```

test/unit/book_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'

class BookTest < Test::Unit::TestCase
  fixtures :books

  def test_find
    book = Book.find_by_title('HelloWorld')
    assert_kind_of Book, book
    assert_equal book.author, 'Me'
  end

  def test_save
    book = Book.new
    book.author = 'A'
    book.title = 'B'
    book.date = Time.now
    assert book.save
    book_a = Book.find_by_author('A')
    assert_equal book.title, 'B'
  end

  def test_validate
    book = Book.new
    book.author = 'K'
    assert !book.save
    assert_equal 2, book.errors.count
    book_k = Book.find_by_author('K')
    assert_nil book_k
  end
end
```

Testing Controllers

```
class AdminControllerTest < Test::Unit::TestCase
  fixtures :books

  def setup
    @controller = AdminController.new
    @request     = ActionController::TestRequest.new
    @response    = ActionController::TestResponse.new
  end

  def test_index
    get :index
    assert_response :success
    assert_template 'list'
    assert_tag :tag => "table"
  end

  def test_create
    num_books = Book.count

    post :create, :book => { :title => 'Q', :author => 'q', :date => Time.now }

    assert_response :redirect
    assert_redirected_to :action => 'list'

    assert_equal num_books + 1, Book.count
  end
end
```