Ruby on Rails

CSE 190M, Spring 2009
Week 7
Customizing Our Views

• In our blog example, our entries were ordered from oldest to newest

• We can order them newest to oldest if we wish
  • @entries = Entry.all.reverse!

• Maybe we only want to show the 5 most recent entries
  • @entries = Entry.all.reverse!.slice(0,4)
Adding Additional Views

• Last week we showed how Scaffold generates a bunch of standard views and a template layout.

• If we wanted additional actions we:
  – Add the action in our controller
  – Create the corresponding view
  – Update our routes.rb file
    • map.resources :obj_plural => { :action => method }
    • e.g. map.resources :entries => { :preview => get }
Templates with Layouts

- Scaffold creates a Layout that works as a template for our objects
  layouts/entries.html.erb

- This template displays the same thing for each of the actions for that object, and then yields to the corresponding view to display the unique content

- If we want, we can make one single template to be used by all objects by making one named "layouts/application.html.erb"
Partial Layouts

- Sometimes we may want the same bit of code to show up in multiple templates (e.g. navigation bar).
- We can display partial pages in our templates or views, and they are called "partials".
- Partials can be in any of the views or layouts folders.
- By convention, they start with an underscore.
  ```ruby
  views/layouts/_top_nav.html.erb
  ```
- We can render our partial in a template wherever we want it to be displayed.
  ```erb
  <%= render( :partial => "layouts/top_nav" ) %>
  ```
Rails on Webster

- In your public_html directory, make a folder for your Ruby apps
  /home/rctucker/ruby_apps
- Create your Rails app in this folder
  rails --d mysql my_app
- In your public_html folder, make a symlink from the public folder of your app to a folder with the name of your app
  ln -s ruby_apps/my_app/public my_app
- Create/modify .htaccess file in your public_html folder. Add the following line (using your username and application name instead)
  RailsBaseURI /rctucker/my_app
- Add the following line to your environment.rb file
  config.action_controller.relative_url_root = "/rctucker/my_app"
- Restart your app
  touch my_app/tmp/restart.txt
- View app at webster.cs.washington.edu/username/app_name
"Restarting" Your Rails App

• Instead of using WEBrick, we are using a plugin so that Apache can run your Rails apps
• Unlike when you run your application locally, when your application is on Webster, it is always running
• If you make changes to your app that require you to restart it (such as changes to routes.rb or anything else in the config folder) you must restart your application
• Restart your app on Webster by modifying a file named restart.txt in your tmp folder
  touch my_app/tmp/restart.txt
Other Resources

• At the end of the quarter, Webster will disappear, but...

• As students, you have web space on a server called Dante through your UW accounts
  – dante.u.washington.edu

• Dante already has Ruby and Rails installed for you (MySQL is not installed)

• You can find documentation on how to get your Rails applications up and running on Dante here:
  http://www.washington.edu/computing/web/publishing/rails.html