



zenika
<animés par la passion>

AsciiDoc & doc-as-code Best Practices

Version 1.18-SNAPSHOT

2018-11-01

Table of Contents

| | |
|---------------------------------------|----|
| 1. Editor | 2 |
| 2. Inline icons | 3 |
| 3. Custom code highlighter | 3 |
| 4. init and closure | 4 |
| 5. Generation using Maven | 4 |
| 6. .doc to .adoc | 7 |
| 7. Publish HTML & PDF to github | 7 |
| 7.1. Initialize Github space | 7 |
| 7.2. Configure Maven plugin | 8 |
| 8. Use in pipelines | 8 |
| 9. Reveal.js | 9 |
| 9.1. Examples | 9 |
| 9.2. Generation | 9 |
| 9.3. Inline options | 9 |
| 10. Appendix | 11 |
| 10.1. Revision marks | 11 |

Table 1. History

| Date | Author | Detail |
|------------|----------|--|
| 2018-11-01 | bcouetil | - reorganized asciidoc best practices - added some Geek & Poke images - reveal.js Zenika dark + light themes - 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 |

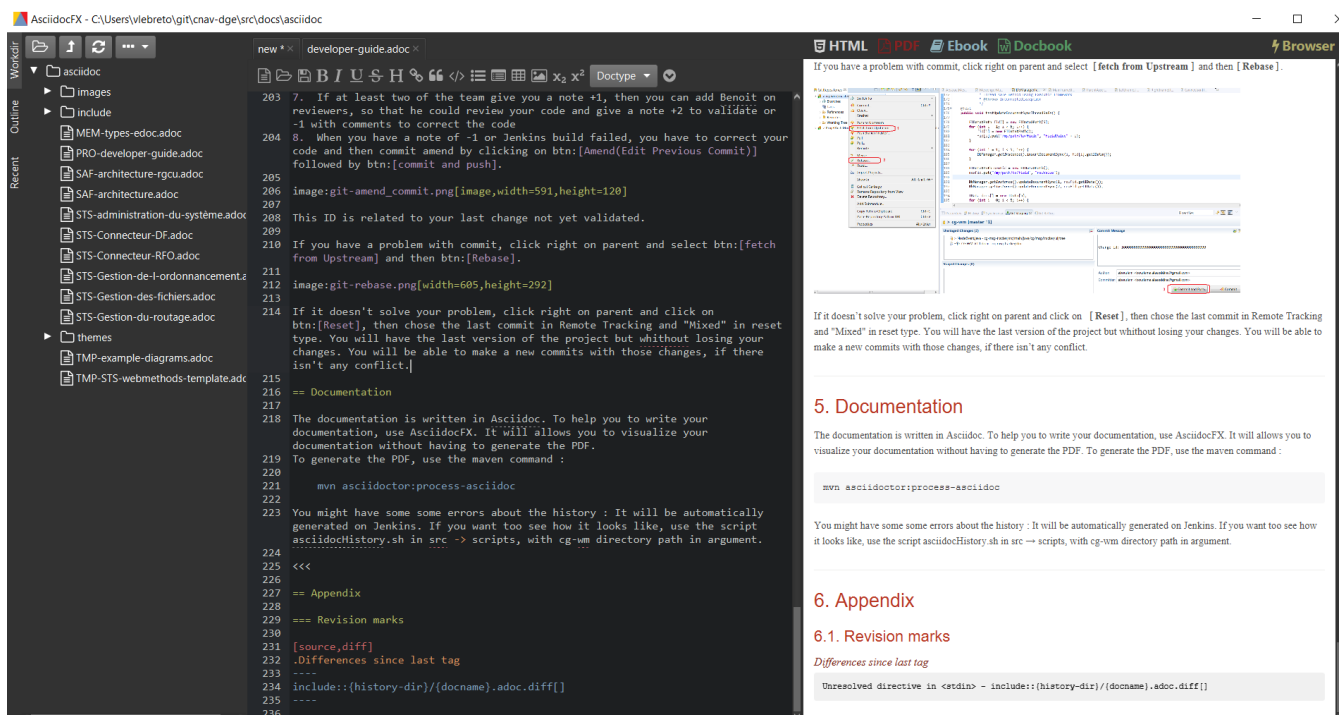
THE HISTORY OF...

GETTING CODERS TO DOCUMENT



1. 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 <<<)

To show a in progress diagram to colleagues, you can generate a web link using [PlantUml generator](#).

2. 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

3. 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



4. 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.

5. Generation using Maven

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

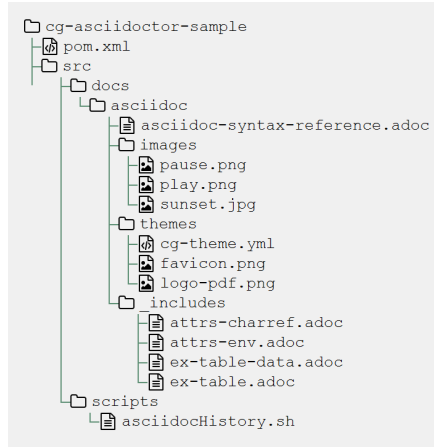


Figure 1. File system tree

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>
  <version>1.5.7.1</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</artifactId>
      <version>1.5.8.1</version>
    </dependency>
  </dependencies>
  <configuration>
    <skip>${adoc.skip}</skip>
    <sourceDirectory>src/docs/asciidoc</sourceDirectory>
    <!-- <requires>
      <require>asciidoctor-diagram</require>
    </requires> -->
    <attributes>
      <icons>font</icons>
      <idseparator>-</idseparator>
      <idprefix />
      <!-- custom -->
      <source-dir>../../main/java</source-dir>
      <test-dir>../../test/java</test-dir>
      <project-version>${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>
    </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>
  <toclevels>5</toclevels>
  <sectanchors>true</sectanchors>
  <docinfo1>true</docinfo1>
  <favicon>themes/favicon.png</favicon>
  <!-- custom style if default is not suitable -->
  <!-- <stylesheet>rubygems.css</stylesheet> -->
  <!-- <stylesdir>themes</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>

    <outputDirectory>${project.slides.directory}</outputDirectory>
    <templateDir>${project.build.directory}/asciidoctor-revealjs-${asciidoctor-
revealjs.version}/templates</templateDir>
    <sourceHighlighter>coderay</sourceHighlighter>
    <attributes>
      <revealjsdir>reveal.js-${revealjs.version}</revealjsdir>
      <!-- default closest to Zenika layout : -->
      <!-- <revealjs_theme>blood</revealjs_theme> -->
      <revealjs_customtheme>themes/reveal-zenika-dark.css</revealjs_customtheme>
      <!-- <revealjs_customtheme>themes/reveal-zenika-light.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>
    </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>${project.build.directory}/generated-docs/images</imagesdir>
      <toc />
      <toclevels>3</toclevels>
      <pdf-style>${user.dir}/src/docs/asciidoc/themes/cg-theme.yml</pdf-style>
      <pagenums />
    </attributes>
  </configuration>
</execution>
</executions>
</plugin>

```


To generate the PDF locally, use the maven command :

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

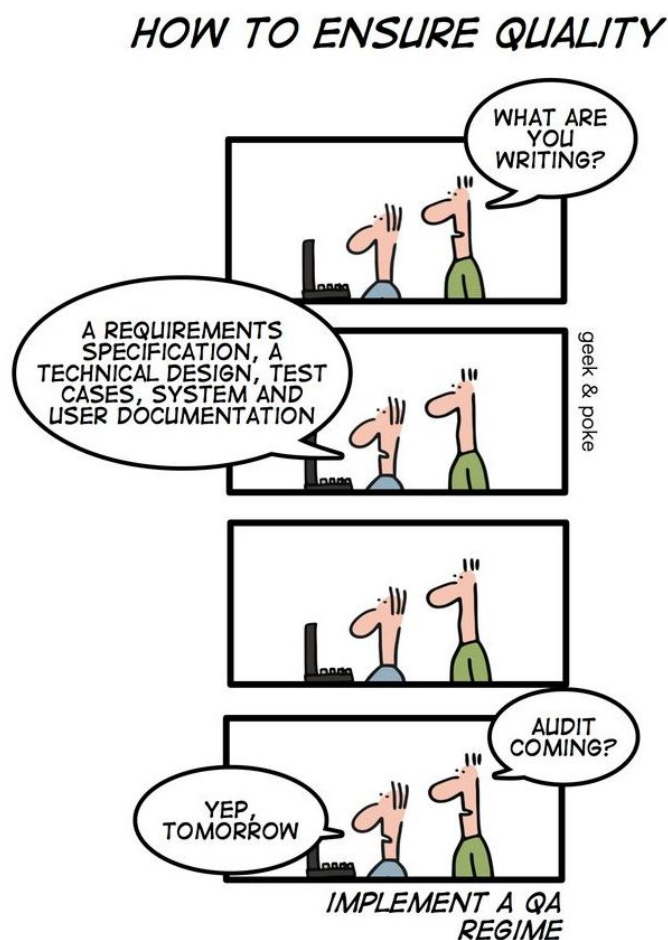
6. .doc to .adoc

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
```



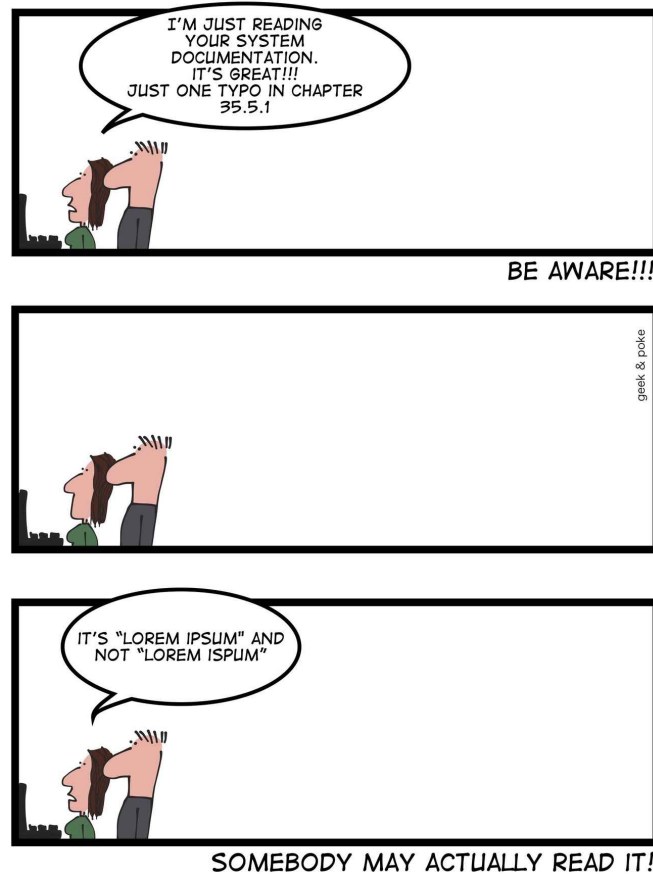
7. Publish HTML & PDF to github

7.1. 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
```



7.2. 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>
```

8. Use in pipelines

To use in Jenkins Pipelines, see [Jenkins Best Practices](#)

To use in Gitlab Pipelines, see [Gitlab Best Practices](#)

9. Reveal.js

See [official documentation](#) on AsciiDoctor website.

9.1. Examples

Official examples [here](#), especially [the G2 Scoring one](#).

Personnal example : [dark on light](#) and [light on dark](#)

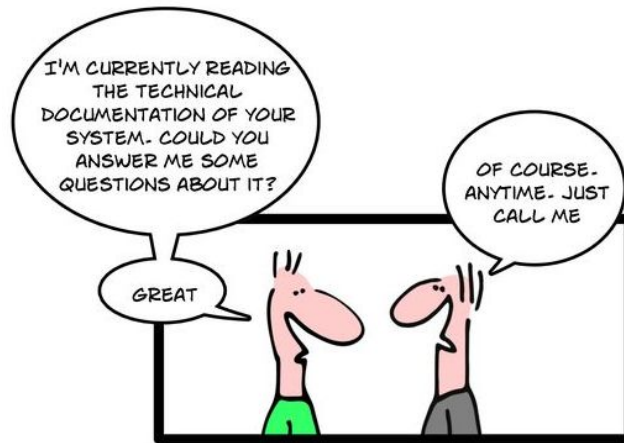
9.2. Generation

Maven plugin showed above include a reveal.js generation.

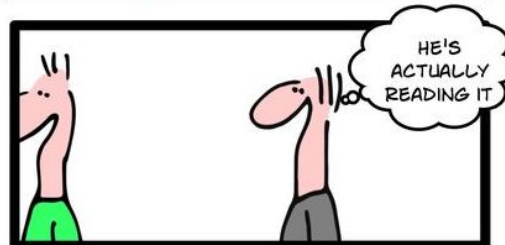
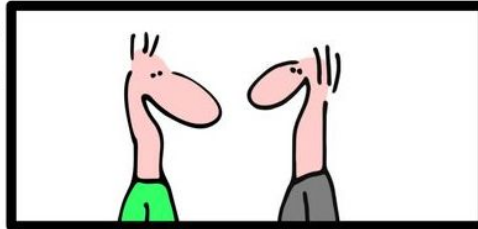
9.3. Inline options

Some options can be added in HTTP address :

- [PRES-asciidoc.html?transition=convex](#)
- [PRES-asciidoc-light.html?slideNumber=false](#)



geek & poke



10. Appendix

10.1. Revision marks

Differences since last tag

