




# Coding Assignment # 5 – Facecook

You are tasked with creating a Rails app that will display a recipe index based on a keyword entered. Following in the footsteps of a little-known company, you decide to name your app `facecook`.


1. Create a rails app called `facecook`
2. Generate `RecipesController` (`recipes_controller.rb`) that will have an `index` action
3. The above-mentioned controller should check if a request parameter `keyword` was passed in.
4. If it was – it should use it, and if not – it should use a default value of “`chocolate`” instead.
5. Create a model, `Recipe` (`recipe.rb`) that will contain a `for` class method that will take a keyword to query and query the Recipe Puppy API. (You should already have the `Recipe` class from the previous assignment. Remember, that unlike the previous assignment – you will not need to require ‘`httparty`’ gem in your code, since loading `HTTParty` gem should be the Bundler’s job.)
6. Create your view (with CSS – use your imagination) that might look something like this:

## Recipes for chocolate

	Title	Ingredients
	<a href="#">Tim and Tracy's Chocolate Cake (Boiled)</a>	baking soda, butter, cocoa powder, eggs, flour, sugar, water
	<a href="#">Chocolate Chocolate Chocolate Butterscotch Chip Cookies Recipe</a>	cocoa powder, baking soda, brown sugar, butterscotch chips, eggs, flour, milk chocolate, salt, semisweet chocolate chips, sugar, vanilla extract
	<a href="#">Chocolate Cake With Fruits Recipe</a>	baking powder, baking soda, chocolate cake, eggs, frosting, vanilla extract, vanilla extract, salt, salt


7. I should be able to click on either the title or the thumbnail and go straight to the actual recipe (out there on the web). Look at `image_tag` helper for defining an `img` tag and use this helper as the first argument to `link_to` helper.
8. Rails automatically escapes HTML in your strings (mainly to avoid XSS attacks [http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting)). Because of this – some of your titles will look wrong.

## Recipes for mocha

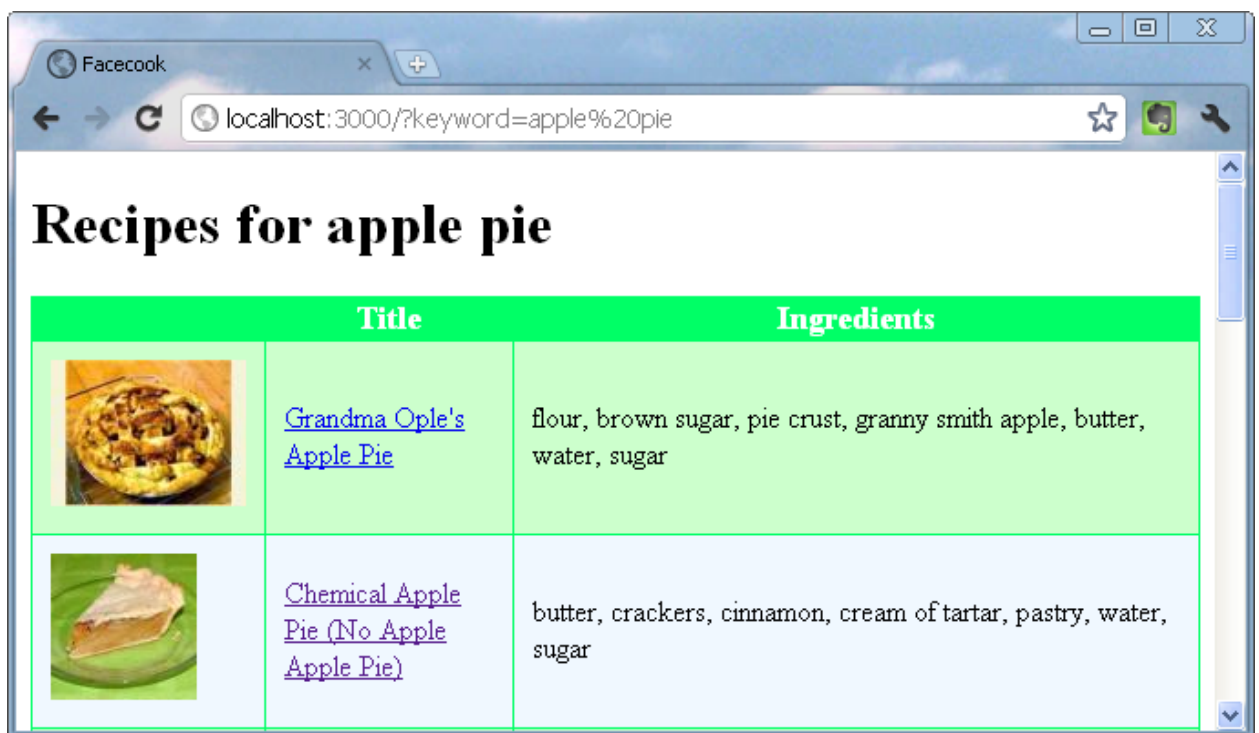
	Title	Ingredients
	<a href="#">Café Mocha Mint or Café Mocha Mandarin Mocha</a>	chocolate syrup, cocoa powder, coffee, milk, mint extract

Rails has a `sanitize` (or `raw`) helper that will help you display HTML characters properly

## Recipes for mocha

	Title	Ingredients
	<a href="#">Café Mocha Mint or Café Mandarin Mocha</a>	chocolate syrup, cocoa powder, coffee, milk, mint extract

9. Instead of having to go to <http://localhost:3000/recipes/index> all the time - you want this page to be the default. This involves 2 simple steps (removing something and creating a route for root) about which you can learn by reading the comments of a newly created `config/routes.rb` file. So, you should be able to go to <http://localhost:3000/?keyword=apple%20pie> and see the following



## Submission Guidelines

1. Commit all the files to your local repo and push your repo out to your private team repo on github creating a remote branch called `assignment5`.

2. For this assignment, it might make the most sense to work on a master branch of a newly created repo and then do `$git push remote_alias_github master:assignment5` to push it out to github.
3. Deploy your homework to heroku (**WARNING:** Heroku only allows pushing to a remote `master` branch) using a URL pattern of `jhu2013-teamX-facecook.herokuapp.com` (where X is your team number).
4. Feel free to include a README.txt with any comments, as well as your deployed homework URL