

参考

https://tech.nri-net.com/entry/spring_boot_multi_project

<https://eng-entrance.com/java-springboot>

<https://intellectual-curiosity.tokyo/2019/04/21/spring-boot%E3%81%A7%E3%83%87%E3%83%BC%E3%82%BF%E3%83%99%E3%83%BC%E3%82%B9%E3%81%AB%E3%82%A2%E3%82%AF%E3%82%BB%E3%82%B9%E3%81%99%E3%82%8B%E6%96%B9%E6%B3%95/>

<https://qiita.com/kuro227/items/a16e22ac12afe7442a3d>

<https://qiita.com/ozaki25/items/3b348874b6db5ab4f04f>

概要

Spring Framework をベースにしたフレームワーク。

コーディング量を削減したり、Eclipse ベースの環境 (STS) があったり、jar にコンテナを含めたりすることができる。

開発環境

STS(<https://spring.io/tools3/sts/all>) をダウンロードするか、Eclipse でヘルプ -> Eclipse マーケットプレイスで STS を検索し、「Spring Tools 4 for Spring Boot」をインストール。

サンプル作成

1. 新規 -> Spring スターター・プロジェクト
2. 利用するライブラリを選択する画面で必要なものを選択
 1. 開発ツール -> Spring Boot DevTools(自動リロードとか)、Lombok
 2. テンプレートエンジン -> Thymeleaf
 3. Web -> Spring Web
3. ライブラリはあとで変更できる
 1. 変更するにはプロジェクトを右クリック -> Spring -> スターターの編集

その他

コマンドでビルド、実行したい場合

■ class ファイルを実行する場合

```
mvn compile
mvn spring-boot:run
```

実行は以下のコマンドでも実行できる

```
java -cp `mvn dependency:build-classpath | grep -v "^\":target/classes` com.example.demo.DemoApplication
```

mvn のクラスパス情報に target/classes を追加して、実行。

■ パッケージ (jar 化) する場合

この場合、devtools の自動リロードは効かない

```
mvn package
```

```
java -jar target/xxxxx.jar
```

```
mvn compile
```

自動リロード

Spring Devtools を使ってホットデプロイする場合は、pom.xml に以下を追加する

```
<dependencies>
  <!-- ここから -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
  </dependency>
  <!-- ここまでを追加 -->
</dependencies>
```

resources/application.properties or resources/application.yml のファイルに設定を追加。

■ application.properties の場合

```
spring.devtools.remote.restart.enabled=true
spring.devtools.livereload.enabled=true
```

■ application.yml の場合

```
spring:
  devtools:
    livereload:
      enabled: true
    restart:
      enabled: true
```

クラス名がかぶった場合

<https://layerprogram.com/springcontrollerdi/>