

HW03 instructions

Ruby On Rails Course



Prerequisites

Create a new branch for working on this assignment

- ✦ Checkout to the master branch so the new branch doesn't contain unmerged changes from HW2:
- ✦ `git checkout master`
- ✦ Please create a new branch from the master branch for working on this assignment called HW3 with the following command:
- ✦ `git checkout -b HW3`

Ensure you have the right Rails version

- ✦ Please run in your terminal: ``rails -v``. If the command says 'Rails 4.2.2' go to the next slide, otherwise run:
- ✦ `gem install rails -v 4.2.2`
- ✦ reload your terminal
- ✦ once again when you run ``rails -v`` it should say ``Rails 4.2.2``.

Assignments

Assignment #1: Setup the database

Make a new rails application called reddit_homework_03

✦ rails new reddit_homework_03



Subreddit model

- ✦ The subreddit model should contain (along with the default id and timestamps):
 - ✦ name: string
 - ✦ description: text

Subreddit model validations

- ✦ The subreddit model should have the following validations:
 - ✦ name:
 - ✦ must be present
 - ✦ must have a minimum of 3 characters and max of 20
 - ✦ must be unique
 - ✦ description:
 - ✦ maximum of 500 characters



Post model

- ✦ The post model should contain (along with the default id and timestamps):
 - ✦ title: string
 - ✦ content: text
 - ✦ a reference to the subreddit it belongs to (i.e. the subreddit_id)



Post model validations

- ✦ The post model should have the following validations:
 - ✦ title:
 - ✦ must be present
 - ✦ description:
 - ✦ must be present
 - ✦ subreddit_id:
 - ✦ must be present

Comment model

- ✦ The comment model should contain (along with the default id and timestamps):
 - ✦ content: text
 - ✦ a reference to the post model it belongs to (i.e. the post_id)

Comment model validations

- ✦ The comment model should have the following validations:
 - ✦ content:
 - ✦ must be present
 - ✦ post_id:
 - ✦ must be present

Associations

- ✦ Please create the appropriate associations between the
 - ✦ subreddit model and post model
 - ✦ the post model and comment model



Assignment #2: Show the data in the views

Fill your database

- ✦ To show something in your views, you need to create some database records. Create a few records through the rails console:
 - ✦ a couple of subreddits
 - ✦ create some posts for some (or all) subreddits
 - ✦ create some comments for some (or all) posts
- ✦ We will not be able to see how many records you've created, but you need some so you can see what you're doing. Checkout the slides from the 3rd lecture for references on how to do that (or just google it :).



Routes

- ✦ We will create 3 routes:
 - ✦ The root route which will show all the posts (no matter in which subreddit they belong to).
 - ✦ A route where we can view details about a subreddit with all of it's belonging posts.
 - ✦ A route where we can show data about the post as well as it's belonging comments.
- ✦ Check the instructions for more details...

Routes

```
Rails.application.routes.draw do
  root 'static_pages#index'
  get 'subreddits/:id', to: 'subreddits#show',
                        as: :subreddit
  get 'subreddits/:subreddit_id/posts/:id', to: 'posts#show',
                                              as: :subreddit_post
end
```

The `:id` and `:subreddit_id` are dynamic path parameters.

Usage of path helpers:

```
subreddit_path(id: subreddit.id)
```

```
subreddit_post_path(subreddit_id: post.subreddit.id, id: post.id)
```

StaticPagesController#index

- ✦ Create a controller named `StaticPagesController` with an index action that the root route will point to (do this by hand, not with a generator).
- ✦ In the index action you should have a `@posts` instance variable which will hold all the posts.
- ✦ Create a `static_pages/index.html.erb` view
- ✦ Follow the instructions on the next slide for the details on what you should show in the view



The static_pages/index view

- ✦ The view should list all the posts with each item containing:
 - ✦ a link to the post with its title using the `subreddit_post_path` helper
 - ✦ the time that has passed since it was created - in words. Checkout this [nifty helper](#). If the post was created 6 hours ago it should state 'created 6 hours ago'.
 - ✦ show in which subreddit it was created as the link to the subreddit itself. Use the `subreddit_path_helper` in which you can pass the id of the subreddit.
 - ✦ show the number of comments on the post.
 - ✦ checkout how that looks like on reddit: <https://www.reddit.com/>

(it doesn't have to look like that exactly, you will get the design of your reddit clone on one of the next lectures)



SubredditsController

- ✦ Create a controller named `SubredditsController` with an show action:
 - ✦ In the show action you should have a `@subreddit` instance variable (of the Subreddit model) which you can initialize with the data from `params[:id]`
 - ✦ when fetching the Subreddit model use the `find(id)` method instead of the `find_by(id: id)`. The difference is that the `find` method will raise an error if there is no record found.
- ✦ Create a `subreddits/show.html.erb` view
- ✦ Follow the instructions on the next slide for the details on what you should show in the view



subreddits/show view

- ✦ We need a show page for every one of our subreddits. It must contain:
 - ✦ the name of the subreddit
 - ✦ list of all the posts in that subreddit:
 - ✦ a link to the post with it's title using the `subreddit_post_path` helper
 - ✦ the time that has passed since it was created - in words. Checkout this nifty helper. If the post was created 6 hours ago it should state 'created 6 hours ago'.
 - ✦ show the number of comments on the post.
 - ✦ the description of the subreddit on the right
 - ✦ checkout how that looks like on reddit: <https://www.reddit.com/r/ruby>



PostsController

- ✦ Create a controller named `PostsController` with an show action:
 - ✦ In the show action you should have a `@subreddit` instance variable (of the Subreddit model) which you can initialize with the data from `params[:subreddit_id]`.
 - ✦ You will also need to instantiate a `@post` instance variable (of the Post model using `@subreddit.posts.find`) with the data from params[:id]`
- ✦ Create a `posts/show.html.erb` view
- ✦ Follow the instructions on the next slide for the details on what you should show in the view



post/show view

- ✦ We need a show page for every one of our post. It must contain:
 - ✦ the name of the subreddit
 - ✦ the title of the post, the content of the post, comment count
 - ✦ each of the comments:
 - ✦ the content of the comment
 - ✦ the time that has passed since it was created - in words. Checkout this [nifty helper](#). If the post was created 6 hours ago it should state 'created 6 hours ago'.
 - ✦ if there are no comments, show 'No comments yet.'
- ✦ checkout how that looks like on reddit: https://www.reddit.com/r/ruby/comments/1i72ww/on_monday_i_start_a_new_career_as_a_web_developer/



Submit your solutions

Push to the remote repository on Github

- ✦ NOTE: I'm not going in depth how to do commits, please consult with already provided online resources.
- ✦ Once you finish each assignment create a final push to your branch:
- ✦ `git push origin HW3`
- ✦ You don't have to push it only once (i.e. when the whole assignment is over), you can do commits and pushes every time you make any significant progress such as finish an assignments.

Create a pull request

- ✦ Once you have pushed your finished assignments, you will have to create a pull request from your HW3 branch to your master branch
- ✦ Go to Github and position yourself in the HW3 branch
- ✦ If you can't find a way how to go to your branch, the URL should be: <https://github.com/InfinumAcademy/academy-#{name}-#{surname}/tree/HW3>
- ✦ Find the little green button on the left and create a pull request - it should point from branch HW3 to master

Reviewing your pull request

- ✦ if everything is good, we will merge it to the master branch and your work is done (awesome!)
- ✦ if we have some constructive comments about your solution, we will ask you to modify your solution and push again to the HW3 branch again (the pull request will be updated automatically with every commit pushed to the HW3 branch). After you've pushed your changes, just make a comment on Github mentioning us, (e.g. 'Hey @#{reviewer}, I've improved my solution!')