

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>をクローンします。（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ファイルを解凍してください）

```
$ 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](#)をしてください。

- プロジェクト直下で、下記を実行します。

```
$ ./gradlew asciidoctor
```

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

```

$$ tree /f build/docs/asciidoc/
build/docs/asciidoc/
├── html5 ①
│   ├── css
│   │   └── index.css
│   ├── images
│   ├── index.html
│   └── js
│       └── index.js
├── images ②
│   ├── 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 ③
    ├── css
    │   └── index.css
    ├── images
    ├── index.pdf
    └── js
        └── index.js

```

① HTML形式のドキュメント

② HTML形式のドキュメントのテーマ用の画像

③ PDF形式のドキュメント

3.2. ライブリロードを使う場合

adocファイルを修正するとHTMLを出力し、ブラウザに修正がリアルタイムに反映されるようにします。

3.2.1. 手順

- プロジェクト直下で下記を実行します。

```
$ ./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します。

```
$ ./gradlew liveReload
```

- Chromeで <http://localhost:35729/html5/> を開きます。
- ChromeのLiveReload機能をONにします。（右上にあるLiveReloadアイコンをクリックします）
- この状態でadocファイルを編集するとブラウザに編集内容がリアルタイムに反映されます。

4. Gradleのプロキシ設定

- プロジェクト直下の`gradle.properties`を編集します。

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します。
- ② コメントアウトしてプロキシのポートを指定します。
- 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します。
- 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します。
- ⑤ プロキシ除外対象のホストがあれば区切りで指定します。
- ⑥ httpsも同様に設定が必要であればコメントアウトして、それぞれ値を指定します。

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。公開する場合は自分のリポジトリで書き手順を実施してください。

- 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。