Review on Ruby on Rails Developers–An Agile Developer's Framework

Shubham Devgire, Rohan Jagtap

Department of CSE, Jawaharlal Nehru Engineering College,
Aurangabad, India

E-mail: shubhamdevgire@gmail.com, rohan.jagtap3333@gmail.com

Abstract

Since its debut in 2004, Ruby on Rails has speedily become one in all the foremost powerful and standard frameworks for building dynamic net applications. Most of the add scientific computing nowadays is finished in parallel algorithms, typically via message-passing architectures admire the message-passing interface (MPI). A new rising language referred to as Ruby that maintains a strict adherence to object homeward-bound principles and a clean, intuitive syntax. The author created MPI Ruby, an entire binding of MPI to Ruby. During this article, he introduces Ruby and MPI Ruby. Some applications and data on the project's current standing and its convenience are represented.

***Keywords:** Java architecture, ruby on rail, web 2.0 framework, object-oriented ruby programming language, open source framework, MVC architecture*

HISTORY

Version History	
Version	**Date**
1.0	December 13, 2005
1.2	January 19, 2007
2.0	December 7, 2007
2.1	June 1, 2008
2.2	November 21, 2008
2.3	March 16, 2009
3.0	August 29, 2010
3.1	August 31, 2011
3.2	January 20, 2012
4.0	June 25, 2013
4.1	April 8, 2014
4.2	December 19, 2014

INTRODUCTION

Ruby–Scripting language, dynamic and object orienting language. Ruby permits speedy development and speedy prototypes of software system. It consists of helpful string operations and regular expressions. Everything in Ruby is associate in nursing object. It is automatic pickup. It provides Instant feedback once developing Multiple OS support and open supply [1]. Like several net frameworks, Ruby on Rails uses the MVC design pattern to arrange application programming. In a default configuration, a model within the Ruby on Rails framework maps to a table during an information, and to a Ruby file. Let us say, a model category, user, can typically be outlined within the file, 'user.rb', within the app/models directory, and connected to the table, 'users', within the information. Whereas developers area unit unengaged to ignore this convention and opt for differing names for his or her models, files, and information table, this can be not common apply and is sometimes discouraged in accordance with the "convention-over-configuration" philosophy. A controller may be a server-side element of Rails that responds to external requests from the online server to the appliance, by decisive that read file to render. The controller may got to question one or additional models directly for data and pass these on to the read. A controller might give one or additional actions [2].

In Ruby on Rails, an action is often a basic unit that describes the way to reply to a selected external web-browser request. Additionally, note that the controller/action are going to be accessible for external net requests given that a corresponding route is mapped to that [3]. Rails encourage developers to United States of America routes that embrace actions such as: produce, new, edit, update, destroy, show, and index. These mappings of incoming requests/routes to controller actions are often simply come upon within the routes configuration file. A read within the default configuration of Rails is an erb file that is evaluated and reborn to hypertext mark-up language at run-time. Instead, several different templating systems are often used for views [4, 5].

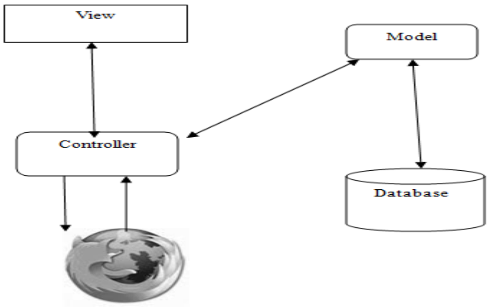

Fig. 1: *A Schematic Representation of the Model-View-Controller (MVC) Architecture.*

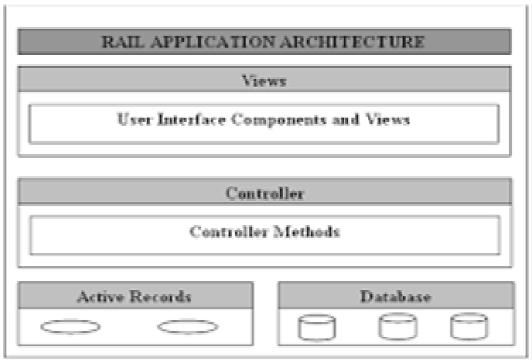
RAIL APPLICATION ARCHITECTURE
Views
User Interface Components and Views
Controller
Controller Methods
Active Records
Database

Fig. 2: *Rail Application Architecture.*

Table 1: *A Summary of Default Rails Directory Structure.*

File/Directory	**Purpose**
code, app/	Core Application (app) including Models, Views, Controllers and Helpers
app/assets	Applications Assets such as Cascading Style Sheets (CSS), JavaScript Files and Images
config/	Application Configuration
db/	Database Files
doc/	Documentation for the Application

ADVANTAGES OF RAILS

1. The method of programming is far quicker than with different frameworks and languages, part thanks to the object-oriented nature of Ruby and also the huge assortment of open ASCII text file obtainable at intervals the Rails community.
2. Rails is sweet for fast application development (RAD), because the framework makes it simple to accommodate changes.
3. Rails has developed a powerful specialize in testing, and has sensible testing frameworks.
4. Rails and most of its libraries area unit open supply, thus not like different business development frameworks there are not any licensing prices concerned.

LIMITATIONS OF RAILS

1. Not all web site hosts will support Rails.
2. Java and PHP square measure additional wide used, and there square measure additional developers in these languages.
3. Performance and measurability.
4. There are issues that Rails applications are not as quick as Java or C that is true, except for the bulk of applications it is quick enough. There square measure many status organizations that place confidence in Rails to power their sites.

CONCLUSION

Our decision to use Rails is especially one among personal preference-we discover that it works well for us by sanctionative the speedy development of dynamic internet applications.

REFERENCES

1. Rafael Franca. *Ruby on Rails*; 2015.
2. Repinel. This week in Rails: Rails 4.2.4 RC1, 4.1.13 RC1 and more!" *Ruby on Rails*; 2015.
3. Ruby on Rails 3.0 Release Notes.
4. Ruby on Rails 3.1 Release Notes.
5. Ruby on Rails 3.2 Release Notes.