Website Editing

Using Ruby on Rails

Presenter

- Brian Jones
- Web Programmer / Technology Coordinator
- Luther College
- Decorah, Iowa
- Email: brian.jones@luther.edu

Defining Goals

- To make editing a Website as easy as possible for a user.
- To simplify the process of placing, sizing, and captioning images in an article.

Outline

- What is Ruby on Rails?
- Test drive RubyCMS, a working Rails project.
- Look under the hood.

Part I:
What is Ruby on Rails?

- Ruby is a dynamic scripting language written in the C programming language.
- Rails is a Web framework written in Ruby.

Ruby

- Has been claimed to be both elegant and easy to read and write.
- Usually requires less code to complete a given task.
- Once you grasp Ruby it will probably become your scripting language of choice.

Ruby

- Won't necessarily make you a better programmer.
- There is a definite learning curve to come up to speed.
- Some of the syntax may not be as intuitive as claimed.

Rails

- Favors convention over configuration.
- Is Agile.
  - minimal planning
  - rapid prototyping
  - frequent inspection and revision
  - working software is more important than comprehensive documentation

Rails

- Has many built-in features.
- Uses Model-View-Controller (MVC) architecture to organize application programming.
  - Model stores data and enforces business rules.
  - View generates what the user will see.
  - Controller receives requests, interacts with the model, and invokes the view.

Websites using Rails

- Basecamp, a project management tool, out of which Rails was made.
- Yellowpages, a large-scale commercial site.
- Twitter, a social networking site.
- A List Apart, "for people who make Websites."
- RubyCMS.

Part II:
Test Drive RubyCMS

RubyCMS

- Website editor built using Ruby on Rails.
- Editing occurs directly on the page itself not in a separate console.
- Page is the basic building block.
- Focus is on making image uploading, sizing, and captioning as easy as possible.

RubyCMS

- Authentication is both local and through LDAP.
- Full version control.
- Publish and draft modes.
- Ability to limit page read access to given users or groups.
- Database encryption.
- Administrative interface.

RubyCMS Page

- Components
  - Title
  - Main content
  - Navigation
  - Contact Information
  - Breadcrumbs
  - Layouts
  - Keywords

RubyCMS Page

- Components (continued)
  - Valid date range
  - Top images
  - Banners
  - Sidebar images
  - Galleries

Editing

- Navigate to the page.
- Preface the url root with "/edit."
  - *rubycms.org/rubycms/examples* becomes:
  - *rubycms.org/edit/rubycms/examples*
- Login at the prompt.

Editing (as a logged-in user)

- Click the right-most breadcrumb.
- Select the dot to the left of the username.
- Preface the url root with “/edit.

Editing

- Editing is only supported in Firefox.

Creating a new page

- Navigate to an existing page similar to the page being created.
- Remove unneeded images.
- Enter new filename.
- Click save.

Staging mode

- “/staging” appears before the url root.
- Shows the most recent draft, (which could be be published).
- Lists versions.
- Must be a logged in user to use staging mode.

Editing the Title

- Click anywhere on the title.
- Change text.

Editing Breadcrumbs

- Click anywhere on the last breadcrumb.
- Change text.
- Note: the preceding breadcrumbs are still links.
- Note: The last breadcrumb will only trigger edit mode when in normal viewing or staging.

Editing Main Content

- RubyCMS employs TinyMCE editor.
- Behaves like most word processors.
- Supports:
  - Ordered and unordered lists
  - Tables
  - Links
  - Paragraph, headings 1 to 6, and preformatted font sizes
  - Html editing

Images

- RubyCMS utilizes *Imagemagick* and *Attachment fu* to upload and process images.
- Use *Image & Document Manager* to upload, caption, and place images in TinyMCE editor.
- Jpg, gif, and png formats supported.
- May upload a single image or zip file containing many images.

Captioning images

- Use *Image and Document Manager* to caption images.
- Captions will appear below image after page is saved.
- Don't use TinyMCE editor for captioning.

Image placement

- Drag the image from the *Image and Document Manager* to desired spot in TinyMCE editor.
- Resize image to one of three sizes (small, medium or large) by dragging image handles.
- Use html editor to fine-tune image placement.

Documents

- Use *Image & Document Manager* to upload and place documents in TinyMCE editor.
- Pdf, doc, rtf, txt, and xls formats supported.
- Drag the document from the *Image and Document Manager* to desired spot in TinyMCE editor.

Navigation

- Contains links to other pages.
- Click a link to change its name or url.
- Onsite urls have absolute paths like:
  - /examples/3column
  - /contact
- Offsite urls must begin with a protocol (http:// or https://).

Editing Navigation

- Reposition the link by clicking and holding the mouse on the diamond to the left of the link name.
- Delete a link by clicking the X to the right of the link name.
- Import links from another site.
- Revert if mistakes made.

Editing Contact Information

- Invoked by clicking Contact Information link if no contact information is present, or the Contact Information title.
- Add a line by clicking +.
- Delete by picking X.
- Reorder with dot.
- New section with diamond.

Editing Contact information

- Type the full url of another page to use its contact information.
- Check *hide* if you don't want to display contact information.
- Note that it is often necessary to hit the *Enter* key to register changes for a given line.

Layouts

- Determine what is shown on the page and what the page will look like.
- Most of development effort for a new site will consist of writing templates and style sheets for its layout.
- As layouts are added they become automatically available for use.

Keywords

- Used to filter information.
- Available to search engines.
- Suggestions from a site-wide database of categories appear as user types.
- Global list is maintained by users with administrative privileges.

Page Validity

- Pages are always available unless specified by a date/time range.
- Most ways of specifying a date and time are recognized (except European decimal dd/mm/yyyy format).
- To take a page offline, make sure it is published, and the *valid to* entry is before the present.

Top Image

- Feature image for the page.
- Allow for flexibility by starting with an image that will fit on the widest layout (around 1000 pixels).
- Image size does not have to be exact.
- Use css to size and crop image in a particular layout.

Banners

- Advertisement nudging a user to navigate to another page.
- Contains both an image and up to four lines of text.
- Banners can be sorted.

Sidebar Images

- Sidebar images can complement or replace images in an article.
- Can link to another page if url is specified.
- Otherwise a larger image appears when a sidebar image is clicked.
- Sidebar images can be sorted.

Image Galleries

- RailsCMS utilizes the Highslide Javascript Thumbnail viewer for photo galleries.
  - There is a small fee for commercial use of Highslide.
  - Comprehensive API.
  - Actively being developed and enhanced.
  - Can display html content in a separate window.

Image Galleries

- When compiling a digital collection use fewer images of higher quality.
- Use a photo editor such as Picasa or Photoshop to reduce a folder of images to around 1200 by 800 pixels.
- Compress the folder of images using zip.
  - On a PC right click the folder containing the images and choose *Send to -> Compressed (zipped) Folder*.

Image Galleries

- On a Mac using the Finder to make a zip file produces an unrecognizable format.
- Instead open the Terminal program.
- Change to the directory with the folder containing the images.
- Type: zip -r *foldername*.zip *foldername* (where *foldername* is the name of the folder containing the images).

Image Galleries

- Once the images are compressed upload the zip file using the *Image Gallery* module.
  - Please be aware that it may take several minutes for the upload process to complete and that a progress indicator is lacking.
- When images have been uploaded, click an image to caption or toggle hide/show.
- Reorder by dragging the arrow.
- Delete by selecting X.

Forms

- Ruby on Rails was designed to process forms.
- Active Record class uses a simple command set that integrates with popular databases:
  - MySQL
  - Oracle
  - SQLServer
  - PostgresSQL
  - Sybase
  - Sqlite.

Forms

- Form development is just getting started in RubyCMS.
- There is no automated way of generating forms at this time.
- Rails programming is required.
- Users can easily choose a form once it is available.

Forms

- RubyCMS has built-in
  - Database encryption
  - Auto-completion
- Cannot encrypt a column in a table after data already exists.

Administration

- Type /admin after the hostname to invoke administrator mode.
- Defaults to a list of pages.
- Administrator can delete pages.

Administration

- Create and manage local users.
  - Note that groups are only possible in LDAP.
- Set publish, write, and read privileges using regular expressions.
- Enter site-wide navigation.
- Maintain a list of keywords.

Part III:
RubyCMS: Under the Hood

Ruby Gems

- A gem is a packaged Ruby application or library.
- Extends or adds functionality to Ruby.
- Installed Ruby gems can be used by every Rails application.

Ruby Gems

- RubyCMS uses:
  - Capistrano, a utility for deploying rails applications to remote servers.
  - Mongrel is an HTTP server that runs Rails.
  - Passenger is an Apache module that runs Rails.

Ruby Gems

- Rmagick has Ruby binding to ImageMagick.
- Ruby-net-ldap is a Ruby LDAP client library.
- Crypt contains popular encryption algorithms.
- Hpricot, an HTML parser.

Hpricot Gem

```
f = Hpricot(open(url),
  :fixup_tags => true)
n = (f/"a")
n.each { |i| puts i }
```

Rails plugins

- Plugins are self-contained libraries made especially for Rails.
- Unlike gems, plugins are installed directly into a specific Rails application.

Rails plugins

- RubyCMS uses:
  - Attachment fu
  - TinyMCE

Stylesheets

- Blueprint css
  - Provides an easy-to-use grid and sensible typography.

Routing

- Elegantly designed.
- Code samples:

```
map.connect 'edit/*url', :controller
  => 'site', :action => 'edit'
map.connect 'staging/*url',
:controller => 'site', :action =>
  'show'
map.connect
  'rubycms_login/:action/:id',
  :controller => 'login'
```

Hashes in Main Content

- A hash is a key that returns a value.
- Modeled after a Twitter hash.
- RubyCMS keys include:
  - #rubycms_news
  - #rubycms_headlines
- Gather and display child pages of any directory named *news*.
  - Filter using keywords.

Conclusion

- RubyCMS is an example of Ruby on Rails in action.
- Like Rails, RubyCMS follows the *Convention over Configuration* mantra.
- RubyCMS is a work in progress.

Additional information

- rubycms.org Website to launch on Halloween of 2009.
- Open source code will be available for download on Thanksgiving day 2009.
- Follow *rubycms* on Twitter.