



★ Catch Up with Ruby on Rails 4

March 4, 2013

Web Development \ Ruby

Tweet 5

+1 0

Like 43

Instructor



José Mota

Ruby on Rails, one of the most innovative frameworks in the world, is now [marching its way](#) into 4.0 territory. As with any major new release, there will most certainly be a variety changes, new additions, and improvements. But don't worry; let me take you through all of these changes, step by step. You'll be prepped in no time!

Catch Up with Ruby on Rails 4	7m 46s
Introduction	7m 46s
Controllers and Views	39m 38s
HTML5 Form Helpers	11m 52s
Collection Form Helpers	7m 40s
Custom Flash Types	6m 28s
Before_action	2m 56s
Caching	10m 42s
New features	34m 12s
Live Streaming	11m 4s
Turbolinks	10m 24s
Dynamic index.html	3m 28s
Strong Parameters	9m 16s
Active Record	45m 22s
PostgreSQL hstore	17m 2s
PostgreSQL arrays	6m 12s
ActiveModel::Model	9m 6s
ActiveRecord's `not` Method	6m 54s
Concerns in the Models	6m 8s
Router	11m 34s
Concerns in the Router	5m 54s
PATCH Verb	5m 40s
Upgrade	26m 36s
No More Plugins	5m 36s
Tests	3m 56s
Gem Extractions: ActiveRecord, Observers	8m 20s
How to Upgrade	8m 44s
Conclusion	4m 18s
Final Tips	4m 18s

Catch Up with Ruby on Rails 4

Introduction

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 7m 46s

Published: March 4, 2013

This is the course on how to catch up to Rails 4, one of the best web frameworks in the world. This course will prepare you for the main changes and new features of the new major release of Ruby on Rails.

This lesson will get you up and running with the entire agenda for the course and teach you how to install Rails 4 on your machine.

Useful comments posted on this video:

Rails 4 is now available at rubygems.org. See <http://rubygems.org/gems/rails/versions/4.0.0.beta1>.

Controllers and Views

HTML5 Form Helpers

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 11m 52s

Updated: March 5, 2013

Rails 4 comes packed with a lot of new form helper methods for you to use. Most of them are related to the new HTML5 tags. You'll be taught how to use each one of these new methods in your project.

Useful comments posted on this video:

See <https://github.com/Modernizr/Modernizr/wiki/HTML5-Cross-Browser-Polyfills> for shims, fallbacks, and polyfills that implant html5 functionality in browsers that don't natively support them. Web Forms included.

Collection Form Helpers

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 7m 40s

Updated: March 5, 2013

This lesson will help you manage your ActiveRecord relationships through your interface. Previously in Rails 3 you'd have to use something like Simpleform in order to have checkboxes and radio buttons that represent your relationships. Now it's possible to accomplish that with one single method.

Custom Flash Types

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 6m 28s

Updated: March 5, 2013

Rails 3 advocates the convention of using just two flash types: notice and alert. Rails 4 now allows you to register and use custom flash types the same way you use the first two. This brings greater flexibility to your application workflow.

This lesson will teach you how to register and use custom flash types.

Before_action

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 2m 56s

Updated: March 5, 2013

This lesson shows one of the key philosophical points that makes Rails such a great ecosystem. In Rails 3 there is a `before_filter` method that allows you to run a method before a common set of actions in your controller is executed. In Rails 4 that method has changed to `before_action` to accommodate its true meaning and purpose.

Caching

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 10m 42s

Updated: March 5, 2013

Caching has changed a lot in Rails 4. Some things were deprecated and extracted, while other caching features have been greatly improved.

In this lesson you'll learn how to regain access to the deprecated features and also how to use nested caching properly.

New Features

Live Streaming

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 11m 4s

Updated: March 5, 2013

Live streaming is one of the most anticipated features in Rails 4. It allows you to stream data in a push notification fashion, in real time.

This lesson will teach you how to use ActionController::Live to enable your controllers to stream live data to your client. You'll also learn how to interact with that data in the browser side as well.

Turbolinks

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 10m 24s

Updated: March 5, 2013

Turbolinks is a brand new way of handling assets in Rails 4. Rather than requesting page after page, including all the assets, Turbolinks requests the page via AJAX and updates the current document, without requesting all the parameters.

This lesson will help you understand and get started with this new gem.

Dynamic index.html

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 3m 28s

Updated: March 5, 2013

Every time you start a new Rails application, you have to remove the `public/index.html` file so it won't mess up your homepage. In Rails 4 this page was extracted so you don't have to.

Strong Parameters

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 9m 16s

Updated: March 5, 2013

The topic of mass assignment protection has always been really important in Rails. Its 3.2 release brought the `attr_accessible` method as a way to protect against malicious attempts to modify data.

Rails 4 introduces a smarter way to protected data coming from the outside with the new `strong_parameters` gem. This lesson will teach you how to install it and use it.

Active Record

PostgreSQL hstore

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 17m 2s

Updated: March 5, 2013

PostgreSQL is the de facto standard in Rails production databases. It has been given a lot of care lately, as you'll see in this lesson where you'll learn how to use the `hstore` extension and store schemaless hashes in your data.

PostgreSQL arrays

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 6m 12s

Updated: March 5, 2013

Besides storing hashes with `hstore`, PostgreSQL allows you to store arrays. This lesson will teach you how to store and read array data from your ActiveRecord models.

ActiveModel::Model

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 9m 6s

Updated: March 5, 2013

Using Rails to generate forms that perfectly map your ActiveRecord models on a page is one of the most awesome features the framework has. However it's been historically difficult to use those forms in models that don't necessarily need to inherit from ActiveRecord.

This lesson will explain you how to use the new ActiveModel::Model mixin into your classes so that they can now use what made ActiveRecord so shiny, like validations, form building and much more.

ActiveRecord's `not` Method

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 6m 54s

Updated: March 5, 2013

In order to prevent the use of direct SQL in your queries, one more method was introduced in the ActiveRecord query interface: the not method. This also allows for reusability in scopes.

In this lesson you will learn how to use the not method in your development.

Concerns in the Models

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 6m 8s

Updated: March 5, 2013

Concerns have been introduced in Rails 3.2 as a way of mixing behavior into classes. In Rails 4, this behavior has been further extended in the way you can register relationships and methods in one single place and then declaratively inject those features in your models.

This lesson will teach how to extend your ActiveRecord models with concerns.

Router

Concerns in the Router

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 5m 54s

Updated: March 5, 2013

If you think concerns in models and controllers weren't enough, wait until you realize in this lesson that you can also apply concerns in your routes.

This lesson will teach you how to declare routing concerns and then apply them to resource routes.

PATCH Verb

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 5m 40s

Updated: March 5, 2013

As you might know, Rails follows the REST pattern pretty tightly. However, it was suggested some time ago the PATCH verb should replace the PUT verb for updating a resource. This lesson will explain you how that decision came through and the reasons behind it.

Upgrade

No More Plugins

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 5m 36s

Updated: March 5, 2013

In Rails 4, there have have some changes that require some manual procedure. One of them is the total removal of the plugins functionality. This lesson will guide you in moving away from plugins and start using the Gemfile.

Tests

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 3m 56s

Updated: March 5, 2013

Since Ruby 1.9.3 is now the default minimum required version to use Rails 4, Minitest now replaces Test::Unit and with it come some minor changes to the `test/` folder structure, which you'll see in this lesson.

Gem Extractions: ActiveResource, Observers

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 8m 20s

Updated: March 5, 2013

Some features that existed in Rails 3 are now being extracted out to their own gems, in order to make Rails 4 lighter. ActiveResource and Observers are two of these features.

In this lesson you'll explore their functionality and how to restore them back into Rails 4 if you need them.

How to Upgrade

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 8m 44s

Updated: March 5, 2013

This lesson is a brief summary of all the big changes coming to Rails 4, with the upgrade of your Rails 3 application in mind. If you need a quick reference for the upgrade, then this is the right lesson to go to.

Conclusion

Final Tips

Instructor: [José Mota](#)

Difficulty: Intermediate

Length: 4m 18s

Updated: March 5, 2013

So that is it for this course. Rails 4 is proving to be a great release and it's fast sailing to an even better ecosystem and community.

I hope to see you soon in a next course. Thanks for taking the time in improving your knowledge by watching this series. If you have any questions about the contents I gave you here or if you have any suggestions on how to improve my recordings, please visit the forums and post a new question under the topic "Catch up with Rails 4".