





DOCUMENTATION (Source)

=====

Project: jAsciiDoctor

Folder: G:/_PROJECTS/Ruby/jAsciiDoctor/src

Status: 19.03.2014

jAsciiDoctor.jrb 19.03.2014 11:19 # jAsciiDoctor
style.css 12.03.2014 08:55 /*controlStyle.css */

```
1: # jAsciiDoctor
2: #!/usr/bin/env jruby
3: #encoding: utf-8
4: #
5: include Java
6: import javax.swing.JOptionPane
7: require 'jrubyfx'
8: require 'asciidoctor'
9: require 'erb'
10:
11: fxml_root File.dirname(__FILE__)
12: PROG = Dir.pwd
13: AUTHOR = "Andreas Weber"
14: MAIL = "ruby@gruena.net"
15: VERSION = "1.0"
16: TITLE = "jAsciiDoctor" + " #{VERSION}"
17:
18: class GUI < JRubyFX::Application
19:
20:     def start(stage)
21:         stage.title = TITLE
22:         stage.fxml GUIController
23:         stage.show()
24:     end
25: end
26:
27: class GUIController
28:     include JRubyFX::Controller
29:     fxml "jAsciiDoctor.fxml"
30:
31:     def about
32:         msg = "#{TITLE}\nby #{AUTHOR}\n<#{MAIL}>"
33:         JOptionPane.showMessageDialog(nil, msg, "About jAsciiDoctor", JOptionPane::INFORMATION_MESSAGE)
34:     end
35:
36:     def help
37:         html = 'help/help.html'
38:         system("start #{html}")
39:     end
40:
41:     def newFile
42:         @editor.text = "= Title\nauthor <email>\nrevision, date"
43:         @file = saveFileAs()
44:         @txtFile.text = @file.to_s
45:     end
46:
```

N)

```
47:      def openFile(f)
48:          @file = f
49:          @txtFile.text = @file.to_s
50:          File.open(@file.to_s, "r:utf-8") { |f| @text = f.read }
51:          @editor.text = @text
52:          @mnClean.disable = false
53:          @mnXML.disable = false
54:          @mnPDF.disable = false
55:      end
56:
57:      def openFileDialog
58:          chooser = FileChooser.new()
59:          chooser.setTitle("Select AsciiDoc file")
60:          chooser.add_extension_filter("ADOC", "*.adoc")
61:          file = chooser.showOpenDialog(nil)
62:          openFile(file)
63:      end
64:
65:      def saveFile
66:          File.open(@file.to_s, "w") { |f| f.puts @editor.text}
67:      end
68:
69:      def saveFileAs
70:          chooser = FileChooser.new()
71:          chooser.setTitle("Save file as (*.adoc)")
72:          chooser.add_extension_filter("ADOC", "*.adoc")
73:          file = chooser.showSaveDialog(nil)
74:          file = file.to_s + ".adoc" if File.extname(file.to_s) == ""
75:          File.open(file.to_s, "w+") { |f| f.puts @editor.text}
76:          return file
77:      end
78:
79:      def closeFile
80:          ret = JOptionPane.showConfirmDialog(nil, "Do you want to save ?", "jAsciiDoctor", JOptionPane::YES_NO_OPTION
81:
82:          saveFile() if ret == 0
83:          @txtFile.text = "Open AsciiDoc file ..."
84:          @editor.text = ""
85:          @html = ""
86:          @preview.getEngine.loadContent(@html)
87:      end
88:
89:      def preview
90:          saveFile()
91:          asd = @file.to_s.gsub(/\\/, '/')
          html = asd.gsub(/.adoc/, '.html')
```

```
92:         AsciiDoctor.render_file(asd,
93:             :in_place => true,
94:             :icons => true,
95:             :backend => 'html5'
96:         )
97:         File.open(html, "r:utf-8") { |f| @html = f.read }
98:         @preview.getEngine.loadContent(@html)
99:     end
100:
101:     def clean
102:         asd = @file.to_s.gsub(/\\/, '/')
103:         File.delete(asd.gsub(/.adoc/, '.html')) if File.exists?(asd.gsub(/.adoc/, '.html'))
104:         File.delete(asd.gsub(/.adoc/, '.xml')) if File.exists?(asd.gsub(/.adoc/, '.xml'))
105:         File.delete(asd.gsub(/.adoc/, '.pdf')) if File.exists?(asd.gsub(/.adoc/, '.pdf'))
106:     end
107:
108:     def fullview
109:         asd = @file.to_s.gsub(/\\/, '/')
110:         html = asd.gsub(/.adoc/, '.html')
111:         system("start #{html}")
112:     end
113:
114:     def exportXML
115:         saveFile()
116:         asd = @file.to_s.gsub(/\\/, '/')
117:         xml = asd.gsub(/.adoc/, '.xml')
118:         AsciiDoctor.render_file(asd,
119:             :in_place => true,
120:             :icons => true,
121:             :backend => 'docbook5'
122:         )
123:         JOptionPane.showMessageDialog(nil, "Docbook export finished => #{xml}",
124:             "jAsciiDoctor", JOptionPane::INFORMATION_MESSAGE)
125:     end
126:
127:     def exportPDF
128:         exportXML()
129:         asd = @file.to_s.gsub(/\\/, '/')
130:         xml = asd.gsub(/.adoc/, '.xml')
131:         #~ xsl = "g:/Tools/OpenWriting/docbook_xsl/fo/docbook.xsl"
132:         xsl = File.dirname(__FILE__) + "/xsl/fo/docbook.xsl"
133:         pdf = asd.gsub(/.adoc/, '.pdf')
134:         cmd = "fop -xml #{xml} -xsl #{xsl} -pdf #{pdf}"
135:         system(cmd)
136:         JOptionPane.showMessageDialog(nil, "PDF export finished => #{pdf}",
137:             "jAsciiDoctor", JOptionPane::INFORMATION_MESSAGE)
```

```
138:                system("start #{pdf}")
139:            end
140: end
141:
142: GUI.launch
```

```
1: /*controlStyle.css */
2: .root{
3: -fx-background-color: #67644e;
4: -fx-background-radius: 10;
5: -fx-padding: 10;
6: }
7: .menu-bar{
8: -fx-background-color: orange;
9: -fx-padding: 5;
10: }
11: .menu{
12: -fx-padding: 2;
13: }
14: .menu-item{
15: -fx-font: 14pt "Arial Black";
16: -fx-text-fill: white;
17: }
18: .label{
19: -fx-font: 14pt "Arial Black";
20: -fx-text-fill: white;
21: -fx-background-color: tomato;
22: -fx-padding: 5;
23: }
24: .file{
25: -fx-font: 12pt "Arial";
26: -fx-text-fill: black;
27: -fx-background-color: tomato;
28: }
29: .button{
30: -fx-font: 14pt "Arial Black";
31: -fx-text-fill: white;
32: -fx-background-color: tomato;
33: -fx-background-radius: 30;
34: -fx-background-insets: 0;
35: }
36: .text-area{
37: -fx-color: orange;
38: -fx-background-color: lightyellow;
39: }
40: .text-input{
41: -fx-font: 14pt "Arial";
42: -fx-text-fill: navy;
43: }
44: .web-view{
45: -fx-background-color: wheat;
46: }
```