AsciidoctorとGradleで作る文書執筆環境

http://github.com/kazurayam/DAE

2020-04-13

目次

1. はじめに

本文書は `Asciidoc` とそのRubyによる実装である `Asciidoctor` を用いてAsciidoc文書を執筆する環境を構築する手順を示します。実行環境はWindows、Linux、macOSの各OSに対応しています。

この文書の手順により以下のことができるようになります。

Asciidoc形式で執筆した文書をHTML/PDF形式に変換する。	Asciidoc文書変換用スクリプト
変換結果をリアルタイムにプレビューしながら、エディターで文書を編集する。	Visual Studio Code拡張設定

Asciidocは表現力の高い文書をテキストファイルベースで執筆できるテキストプロセッサーです。他の軽量テキストプロセッサーが持たない文書間のインクルードやソースコードの挿入などの機能も有し、かつ簡潔です。特に技術文書の執筆には大きな力を発揮することでしょう。

Asciidocでこのような脚注を表現することができます。Asciidocの高い表現力を示す一つの例です。

一般的にこのようなテキストプロセッサーを用いた執筆環境を構築するためには多くの準備が必要となりますが、本文書の手順は極力初期導入するプロダクトを少なく、簡単に快適な執筆環境を整えられるよう考えています。

具体的には文書の変換に、実行をJVM環境だけに依存する `Asciidoctorj` と `Gradle` を活用し、執筆環境については `Visual Studio Code` を用いることでリアルタイムに文書をプレビューしながら、最後にコマンド一つでPDF化できるように準備してあります。

本文書がみなさんの執筆活動のお手伝いになれば幸いです。

1.1. 謝辞

本文書の手順の実装であるビルドスクリプトやテーマでは次のプロダクトと技術資料が使われています。

プロダクト名の隣にライセンスを併記しました。商用利用等で制限のあるプロダクトはありませんが、それぞれライセンスを確認してください。

Font

- 源真ゴシック - SIL Open Font License 1.1 - http://jikasei.me/font/genshin/

2. Asciidoc 文書変換用スクリプトを使う準備をする

本手順で用いる Asciidoc 文書変換用スクリプトはビルドツールである Gradle を活用しており、実行するためには Java 実行環境が必要です。

Java実行環境は、文書変換スクリプトを動作させる過程であなたが自分のコンピュータの OS 環境に手動で導入する必要がある唯一のプロダクトです。それ以外のものは Gradle によりプロジェクトとして独立した形で自動的に導入されます。

あなたがお使いのコンピュータのコマンドライン環境（macOS/Linuxではターミナル 、Windowsではcmd.exeかpwsh.exe）で `java -version` コマンドを入力し、Java 8 以上のバージョンが表示されるようであれば既に準備は整っています。

macOS/Linuxの場合

```
$ java -version
openjdk version "1.8.0_242"
OpenJDK Runtime Environment (AdoptOpenJDK)(build 1.8.0_242-b08)
OpenJDK 64-Bit Server VM (AdoptOpenJDK)(build 25.242-b08, mixed mode)
```

Windowsの場合

```
C:\> java -version
openjdk version "1.8.0_242"
OpenJDK Runtime Environment (AdoptOpenJDK)(build 1.8.0_242-b08)
OpenJDK 64-Bit Server VM (AdoptOpenJDK)(build 25.242-b08, mixed mode)
```

現在 Java 9 以降の環境ではビルド時にワーニングが出力されるため本手順では Java 8 を使って解説しています。筆者の Java 11 環境で変換の動作は正しいことが確認できていますので適宜読み替えて Java を導入してください。この問題は将来解消されるでしょう。

=== Java 実行環境の導入(macOS / Linuxの場合)

もしあなたの macOS / Linux 環境に Java 実行環境がなければ `SDKMAN` を利用することで、ターミナルから簡単に導入することができます。

```
SDKMAN! is a tool for managing parallel versions of multiple Software Development Kits on most
Unix bases systems.

https://sdkman.io/
```

手順.SDKMANを用いたJavaの導入

```
$ curl -s "https://get.sdkman.io" | bash       ❶
$ source "$HOME/.sdkman/bin/sdkman-init.sh"    ❷
$ sdk list java                                ❸
================================================================================
Available Java Versions
================================================================================
 Vendor        | Use | Version      | Dist    | Status     | Identifier
--------------------------------------------------------------------------------
 AdoptOpenJDK  |     | 14.0.0.j9    | adpt    |            | 14.0.0.j9-adpt
               |     | 14.0.0.hs    | adpt    |            | 14.0.0.hs-adpt
               |     | 13.0.2.j9    | adpt    |            | 13.0.2.j9-adpt
               |     | 13.0.2.hs    | adpt    |            | 13.0.2.hs-adpt
               |     | 12.0.2.j9    | adpt    |            | 12.0.2.j9-adpt
               |     | 12.0.2.hs    | adpt    |            | 12.0.2.hs-adpt
               |     | 11.0.6.j9    | adpt    |            | 11.0.6.j9-adpt
               |     | 11.0.6.hs    | adpt    |            | 11.0.6.hs-adpt
               |     | 8.0.242.j9   | adpt    |            | 8.0.242.j9-adpt
               |     | 8.0.242.hs   | adpt    |            | 8.0.242.hs-adpt
 Azul Zulu     |     | 14.0.0       | zulu    |            | 14.0.0-zulu
               |     | 13.0.2       | zulu    |            | 13.0.2-zulu
               |     | 12.0.2       | zulu    |            | 12.0.2-zulu
               |     | 11.0.6       | zulu    |            | 11.0.6-zulu
               |     | 11.0.6.fx    | zulu    |            | 11.0.6.fx-zulu
               |     | 10.0.2       | zulu    |            | 10.0.2-zulu
               |     | 9.0.7        | zulu    |            | 9.0.7-zulu
               |     | 8.0.242      | zulu    |            | 8.0.242-zulu
               |     | 8.0.232.fx   | zulu    |            | 8.0.232.fx-zulu
               |     | 8.0.202      | zulu    |            | 8.0.202-zulu
               |     | 7.0.181      | zulu    |            | 7.0.181-zulu
 ================================================================================
Use the Identifier for installation:

    $ sdk install java 11.0.3.hs-adpt
================================================================================
$ sdk install java 8.0.242.hs-adpt          ❹
```

❶ SDKMANを導入します。

❷ SDKMANを起動するシェルスクリプト `sdkman-init.sh` を現在実行中のシェルにロードします。

❸ 今現在導入可能なJavaのバージョンを一覧します。

❹ 8.0系の最新バージョンを指定して Java を導入します。

また、Gradle は `JAVA_HOME` 環境変数に実行環境の Java のパスが設定されていることを期待していますので、`.bash_profile` で次のように `JAVA_HOME` を設定します。

手順. JAVA_HOME の設定

```
$ vi ~/.bash_profile ❶
export JAVA_HOME=~/.sdkman/candidates/java/current ❷
$ source ~/.bash_profile ❸
```

❶ vi エディタで `.bash_profile` を開きます。

❷ 本ラインをファイルの最下部に追加し vi を保存終了します。

❸ 設定を適用します。

これで準備完了です。

SDKMAN について

SDKMAN は主に Java エコシステムの開発環境をコマンドラインから簡単に導入・設定するためにつくたれた管理ソフトウェアです。

たとえば簡単に各種 Java のバージョンを導入し切り替えることができます。

手順. SDKMAN による Java のバージョン切り替え

```
$ sdk install java 11.0.1-open ❶
$ sdk default java 11.0.1-open ❷
$ sdk default java 8.0.192-zulu ❸
```

❶ Java 11 を導入

❷ Java 11 をデフォルトに設定

❸ Java 8 をデフォルトに設定

2.1. Java 実行環境の導入（Windowsの場合）

TODO

3. AsciidocからHTML/PD文書を作成する

3.1. サンプル文書の変換を試す

環境の準備ができましたので Asciidoc 文書を HTML/PDF に変換してみます。

変換に使うスクリプトは github のリポジトリに公開されており、リポジトリには HTML/PDF 変換に使うファイル一色と、文書サンプルとして この文書 のAsciidocファイルが置かれています。まずはサンプル文書が正しく変換できるかを試してみましょう。

macOS / Linux の場合は次のようにします。

手順.PDF 変換ビルドスクリプトを取得して実行する

```
$ curl -L -O https://github.com/kazurayam/DAE/archive/master.zip    ❶
$ unzip master.zip    ❷
$ cd DAE              ❸
$ ./gradlew docs      ❹
...
BUILD SUCCESSFUL in 19s   ❺
2 actionable tasks: 2 executed
```

❶ リポジトリのファイルをダウンロードします。

❷ ダウンロードした.zipファイルを展開します。

❸ 展開してできたディレクトリのなかにcdします。

❹ Gradleのビルドを実行します。初回実行時はビルドに必要なファイル群をダウンロードするため少し時間がかかります。次回からは数秒で完了します。

❺ `BUILD SUCCESSFUL` が出力されればビルド成功です。

Windowsをお使いの場合に同等の操作をブラウザとエキスプローラーを使って行います。

TODO

プロキシーサーバーの設定

もしお使いのコンピューターがプロキシーサーバーを経由してインターネットにアクセスする場合は、次のコマンドを `./gradlew docs` する前に投入してください。インターネットからライブラリをダウンロードして取得するのに必要です。ホスト名(`example.com`)とポート番号(`8080`)はそれぞれの環境に合うように変更してください。

```
プロキシーサーバーの設定(Windows)
----
set JAVA_OPTS=-DproxyHost=example.com -DproxyPort=8080
----
```

```
プロキシーサーバーの設定(macOS / Linux)
----
export JAVA_OPTS=-DproxyHost=example.com -DproxyPort=8080
----
```