4/29/2020

Ruby on Rails

IT 3203

Robert Somers, Amanda Zheng, Yusuf Ahimed

GROUP5

Ruby on Rails is a server-side web application framework written with the Ruby programming language. The original creator of this framework is David Heinemeier Hansson. Ruby was intended to make programming web applications simpler by making presumptions about what each designer's needs. Rails, or Ruby on Rails, is a server-side web application framework written in Ruby under the Massachusetts Institute of Technology License. Rails works by providing default structures for a database, model–view–controller (MVC) framework, web page, and web services. IT (information technology) facilitates and supports make use of web standards such as JavaScript Object Notation (JSON) or Extensible Markup Language (XML) for data transfer. They also use Cascading Style Sheets (CSS), Hypertext Markup Language (HML) and JavaScript for user interfacing. In addition to Model-View-Controller (MVC), Rails emphasizes the use of other well-known software engineering patterns and paradigms. This includes do not repeat yourself (DRY), convention over configuration (CoC), and the active record pattern.

Similar to other web frameworks, Ruby on Rails works the model–view–controller (MVC) pattern to create application programming. In a default configuration, a model in the Ruby on Rails framework outlines to a table in a database and a Ruby file. For example, a model class user will usually be specified in the file 'user.rb' in the app/models linked and directory to the table 'users' in the database. While developers are free to neglect this convention and choose differing names for their database models, files, and table, this is not a regular practice and is usually managed following the "convention-over-configuration (CoC)" philosophy. Ruby on Rails includes tools that make everyday development tasks easier "out-of-the-box," such as scaffolding that can automatically create some of the models and views needed for an essential website. Also added a simple Ruby web server that is shared with Ruby, a build system, WEBrick, and Rake, issued as a gem. Together with Ruby on Rails, these tools help give primary development context.

Commented [AZ1]: I don't understand this sentence

Commented [AZ2]: unsure

Ruby on Rails is not connected to the Internet directly but through a front-end web server. Mongrel was generally favored over WEBrick in the early days. Still, it can also run on Apache, Lighttpd, Cherokee, Nginx (either as a module – Phusion Passenger, for example – or via CGI, mod_ruby, or FastCGI), Hiawatha, and many others. From 2008 onward, Passenger succeeded Mongrel as the most-used web server for Ruby on Rails. Ruby is also carried natively on the International Business Machines (IBM). The Ruby on Rails is also noteworthy for its widespread use of the Script.aculo.us and Prototype, JavaScript libraries for scripting Ajax actions. Ruby on Rails initially utilized lightweight Simple Object Access Protocol (SOAP) for web services; RESTful web services later replaced this. Ruby on Rails 3.0 uses a technique called Unobtrusive JavaScript to divide the functionality (or logic) from the formation of the web page. JQuery is fully sustained as the default JavaScript library in Rails 3.1 and a replacement for Prototype, reflecting an industry-wide move towards jQuery.jquery. JQuery is fully sustained as the default JavaScript library in Rails 3.1 and a replacement for Prototype, reflecting an industry-wide move towards jQuery. CoffeeScript is also included in Rails as the default JavaScript language.

Ruby on Rails is intended to emphasize the Don't Repeat Yourself (DRY) and Convention over Configuration (CoC) principle. Convention over Configuration means a developer-only needs to define unique features of the application. For example, if there is a class Sale in the model, the same table in the database is called *sales* by default. It is only if one varies from this convention, such as calling the table *products sold*, that the developer is required to write code concerning these names—generally, Ruby on Rails conventions guide to less system and less repetition. Do not repeat yourself (DRY) means that information is found in a single, unambiguous position. For example, using the Active Record module of Rails, the developer does not need to specify database column names in class definitions.

Alternatively, Ruby on Rails can reclaim this information from the database based on the class name. Fat models, skinny controllers mean that greatest of the application thesis should be placed within the model while splitting the controller as light as possible. Rails running on Matz's Ruby Interpreter (the de-facto reference interpreter for Ruby) were criticized for scalability issues. These analysts often mentioned several Twitter outages in 2007 and 2008, which drove Twitter's partial transition to Scala (which runs on the Java Virtual Machine) for other middleware and their queueing system. The site's user interface aspects continued to run Ruby on Rails until 2011 when it was replaced due to concerns over performance. In 2011, Gartner Research noted that comparisons to Java and despite criticisms, many high-profile user web firms are using Ruby on Rails to create scalable web applications. Some of the largest sites running Ruby on Rails include Basecamp, Airbnb, GitHub, Shopify, Scribd, and Hulu.

As of January 2016, it is calculated that more than 1.2 million web sites are running Ruby on Rails. It used to be popular before 2014 but now the popularity of Ruby has declined dramatically. Ruby on Rails was originally popular because Ruby code is mostly self-documenting and easy to read. This increases productivity, as there is less needed to write out separate documentation, making it more accessible for other developers to pick up existing projects. Rails had developed a strong focus on testing and having proper testing frameworks. Rails are useful for rapid application development (RAD), as the structure makes it easy to accommodate changes.

Ruby is a dynamic, open-source, object-oriented, and reflective programming language. Ruby is considered to be similar to Smalltalk and Perl programming languages. Ruby is also able to run on all kinds of platforms like all versions of UNIX, Windows, and Mac OS (Operating System). Python is a simple, powerful, easy to learn, prominent level, and object-oriented programming language. It is also an interpreted writing language. Guido Van Rossum is known as the founder of python programming. Python is a simple programming language. The popularity of python has increased because people have begun to largely favor simplicity over complexity.

Ruby, on the other hand, is a dynamic programming language. Both programming languages can be used to develop web applications. But the use of Python is not only up to web applications. Python is publicly available, and they ready-to-use code is a relevant factor when choosing a programming language. Python's use in the field of data science and artificial intelligence is increasing. This is causing an increase demand for Python developers. In our opinion, and from the stats of the most trending language, Python is not only easier to use but also more convent.

Commented [AZ3]: Unsure about how to correct this

Similarities both languages have been: They both are high-level language. They both are a server-side scripting language. Both are used for web applications. Both work on multiple platforms. Both have clean syntax and are easily readable. Both use an interactive prompt called IRB (Institutional Review Board). Objects are strongly and dynamically typed. Both use embedded doc tools. A GNU Debugger(gdb) style is available for each language. Emacs modes support both languages.

Commented [AZ4]: Turn this into a paragraph

Organizations using Python are Google, Drobox, Instagram, Mozilla, Yahoo, Venom, YouTube. Organizations using Ruby are Apple, GitHub, Twitter, Hulu, ZenDesk, Urban

Dictionary. When it comes to developing web applications, Ruby on Rails is preferred. When it gets to all in one use like web applications, data science, some programming, Python is better. It depends on where to use and how to use it.

Python is an official language of Google. Most of Google's algorithms are written in Python and C++. Google App Engine was originally designed for the use of Python. It allows applications to be built with Python and gives access to a verity of libraries. Google Data Python Client Library, Google APIs Client Library for Python and Google AdWords API Python Client Library are just a few examples of their large pool of resources. One of the biggest ways that Google has utilized Python is through its platform YouTube. There Python is used for viewing videos, controlling templates for the websites as well as access to data. Google also utilizes Python for generating reports and analysis. Python's flexibility and ability for rapid development and scalability is the main reason Google has chosen to use it as their primary language over Ruby.

Ruby is similar to JavaScript in many ways but depending on the person, could be slightly easier to learn and use. We can compare the two to see why. Both languages are object oriented. JavaScript allows you to process and maintain data within the browser. JavaScript can also only be running on web-browsers. Ruby on the other hand is an open-source language. This allows us to be able to freely modify and edit to meet our own needs and uses. Another reason Ruby seems to be easier is the use of syntax. Ruby's variable declaration and assignment is easier compared to JavaScript. They are not only shorter but also more intuitive. Ruby can be very flexible with syntax that is not hard to remember.

There are many other reasons why a person would prefer Ruby over JavaScript. Unlike JavaScript, Ruby is a back-end programing language. This is better for developing the server side

Commented [RS5]: Going to fix that later

Commented [RS6]: Made the bulleted list into a paragraphed form list and adjusted grammar to be correct and not justpasted

Commented [AZ7]: Need to insert more of my stuff above here

Commented [AZ8R7]: Opted to stick it in the conclution instead. Made more sense

Commented [AZ9R7]:

of an application and everything that communicates between the database and the browser. JavaScript is more so used for front-end development. This is used for the look and feel of a website. There are many cons that also need to be pointed out on the use of Ruby. For one, Ruby is awfully slow in terms of performance. Simply debugging can be time consuming. Ruby is also not as scalable as JavaScript and that is why many companies have migrated to JavaScript. So, to summarize Ruby is great and easy to use in specific uses. Ruby can be used to develop applications that are CPU heavy and when the user wants to develop applications rapidly with fewer lines of code. JavaScript is better used for performance and the ability to scale in size.

While *Ruby on Rails* has been a seasoned framework, it has been on a downward spiral in terms of popularity lately. Many other languages and frameworks about have been taking charge for demand such as Java, C, and Python being in the top three for demand; but *Ruby* does still fit in the most popular list being in thirteenth place, per the *Tiobe index.* Although this popularity has been dropping, it is still widely used by several major trafficked websites such as *Hulu* and *GitHub*. Development with Rails has been slowing immensely due to it not having a speedy runtime. Compared to Node.js, it is not optimized for slower and smaller infrastructure, but is better suited for larger scale enterprises as mentioned earlier. This leads to another issue, since it is unoptimized for smaller infrastructures, there are not many uses for the inexperienced developers trying to gain experience in this area. They will only have access to consumer grade infrastructure, not showing them much leeway. An experienced developer with more enterprise grade hardware will have more work done for them compared to the inexperienced one since they are used to the framework altogether. Overall, Rails is on a downward spiral due to its maturity but is still in a stable demand due to being maintained for aging software.

As of recently, *Ruby* and Rails each have been updating for better usability, stability, and optimization per mass demand by the development industry. Since January, *Ruby* updated to 2.7 offering major improvements over 2.6 in terms of memory fragmentation and higher memory usage that 2.6 had issues with. 2.7 as well has newer powerful methods such as *Self* which allows calling on private methods. Rails itself had a major update in August 2019, their 5.0 update, allowing a stronger online text editor to be implemented and allowing *Webpack*, a JavaScript module bundler, to be default in its application. For experienced developers, this means a stronger support framework when they are preparing their innovative programs and maintained programs for higher optimization speeds and more features for already aging websites and for high trafficked websites. This allows more users on their servers. With current events effectively making the internet used now more than ever, these upgrades are incredibly necessary and able to hold up to tests in production. For newer developers, it can be considered more user-friendly and powerful to get a jump into *Ruby on Rails* Development, but as stated it may not be particularly the best choice especially if they are on a smaller infrastructure. Though there are still issues that people will find, even though this framework is still popular, many would enjoy an alternative to Rails based on their programming preferred languages.

Ruby on Rails while being a powerful framework, it is unfortunately outdated and slowing. Even though updates are rolling out to keep up with the modern times and hardware optimization (which is necessary to work properly on current hardware), it is not enough. People may not be inclined to get into this framework but can seek an alternative that is more appealing to them. One in particular that can be considered easier is *Django*, a Python web framework. Django being a python framework, it is newer and more user friendly than Rails; Python itself is an easier and more applicable language than Ruby. In terms of popularity, Python is in third place on the *Tiobe index* and is known for its simpler syntax. Many companies use Django such as Instagram, Spotify, even the media giant YouTube. There are a few issues though when it comes to the usage of this alternative framework, a few are more preferred by developers. Django does not have direct support for JavaScript unlike rails, as well as being slower than rails in most testing for raw performance on the server-side. Though the actual difference is within 0.7%, that can mean the world in modern loading times. However, each have a large community supporting them in the open source world, and that can be the real decision maker by the amount of contributions each side has.

In the modern development world, there are many issues to be run into such as not understanding all that is being asked of a client; open-source communities such as GitHub and GitLab are able to host repositories of programs for any other users to use at their own request and use it for their benefit. Ruby on Rails is an open-source project, and people from many different development communities come in and contribute what they believe is a momentous change that can make rails better. On the rail's website, they mention how it is open-source and over 5,000 people have contributed to making it better. Just checking GitHub alone and following both Rails and Ruby tags, there are over 25,000 public repositories that people can pull from and use to help their own works. One person, Rafael Mendonça França, has contributed to almost 9,000 commits to rails since 2010; that is double what the founder and creator of rails, David Heinemeier Hansson, has committed since 2004. People that appreciate their daily used software are more inclined to help make it better tenfold as França has proven. Rails, without the open-source community input, would be farther behind than most frameworks if it were not licensed like this; any company can put a proprietary license on any of their software, but to have it grow as rapidly and function better by the contributions of end users, that is incredible.

In conclusion, Ruby on Rails is on a decline. There are still may people would continue and will continue to use it for a variety of reasons. While it is a widespread use language, Python seems to be more preferred. Because Ruby has such a large variety or libraries and resources, it is a great language to use for web development. The most common framework being Ruby on Rails. It is a language that is focused on ease of use and many companies have, and still continue to, use it to build and secure websites. Maybe Ruby will make a comeback in the near future.

Work Cited

"Latest News." TIOBE, www.tiobe.com/tiobe-index/.

Noria, Xavier. "Rails Contributors - All Time Showing 5573 People." Rails Contributors - All Time, contributors.rubyonrails.org/.

"Ruby on Rails." Ruby on Rails, rubyonrails.org/.

"Ruby on Rails Alternatives and Similar Software." AlternativeTo, 22 Dec. 2019, alternativeto.net/software/ruby-on-rails/.

Sidorenko, Vladimir. "Recent Works." Gearheart, 22 Jan. 2017, gearheart.io/blog/ruby-on-rails-vs-django-performance-comparison/.

"Learn Python Tutorial - Javatpoint." *Www.javatpoint.com*, www.javatpoint.com/python-tutorial.

"Python vs Ruby: What's the Difference?" *Guru99*, www.guru99.com/python-vs-ruby-difference.html.

"ROR Application Development." *LotzAp Solutions*, 27 Feb. 2020, www.lotzap.com/ror-application-development/.

"Ruby on Rails: What It Is and Why You Should Use It for Your Web App." *Bit Zesty*, 12 Nov. 2019, bitzesty.com/2014/03/03/ruby-on-rails-what-it-is-and-why-we-use-it-for-web-applications/.

"Rubyonrails.org." *Wikipedia*, Wikimedia Foundation, 27 Feb. 2020, en.wikipedia.org/wiki/Rubyonrails.org.

Globalluxsoft. “The Best Sources to Learn Ruby on Rails.” *Medium*, Globalluxsoft, 29 Mar. 2018, medium.com/globalluxsoft/the-best-sources-to-learn-ruby-on-rails-6eecbe4001c6.

Employcoder. “Top Web Development Frameworks 2020.” *Hire Dedicated Coders for Web, Mobile, Cloud, IoT and More*, www.employcoder.com/top-web-development-frameworks-2020.