Tute/Lab Session (Week 2)

Welcome to RAD! In this course, you will be learning Ruby programming, Ruby on Rails framework, MVC, TDD and many other concepts, methodologies, tools and skills widely used in industry. The key to a successful completion of this course is practice, practice and practice!

PART 1: Ruby Syntax

1. Are the following statements syntactically correct?

```
a) sum = assignment +
         exam

b) tax = income - threshold
         * tax_rate

c) average = a + b \
                  2;
```

2. Assume we have a global variable `tax_rate` defined and set to 10%, define a class `Car` so it supports the following use:

```
car1 = Car.new("Nissan", 15000)
car2 = Car.new("BMW", 45000 )
puts car1             #print=> Car model: Nissan, price: 16500
puts car2             #print=> Car model: BMW, price: 49500

car1.price= 60000;
puts car1             #print=> Car model: Nissan, price: 60000
```

PART 2: Setting up and simple exercises

3. Set an account with Github or Bitbucket if you don't have one yet.

https://github.com/

https://bitbucket.org/

4. Set an account with Heroku if you don't one yet.

https://www.heroku.com/

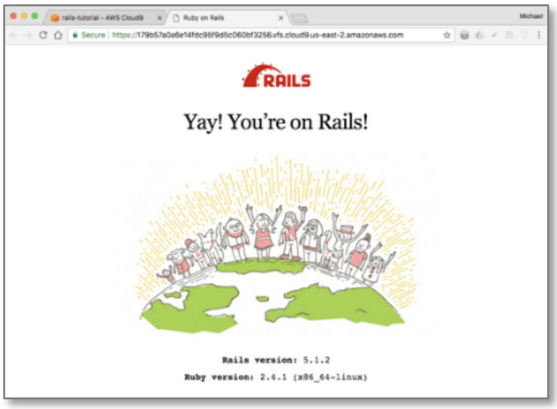
5. Use your favourite code editor to explore the file directory that you created above. Some editors/IDEs include Visual Studio Code, RubyMine, Aptana Studio, Notepad++, Sublime, Komodo and so on.

6. **AWS Cloud 9** This semester we will be using AWS Cloud 9. You can also use docker to setup local installation of Ruby on Rails.

You should have received an invitation from AWS to your student email. You could also sign up for free https://aws.amazon.com/free/ (credit card required).

`https://www.learnenough.com/ruby-on-rails-4th-edition-tutorial/beginning#sec-development_environment`

Follow the instructions to set up your Cloud 9 environment, install Rails and start your first app.


rails-tutorial - AWS Cloud9
Ruby on Rails
Michael
Secure
https://179b57a0a6e14fdc95f9d5c060bf3256.vfs.cloud9.us-east-2.amazonaws.com
RAILS

Yay! You're on Rails!

Rails version: 5.1.2
Ruby version: 2.4.1 (x86_64-linux)

Alternatively you can follow the instructions below:

Make sure you are using the right version of rails. You may need to remove rails and rail-ties and install rails 5.2. Check online to see how to reinstall rails with a specific version. To create a new app in AWS Cloud9, the directions are basically the same, but you need to use `rails new` (see note below first) first and move all code out of the new folder and into the workspace (root) folder, so you don't need to CD into your project.

Note: When using "Rails new" it is useful to review the `rails new -h` to see what flags you can use. Since we use postgres, you'll need to use: `rails new app_name --database postgresql`

Workspace Settings

When creating a Cloud9 Environment, use **Ubuntu** not Amazon Linux

Cloud9 Editor Settings

- Strip Whitespace
- Soft tabs (2)

Postgres

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
sudo apt-get update
sudo apt install postgresql postgresql-contrib libpq-dev

# ubuntu User setup
sudo service postgresql start
sudo sudo -u postgres psql
CREATE USER ubuntu SUPERUSER PASSWORD 'password'; # You may want to use something more secure
```

Ruby

First install the correct version of Ruby. Remove the default version if a downgrade is needed.
`rvm list`

Remove unwanted version: `rvm remove <not required version>`

Install and use default version: `rvm use --default --install <required version>`

Rails

run `rails -v` to check your current version of rails. If that isn't the version you want, remove the current railties (you may need to add `-v '5.0.0'` at the end) or something if you need to uninstall a specific version.

`gem uninstall railties`

Install the correct railties, in this case, anything that is at least 5.2.X but less than 5.3 (`~>` limits versions) `gem install railties -v '~> 5.2.0'`

Here is an example for an exact version: `gem install railties -v '5.2.4.1'`

Once that installs, you should update the gemfile to use the version of rails installed by the railties Update the gemfile to use the correct version (if needed), e.g. `gem rails, '~> 5.2.0'`

You should replace the ~> 5.2.0 with the actual version it installs e.g. `gem rails, '5.2.4.1'`

Rails Setup

- For New Rails App

Create a new rails app if needed Add Setup Rails if needed (if not migrating)
https://guides.rubyonrails.org/v5.2/getting_started.html

Use `rails new -h` to get the options for setup.
For simple rails apps, one can use these options:

- `--database=postgresql` for postgres
- `--skip-test` if you won't be following TDD or plan to use `rspec` instead of the built-in test lib.

- For Existing Repo

Use `git clone.` Once the repo is cloned, make sure to move all the files (check that you have turned on viewing of hidden files in cloud9) Note: It is possible to git clone from a heroku app, but it is not ideal to rely on this as it will only contain the branch that was pushed to heroku (typically master). This is especially the case as GitHub now offers private as well as public repos for free.

- Bundler Setup

```
gem install bundler
bundle install
```

Postgres Rails DB Setup

```
# Do the postgres setup for cloud9
rake db:create
rake db:migrate
rake db:seed # Optional, may not always be needed or desired. Use caution.
             # For new rails apps, this is safe to run as it does nothing.
```

You can run them all at once using `rake db:create db:migrate db:seed`

Rails Console

Solves `FATAL: Listen error: unable to monitor directories for changes.` error if it comes up.

```
 echo fs.inotify.max_user_watches=524288 | sudo tee -a /etc/sysctl.conf && sudo sysctl -p
```

Gitignore

```
# See https://help.github.com/articles/ignoring-files for more about ignoring files.
#
# If you find yourself ignoring temporary files generated by your text editor
# or operating system, you probably want to add a global ignore instead:
#   git config --global core.excludesfile '~/.gitignore_global'

# Ignore bundler config.
/.bundle

# Ignore all logfiles and tempfiles.
/log/*
/tmp/*
!/log/.keep
!/tmp/.keep

# Ignore uploaded files in development
/storage/*
!/storage/.keep

/node_modules
/yarn-error.log

/public/assets
.byebug_history

# Ignore master key for decrypting credentials and more.
/config/master.key

/public/packs
/public/packs-test
/node_modules
/yarn-error.log
yarn-debug.log*
.yarn-integrity

.c9/
```

DOCKER (not recommended)

Make sure install the docker and docker-compose across Mac, Windows, and Linux.

https://docs.docker.com/compose/install/

Install Compose on macOS

Docker Desktop for Mac and **Docker Toolbox** already include Compose along with other Docker apps, so Mac users do not need to install Compose separately. Docker install instructions for these are here:

- Get Docker Desktop for Mac https://docs.docker.com/docker-for-mac/install/
- Get Docker Toolbox (for older systems) https://docs.docker.com/toolbox/overview/

Install Compose on Windows desktop systems

Docker Desktop for Windows and **Docker Toolbox** already include Compose along with other Docker apps, so most Windows users do not need to install Compose separately. Docker install instructions for these are here:

- Get Docker Desktop for Windows https://docs.docker.com/docker-for-windows/install/
- Get Docker Toolbox (for older systems) https://docs.docker.com/toolbox/overview/

If you are running the Docker daemon and client directly on Microsoft Windows Server, follow the instructions in the Windows Server tab.

Install Compose on Linux systems

On Linux, you can download the Docker Compose binary from the Compose repository release page on GitHub. https://github.com/docker/compose/releases
Follow the instructions from the link, which involve running the curl command in your terminal to download the binaries. These step-by-step instructions are also included below.

Attention ! If you are running Docker on **Linux**, the files rails new created are owned by root. This happens because the container runs as the root user. If this is the case, change the ownership of the new files.

```
sudo chown -R $USER:$USER
```

This requires user to have admin access on Linux Machine.

Installing Ruby on Rails distribution on Docker, using the link listed for both Bitbucket and Github.

https://github.com/pigfly/RMIT-RAD/tree/master/2021_s1

Download the repository to your local machine and unzip the directory. Enter the directory (you may rename the directory first) on command line environment, then use the following command to download the rails docker image and to build

```
> docker-compose run web rails new . --force --no-deps --database=postgresql
```

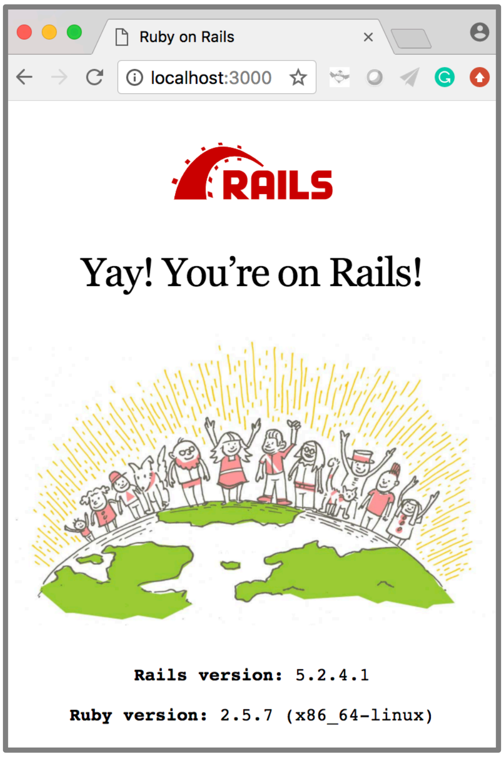
```
> docker-compose build
```

Then you can check file `config/database.yml` which should show

```
default: &default
  adapter: postgresql
  encoding: unicode
  host: db
  username: alex
  password:
  pool: 5

development:
  <<: *default
  database: myapp_development


test:
  <<: *default
  database: myapp_test
```



Yay! You're on Rails!

Rails version: 5.2.4.1

Ruby version: 2.5.7 (x86_64-linux)

Run the following commands on two separate terminals:

```
> docker-compose up
```

```
> docker-compose run web rake db:create
```

Then on your web browser, you should see your first Rails application running at this URL

http://localhost:3000

To shut it down, use this command `> docker-compose down`