Ruby on Rails

Александр Красс
Alexander.Krass@gmail.com

Что такое Ruby on Rails?

- Ruby
- Мощнейший фреймворк

Ruby

На язык очень сильно повлияли:

- Perl
- Smalltalk
- Lisp

Разработан в 1993 году.

Ruby

“Ruby is designed to make programmers happy”

Yukihiro Matsumoto aka Matz

Ruby

- Важно, чтобы удобно было человеку, а не машине, тогда программирование по-настоящему продуктивно.

В Ruby всё – это объекты

```ruby
# Выводит "I'm Ruby ninja!" 5 раз
5.times do |n|
  puts "I'm Ruby ninja!"
end

# Ruby
"ruby".capitalize

# 5
-5.abs

# [1, 2, 3, 4]
[1, 2, [3, 4]].flatten!
```

If без всяких хитростей

```
if account.total > 100000
  puts "large account"
else
  puts "small account"
end
```

If без всяких хитростей

```
if account.total > 100000
  puts "large account"
elsif account.total > 25000
  puts "medium account"
else
  puts "small account"
end
```

Case

```
case name
when "John"
  puts "Howdy John!"
when "Ryan"
  puts "Whatz up Ryan!"
else
  puts "Hi #{name}!"
end
```

Регулярные выражения

```
phone = "123-456-7890"

if phone =~ /(\d{3})-(\d{3})-(\d{4})/
  ext  = $1
  city = $2
  num  = $3
end
```

Блоки

```ruby
# [1, 2, 3]
[2,1,3].sort! { |a, b| b <=> a }

# Выводит "I'm Ruby ninja!" 5 раз
5.times do |n|
  puts "I'm Ruby ninja!"
end
```

Как принимать в функции блок?

```ruby
def make_coffee
  puts "Take a cup."
  yield
  puts "Use a coffee-mashine."
end

make_coffee { puts "Wash it." }

#или так

make_coffee do
  puts "Wash it."
end
```

Посмотрим внимательно на динамическое типизирование

```ruby
class Duck
  def quark
    puts "Quack!"
  end
end

class Hunter
  def quark
    puts "Quack!"
  end
end
```

```ruby
array = [Duck.new, Hunter.new]

array.each do |o|
  o.quark
end
```

Duck typing

- Если что-то ходит как утка, крякает как утка, выглядит как утка, то это утка
- Это называется “Duck typing” (“утиным типизированием”)

Посмотрим ещё внимательнее

```
class Dog
end

[Duck.new, Hunter.new. Dog.new].each do |o|
  o.quark
end
```

Посмотрим ещё внимательнее

```
class Dog
end

[Duck.new, Hunter.new. Dog.new].each do |o|
  o.quark
end
```

У объекта класса `Dog` вызовется `method_missing`, который сгенерирует исключение.

method_missing

```
class Dog
  def method_missing
    puts "Woof!"
  end
end

# Quark!
# Quark!
# Woof!
[Duck.new, Hunter.new. Dognew].each do |o|
  o.quark
end
```

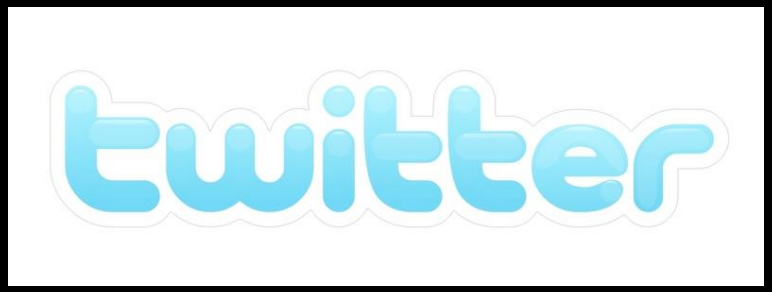
Это можно, например, использовать так...

```ruby
class Proxy
  def initialize(object)
    @object = object
  end

  def method_missing(symbol, *args)
    puts "Invoking the #{symbol} method"
    @object.send(symbol, *args)
  end
end

object = ["a", "b", "c"]
proxy = Proxy.new(object)
puts proxy.first
# Invoking the first method
# a
```

Ruby

- Очень активное сообщество
- Постоянно совершенствуется

Rails. Кто, например, использует?



Rails. Основные принципы

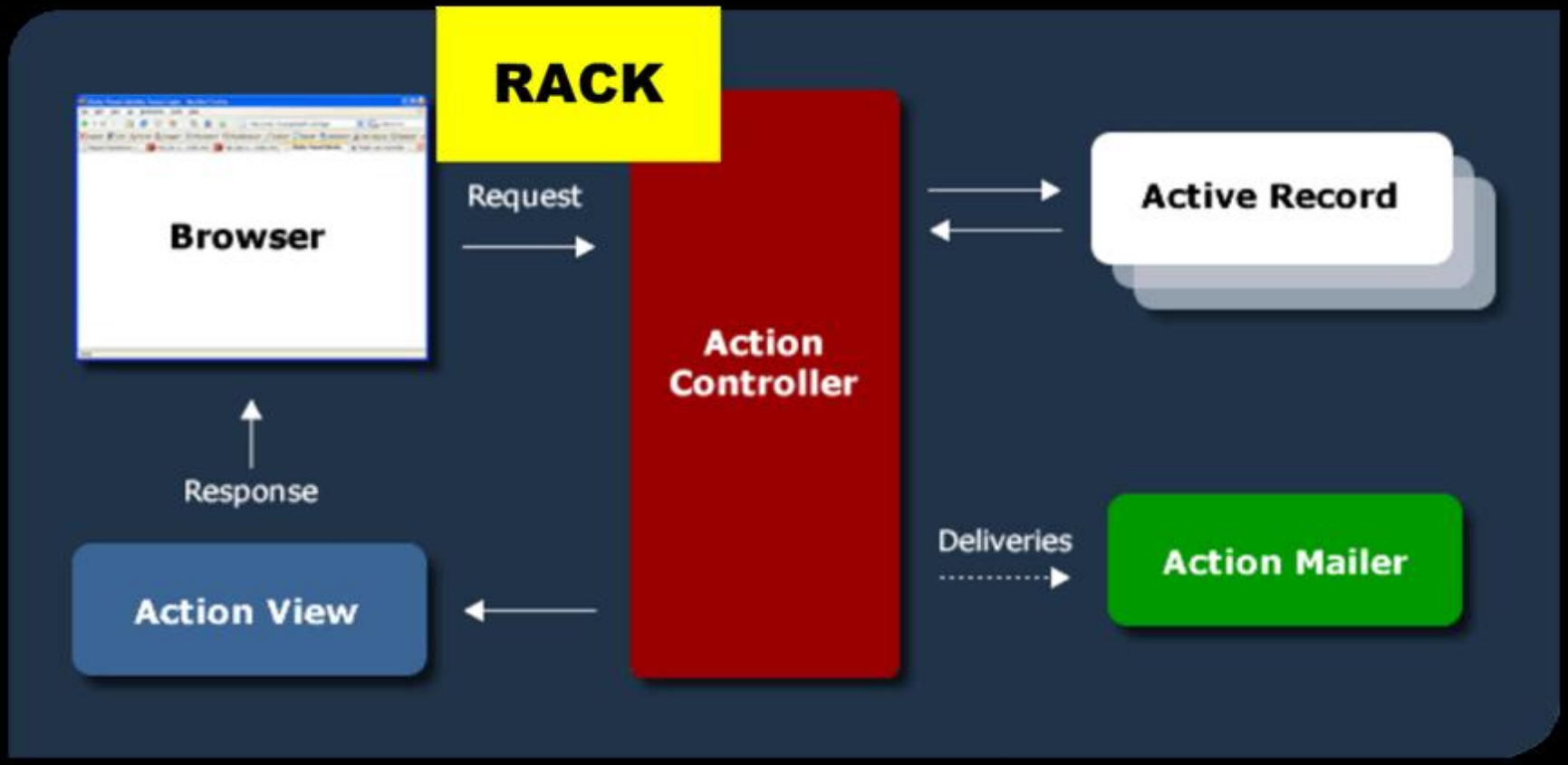
Convention over Configuration

Rails. Основные принципы

Don’t Repeat Yourself

Rails. Основные принципы

Model-View-Controller



Active Record

Таблица tasks

Поле	Тип
id	int(11)
title	varchar(255)
due_date	date

```
CREATE TABLE tasks
  id int(11) NOT NULL auto_increment,
  title varchar(255),
  due_date date,
  PRIMARY KEY (id)
```

Создаём модель

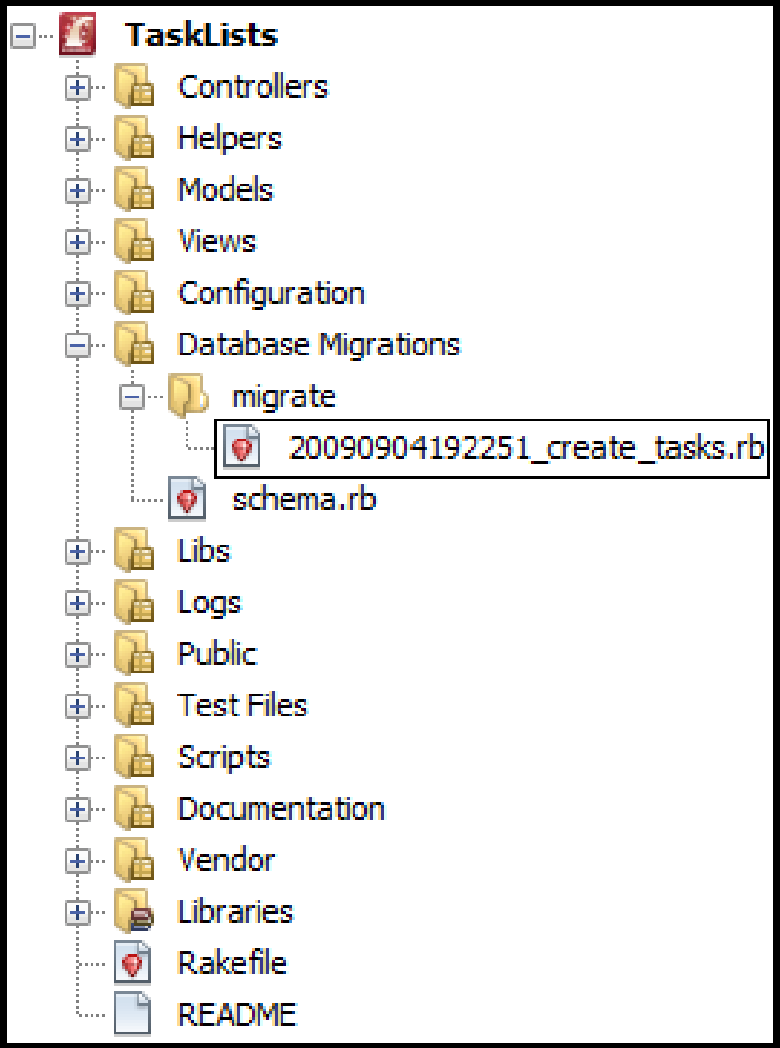
```
> ruby script/generate model task
title:string due_date:date
```

Database Migration

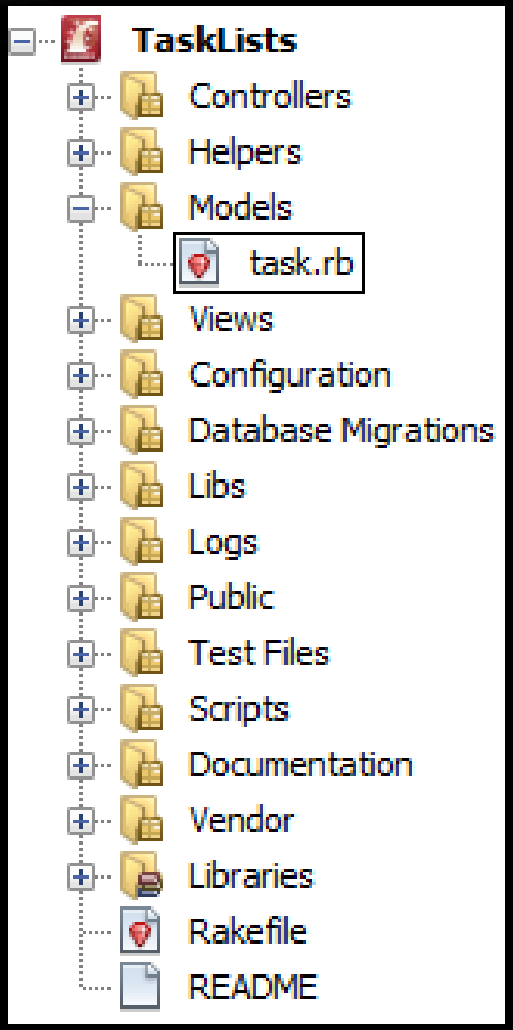
```
# 20090904192251_create_tasks.rb
class CreateTasks < ActiveRecord::Migration
  def self.up
    create_table :tasks do |t|
      t.string :title
      t.date :due_date

      t.timestamps
    end
  end

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


RAILS

db/migrate


TaskLists
Controllers
Helpers
Models
Views
Configuration
Database Migrations
migrate
20090904192251_create_tasks.rb
schema.rb
Libs
Logs
Public
Test Files
Scripts
Documentation
Vendor
Libraries
Rakefile
README

Active Record




```
class Task < ActiveRecord::Base
end
```

Active Record

```
task = Task.new
task.title = 'Go to the HackDay'
task.date = '2009-09-5'
task.save
```

```
t = Task.find(1)
puts t.title # Go to the HackDay

t = Task.find_by_title('Go to the HackDay')
t = Task.find_by_due_date('2009-09-05')
```

Action Controller

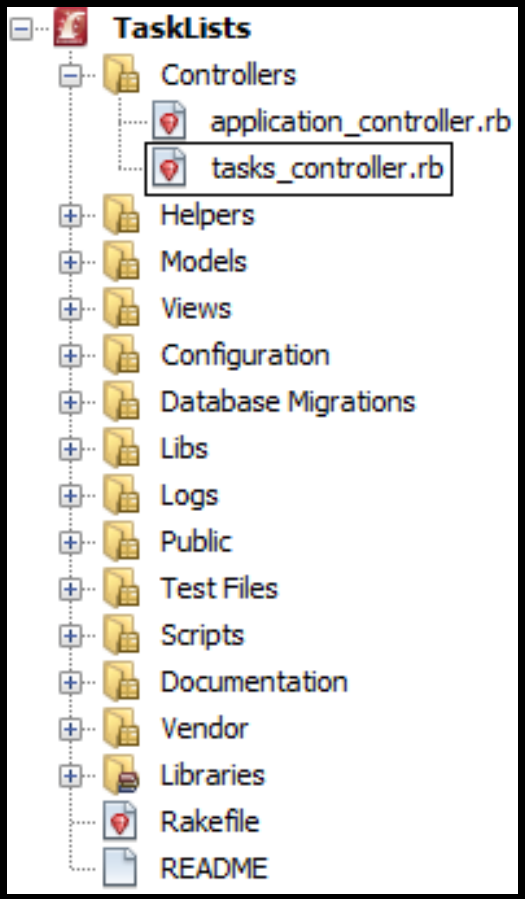
- http://www.../tasks/index

- Здесь:
  - `tasks` – это `TaskController`
  - `index` – это метод класса `TaskController`

Для этого создаём ресурс

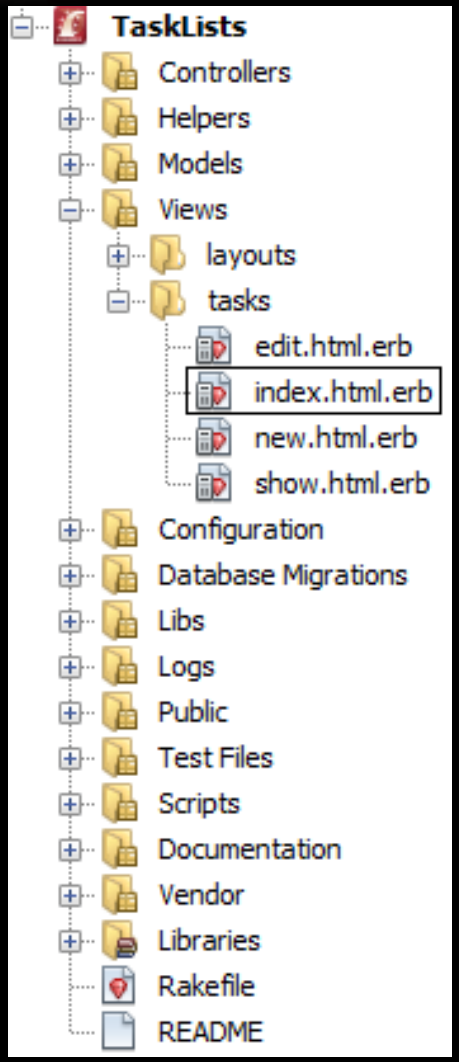
```
> ruby script/generate scaffold task
title:string due_date:date
```

Как это выглядит



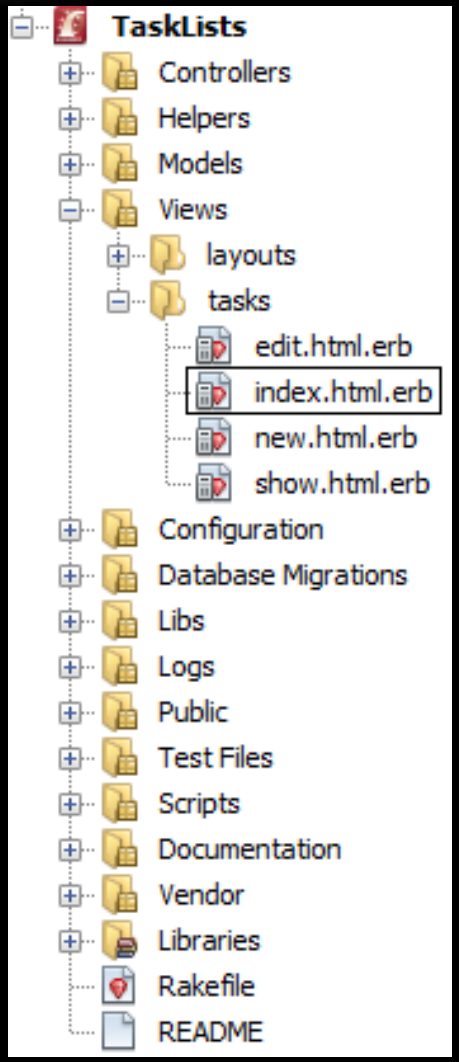

```ruby
class TasksController < ApplicationController
  # GET /tasks
  # GET /tasks.xml
  def index
    @tasks = Task.all

    respond_to do |format|
      format.html # index.html.erb
      format.xml  { render :xml => @tasks }
    end
  end

  ...

end
```

Action View




```
<h1>Обычный html-файл</h1>

<% puts "Только с примесью кода на Ruby" %>

<table>
<% @tasks.each do |task| %>
  <tr>
    <td><%=h task.title %></td>
    <td><%=h task.due_date %></td>
  </tr>
<% end %>
</table>

<br />
```

Action View




```
<h1>Listing tasks</h1>

<table>
  <tr>
    <th>Title</th>
    <th>Due date</th>
  </tr>

<% @tasks.each do |task| %>
  <tr>
    <td><%=h task.title %></td>
    <td><%=h task.due_date %></td>
    <td><%= link_to 'Show', task %></td>
    <td><%= link_to 'Edit', edit_task_path(task) %></td>
    <td><%= link_to 'Destroy', task,
            :confirm => 'Are you sure?',
            :method => :delete %></td>
  </tr>
<% end %>
</table>

<br />

<%= link_to 'New task', new_task_path %>
```

Демонстрация

Они сделают за Вас многое – плагины и gems

Аутентификация и авторизация

- restful_autentification
- authlogic
- authlogic_openid
- restful_acl
- …

Загрузка файлов и изображений

- papaperclip
- attachments_fu
- file_column
- …

И многое, многое другое

- will_paginate
- recaptcha
- act_as_commentable
- act_as_taggable / act_as_taggable_on_steroids
- act_as_rateable
- act_as_state_machine
- …

Популярные IDE

- Aptana RadRails (Eclispse)
- IntelliJ IDEA / RubyMine
- Microsoft Visual Studio (Ruby in Steel)
- NetBeans
- TextMate, Vim и другие мощные текстовые редакторы

Что же нам даёт Rails в двух словах?

Rails делает за Вас много рутинных задач, сохраняя Ваше время для творчества.

Что посмотреть?

- rubyonrails.com
- railscasts.com
- slideshare.net
- Agile Web Development with Rails. 3nd Edition

Ruby on Rails

Александр Красс
Alexander.Krass@gmail.com