



Ruby on Rails Training - 1

Base Rails

Summary: Rails is a powerful opensource framework combining high productivity and quality code while benefiting the support of a large community of devoted users.

With this D04, you will learn to domesticate its most accessible sides.

Version: 1

Contents

I	Preamble	2
II	General rules	3
III	Today's specific instructions	4
IV	Exercise 00: CheatSheet	5
V	Exercise 01: Moar CheatSheet	7
VI	Exercise 02: Quick search	9
VII	Exercise 03: Diary	10
VIII	Submission and peer-evaluation	11

Chapter I

Preamble

Philosophy

- DRY : Don't Repeat Yourself
- Convention over configuration
- REST : Representational State Transfer
- CRUD : Create, Read, Update, Delete

Library

- [Your new bible](#)
- [Your new toolbox](#)
- [Your new evening news program](#)
- [Your new TV](#)

Chapter II

General rules

- Your project must be realized in a virtual machine.
- Your virtual machine must have all the necessary software to complete your project. These softwares must be configured and installed.
- You can choose the operating system to use for your virtual machine.
- You must be able to use your virtual machine from a cluster computer.
- You must use a shared folder between your virtual machine and your host machine.
- During your evaluations you will use this folder to share with your repository.
- Your functions should not quit unexpectedly (segmentation fault, bus error, double free, etc) apart from undefined behaviors. If this happens, your project will be considered non functional and will receive a 0 during the evaluation.
- We encourage you to create test programs for your project even though this work **won't have to be submitted and won't be graded**. It will give you a chance to easily test your work and your peers' work. You will find those tests especially useful during your defence. Indeed, during defence, you are free to use your tests and/or the tests of the peer you are evaluating.
- Submit your work to your assigned git repository. Only the work in the git repository will be graded. If Deepthought is assigned to grade your work, it will be done after your peer-evaluations. If an error happens in any section of your work during Deepthought's grading, the evaluation will stop.

Chapter III

Today's specific instructions

- All the turned-in files will be Ruby On Rails web applications.
- Today's projects' Gemfile must include:


```
gem 'rubycritic', :require => false
```

When running the "rubycritic" command, the html report page's "smells" must be EMPTY!!!

- Except if specified otherwise in the subject, NO additional Gem will be authorized to complete these exercises.
- If there is no gem you can use yarn to install what you need.

Chapter IV

Exercise 00: CheatSheet

	Exercise 00
Exercise 00: CheatSheet	
Turn-in directory : <i>ex00/</i>	
Files to turn in : CheatSheet	
Allowed functions : bootstrap , rubycritic	

Satisfied with your first "Hello World" in rails, you feel like you've grown wings and choose to push what you've learned from your new companion a little further.

Not too far, though... As a matter of fact, we're gonna spend the day using rails' most simple mechanisms.

If you spent the majority of your time on Wikipedia yesterday, I suggest you take D03-Ex04 before going any further.

NO DATABASE, NO 'CRUD' for now. We will just focus on views.

Go [here](#) and feast your eyes on this wonderful site that will remind you of the useful commands...

You must replicate this page, including footer. It will just be the page this time. The navigation system will be for some time later. By the way, don't waste time making pixel perfect css. However, your result will have to remotely look like this page.

You will have to observe this few instructions:

- The app will be named "CheatSheet"
- It just contains a controller of your own design, `application_controller` which must be left as is
- The main page's title is "CheatSheet"
- You don't have to replicate the nav-bar (the red thing at the beginning of a page) for the moment
- You will have to design your page a little. We don't demand a professional ready-to-produce design, but your page will have to be pleasant to look at.


Your HTML should be where it belongs. The controller should be in its folder. No CSS will be tolerated in an HTML file! Use Rails like you ought to.

Don't be silly. The CheatSheet is aptly named. It will help you in the arborescence of a Rails application as well as in the useful commands.

You should also use the "inspect element" command of your browser and be smart... Remember yesterday's crawler.

Chapter V

Exercise 01: Moar CheatSheet

	Exercise 01
Exercise 01: Moar CheatSheet	
Turn-in directory : <i>ex01/</i>	
Files to turn in : NewCheatSheet	
Allowed functions : bootstrap , rubycritic	

Ok. Now you know how to make a One-Page with Rails, it's time to talk serious business.

Pick your previous application. You will now implement the nav-bar (the red thing...) and change you CheatSheet navigation a little bit.

Practically, you're gonna have to separate each part of this CheatSheet into independent pages and assign a nav-bar button to the matching page.


What's expected:

- The app will be named "NewCheatSheet"
- It stills includes just one controller of your own design, **application_controller** you will leave as is
- Each page will have a dedicated "title" tag
- You must write the "nav-bar" code just once!
- The first page is the "Convention" page. It will be your root page

- All the following pages must be represented:
 - convention
 - console
 - ruby
 - ruby-concepts
 - ruby-numbers
 - ruby-strings
 - ruby-arrays
 - ruby-hashes
 - rails-folder-structure
 - rails-commands
 - rails-erb
 - editor
 - help
- "Resources" and "About" pages won't have to be replicated nor included in the nav-bar.
- You will have to design your page a little. We don't demand a professional ready-to-produce design, but your page will have to be pleasant to look at.

Chapter VI

Exercise 02: Quick search

	Exercise 02
Exercise 02: Quick search	
Turn-in directory : <i>ex02/</i>	
Files to turn in : CheatSheet	
Allowed functions : <code>jquery</code> , <code>bootstrap</code> , <code>datatables.net-bs4</code>	

This CheatSheet is useful, right? But it still can be enhanced.

It will take a `jQuery` plugin and a little content reformatting... Add a thumbnail named "Quick search" that will include a table incorporating the content of ALL the other tables of the CheatSheet.

This table will be **properly formatted**, with a header, the proper number of columns, properly located tags etc.


To be clear, you will have to:

- Implement a new "Quick search" thumbnail in the nav-bar.
- Let this thumbnail have its personalized title tag, like the rest of the site.
- Let its content be a recap table of ALL the other pages' commands.

NB: If an error message pops up or Javascript appears in the `html.erb` file, the exercise will be invalid.

Chapter VII

Exercise 03: Diary

	Exercise 03
Exercise 03: Diary	
Turn-in directory : <i>ex03/</i>	
Files to turn in : CheatSheet	
Allowed functions : bootstrap, rubycritic, jquery	

The CheatSheet project starts looking good. But as any seasoned developer, you're keeping a technical journal, and it would be really neat to include it in your project.

Create a new thumbnail called "Log Book" that features a form with a "Write" button and a field to include text in the page header. Once the form is filled and submitted, your app will add the text in the form to a "entry_log.txt" file, located at the root of your CheatSheet project's folder.

The text will have to be formatted on the fly adding date and time as follows at the beginning of the line:

```
15/07/2016 23:42:04 : The whole team is asleep, Jim.  
15/07/2016 23:41:26 : The whole team is yawning, Jim.
```

A loop manages the text display: each entry must be displayed after the form, with its own tag from the oldest to the newest entry.

Each time you click on the form's "Write" button, you must be redirected on the same page, with the new line showing on top.

Chapter VIII

Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your folders and files to ensure they are correct.



The evaluation process will happen on the computer of the evaluated group.