**References:**

<http://railscasts.com/episodes/272-markdown-with-redcarpet>

**Gemfile**

gem 'redcarpet'

gem 'albino'

gem 'nokogiri'

**Application.css**

//= require pygments

**Application\_helper.rb**

def markdown

options = { autolink: true, space\_after\_headers: true, hard\_wrap: true, filter\_html: true, no\_intraemphasis: true, fenced\_code: true, gh\_blockcode: true }

Redcarpet::Markdown.new(Redcarpet::Render::HTML, options)

end

def syntax\_highlighter(html)

doc = Nokogiri::HTML(html)

doc.search("//pre[@lang]").each do |pre|

pre.replace Albino.colorize(pre.text.rstrip, pre[:lang])

end

doc.to\_s

end

**app/assets/stylesheets/pygments.css.scss**

For Syntax\_hightlighter add pygments.css.scss file in stylesheets:

.hll { background-color: #ffffcc }

.c { color: #408080; font-style: italic } /\* Comment \*/

.err { border: 1px solid #FF0000 } /\* Error \*/

.k { color: #008000; font-weight: bold } /\* Keyword \*/

.o { color: #666666 } /\* Operator \*/

.cm { color: #408080; font-style: italic } /\* Comment.Multiline \*/

.cp { color: #BC7A00 } /\* Comment.Preproc \*/

.c1 { color: #408080; font-style: italic } /\* Comment.Single \*/

.cs { color: #408080; font-style: italic } /\* Comment.Special \*/

.gd { color: #A00000 } /\* Generic.Deleted \*/

.ge { font-style: italic } /\* Generic.Emph \*/

.gr { color: #FF0000 } /\* Generic.Error \*/

.gh { color: #000080; font-weight: bold } /\* Generic.Heading \*/

.gi { color: #00A000 } /\* Generic.Inserted \*/

.go { color: #808080 } /\* Generic.Output \*/

.gp { color: #000080; font-weight: bold } /\* Generic.Prompt \*/

.gs { font-weight: bold } /\* Generic.Strong \*/

.gu { color: #800080; font-weight: bold } /\* Generic.Subheading \*/

.gt { color: #0040D0 } /\* Generic.Traceback \*/

.kc { color: #008000; font-weight: bold } /\* Keyword.Constant \*/

.kd { color: #008000; font-weight: bold } /\* Keyword.Declaration \*/

.kn { color: #008000; font-weight: bold } /\* Keyword.Namespace \*/

.kp { color: #008000 } /\* Keyword.Pseudo \*/

.kr { color: #008000; font-weight: bold } /\* Keyword.Reserved \*/

.kt { color: #B00040 } /\* Keyword.Type \*/

.m { color: #666666 } /\* Literal.Number \*/

.s { color: #BA2121 } /\* Literal.String \*/

.na { color: #7D9029 } /\* Name.Attribute \*/

.nb { color: #008000 } /\* Name.Builtin \*/

.nc { color: #0000FF; font-weight: bold } /\* Name.Class \*/

.no { color: #880000 } /\* Name.Constant \*/

.nd { color: #AA22FF } /\* Name.Decorator \*/

.ni { color: #999999; font-weight: bold } /\* Name.Entity \*/

.ne { color: #D2413A; font-weight: bold } /\* Name.Exception \*/

.nf { color: #0000FF } /\* Name.Function \*/

.nl { color: #A0A000 } /\* Name.Label \*/

.nn { color: #0000FF; font-weight: bold } /\* Name.Namespace \*/

.nt { color: #008000; font-weight: bold } /\* Name.Tag \*/

.nv { color: #19177C } /\* Name.Variable \*/

.ow { color: #AA22FF; font-weight: bold } /\* Operator.Word \*/

.w { color: #bbbbbb } /\* Text.Whitespace \*/

.mf { color: #666666 } /\* Literal.Number.Float \*/

.mh { color: #666666 } /\* Literal.Number.Hex \*/

.mi { color: #666666 } /\* Literal.Number.Integer \*/

.mo { color: #666666 } /\* Literal.Number.Oct \*/

.sb { color: #BA2121 } /\* Literal.String.Backtick \*/

.sc { color: #BA2121 } /\* Literal.String.Char \*/

.sd { color: #BA2121; font-style: italic } /\* Literal.String.Doc \*/

.s2 { color: #BA2121 } /\* Literal.String.Double \*/

.se { color: #BB6622; font-weight: bold } /\* Literal.String.Escape \*/

.sh { color: #BA2121 } /\* Literal.String.Heredoc \*/

.si { color: #BB6688; font-weight: bold } /\* Literal.String.Interpol \*/

.sx { color: #008000 } /\* Literal.String.Other \*/

.sr { color: #BB6688 } /\* Literal.String.Regex \*/

.s1 { color: #BA2121 } /\* Literal.String.Single \*/

.ss { color: #19177C } /\* Literal.String.Symbol \*/

.bp { color: #008000 } /\* Name.Builtin.Pseudo \*/

.vc { color: #19177C } /\* Name.Variable.Class \*/

.vg { color: #19177C } /\* Name.Variable.Global \*/

.vi { color: #19177C } /\* Name.Variable.Instance \*/

.il { color: #666666 } /\* Literal.Number.Integer.Long \*/

need to add syntax\_highlighter method and markdown method where ever the markdown support required. Eg:

**raw syntax\_highlighter(markdown.render(**@comment.content)**)**

**Markdown Editor (Markitup)**

**Gemfile:**

gem ‘markitup-rails’

**application.js**

//= require markitup

//= require markitup-markdown

**projects.js.coffee**

$(document).ready ->

# mIu nameSpace to avoid conflict.

miu = markdownTitle: (markItUp, char) ->

heading = ""

n = $.trim(markItUp.selection or markItUp.placeHolder).length

i = 0

while i < n

heading += char

i++

"\n" + heading + "\n"

myMarkdownSettings =

nameSpace: "markdown" # Useful to prevent multi-instances CSS conflict

previewParserPath: "/static\_pages/preview"

onShiftEnter:

keepDefault: false

openWith: "\n\n"

markupSet: [

name: "First Level Heading"

key: "1"

placeHolder: "Your title here..."

closeWith: (markItUp) ->

miu.markdownTitle markItUp, "="

,

name: "Second Level Heading"

key: "2"

placeHolder: "Your title here..."

closeWith: (markItUp) ->

miu.markdownTitle markItUp, "-"

,

name: "Heading 3"

key: "3"

openWith: "### "

placeHolder: "Your title here..."

,

name: "Heading 4"

key: "4"

openWith: "#### "

placeHolder: "Your title here..."

,

name: "Heading 5"

key: "5"

openWith: "##### "

placeHolder: "Your title here..."

,

name: "Heading 6"

key: "6"

openWith: "###### "

placeHolder: "Your title here..."

,

separator: "---------------"

,

name: "Bold"

key: "B"

openWith: "\*\*"

closeWith: "\*\*"

,

name: "Italic"

key: "I"

openWith: "\_"

closeWith: "\_"

,

separator: "---------------"

,

name: "Bulleted List"

openWith: "- "

,

name: "Numeric List"

openWith: (markItUp) ->

markItUp.line + ". "

,

separator: "---------------"

,

name: "Picture"

key: "P"

replaceWith: "![[![Alternative text]!]]([![Url:!:http://]!] \"[![Title]!]\")"

,

name: "Link"

key: "L"

openWith: "["

closeWith: "]([![Url:!:http://]!] \"[![Title]!]\")"

placeHolder: "Your text to link here..."

,

separator: "---------------"

,

name: "Quotes"

openWith: "> "

,

name: "Code Block / Code"

openWith: "(!(\t|!|`)!)"

closeWith: "(!(`)!)"

,

separator: "---------------"

,

name: "Preview"

call: "preview"

className: "preview"

]

$("textarea.markdown").markItUp myMarkdownSettings

$("a[title=\"Preview\"]").trigger "mouseup"

**application.css.scss**

//= require markitup

//= require markitup-markdown

**bare.css.scss**

@import "compass";

html, body {

font: 13px/18px 'Open Sans';

height: 100%;

}

.sidebar {

padding: 15px;

background: #EEE;

@include border-radius(5px);

}

.slash {

color: #677E30;

}

.shadow {

@include box-shadow(skyblue 2px 2px 10px);

}

**markdown\_preview.css.scss**

//= require bootstrap\_and\_overrides

//= require bare

//= require font

//= require pygments

**static\_pages\_controller.rb**

def preview

render partial: 'preview'

end

**app/views/static\_pages/\_preview.html.haml**

= stylesheet\_link\_tag "markdown\_preview", :media => "all"

=raw syntax\_highlighter(markdown.render(params[:data]))

**app/views/projects/\_form.html.haml**

= f.input :description, :input\_html => { class: **'markdown'** }

**routes.rb**

post 'static\_pages/preview', controller: :static\_pages, action: :preview