AsciiDoc & doc-as-code Best Practices

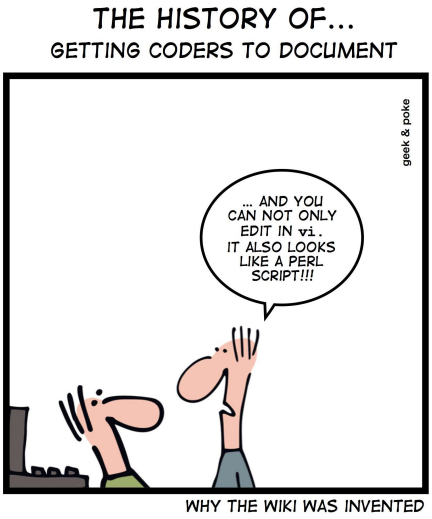
Version 1.18-SNAPSHOT

2019-01-06

Table of Contents

Table 1. History

Date	Author	Detail
2019-01-06	bcouetil	updated reveal theme + reveal my asciidoc + css c3js fix
2018-12-18	bcouetil	Moved Gerrit plugin configuration from Jenkins to Gerrit page + Enhanced HTML CSS + Cropped some images + Improved PlantUML skin
2018-12-11	bcouetil	- Added reveal plugins and background - Fixed reveal css following change in structure in asciidoc-reveal master (from previous version : 1.1.3) - Implemented Zenika layout in HTML and PDF - Reported back reveal-js enhancements
2018-11-28	bcouetil	- Updated reveal css for all syntax elements, for both light and dark themes : asciidoc-syntax-quick-reference is now nicely output - Refactored light and dark css, extracting common items into reveal-zenika.css - Added syntax quick reference in all format generated and linked in asciidoc page
2018-11-13	bcouetil	- Updated sample project with Reveal.js generation - Duplicated Reveal.js execution to have multiple styles - Compromised layout between 4/3 and 16/9 - Minor changes in Reveal.js css - Added some web comics
2018-11-02	bcouetil	- reorganized asciidoc best practices - added some Geek & Poke images - reveal.js themes Zenika dark + light + parallax - updated asciidoc/reveal versions in pom.xml
2018-09-19	bcouetil	- Sample asciidoctor maven project published on Github - Github & LinkedIn links - Sample project tree - new images + resizing and positioning
2018-08-29	bcouetil	Asciidoc HTML look & feel changes
2018-08-24	bcouetil	Icones added for download + favicon added for webpage
2018-08-23	bcouetil	Initial commit



1. Demo

And also an advanced everything-as-code Reveal.js demo : Reveal my Asciidoc.

2. Editor

To help you to write your documentation, use AsciidocFX. It will allows you to visualize your documentation without having to generate the PDF. You can consider Visual Studio Code too, it has a nice plugin but with less features.

Download and install :

- AsciidocFX, to create and edit your documents
- Graphviz, to add the extension for your AsciidocFX installation

Rules to produce AsciiDoc files :

- File names are in kebab-case, except the type at the beginning

- Use **kbd:[myShortcut]** for keyboard shortcuts
- Use **\btn[myButton]** for buttons
- High level title should be on new pages (use **<<<**)

Diagrams are generated using PlantUML. Here are some examples with Zenika's style.

To show a in progress diagram to colleagues, you can generate a web link using PlantUml generator. There is also PlantText.

2.1. Graphviz

Graphviz has to be installed to use PlantUML diagrams.

On Windows

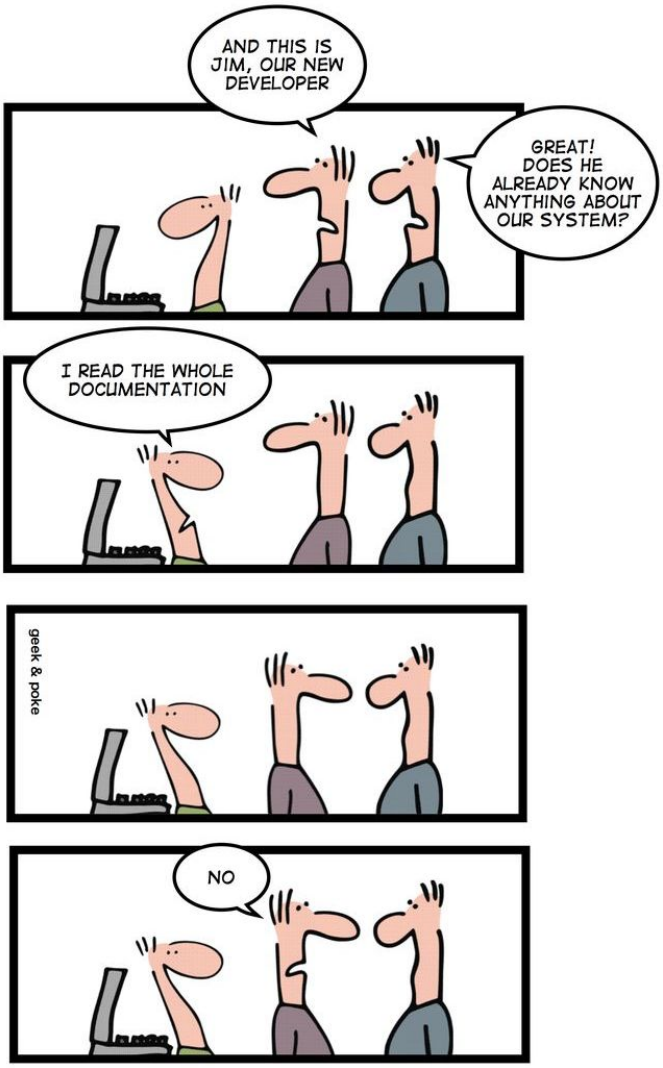
- Download and install : http://www.graphviz.org/download/
- add bin directory to windows path
  - C:\Program Files (x86)\Graphviz\bin

3. Inline icons

- See https://asciidoctor.org/docs/user-manual/#icons
- Font Awesome 4.6.3 Class Explorer : https://lab.artlung.com/font-awesome-sample/
  - no need for :icons: font to use font awesome-font for icons

4. Custom code highlighter

- To use a custom code highlighter, see the maven configuration in next paragraphs
- Default highlight.js processor is too basic, instructions here to change
- Highlight.js demo with all languages and all styles



5. init and closure

Here are the sources of init.adoc and closure.adoc included in various sources in this documentation.

They are here for generic configuration and data accross files.

6. Generation using Maven

You will find a sample Maven project generating both PDF and HTML, with current doc's layout, on my Github.

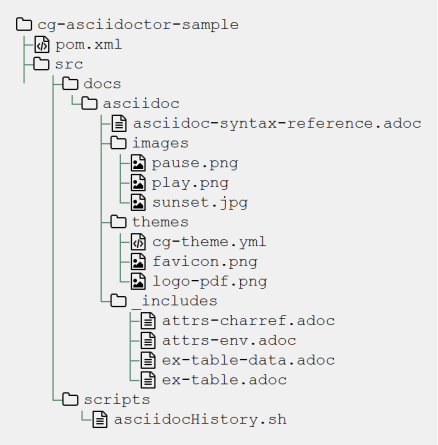


Figure 1. File system tree

AsciiDoctor maven plugin configuration

```
<!-- to generate asciidoc PDF + HTML documents -->
<!-- single usage : mvn generate-resources -Dadoc.skip=false [- -non-recursive] -->
<plugin>
    <groupId>org.asciidoctor</groupId>
    <artifactId>asciidoctor-maven-plugin</artifactId>
    <!-- to increment versions, see pom.xml examples on Asciidoctor Github -->
    <!-- 1.5.7 and 1.5.7.1 show too many warnings on PDF -->
    <version>1.5.6</version>
    <!-- if we don't want to execute in modules (this would work though) -->
    <inherited>false</inherited>
    <dependencies>
        <dependency>
            <groupId>org.asciidoctor</groupId>
            <artifactId>asciidoctorj-pdf</artifactId>
            <version>1.5.0-alpha.16</version>
        </dependency>
        <dependency>
            <groupId>org.asciidoctor</groupId>
            <artifactId>asciidoctorj-diagram</artifactId>
            <version>1.5.9</version>
        </dependency>
        <!-- comment to use the default version -->
        <dependency>
            <groupId>org.jruby</groupId>
            <artifactId>jruby-complete</artifactId>
            <version>9.1.17.0</version>
        </dependency>
        <!-- comment to use the default version -->
        <!-- <dependency>
            <groupId>org.asciidoctor</groupId>
            <artifactId>asciidoctorj</artifactId>
            <version>1.5.7</version>
        </dependency> -->
    </dependencies>
    <configuration>
        <skip>adoc.skip</skip>
        <sourceDirectory>src/docs/asciidoc</sourceDirectory>
        <requires>
            <require>asciidoctor-diagram</require>
            <require>./src/docs/asciidoc/lib/chart-block-macro.rb</require>
        </requires>
        <attributes>
            <!-- for Font Awesome icons -->
            <icons>font</icons>
            <idseparator>-</idseparator>
            <idprefix />
            <!-- custom -->
            <source-dir>../../main/java</source-dir>
            <test-dir>../../test/java</test-dir>
```

```
            <project-version>${project.version}</project-version>
            <root-project-dir>${user.dir}</root-project-dir>
            <history-dir>${project.build.directory}/generated-docs/history</history-dir>
            <project-images-dir>${project.basedir}/src/main/resources/images</project-images-dir>
            <!-- 'plantuml-config' for plantuml, 'salt-config' for salt, 'plantumlconfig' for both -->
            <plantuml-config>themes/plantuml.cfg</plantuml-config>
        </attributes>
    </configuration>
    <executions>
        <execution>
            <id>asciidoc-to-html</id>
            <phase>generate-resources</phase>
            <goals>
                <goal>process-asciidoc</goal>
            </goals>
            <configuration>
                <backend>html5</backend>
                <!-- coderay, highlight.js, prettify -->
                <!-- coderay/prettify have only a bright theme -->
                <!-- prettify is nice but comments are red -->
                <!-- pygments/rouge is not integrated yet : https://github.com/asciidoctor/asciidoctor/issues/1040 -->
                <sourceHighlighter>highlight.js</sourceHighlighter>
                <attributes>
                    <!-- local version to go beyond basic languages (for ex groovy) -->
                    <highlightjsdir>highlight</highlightjsdir>
                    <!-- explore here https://highlightjs.org/static/demo/ -->
                    <highlightjs-theme>gruvbox-dark</highlightjs-theme>
                    <imagesdir>./images</imagesdir>
                    <toc>left</toc>
                    <toclevels>5</toclevels>
                    <sectanchors>true</sectanchors>
                    <docinfo1>true</docinfo1>
                    <favicon>themes/favicon.png</favicon>
                    <linkcss />
                    <stylesheet>html-zenika.css</stylesheet>
                    <stylesdir>themes/css</stylesdir>
                </attributes>
            </configuration>
        </execution>
        <execution>
            <id>asciidoc-to-revealjs</id>
            <!-- after html because it's also an html -->
            <phase>generate-resources</phase>
            <goals>
                <goal>process-asciidoc</goal>
            </goals>
            <configuration>
                <backend>revealjs</backend>
                <sourceDocumentName>PRES-asciidoc.adoc</sourceDocumentName>
                <outputFile>${project.slides.directory}/PRES-asciidoc.html</outputFile>
                <templateDir>${project.build.directory}/asciidoctor-reveal.js-${asciidoctor-
revealjs.version}/templates</templateDir>
                <attributes>
                    <source-highlighter>highlightjs</source-highlighter>
                    <highlightjs-theme>highlight/styles/gruvbox-dark.min.css</highlightjs-theme>
                    <revealjsdir>reveal.js-${revealjs.version}</revealjsdir>
                    <!-- default closest to Zenika's graphic chart : -->
                    <!-- <revealjs_theme>blood</revealjs_theme> -->
                    <revealjs_customtheme>themes/reveal-zenika-dark.css</revealjs_customtheme>
                    <!-- none, fade, slide, convex, concave, zoom -->
                    <!-- dynamic : PRES-asciidoc.html?transition=convex -->
                    <revealjs_transition>slide</revealjs_transition>
                    <revealjs_slideNumber>true</revealjs_slideNumber>
                    <!-- does not work T_T only solution is put a favicon.ico in the root folder of the website -->
                    <!-- <favicon>themes/favicon.png</favicon> -->
                    <revealjs_width>1100</revealjs_width>
                    <revealjs_height>700</revealjs_height>
                    <revealjs_plugins>revealjs-plugins/revealjs-plugins.js</revealjs_plugins>
                    <revealjs_plugins_configuration>revealjs-plugins/revealjs-plugins-
conf.js</revealjs_plugins_configuration>
```

```
                </attributes>
            </configuration>
        </execution>
        <execution>
            <id>asciidoc-to-pdf</id>
            <phase>generate-resources</phase>
            <goals>
                <goal>process-asciidoc</goal>
            </goals>
            <configuration>
                <backend>pdf</backend>
                <sourceHighlighter>rouge</sourceHighlighter>
                <attributes>
                    <imagesdir>${project.build.directory}/generated-docs/images</imagesdir>
                    <toc />
                    <toclevels>3</toclevels>
                    <pagenums />
                    <pdf-style>${user.dir}/src/docs/asciidoc/themes/pdf-theme.yml</pdf-style>
                    <pdf-fontsdir>${user.dir}/src/docs/asciidoc/themes/fonts/pdf</pdf-fontsdir>
                </attributes>
            </configuration>
        </execution>
    </executions>
</plugin>
```

To generate the PDF locally, use the maven command :

```
mvn generate-resources --non-recursive
```

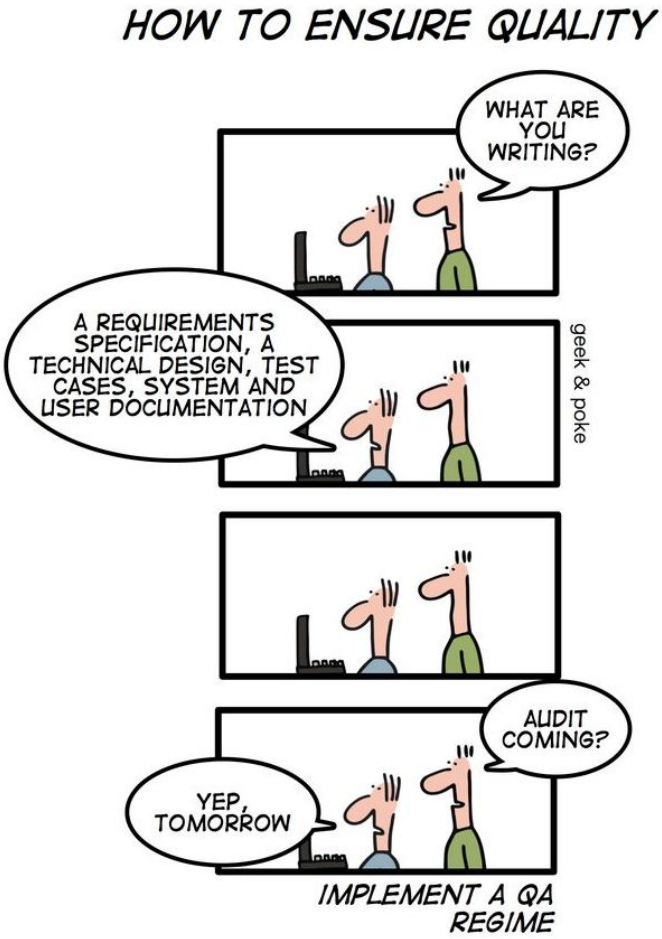
7. .doc to .adoc

If the initial documentation is of Microsoft Word, you have to first migrate it to AsciiDoc file.

Download and install pandoc.

To migrate a document, use it in command line

```
pandoc --from=docx --to=asciidoc --wrap=none --atx-headers --normalize --extract-media=images monDoc.docx > monDoc.adoc
```



8. Publish HTML & PDF to github

8.1. Initialize Github space

```
mkdir docs
cd docs
git init
git checkout --orphan gh-pages
```

Copy a first version of the site in the directory, then :

```
git add *
git commit -m "initial site content"
git remote add origin "https://github.com/NeVraX182/docs.git"
git push --set-upstream origin gh-pages
```

8.2. Configure Maven plugin

Maven SCM plugin configuration

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-scm-publish-plugin</artifactId>
    <version>3.0.0</version>
    <inherited>false</inherited>
    <configuration>
        <scmBranch>gh-pages</scmBranch>
        <!-- token generated from github > settings > Developer settings > Personal access tokens > public_repo -->
        <pubScmUrl>scm:git:https://USER:TOKEN@github.com/USER/docs.git</pubScmUrl>
        <content>target/generated-docs</content>
    </configuration>
</plugin>
```

9. Use in pipelines

To use in Jenkins Pipelines, see Jenkins Best Practices

To use in Gitlab Pipelines, see Gitlab Best Practices

10. Confluence to Asciidoc

From link::https://confluence.atlassian.com/doc/export-content-to-word-pdf-html-and-xml-139475.html[Confluence HTML export] and

- Go to the space and choose **[Space tools]** → **[Content Tools]** from the bottom of the sidebar

- Choose [**Export**] (you may need to zoom out to see it)
- Choose [**HTML**]
- Click [**Unselect All**] and choose specific pages
- Click [**Export**]

TODO: https://github.com/asciidoctor/asciidoctor.org/blob/master/docs/_includes/convert-from-confluence-xhtml.adoc

11. Reveal.js

See official documentation on Asciidoctor website.

11.1. Examples

Official examples here, especially the G2 Scoring one.

Personnal examples : see at the end of the homepage.

11.2. Fullscreen on smartphone

On Android, common browsers have adress bar preventing reveal.js sites to go fullscreen. To get a real fullscreen, install **Full Screen Browser**.

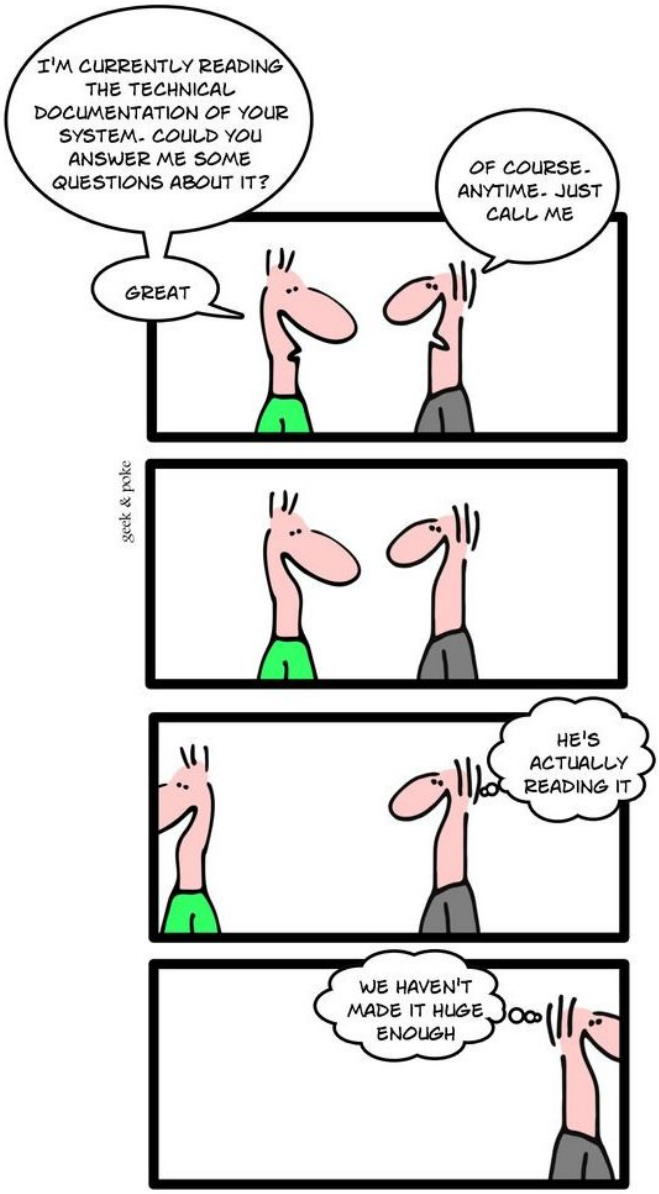
11.3. Generation

Maven plugin showed above include a reveal.js generation.

11.4. Inline options

Some options can be added in HTTP address :

- PRES-asciidoc.html?transition=convex
- PRES-asciidoc-light.html?slideNumber=false
- https://revealjs.com/?parallaxBackgroundImage=https%3A%2F%2Fs3.amazonaws.com%2Fhakim-static%2Freveal-js%2Freveal-parallax-1.jpg&parallaxBackgroundSize=2100px%20900px


I'M CURRENTLY READING THE TECHNICAL DOCUMENTATION OF YOUR SYSTEM. COULD YOU ANSWER ME SOME QUESTIONS ABOUT IT?
GREAT
OF COURSE. ANYTIME. JUST CALL ME
geek & poke
HE'S ACTUALLY READING IT
WE HAVEN'T MADE IT HUGE ENOUGH

12. Appendix

12.1. Revision marks

Differences since last tag