Ruby on Rails

Course Summary

Description

Students will learn to create database-driven websites using Ruby on Rails. They will learn to install Ruby and Rails, to program in Ruby and use Ajax in their Ruby on Rails applications.

Objectives

At the end of this course, students will be able to:

- Install and configure Ruby and Rails
- Understand the Model-View-Controller framework architecture
- Program in Ruby
- Build database-driven applications with Ruby on Rails
- Work with templates
- Use Ajax in Ruby on Rails applications
- Test Ruby on Rails applications

Topics

- Getting Started with Rails
- Ruby Language Skills and Techniques
- Model Design and Database Management
- Controller Actions and View Templates
- Application Testing
- Ruby and Rails Resources

Prerequisites

Students should have experience with HTML. Experience in the following areas would also be beneficial:

- Server-side programming experience
- SQL

Duration

Four days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Ruby on Rails

Course Outline

I. Getting Started with Rails
- A. Installing RailsPlatform requirements
  1. Supported database backends
  2. Ruby installation
  3. InstantRails for Windows
- B. Creating a new application
  1. The 'rails' command
  2. Tour of the Rails application layout
- C. Application configuration and bootstrapping
  1. Creating the databases
  2. Configuring database.yml
  3. Setting the default route
- D. Connecting to a Rails application
  1. controller/action/id URL semantics
  2. Dispatcher and routing roadmap
  3. Examining the log files
- E. Rails tools
  1. Managing tasks with rake (Ruby make)
  2. The application console
- F. The Model/View/Controller (MVC) framework architecture
  1. Separation of concerns
  2. MVC in the application directory layout
- G. The Rails philosophy: "Convention over configuration"

II. Ruby Language Skills and Techniques
- A. Running Ruby
  1. Command-line Ruby
  2. The Interactive Ruby (irb) console
- B. Objects, variables, and methods
  1. Basic object-orientation concepts
  2. Naming conventions
  3. Variable assignment
  4. Local vs. instance variables
  5. Method-calling semantics
  6. Method arguments and return values
- C. Classes and modules
  1. Instantiating classes
  2. Mixing in modules
  3. Modules as namespaces
- D. Built-in classes
  1. String
  2. Array
  3. Hash
  4. Symbol
  5. Numerics
- E. Blocks and iterators
  1. Blocks vs. methods vs. lambdas
  2. Collection iteration
  3. Single-object iteration
- F. Exceptions
  1. Built-in exceptions
  2. Writing your own exception classes
  3. Exceptions in Rails

III. Model Design and Database Management
- A. Domain modeling for Rails and Ruby
  1. Describing the "what" of the application
  2. Creating ActiveRecord models with "generate"
- B. Object-relational mapping with ActiveRecord
  1. ActiveRecord models and Ruby classes
  2. ActiveRecord associations
     - a. One-to-one, one-to-many
     - b. Many-to-many
- C. "Rails-friendly" SQL
  1. Table-naming conventions
  2. Primary and foreign keys
- D. Database maintenance with ActiveRecord Migrations
  1. Default migrations
  2. Customizing migrations
  3. Migrating backwards

IV. Controller Actions and View Templates
- A. Basics of controller/view interaction
  1. Creating controllers and view with "generate"
  2. Default rendering rules
  3. Shared controller/template instance variables
  4. Separating controller logic from view specifics
- B. HTML templating with Embedded Ruby (ERb)
  1. Rules of ERb processing
  2. Layouts
  3. Using master and partial templates
- C. Fine-tuning controllers and view
  1. Controller filters
  2. Redirecting control

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Ruby on Rails

Course Outline (cont'd)

3. Default and custom view helper methods

D. Writing and processing HTML forms in Rails
   1. Using form helper methods
   2. "Magic" field initialization from instance variables
   3. Accessing CGI data through the "params" hash

E. Ajax calls and RJS (Ruby/JavaScript) templates
   1. Basic DOM updating with Ajax
   2. Using RJS for composite Ajax calls

V. Application Testing

A. Rails unit tests
   1. The Ruby TestUnit framework
   2. Writing fixturesAddressing test failures and errors

B. Ruby and Rails Resources
   1. Mailing lists
   2. Websites
   3. Books
   4. IRC channels

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically