Ruby on Rails, Part 2 of 5: Best Practices

Meet the expert: Jordan Hudgens has certifications for Ruby on Rails, Thinkful; Ruby on Rails, Bloc.io; Front End Development, Thinkful; and AngularJS, Thinkful. He is currently vice president of engineering for TRACKR in Midland, Texas and is working on his PhD in Computer Science from Texas Tech. In addition to Ruby, Jordan works with PHP, JavaScript, MySQL, Postgres, CSS3, C, C++, C#, Objective-C, and Python. He also works with the frameworks Rails (Ruby), Zend (PHP), and Django (Python), plus the libraries AngularJS, jQuery, and Backbone.js.

Prerequisites: This course teaches anyone who is interested in learning the basics of programming computers using Ruby on Rails. This course has no programming language prerequisites and only assumes basic familiarity with computer usage. Anyone with moderate computer end-user experience should be able to take this course. Please view the Ruby on Rails: Intro to Development course before taking this course.

Runtime: 01:23:56

Course description: To begin this course we will cover how the model view controller design pattern works in the Ruby on Rails framework, specifically discussing the rolls and best practices associated with each element. We will walk through the file system so you can become familiar with items such as the respective roles of our config directory, how the asset management sytem is structured, plus many other items, and lastly, how we utilize the Gemfile. The Ruby on Rails development stack works very closely with version control management systems such as Github and we will cover what Git is, why it is important and how to implement it into our application. Routing is one of the core fundamentals of any web application, and we will walk through how to setup a root homepage. We will finish up this course by building a scaffold into our application that interacts with our database, has form elements for the user, renders a database query on the page and is tested using RSpec.

Course outline:

MVC
- Introduction
- Overview
- Model
- View
- Controller
- Demo: MVC
- Summary

File System
- Introduction
- Config
- Assets
- Logic
- Layouts
- Database
- Gemfile
- Demo: Config Directory
- Demo: Asset Directory
- Demo: Layouts Directory
- Demo: Database Directory
- Demo: Gem File
- Summary

Git
- Introduction
- Why use git?
- Basic commands
- Branching & Commands
- Reverting
- Demo: README file
- Demo: Revert Back
- Summary

Routing
- Introduction
- Setting the homepage
- Resources
- Pages
- Usage in Rails
- Demo: Routing
- Demo: Homepage Control
- Demo: Link to
- Demo: Navigation Bar
- Demo: Git Status
- Summary

Testing With Scaffolds
- Introduction
- Setting up Rspec
- Scaffolds
- Demo: Setting up Rspec
- Demo: Create Scaffold
- Demo: Start servers
- Demo: Adding a Tab
- Demo: Rspec Testing
- Demo: Testing Failure
- Summary