

AsciidoctorTemplate

目次

1. 前提条件	1
2. インストール方法.....	1
3. 使用方法	1
3.1. ドキュメントの生成.....	1
3.2. ライブリロードを使う場合.....	2
4. Gradleのプロキシ設定	3
5. ページの公開方法.....	3

Javaさえ動く環境であればAsciidoctorすぐに始められる雛形プロジェクトです。

1. 前提条件

事前にインストールしておくもの

- Java
- Chromeの [LiveReload](#)
- Gitクライアント（必須ではない）

2. インストール方法

- <https://github.com/Takumon/AsciidocTemplate.git>をクローンする。（Gitクライアントをインストールしていない場合はGithubの[Takumon/AsciidocTemplate](#)で **[Clone or downloadzip]** > **[Download ZIP]** をクリックし、ダウンロードしてZIPファイルを解凍する）

```
$ git clone https://github.com/Takumon/AsciidocTemplate.git
```

3. 使用方法

3.1. ドキュメントの生成



プロキシ環境下の場合は、事前に[Gradleのプロキシ設定](#)を行ってください。

- プロジェクト直下で、下記を実行する。

```
$ cd AsciidocTemplate
$ ./gradlew asciidoctor
```

- build/docs/asciidocフォルダ配下にHTMLとPDFが生成される。[index.html](#)をブラウザで開くと生成されたHTMLが見れる。`index.pdf`をPDFビューワーで開くとPDFが見れる。

```
$ tree /f build/docs/asciidoc
build/docs/asciidoc/
├── html5
│   ├── css
│   ├── images
│   ├── index.html
│   └── js
├── images
│   ├── github
│   │   ├── blockquote-arrow.png
│   │   └── li-chevron.png
│   ├── golo
│   │   ├── body-bg.png
│   │   └── pre-bg.png
│   ├── maker
│   │   └── body-bg.png
│   └── riak
│       ├── body-bg.jpg
│       ├── footer-bg.jpg
│       ├── info-bg.jpg
│       ├── pre-bg.jpg
│       └── sidebar-bg.jpg
└── pdf
    ├── css
    ├── images
    ├── index.pdf
    └── js
```

3.2. ライブリロードを使う場合

adocファイルを修正したらリアルタイムにHTMLを出力し、ブラウザに修正が反映されるようにする。

3.2.1. 手順

- プロジェクト直下で下記を実行する。

```
$ cd AsciiDocTemplate
$ ./gradlew -t asciidoctor
```

- もう一つ別のコマンドプロンプト(またはターミナル)を起動し、プロジェクト直下で下記を実行する。

```
$ cd AsciiDocTemplate
$ ./gradlew liveReload
```

- Chromeで <http://localhost:35729/html5/> を開く。
- ChromeのLiveReload機能をONにする（右上にあるLiveReloadアイコンをクリックする）

- この状態でadocファイルを編集するとブラウザにリアルタイムに反映される。

4. Gradleのプロキシ設定

- プロジェクト直下の`gradle.properties`を編集する。

リスト 1. *gradle.properties*

```
#####
# gradlew実行時のプロキシ設定
#####
# http
#systemProp.http.proxyHost = [your proxy host] ①
#systemProp.http.proxyPort = [your proxy port] ②
#systemProp.http.proxyUser = [your proxy user] ③
#systemProp.http.proxyPassword = [your proxy password] ④
#systemProp.http.nonProxyHosts = localhost ⑤

# https ⑥
#systemProp.https.proxyHost = [your proxy host]
#systemProp.https.proxyPort = [your proxy port]
#systemProp.https.proxyUser = [your proxy user]
#systemProp.https.proxyPassword = [your proxy password]
#systemProp.https.nonProxyHosts = localhost

org.gradle.jvmargs = -Dfile.encoding=UTF-8
org.gradle.daemon = true
#org.gradle.java.home = [JDK install dir path]
```

- ① コメントアウトしてプロキシのホストを指定する。
- ② コメントアウトしてプロキシのポートを指定する。
- ③ 認証が必要であれば、コメントアウトしてユーザ名を指定する。
- ④ 認証が必要であれば、コメントアウトしてパスワードを指定する。
- ⑤ プロキシ除外対象のホストがあれば区切りで指定する。
- ⑥ httpsも同様に設定が必要であればコメントアウトして、それぞれ値を指定する。

5. ページの公開方法

[GitHub](#) [Pages](#)を使用してドキュメントを公開できるように、ドキュメント生成時に `docs` フォルダ配下にもドキュメントを出力するようにしています。

- Githubのリポジトリで[**setting**]を選択します。
- GitHub PagesのSourceで`master branch /docs folder`を選択し[**Save**]ボタンをクリックします。
- GitHub PagesのSourceに
URLが記載されているので、そこにアクセスするとドキュメントが見れます。