Ruby Sinatra
Sinatra Web-Based CMS

By: Eric Strausbaugh

Contents

Chapter 1

Introduction

Topics:

- Features
- Required Software

The Sintara web-based content management system (SWCMS) was created to make list generation and content management easy. This user guide describes how to install and use the SWCMS. It also describes the required software and documents some of the code for the SWCMS.

Features

The content management application has the following features:

- Web-based content management system
- Configurationable forms to enter contents
- WYSIWYG editing window
- RSS-like feeds for list of things (like favorites movies, music, etc.)
- Output xml files in DITA compatible format
- Search capabilities (to be added later)

Required Software

The following software is required to run the content management application:

- jQuery TE - Used as a WYSIWYG HTML editor.
- Ruby
  - Sinatra - Used for quickly creating web applications in Ruby.
  - Datamapper - Used as the database
  - erb - Used for creating a Framework for the pages of web application
- DITA OT - Used to output the contents in different formats (xhtml, PDF, PDF2, etc.). This implementation allows the use of markdown language.

jQuery TE

jQuery TE is a lightweight jQuery plugin, which is very useful WYSIWYG HTML editor. jQuery TE is integrated into the web-based content management system. See the following web site for information about jQuery TE:

- jqueryte.com

Ruby

Ruby is an object-oriented, general-purpose programming language that was developed by Yukihiro Matz Matsumoto, Ruby supports a number of different programming paradigms, including functional, object-oriented, and imperative.

Ruby can be downloaded at the following sites:

- www.ruby-lang.org/en/downloads/
- rubyinstaller.org/

Gems which are basically add-ons or packages for Ruby are available by typing *gem install <gem_package>*. Gems are also available at the following site: rubygems.org/

Sinatra

Sinatra is a lanaguage to quickly create web applications in Ruby. www.sinatrarb.com

Datamapper

DataMapper is an Object Relational Mapper written in Ruby. datamapper.org

For the SWCMS, DataMapper is used as the database to store information about topics, topicmaps, bookmaps, images, and metadata information.

DITA Open Tool Kit

DITA Open Toolkit is an open-source publishing engine for DITA xml contents.

- www.dita-ot.org

You must DITA-OT 3 or later so you can use markdown language.

Table 1: DITA OT Version

Version	Description
dita-ot-3.0.1	This is a development version.

Chapter 2

Installation

To install the SWCMS, use a program, like Winzip or 7-zip, to unzip the following file:

sinatra_md.zip

After creating a directory, go to the Setup on page 11 chapter to configure the SWCMS.

Chapter 3

Setup

Topics:

This chapter shows how to set up the SWCMS.

Setting Variables

For cm_app.rb, views/form1_out.erb, views/form2_out.erb, and views/form3_out.erb, set the following variables to match your set up:

```
@sinatra_dir="/home/eric/sinatra_md1"
@ditaot_dir="/home/eric/dita-ot-3.0.1"
@java_dir="/usr/lib/jdk1.7.0_45"
```

To set categories, add the different categories for the following variable:

```
cat_out="<option value=\"form\">Form</option>
  <option value=\"athlete\">Favorite Atheletes</option>
  <option value=\"movies\">Favorite Movies</option>
  <option value=\"movies\">Favorite Movies</option>
  <option value=\"recent_movies\">Recently Viewed Movies</option>
  <option value=\"topicmap\">Topicmap</option>
  <option value=\"bookmap\">Bookmap</option>
  <option value=\"images\">Images</option>
  <option value=\"cm\">Content Management</option>
  <option value=\"misc\">Misc</option>
  <option value=\"baseball\">Favorite Baseball Players</option>"
```

Displaying Set of Data

The Home page is divided into two columns. The first column shows topics that are tagged promote to front. The second columns shows topics associated with a category in list form.

Add the following code to the *display_front.erb* view to view all of the topics associated with a category (in this case *Favorite Movies*):

```
<div id="menu" style="background-color:WhiteSmoke;width:325px;border:2px
 solid;border-radius:5px;font-size:12px;">
 <h3>Favorite Movies</h3>
<ol>
<%
#@dita = Dita.all(:order => :id.desc)
#id_val = 2
@dita = Dita.all(:category1 => "movies")
@dita.each do |var|
%>
<li><a href="display_entry?id_val=<%= var.id %>"><%= var.title %></a>: <%=
 var.short_desc %></li>
<% end %>
 </ul> </div>
```

Chapter 4

Usage

Topics:

This chapter describes how to use the CMS.

Running SWCMS

To run SWCMS, run the following command:

```
ruby cm_app.rb
```

You can also specify the IP address that Sinatra uses with the following command:

```
ruby cm_app.rb -o 192.168.1.13
```

Creating a Form

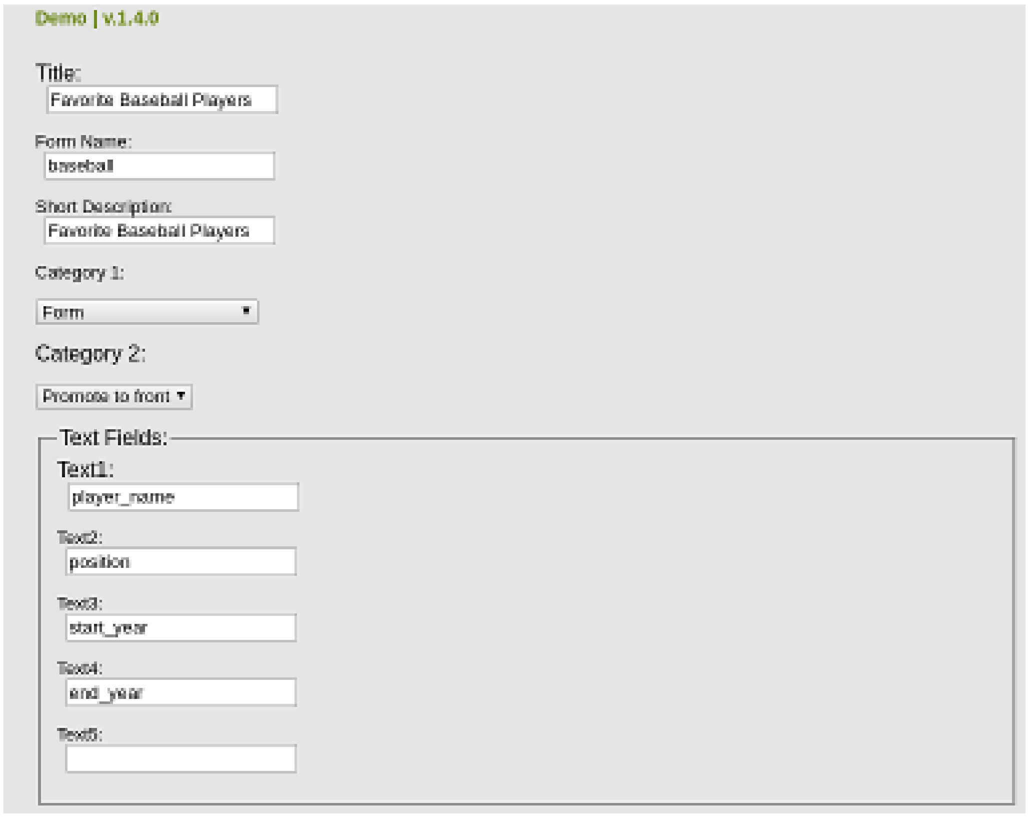
Forms are used to group topics together. These groupings are called categories. For example, you can create a form for your favorite baseball players, which this example will show. There are also the following pre-made forms:

- athlete (Favorite Althelete)
- movies (Favorite Movies)
- recent_movies (Recent Movies)
- topicmap (Topicmaps)
- bookmap (Bookmaps)
- images (Images)
- cm (Content Management - this PDF file was generated using this form)

A template has been setup for creating forms. You just need to enter information to identify the form fields. Perform the following steps to create a form:

1. Select **Forms** > **Enter Forms**.
2. Enter information to create form and identify the fields. See screen capture below for an example.
3. Click submit.
4. To add a form as a category, update the cat_out variable in the cm_app.rb file. See Setting Variables on page 12 for more information. In this example, add`<option value=\"baseball\">Favorite Baseball Players</option>` to cm_app.rb file.

```
cat_out="<option value=\"form\">Form</option>
  <option value=\"athlete\">Favorite Atheletes</option>
  <option value=\"movies\">Favorite Movies</option>
  <option value=\"movies\">Favorite Movies</option>
  <option value=\"recent_movies\">Recently Viewed Movies</option>
  <option value=\"topicmap\">Topicmap</option>
  <option value=\"bookmap\">Bookmap</option>
    <option value=\"images\">Images</option>
  <option value=\"cm\">Content Management</option>
  <option value=\"misc\">Misc</option>
  <option value=\"baseball\">Favorite Baseball Players</option>"
```


Demo | v.1.4.0

Title:
Favorite Baseball Players

Form Name:
baseball

Short Description:
Favorite Baseball Players

Category 1:
Form

Category 2:
Promote to front

Text Fields:

Text1:
player_name

Text2:
position

Text3:
start_year

Text4:
end_year

Text5:

Figure 1: Create Form

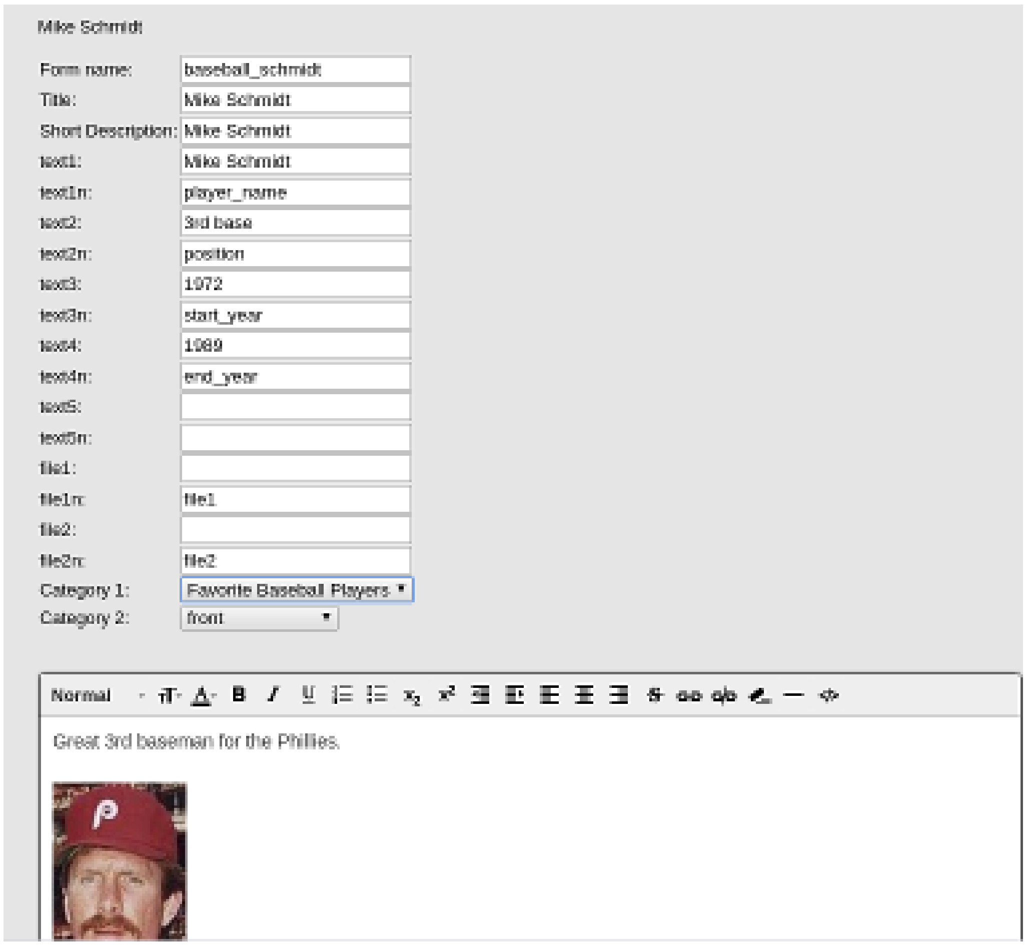
Creating Entries

Perform the following steps to create an entry.

1. Select **Forms** > **Display Forms**.
2. Click on the ID for the form that we created for Favorite Baseball Players.
3. Enter data for form and click submit.
4. Select *Favorite Baseball Players* for Category 1.
5. To show entry on the home page, select *Promote to front* for Category 2.

Note: If you change to *source* mode, you can enter code snippets, like the following to add an image:

```
<img src="images/schmidt.jpg" width="100px">
```



Mike Schmidt

Form name:	baseball_schmidt
Title:	Mike Schmidt
Short Description:	Mike Schmidt
text1:	Mike Schmidt
text1n:	player_name
text2:	3rd base
text2n:	position
text3:	1972
text3n:	start_year
text4:	1989
text4n:	end_year
text5:	
text5n:	
file1:	
file1n:	file1
file2:	
file2n:	file2
Category 1:	Favorite Baseball Players
Category 2:	front

Great 3rd baseman for the Phillies.

Figure 2: Creating Entry

Running DITA OT 3.0

The SWCMS saves the DITA and markdown files in the <DITA_OT_DIR>/samples/topics directory. The bookmap files are stored one level above: <DITA_OT_DIR>/samples,

Run the following command to create a PDF file:

```
../bin/dita --input=bookmap_ditacms.ditamap --format=pdf --output=out1
```

Appendix A

HTML Code Snippets

```
Appendix A. HTML Code Snippets

style
< style> table {style="border-width:1px;border-
color:black}< /style>
image
< img src="images/rollins.jfif" width="100px">
two column table example:
< table border="1"> < tr><td><b>heading 1</b></
td>td><b>heading 2</b></td></tr> < tr><td>cell 1</
td><td>cell 2</td></tr> < /table>
three column table example:
< table border="1"> < tr><td><b>heading 1</b></
td><td><b>heading 2</b></td><td><b>heading 3</
b></td></tr>< tr><td>cell 1</td><td>cell 2</
td><td>cell 3</td></tr> < /table>
four column table example:
< table border="1"> < tr><td><b>heading 1</b></
td><td><b>heading 2</b></td><td><b>heading 2</b></
td><td><b>heading 4</b></td></tr>< tr><td>cell 1</
td><td>cell 2</td><td>cell 3</td><td>cell 4</td></
tr>< /table>
topicmap example:
< map>
<title>DITA work at OASIS</title>
<topicref href="oasis-dita-technical-
committees.dita">
<topicref href="dita_technical_committee.dita"/>
<topicref
 href="dita_adoption_technical_committee.dita" />
</topicref>
<mapref href"oasis-processes.ditamap"/>
</map>
bookmap example:
< bookmap id="taskbook">
< booktitle>
< booklibrary>Retro Tools</booklibrary>
< mainbooktitle>Product tasks</mainbooktitle>
< booktitlealt>Tasks and what they can do</
booktitlealt></booktitle>
< frontmatter>
< booklists><toc/></booklists>
<notices href="taskbook/notices.dita">
<topicref href="taskbook/trademarks.dita"></
topicref>
< /notices>
< /frontmatter>
< chapter href="taskbook/installing.dita">
<topicref href="taskbook/installstorage.dita">
```

```
</topicref>
< /chapter>
< appendix href="taskbook/task_appendix.dita"></
appendix>
< /bookmap>
```

Appendix B

Markdown Code Snippets

```
Appendix B: Markdown Code Snippets

headings
# Topic title

## Nested topic title

# Topic title {#carrot .juice}


# Task {.task}

Context

1.  Command

    Info.


    # Topic title

## Section title {.section}

## Example title {.example}
links
[Markdown](test.md)
[DITA](test.dita)
[HTML](test.html)
[External](http://www.example.com/test.html)
image
An inline ![Alt](test.jpg).

![Alt](test.jpg)

![Alt](test.jpg "Title")
table
| First Header | Second Header | Third Header |
| ------------ | :-----------: | -----------: |
| Content      | *Long Cell*                 ||
| Content      | **Cell**      | Cell         |
[table title]
lists
*   one
*   two
    -   three
    -   four

1.  one
2.  two
```

```
    #.  three
    #.  four


inline
**bold**
*italic*
`code`
~~strikethrough~~
code
```

code 1 code 2

```
topicmap example:
< map>
<title>DITA work at OASIS</title>
<topicref href="oasis-dita-technical-
committees.md" format="markdown">
<topicref href="dita_technical_committee.md"
 format="markdown"/>
<topicref
 href="dita_adoption_technical_committee.md"
 format="markdown" />
</topicref>
<mapref href"oasis-processes.ditamap"/>
</map>
bookmap example:
< bookmap id="taskbook">
< booktitle>
< booklibrary>Retro Tools</booklibrary>
< mainbooktitle>Product tasks</mainbooktitle>
< booktitlealt>Tasks and what they can do</
booktitlealt></booktitle>
< frontmatter>
< booklists><toc/></booklists>
<notices href="taskbook/notices.dita">
<topicref href="taskbook/trademarks.md"
 format="markdown"></topicref>
< /notices>
< /frontmatter>
< chapter href="taskbook/installing.md"
 format="markdown">
<topicref href="taskbook/installstorage.md"
 format="markdown">
</topicref>
< /chapter>
< appendix href="taskbook/task_appendix.md"
 format="markdown"></appendix>
< /bookmap>
```