

バックエンド第1回 新規プロジェクト作成と動作確認

- バックエンド第1回 新規プロジェクト作成と動作確認
 - 新規プロジェクト作成
 - spring initializr で作成する
 - ダウンロードしたzipファイルを展開
 - 設定ファイルダウンロードとフォルダへの配置
 - プロジェクトのビルド
 - 事前準備
 - リモートDB使用方法
 - ビルドコマンド実行
 - 動作確認用ページの作成
 - フォルダ体系を作成
 - 動作確認用ページ作成
 - ビルドします
 - 起動します
 - ダッシュボードから起動
 - gradlewで起動
 - コマンドラインから起動
 - 動作確認
 - WEBブラウザで確認
 - Postmanで確認
 - まとめ

新規プロジェクト作成

spring initializr で作成する

下記URLにアクセスします

<https://start.spring.io/>

以下のようにオプションを選択します。

Project: Gradle Project

Language: Java

Spring Boot: 2.3.1

Project Metadata

Group: com.example

Artifact: webapi

Name: webapi

Description: Demo project for Spring Boot

Package Name: com.example.webapi

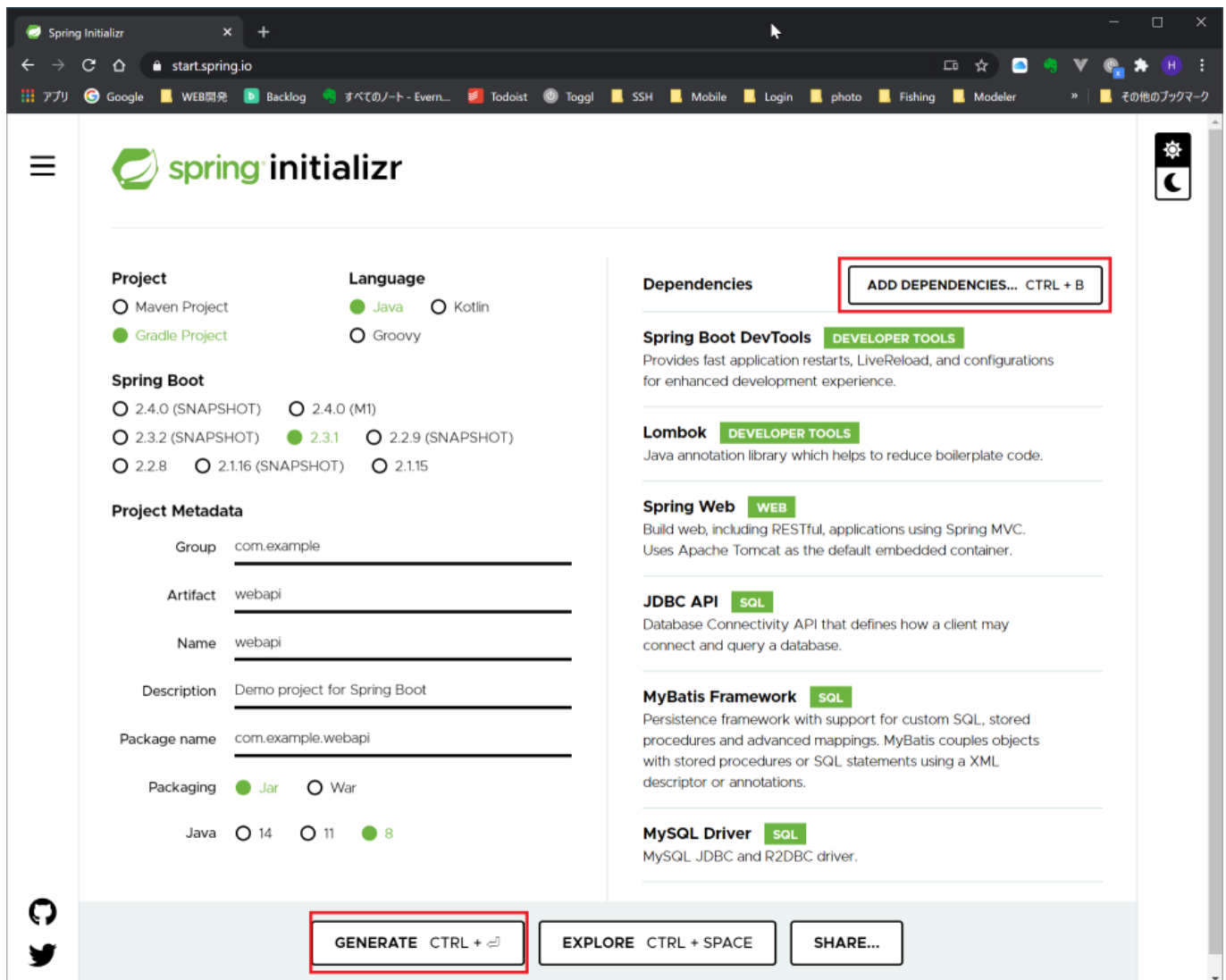
Packaging: jar

Java: 8

Dependencies

Spring Boot DevTools
Lombok
Spring Web
JDBC API
MyBatis Framework
MySQL Driver

DependenciesはADD DEPENDENCIES.. Ctrl + B ボタンをクリックして一覧から選択します。



選択後、GENERATE CTRL+[Enter] ボタンをクリックします。

webapi.zip がダウンロードされます。

ダウンロードしたzipファイルを展開

ダウンロードした webapi.zipを作業フォルダに展開します。

```
.  
./gitignore  
./build.gradle
```

```
./gradle
./gradle/wrapper
./gradle/wrapper/gradle-wrapper.jar
./gradle/wrapper/gradle-wrapper.properties
./gradlew
./gradlew.bat
./HELP.md
./settings.gradle
./src
./src/main
./src/main/java
./src/main/java/com
./src/main/java/com/example
./src/main/java/com/example/webapi
./src/main/java/com/example/webapi/WebapiApplication.java
./src/main/resources
./src/main/resources/application.properties
./src/main/resources/static
./src/main/resources/templates
./src/test
./src/test/java
./src/test/java/com
./src/test/java/com/example
./src/test/java/com/example/webapi
./src/test/java/com/example/webapi/WebapiApplicationTests.java
```

設定ファイルダウンロードとフォルダへの配置

設定ファイル 2種類をダウンロードします。

- <https://hiszuk.github.io/webapi/resource/application.properties>
- <https://hiszuk.github.io/webapi/resource/mybatis-config.xml>

`./src/main/resources` に配置します。

元の`application.properties` は空なので上書きします。

プロジェクトのビルド

事前準備

- インストールしたMariaDBのサービスを起動しておいてください
MariaDBがインストールできていない方は、下記のリモートDB使用方法を参照して設定ファイルを書き換えて下さい。
- プロキシの設定がうまくできませんでしたので、iPhoneをテザリングして実行してください

リモートDB使用方法

`./src/main/resources/application.properties` の下記エントリを書き換えます。

書換前

```
spring.datasource.url=jdbc:mysql://localhost:3306/sampledbs?
useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=JST&
useUnicode=true&characterEncoding=utf-8
spring.datasource.username=sample
spring.datasource.password=sample
```

書換後

```
spring.datasource.url=jdbc:mysql://10.219.2.11:3306/sampledbs?
useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=JST&
useUnicode=true&characterEncoding=utf-8
spring.datasource.username=remote
spring.datasource.password=remote
```

※`spring.datasource.url`のエントリは複数行に見えますが、実際は1行です。改行を入れないでください。

ビルドコマンド実行

webapi のディレクトリで下記コマンドを実行します。

```
PS C:\home\webapi> .\gradlew build
```

下記のようにBUILD SUCCESSFUL in 00s と表示されればOKです。

```
Starting a Gradle Daemon (subsequent builds will be faster)

> Task :test
2020-07-11 16:11:36.225 DEBUG 9364 --- [extShutdownHook]
o.s.w.c.s.GenericWebApplicationContext : Closing
org.springframework.web.context.support.GenericWebApplicationContext@39401536,
started on Sat Jul 11 16:11:33 JST 2020
2020-07-11 16:11:36.231 INFO 9364 --- [extShutdownHook]
o.s.s.concurrent.ThreadPoolTaskExecutor : Shutting down ExecutorService
'applicationTaskExecutor'

BUILD SUCCESSFUL in 49s
5 actionable tasks: 5 executed
```

次のようなフォルダ体系になりましたでしょうか？

```
./.gitignore
./.gradle
./build
```

```
./build/classes
./build/generated
./build/libs
./build/libs/webapi-0.0.1-SNAPSHOT.jar    <-- これが作成モジュールです
./build/reports
./build/resources
./build/test-results
./build/tmp
./build.gradle
./gradle
./gradlew
./gradlew.bat
./HELP.md
./settings.gradle
./src
./src/main
./src/main/java
./src/main/java/com
./src/main/java/com/example
./src/main/java/com/example/webapi
./src/main/java/com/example/webapi/WebapiApplication.java
./src/main/resources
./src/main/resources/application.properties
./src/main/resources/mybatis-config.xml
./src/main/resources/static
./src/main/resources/templates
./src/test
./src/test/java
./src/test/java/com
./src/test/java/com/example
./src/test/java/com/example/webapi
./src/test/java/com/example/webapi/WebapiApplicationTests.java
```

※ここまでくれば、iphoneのデザリングを切っても構いません。

動作確認用ページの作成

フォルダ体系を作成

先ほど説明したレイヤー構造に基づいたフォルダを作成してください。

```
src/main/java/com/example/webapi/controller
src/main/java/com/example/webapi/entity
src/main/java/com/example/webapi/repository
src/main/java/com/example/webapi/service
src/main/resources/com/example/webapi/repository
```

動作確認用ページ作成

controller 配下に **StartPageController.java** を新規ファイルで作成してください。

`StartPageController.java` の中身は次のように記載します。

```
package com.example.webapi.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
public class StartPageController {
    @RequestMapping("/")
    @ResponseBody
    public String home() {
        return "The backend is started...";
    }
}
```

ビルドします

```
PS C:\home\webapi> .\gradlew build

> Task :test
2020-06-27 11:40:05.727 DEBUG 11864 --- [extShutdownHook]
o.s.w.c.s.GenericWebApplicationContext : Closing
org.springframework.web.context.support.GenericWebApplicationContext@39401536,
started on Sat Jun 27 11:40:01 JST 2020
2020-06-27 11:40:05.733 INFO 11864 --- [extShutdownHook]
o.s.s.concurrent.ThreadPoolTaskExecutor : Shutting down ExecutorService
'applicationTaskExecutor'

BUILD SUCCESSFUL in 17s
5 actionable tasks: 3 executed, 2 up-to-date
```

起動します

作成されたjarファイルを起動します。

起動方法は3種類あります。

1. SPRING-BOOT DASHBOARDから起動
開発中はデバッグもでき、おすすめのモードです。ソース変更して保存すると自動でリロードがかかります。
2. gradlewで起動
ダッシュボードに出てこない場合この方法で起動できます。メモリをそれなりに使います。
3. コマンドラインから起動
運用はこの起動方式です。

ダッシュボードから起動



起動すると以下のような画面となります。



gradlewで起動

```
PS C:\home\webapi> .\gradlew bootRun
```

以下のような起動画面になります。

終了するには **Ctrl+C** を入力します。

```
> Task :bootRun
```

[illegible]

```
2020-07-11 16:45:03.419 INFO 16132 --- [ restartedMain]
com.example.webapi.WebapiApplication : Starting WebapiApplication on
DZA201803906 with PID 16132 (C:\home\20\□_RESTAPI\ch2-
1\webapi\build\classes\java\main
:
:
2020-07-11 16:45:06.550 INFO 16132 --- [ restartedMain]
com.example.webapi.WebapiApplication : Started WebapiApplication in 3.496
seconds (JVM running for 3.996)
<=====----> 75% EXECUTING [18s]
> :bootRun
```

コマンドラインから起動

```
PS C:\home\webapi> cd build\libs
PS C:\home\webapi\build\libs> java -jar webapi-0.0.1-SNAPSHOT.jar
```

以下のような起動画面になります。

終了するには **Ctrl+C** を入力します。

[illegible]

```
2020-07-11 16:56:37.359 INFO 13944 --- [main]
```



```
com.example.webapi.WebapiApplication : Starting WebapiApplication on
DZA201803906 with PID 13944 (C:\home\20\□_RESTAPI\ch2-1\webapi\build\libs\webapi-
0.0.1-SNAPSHOT.jar started by katouhis in C:\home\20\□_RESTAPI\ch2-
1\webapi\build\libs)
:
:
2020-07-11 16:56:41.518 INFO 13944 --- [          main]
com.example.webapi.WebapiApplication : Started WebapiApplication in 4.953
seconds (JVM running for 5.633)
```

動作確認

WEBブラウザで確認

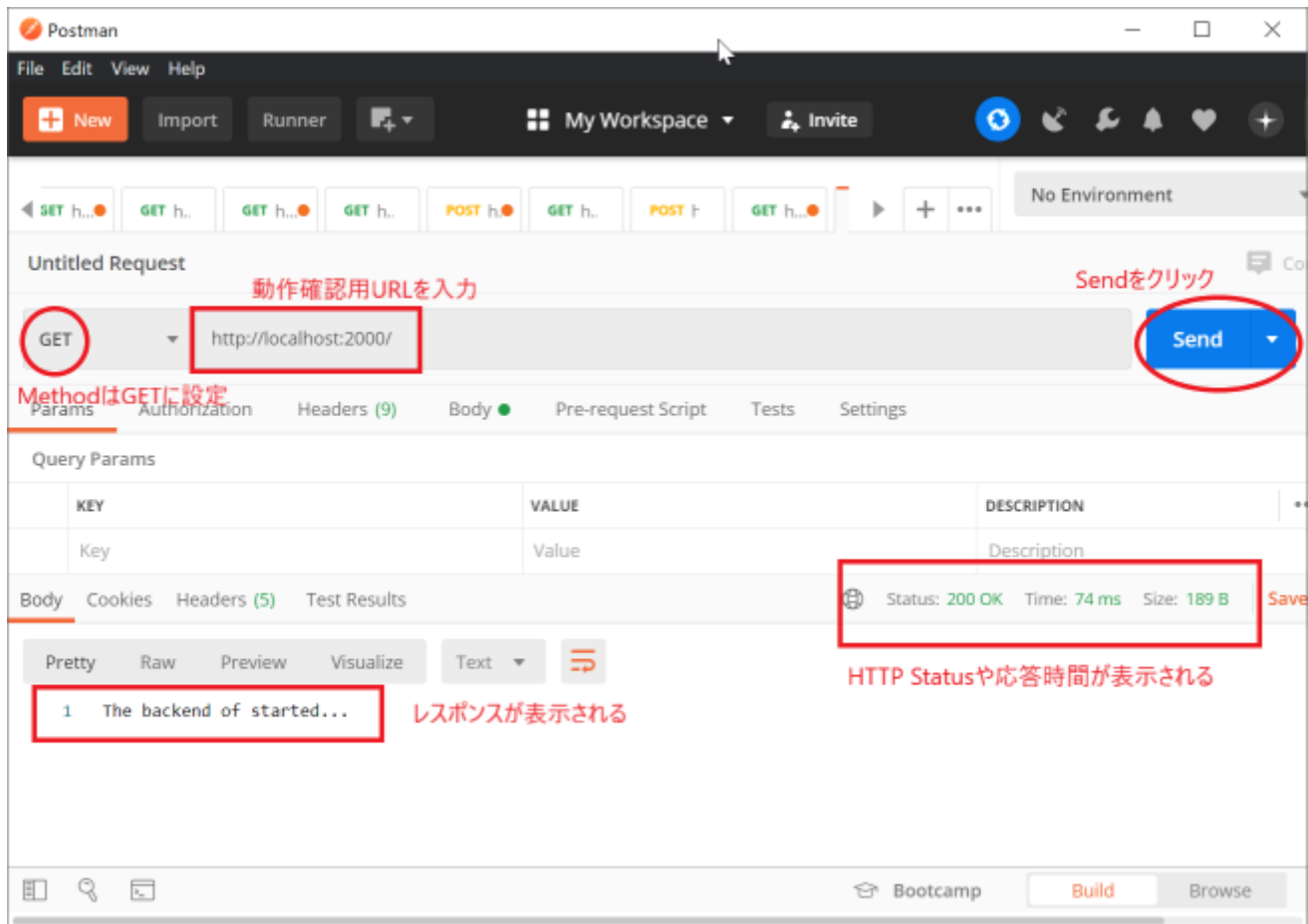
WEBブラウザより、下記URLを入力し画面表示されることを確認する。

```
http://localhost:2000/
```



Postmanで確認

Postmanをインストールしていると思いますので、そちらでも確認します。



まとめ

今回は、Spring Initializrを利用した新規プロジェクト作成から動作確認用ページ作成・動作確認まで行いました。

イニシャライザを使えば、様々なタイプのひな形が作れることが体験できたと思います。

次回より、検索機能などのバックエンドの機能を実装していきます。