Ruby Sinatra
Sinatra Web-Based CMS
By: Eric Strausbaugh

Contents

Chapter 1

Introduction

Topics:

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 how to run DITA Open Tool Kit.

This iteration of the CMS now uses json files to store form data instead of datamapper, which has been deprecated.

Note: This PDF file was created using the SWCMS application.

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

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.
  - Json - Used as the format to store information from forms.
  - 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

Json File

A JSON file (.json) is a lightweight, text-based file format used to store data. It is easy to parse with key/value pairs and can be used with web applications. I chose json file for this application because of its ease of use, and I could reuse a lot of the code that interfaced with datamapper.

DITA Open Toolkit

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-3.6.0	DITA Open Tool Kit version 3.6.0

Chapter 2

Installation

To install the SWCMS, go https://github.com/edstraus23/sinatra_json1/

Click on **Code > Download Zip**.

or use a command like the following to clone the repository:

```
gh repo clone edstraus23/sinatra_json1
```

Unzipping the file will create a sinatra_json1 directory. The sinatra_json1 directory contains the following files at the top-level:

- cm_app.rb - The main sintara ruby application. Starts up the sintara web application and accesses the json file.

There are also three directories at the top-level:

- h2d - This directory contains file for an XSL transform that converts html files into dita.
- public - This directory allows you to post HTML files on the Sinatra web application.
- views - This folder contains the ERB views. See http://sinatrarb.com/intro.html for more details.

After creating the Sinatra_json1 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, set the following variables in the settings.json file to match your set up:

```
[
  {
    "sinatra_dir": "/home/eric/dev/sinatra_json1",
    "ditaot_dir": "/home/eric/dita-ot-3.6/docsrc",
    "java_dir": "/usr/lib/jvm/java-1.21.0-openjdk-amd64",
    "cat_out": "<option value='form'>Form</option>\n <option
 value='topicmap'>Topicmap</option>\n <option value='bookmap'>Bookmap</
option>\n <option value='images'>Images</option> \n<option
 value='cm'>Content Management</option>\n <option value='misc'>Misc</
option>\n <option value='athlete'>Favorite Atheletes</option>
\n <option value='movies'>Favorite Movies</option>\n <option
 value='recent_movies'>Recently Viewed Movies</option>\n <option
 value='baseball'>Favorite Baseball Players</option>",
    "cat_out2": "<option value='inactive'>Inactive</option>\n <option
 value='front'>Promote to front</option>",
    "cat_out3": "<option value='title'>Title</option>\n <option
 value='content'>Content</option>\n <option value='cat_1'>Category 1</
option>\n <option value='cat_2'>Category 2</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.

Figure 1: Home Page

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>
<%
```

```
@data.each do |var|
 if var['category1'] =~/movies/
%>
<li><a href="htm/<%= var['form_name'] %>.htm"><%= var['title'] %></a>: <%=
 var['short_desc'] %></li>
<%
 end
end
%>
 </ul> </div>
```

Chapter 4

Usage

Topics:

This chapter describes how to use the CMS application. The SWCMS has generic forms that you can use to create different categories. Examples of these categories are the following:

- Favorite Athletes
- Favorite Movies
- Recently Viewed Movies
- Topicmap
- Bookmap
- Images
- Content Management
- Misc
- Favorite Baseball Players

These categories can be used to create lists:

Favorite Movies

1. Step Brothers: Great silly movie
2. Old School: This a great movies

Once a form is created, it is easy to create topics related to a category. You can also view all of the topics related to a category, and add the topics to a topicmap or a bookmap.

With the SWCMS, you can use either HTML coding or markdown language to create topics. The SWCMS converts HTML coding into DITA topics. Markdown language is used directly to create PDF files.

Running SWCMS

To begin the SWCMS application, 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 settings.json file.

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

Figure 2: Create Form

Creating Entries - HTML Coding

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">
```

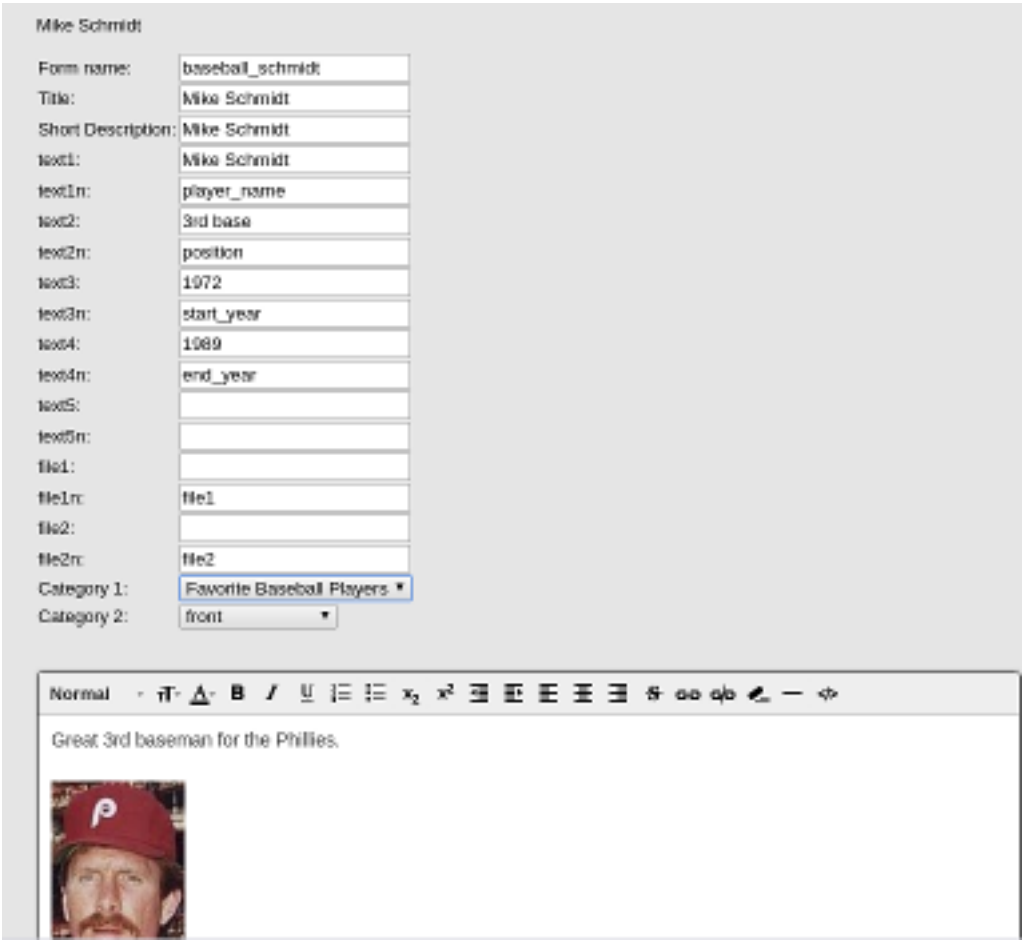

Figure 3: Creating Entry

Creating a Bookmap/Topicmap - HTML Coding

1. Create an entry using the same steps as outlined in the Creating Entries using HTML coding.
2. Select topicmap or bookmap for Category 1.

3, Enter the bookmap or topicmap coding as shown in the sample code:

topicmap example:

```
(map)
(title)DITA Apps(/title)

(topicref href="topics/misc_dita_app.xml")
(topicref href="topics/athlete_chase.xml"/)
(topicref href="topics/misc_h2d.xml" /)
(topicref href="topics/misc_run_ant.xml" /)
(/topicref)

(/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)
```

Note: Parentheses are used instead of < and > symbols.

1. Click Submit.

Creating a Bookmap/TopicMap - Markdown

1. Create an entry using the same steps as outlined in the Creating Entries using HTML coding.
2. Select topicmap or bookmap for Category 1.

3, Enter the bookmap or topicmap coding as shown in the sample code:

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>
```

1. Click submit.

Running DITA OT x.x.x

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=cm_bookmap.ditamap --format=pdf --output=out
```

Run the following command to create a HTML file:

```
../bin/dita --input=topic_map_cms.ditamap --format=xhtml --output=out
```

Note: This example assumes that you have already created a bookmap.

Run the following file to create an HTML file:

```
../bin/dita --input=topicmap_ditacms.ditamap --format=xhtml --output=out1
```

The following are possible formats to use with DITA OT x.x.x:

- dita
- eclipsehelp
- html5
- htmlhelp
- markdown, markdown_gitbook, and markdown_github
- pdf
- xhtml

Refer to External for further details.

Using Search Query

You can perform a search by clicking on the Search button in the upper right-hand corner of the web page. The Search allows you to search by the following:

- Title
- Content
- Category 1
- Category 2

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

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>
```