



RUBY ON RAILS 2.1

WHAT'S NEW ?

```
def temp_p
  p = temp_p.s.sst
  p.respond_to?(:path) ? p.path : p.obj_s
end

def render_partial(partial_path, object_assigns = nil, local_assigns = nil)
  case partial_path
  when String, Symbol, NilClass
    path, partial_name = partial_pieces(partial_path)
    object = extracting_object(partial_name, object_assigns)
    local_assigns = local_assigns ? local_assigns.clone : {}
    add_counter_to_local_assigns!(partial_name, local_assigns)
    add_object_to_local_assigns!(partial_name, local_assigns, object)

    if logger && logger.debug?
      ActionController::Base.benchmark("Rendered #{path}/_#{partial_name}")
      render("#{path}/_#{partial_name}", local_assigns)
    end
  else
    non
  end
end
```

CARLOS BRANDO

REVIEW: MARCOS TAPAJÓS - COVER: DANIEL LOPES

Ruby on Rails 2.1

WHAT'S NEW

Second Edition

Ruby on Rails 2.1

WHAT'S NEW

Second Edition

Carlos Brando
Marcos Tapajós

© Copyright 2008 Carlos Brando. All Rights Reserved.

Second edition: June 2008

Carlos Brando

Website: www.nomedojogo.com

Marcos Tapajós

Website: www.improveit.com.br/en/company/tapajos

Chapter I

Introduction

Around July of 2004 David Heinemeier Hansson publicly released the Ruby on Rails framework. The framework had been extracted from a web application called he was working on called Basecamp. More than three years later, on the December 7th 2007, Ruby on Rails version 2.0 was released with numerous important changes.

Six months have passed since then, and during this time more than **1400 developers** from all around the world have contributed **1600 patches** to the framework. Today, June 1st 2008, version 2.1 of the Ruby on Rails framework was released.

Major new features according to David Heinemeier Hansson:

- Timezones
- Dirty tracking
- Gem Dependencies
- Named scope

Ruby on Rails 2.1 - What's New

- UTC-based migrations
- Better caching

As always, to update or install the new version:

```
gem install rails
```

ACKNOWLEDGMENT

To Marcos Tapajós, co-author of this book. If it wasn't for him, you probably you wouldn't be reading this right now.

To Daniel Lopes who made the beautiful cover for this edition.

To all of the Ruby on Rails Brazilian community that helped directly or indirectly with this book, commenting on blog posts and giving suggestions. It's like I always say, the best of Rails is its community! Keep creating, inventing, and specially sharing.

TRANSLATORS

This book was proudly translated to english by these Brazilian guys:

Pedro Pimentel - <http://www.pedropimentel.com>

Chapters 3-8 and 10-13

Rafael Barbosa - <http://www.act-as-newbie.com>

Introduction and chapter 1

Ricardo S Yasuda - <http://blog.shadowmaru.org>

Chapter 14

Caíke Souza - <http://tech-death.blogspot.com>

Chapter 2

Abraão Coelho - <http://abrcoelho.net>

Chapter 9

Bruno Miranda - <http://brunomiranda.com>

Reviser

Chapter 2

ActiveRecord

ActiveRecord is an object-relational mapping layer responsible for the interoperability among application and database and also responsible for data abstraction. (wikipedia)

Chapter 3

ActiveSupport

Active Support is a collection of useful classes and default libraries extensions which were considered useful for Ruby on Rails Applications. (wikipedia)

Chapter 4

ActiveResource

ActiveResource is a layer responsible by the client side implementation of RESTful systems. Through ActiveResource is possible to consume RESTful services by using objects that work like a proxy for remote services.

Chapter 5

ActionPack

Comprises ActionView (visualization generation for end user, like HTML, XML, JavaScript, and etc) and ActionController (business flow control) (adapted from wikipedia)

Chapter 6

ActionController

ActionController is the layer responsible by receiving web requests and taking decisions of what is going to be run and rendered or to redirect the request to another action. An Action is defined as public methods within controllers which are automatically available through routes.

Chapter 7

ActionView

ActionView is the layer responsible by the generation of the viewable interface visible to users through conversion of ERB templates.

Chapter 8

Railties

Chapter 9

Rake Tasks, Plugins and Scripts

Chapter 10

Prototype and script.aculo.us

Chapter 11

Ruby 1.9

Chapter 12

Debug

Chapter 13

Bugs and Fixes

Chapter 14

Additional Information

Chapter 15

CHANGELOG

RUBY ON RAILS 2.1

WHAT'S NEW ?

```
def temp_partials
  p = temp_partials.first
  p.respond_to?(:path) ? p.path : p.to_s
end

def render_partial(partial_path, object_assigns = nil, local_assigns = nil)
  case partial_path
  when String, Symbol, NilClass
    path, partial_name = partial_pieces(partial_path)
    object = extracting_object(partial_name, object_assigns)
    local_assigns = local_assigns ? local_assigns.clone : {}
    add_counter_to_local_assigns!(partial_name, local_assigns)
    add_object_to_local_assigns!(partial_name, local_assigns, object)

    if logger && logger.debug?
      ActionController::Base.benchmark("Rendered #{path}/_#{partial_name}")
      render("#{path}/_#{partial_name}", local_assigns)
    end
  else
    render(partial_path, object_assigns, local_assigns)
  end
end
```