

RAILS TRAINING PROGRAM

Client Services

October 2020

Content

Content	1
Overall Context	1
Required Profiles	2
Timeline	2
Kickoff Meeting	3
Learning Levels	4
Level 1	4
Level 2	4
Level 3	5
Level 4	7
Closure Meeting	8

Overall Context

Rails is a ruby framework, that allows you to create fast, clean, maintainable systems, from micro services to monolithic applications. In order to meet our clients' increasing demand for this profile, the document aims to introduce a Rails training plan that will help experienced developers from other technologies to become enabled in Rails development.

This program will also help improve our current recruiting tests for Rails developers.

Required Profiles

The execution of this program requires the following roles:

- **Examiner:** Will be the person assessing the candidate Rails knowledge
 - Semi-senior Ruby on Rails developer or better
 - Good communication skills
 - Knowledgeable of Baires' interviewing methodology
- **Mentor:** Will be the person accompanying the candidate through his learning progress and will help the candidate with doubts or guidance when they appear
 - Senior Ruby on Rails developer
 - Good communication skills
 - Patience
- **Trainee:** Will be the person learning Ruby on Rails
 - Semi-senior developers or better at any technology
 - Motivated to learn Ruby on Rails

Timeline



TIME EXECUTION: 44 Hours

Kickoff Meeting

In order to trigger the training program, the Project Manager must schedule a meeting with the selector Examiner and the candidate under two main objectives, the first one is to explain the entire process, expectations, logistics, etc, and the second one is to execute the initial assessment for the scoring value. The steps should be the following:

1. First the new developer must be submitted to the Ruby training and examination and must be approved by the examiner
2. He/She should have a 1h call with an experienced Ruby on Rails developer to assess his/her seniority in Ruby on Rails and define the starting level.
3. It should be to evaluate his/her technical background and past experience with Ruby on Rails. Use this scorecard to estimate the technical knowledge: **TBD**
4. Each level requires some knowledge of the previous level, except for the Level 1
5. Based on the final score, we can determine where we can start the training.

NOTE: In case the developer executing this program does not possess any knowledge of Ruby, he/she must execute Level 1, 2, and 3 of the Ruby Training Program.

Learning Levels

- **Level 1**

- **Expected durations: 2 hours**
- **Topics:**
 - Getting started with Ruby on Rails:
 - <https://rvm.io/>
 - https://guides.rubyonrails.org/getting_started.html
 - Installing and setting up Ruby on Rails environment:
 - https://guides.rubyonrails.org/getting_started.html
 - Ruby in 20 minutes:
 - <https://www.ruby-lang.org/en/documentation/quickstart/>
- **Evaluation:**
 - Validate if the environment is correctly setup
 - Validate if the student can create a hello world application
 - Run the exercises proposed by the documents above

- **Level 2**

- **Expected duration: 20 hours**
- **Topics:**
 - Rails Architecture
 - Model
 - View
 - Controller
 - ApplicationController
 - Migrations
 - Configs
 - Working with Models
 - Connecting a database
 - Migrations
 - Creating migrations
 - Applying migrations
 - Rolling back migrations
 - Validations
 - Custom validators
 - Associations
 - has_many
 - belongs_to
 - has_one
 - has_and_belongs_to_many

- Creating a record
 - Updating a record
 - Destroying a registry
 - Working with Controllers
 - Default methods
 - Index
 - New
 - Create
 - Show
 - Edit
 - Update
 - Destroy
 - Permitted parameters
 - Using routes
 - Rendering page
 - Root page
 - Error pages
 - Named Route helpers
 - Working with Views
 - Creating an product form
 - Creating the product list page
 - Updating a product
 - Deleting a product
 - Navigating between pages
- **References**
 - https://guides.rubyonrails.org/getting_started.html
 - https://guides.rubyonrails.org/action_controller_overview.html
 - <https://guides.rubyonrails.org/routing.html>
- **Evaluation**
 - Exercises: See Level 2 - Exercises to practice document
 - Evaluation: See Level 2 – Evaluation document
 - Test: See scorecard test questions
- **Level 3**
 - **Expected duration: 12 hours**
 - **Topics**
 - Creating an API
 - Models
 - Changing database
 - Update tables

- Scopes
 - Custom scopes
 - Default scopes
- Callbacks
 - before_validation
 - after_validation
 - before_save
 - around_save
 - before_create
 - around_create
 - after_create
 - around_update
 - before_update
 - after_update
 - after_save
 - after_commit/after_rollback
 - before_destroy
 - around_destroy
 - after_destroy
 - after_commit/after_rollback
- Difference between delete and destroy
- Controllers
 - Lifecycle methods
 - before_action
 - after_action
 - Around_action
 - Methods
 - Creating a registry
 - Working with parameters
 - Updating a registry
 - Destroying a registry
 - Parameters
 - Database transactions
 - Handling Errors
- Routes
 - HTTP methods
 - GET
 - POST
 - PUT
 - PATCH
 - DELETE
 - OPTIONS

- Resources
 - Creating resource
 - Nested resources
 - Views
 - Jbuilder
 - Working with simple objects
 - Working with arrays
 - Testing your api with Postman
 - Mailer
 - Sending an email with rails
 - **References**
 - https://guides.rubyonrails.org/active_record_callbacks.html
 - <https://github.com/rails/jbuilder>
 - **Evaluation**
 - Exercises: See Level 3 - Exercises to practice document
 - Evaluation: See Level 3 – Evaluation document
 - Test: See scorecard test questions
- **Level 4**
 - **Expected duration: 4 hours**
 - **Topics**
 - Rails Way
 - 12 factor app
 - Loading environment variables
 - DRY
 - SOLID
 - Paranoia
 - Tools
 - Rubocop
 - Rubycritic
 - Models
 - Creating concerns
 - Controllers
 - Creating modules
 - Api versioning
 - HTTP headers
 - Views
 - Creating json partials
- **Level 5**
 - **Expected duration: 4 hours**

- **Topics**
 - Rails Gems
 - Devise
 - Kaminari
 - Creating tasks
 - Unit testing with RSpec
 - Testing models
 - Testing controllers
 - Testing concerns
 - Factory bot
 - Shoulda matchers
- **References**
 - <https://12factor.net/>
 - <https://github.com/kaminari/kaminari>
 - <https://github.com/heartcombo/devise>
 - <https://github.com/rubocop-hq/rubocop>
 - <https://github.com/whitesmith/rubycritic>
 - https://github.com/thoughtbot/factory_bot
- **Evaluation**
 - Exercises: See Level 4 - Exercises to practice document
 - Evaluation: See Level 4 – Evaluation document
 - Test: See scorecard test questions
- Devise JWT

Closure Meeting

At the end of the process, the Mentor must schedule a final meeting with the trainee to provide overall feedback and results. This is the moment not only to share scores and technical suggestions moving forward but also to ask for potential improvements on the program about methodology, content, exams, etc. that we could incorporate into to make it better every time. The PM must participate in this meeting to write down the actions items.