目次

まえがき

以下の JDK を使用しています。

- openjdk 11.0.13 2021-10-19
- OpenJDK Runtime Environment Temurin-11.0.13+8 (build 11.0.13+8)
- OpenJDK 64-Bit Server VM Temurin-11.0.13+8 (build 11.0.13+8, mixed mode)

以下のパッケージを使用しています。

- Spring Boot 2.6.1

お気付きの点がありましたらこの原稿のリポジトリの Issues までお願いいたします。
https://github.com/CookieBox26/notes/issues

第 1 話　クッキー、Spring Boot 開発に参加する

Spring Boot (Java) による REST API 開発に参加することになりました。Python でならいくつかのやり方で REST API を実装したことがありますが、Spring Boot は触ったことがありません。そもそも Java での開発も初めてに等しいです。Java モジュールに数行の修正 PR をしたことは何度かありますが、

機能追加をするなどは経験がないので、Java における開発プラクティスの知見もまるでありません。C# での開発経験はありますが昔すぎて記憶が……。

……まあしかし、REST API であるならばコードをみれば何が何なのかは概ねわかるでしょう。今回の私のタスクは何もスクラッチで REST API を構築することではありません。現在既に「弁当」を返却しているベントウ API があり、リクエストに応じてその「弁当」に「卵焼き」を追加するだけです。外部 API から新たに「卵」を取り寄せる必要はありますが――。

ともかく適当なソースファイルをみてみましょう――どのファイルも 1 行目に

src/main/java/com/example/bentou/BentouApplication.java

```
package com.example.bentou;
```

などとありますね。これは「このソースファイル内に定義されているクラスは com.example.bentou なるパッケージに所属しています」という意味ですか。クラスを適宜パッケージに整理せよということですね。

では、HTTP リクエストを受け取るハンドラはどこにあるのでしょうか? 現在の仕様書によると /bentou なるエンドポイントに HTTP GET することで「弁当」が返却されるはずですが……あっ、このクラスでしょうか。

src/main/java/com/example/bentou/BentouController.java

```
package com.example.bentou;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class BentouController {
  private final OnigiriService service;

  @Autowired
  public BentouController(@Qualifier("tuna") OnigiriService
    service) {
    this.service = service;
  }

  @GetMapping("/bentou")
  public Onigiri provide() {
    Onigiri onigiri = this.service.provideOnigiri();
    return onigiri;
  }
}
```

@GetMapping(”/bentou”) が付加されたメソッド (Python でいうデコレータのようですが――?) がリクエストを受け取っておにぎりを返却しているようにみえます。このメソッドがリクエストハンドラとみていいでしょう。

第 2 話　クッキー、最大の謎に直面する

リクエストハンドラがわかればリクエストに応じて返却フィールドを追加する変更は実装できそうです。

……しかし、リクエストハンドラをもつ BentouController クラスがどこでインスタンス化されているのかさっぱりわかりません。リポジトリを grep してもそれらしき箇所がありません。BentouController クラスにはコンストラクタもあるようですし、やはりこれはインスタンス化して利用するものでしょう。であれば、何がどうなって――

第 3 話　クッキー、Controller の単体テストで行き詰まる

第 4 話　クッキー、RestTemplate の利用に行き詰まる

第 5 話　クッキー、Service の単体テストで行き詰まる

第 6 話　クッキー、サーキットブレーカーに嫌われる

第 7 話　クッキー、接続プールが不足する

第 8 話　クッキー、メトリクスを隠される

第 9 話　クッキーと素敵な Spring Boot

参考文献

[1] https://javadoc.io/doc/org.mockito/mockito-core/4.0.0/org/mockito/Mockito.html#10