

Lab Assignment 2

Fall Semester 2019

March 20th

1 Goals

On this second lab assignment you will be introduced to the Ruby programming language, and the web development framework Ruby on Rails. By the end of this assignment you will have a basic-intermediate command of the Ruby language, and a glimpse on how Ruby and Ruby on Rails works.

You will be using the same GitHub account with which you submitted last week's assignment, so you should remember your credentials to log in.

2 Initial Steps

Firstly, you need to sign up at the classroom assignment on Github, following the same steps as last week. Go to the [Github classroom assignment](https://classroom.github.com/a/zFRkBUZZ) at <https://classroom.github.com/a/zFRkBUZZ> (copy and paste this URL in your web browser) and accept the invitation to the assignment.

After your personal repository has been created, issue a `git clone` command on the terminal specifying the URL provided by Github, to your Desktop location in you computer. You will find the clone URL in the button "Clone or Download" located at your repository page.

```
# switch to the ~/Desktop
cd ~/Desktop
git clone <place the url for the repository here>
```

This project does not contain a README.md file as the one of the past assignment, so you have to create one and upload it to the remote repository:

```
cd <repository name>
touch README.md
git commit -a -m "Add README.md"
git push origin master
```

3 Tools

If you use Windows, it is highly recommended that you install a virtual machine based on VirtualBox, and install Ubuntu Desktop 18.04 LTS on it (see specific instructions below). If you use

macOS, please install the [homebrew package manager](#). Lastly, if you use Linux, you are almost set. After having a working operating system environment, see below how to install RVM, Ruby and Rails.

3.1 Windows-specific Instructions

To install a working virtual machine on Windows, you will need the following downloadable tools and software:

1. VirtualBox: Free virtualization software provided by Oracle. Get it at <http://www.virtualbox.com>.
2. Ubuntu 18.04 LTS desktop image: <http://releases.ubuntu.com/18.04/>.

It is strongly recommended that you install VMWare Tools. See this video and follow the instructions <https://www.youtube.com/watch?v=qNecdUsuTPw>.

3.2 Install RVM, Ruby, and Rails

Once you have a working Unix-based environment on your computer, you must ensure that ruby is installed. Open a terminal and issue the following command:

```
ruby -v
```

Depending on the operating system installed, the version of ruby provided should be at least 2.1. Now, install RVM by issuing the following commands (see <http://www.rvm.io> for further details):

```
$gpg2 --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E3 \
7D2BAF1CF37B13E2069D6956105BD0E739499BDB
$\curl -sSL https://get.rvm.io | bash -s stable
$\curl -sSL https://get.rvm.io | bash -s stable --rails
```

Now install ruby version 2.5.1 using rvm:

```
$rvm install 2.5.1
$rvm use 2.5.1 --default
```

Finally, create a gemset for your project:

```
$rvm gemset create miniblog
$rvm use 2.5.1@miniblog
```

Finally, install Ruby on Rails 5.2.2:

```
gem install rails --version 5.2.2
```

By now you should be able to create your first rails application. But first, you should learn some ruby.

4 Ruby Warmup

In this first activity you will get acquainted with ruby and have some practice by following the introduction tutorial available at the [RubyMonk](http://rubymonk.com) site. See the introductions to Objects, Strings, Conditions, Arrays and Hashes. If you already know how to program in the Ruby language you can skip this part and head to the programming problems described below.

4.1 Ruby Tutorial

Please complete the following chapters and sub-chapters of the RubyMonk tutorial:

1. Introduction To Ruby Objects: 00 - 01
2. Introduction to Strings: 00 - 01 - 02
3. Conditions and Loops: Control Structures in Ruby: 00 - 01
4. Arrays in Ruby: 00 - 01 - 02
5. Hashes in Ruby: 00 - 01

RubyMonk site: <http://rubymonk.com/learning/books/1-ruby-primer/>

4.2 Programming Problems

1. `sum_of_cubes`:
http://rubymonk.com/learning/books/1-ruby-primer/problems/144-sum_of_cubes
2. `frequency_finder`:
<http://rubymonk.com/learning/books/1-ruby-primer/problems/6-frequency-finder>

For each of the problems listed above, you need to create a Ruby file named as the corresponding problem title, e.g., `sum_of_cubes.rb`. Note that the file name must include the ruby extension `.rb`. Place the files within a folder called `problems`. This folder must be part of your Git repository,

thus place it under your root repository folder.

You may work on local source files created with any text editor, or directly code on the Ruby-Monk online editor. Please always test your code on your local machine.

To run your code, please issue the following commands in a terminal window:

```
cd ~/Desktop/<your_repository_name>/problems
ruby <problem_name>.rb
```

4.3 Ruby on Rails (RoR)

4.3.1 Ruby Environment

We will be using the RVM (Ruby Version Manager) command line tool, which simplifies working with multiple versions of the ruby language as well as multiple third party libraries (known as ‘gems’ in the ruby language). This is really helpful when you are testing different libraries and want to switch to a newer (or older) version of ruby. To see which instances of Ruby are installed, issue the following command:

```
rvm list
```

You should be able to see that ruby is installed and currently in use. Recall that we previously set version 2.5.1 as default. As we work on this assignment (and on the next ones) we want to have a common place where to put all of our third party and dependency libraries, so for that we will use “Gemsets”. Gemsets are the way in which RVM allows working with independent and self-contained library dependencies, facilitating simultaneous development of different applications. By default, there exists a ‘global’ gemset, which is created once you install a new ruby version. We created a gemset called miniblog previously. To ensure that version 2.5.1 is being used with the miniblog gemset, issue the following comand:

```
$rvm use 2.5.1@miniblog
```

4.3.2 Initialize Application

Now that we understand how our Ruby and Ruby on Rails environment and tools work, we can move on to create our first web application. Go to the repository folder and execute the following command:

```
$rails new microblog
```

This will create a new Ruby on Rails project, using version 5.2.2 that you already installed, under the 'microblog' gemset. After your project has finished initial setup, commit your changes to your working copy and push them to GitHub. Use the commit description 'Ruby on Rails project added'.

```
git commit -a -m "Ruby on Rails project added"
git push origin master
```

You will notice a bunch of new folders and files that the Ruby on Rails framework has created for you. Let's do a quick review of the most important ones for a basic understanding:

- **Gemfile:** located at the root of your app folder. This file specifies all the third party and dependency libraries that your RoR application depends on.
- **config** folder: inside the **config** folder you will find all the credentials (as for DB access) or initialization variables needed for different modules and working setups.
- **config/routes.rb:** set the convention for the URLs configuration in our application. So basically, here is where we define how our application will respond to a particular URL. A common pattern is to match a particular route to a specific controller method/action.
- **app/controllers** folder: here you will define all the controllers that your applications relies on. Usually, the naming convention for a controller will be **module_controller.rb**, where the module can be the name of a resource or a module with a specific purpose.
- **app/views:** here you will find all the views related to the different HTML endpoints that your controllers will respond to. You will find some **.html.erb** files. These files are composed of HTML, as well as ruby code, which is processed in the server side, before the resulting view is sent back to the client.
- **app/models** folder: here you can define the models of the data that your application will work with.
- Rails use Convention over Configuration and most of the things you would like to do will be done with just a couple of commands in the terminal.

Now, execute the following commands on a terminal window inside your root folder application of the RoR project:

```
cd ~/Desktop/<repository_name>/microblog
bundle install #install dependencies from Gemfile
rails s #run the localhost server to host the application
```

Go to your prefer web browser and visit <http://localhost:3000>. You will see your first Ruby on Rails application working :).

Now please follow the instructions in the following tutorial which will lead you to create your first Ruby on Rails application, a microblog:

https://www.railstutorial.org/book/toy_app. Follow sections 2.1.1 up to 2.2.3. All the work has to be done in your previous rails applications created in your repository.

5 Way of Working

This work must be done individually. The submission must comprise at least one of the problems listed in the Ruby Warmup section of this document, and the Ruby on Rails microblogging application. Attendance points will be awarded upon submission of your work to your personal GitHub repository.