Extra Resources

[Status Codes](https://developer.themoviedb.org/docs/errors)

[Docker Image Pull](https://docs.docker.com/reference/cli/docker/image/pull/)

[Database Rails Commands](https://learnetto.com/tutorials/rails-database-commands-cheatsheet)

To access the container we created

* Get into docker and sign in
* Get into the DockerWorkspace
  + Cd DockerWorkspaceor the path to get into the DockerWorkspace
* Docker attach container\_name(the container name)
* Cd into the portfolio app
  + You can do this by going into your finder and right click on the path and paste it into the terminal

If the container isn’t running this command “docker container\_name”

To start a new shell session inside the running container

* “docker exec -it container\_name /bin/bash”

To delete/remove/create a database

* Start the container(docker start container\_name)
* cd portfolio\_app
* Whatever command your trying to run
* Rails server -b 0.0.0.0
* Go back to the docker app
* Hit the 3000:3000 that brings you to a local host
* To see if it got deleted/removed/created go to db file students then sqlite viewer

Create a Docker Container

* Create a directory for your ruby exploration files(Windows: C drive, Mac: documents)
* Command line
  + Go to that directory and run docker to create a container
  + Docker run -it -p 3000:3000 -v $(pwd):/workspace dockerUserName/ dockerImage

Rails Scaffold

* Access a container
* Use an IDE and open your directory/folder
* Create a branch and “checkout” so you can commit and push to it
* Run the scaffold command(below-first line) then migrate the database(below-second line)
  + #rails generate scaffold portfolio\_app(directory) name:string school\_email:string minor:string graduation\_date:date
  + #rails db:migrate

To start the testing

* docker exec -it user\_name/image\_name
* docker attach container\_name
* cd portfolio\_app
* rails test test/models/student\_test.rb -v
* rails server -b 0.0.0.0
* Click on 3000:3000 button on app(docker app)

Commands

Docker

* “docker ps -a” :process status show all containers
* “docker stop <container\_id\_or\_name>” :terminates the running container
* “docker restart <container\_id\_or\_name>” :restarts the running container
* “docker exec -it <container\_id\_or\_name> /bin/sh” :runs a new command in a running container

Ruby

* “ruby -v” :ruby’s version
* /workspace# “irb” :interactive ruby console

Rails

* # “rails server -b 0.0.0.0” :start the server
* “rails db:migrate” :add/remove/modify the database schema
* “rails db:create” :create the database
* “rails db:rollback” :revert to the most recent modification to the schema
* “rails db:truncate\_all” :to delete all data from your database without dropping the schema
* “rails db:drop” :if you want to completely remove the database, including all tables
* “Rails destroy scaffold portfolio\_app” :after rolling back the migration, you can remove the scaffold by deleting the generated files. Rails provides a way to destroy a scaffold using the destroy command

Git

* “git tag” :creates a tag on the commit HEAD
* “git status” :tells you what branch your on
* “git branch name” :creates a branch “name”
* “git log” :lists the project history
* “git checkout name” :sets as working branch
* “git add .filename” :adds that “filename”
* “git commit -m ‘message’ “ :captures a snapshot of the projects currently stages changes with a message
* “git push –set-upstream origin name” :git pushing the new information upstream to “name” branch
* “git merge main” :merges the new changes with the branch “main”
* “git push” :uploads local repo content to a remote repo
* “git fetch –all” :downloads commits, files, and refs from a remote repo into your local repo
* “git reset –hard main” :undoes changes to files in the working directory
* “git pull” :fetch and download content from a remote repo and immediately update the local repo to match that content

Linux

* “ls” :list files and directories at path
* “pwd” :print present working directory
* “cd” :change current directory
* “sudo” :run command with superuser privileges
* “nano” :open files with Nano text editor
* “uname” :display system information
* “curl” :transfer data from or to server
* “find” :find files and directories at specified path
* “mkdir” :create new directory
* “rmdir” :remove empty directory
* “ps” :display process status information
* “df” :display disk space usage
* “touch” :create new file or update timestamp
* “cp” :copy files or directories from source to destination
* “chmod” :change file or directory permissions
* “cat” :display the contents of a file
* “head” :display first N lines of file
* “tail” :display last N lines of file
* “wc” :print the number of lines, words, and characters in a file
* “tar” :archive file together into a single file
* “zip” :compress files into a zip archive
* “unzip” :uncompress files from a zip archive
* “env” :print all environment variables and their values
* “~” :home directory(special path)
* “/” :root directory(special path)
* “ls” :list files and directories
* “rm” :remove file or directory
* “mv” :moving a file or directory
* “more” :print contents page by page
* “uniq” :filters out repeated lines