

クッキーと素敵な Spring Boot

目次

クッキー、Spring Boot 開発に参加する	1
クッキー、謎に直面する	3
クッキー、単体テストを実装する	3
参考文献	3

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」を取り寄せる必要がありますが……。



……何にせよ何でもいいのでコードをみていきましょう。1 行目のこれは――

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

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

これは「このソースファイル内に定義したクラスは 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。クラス名を適宜パッケージ

に整理せよということですね。



では、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 リクエストすることで「おにぎり」入りの「弁当」が返却されるはずですが……あっ、ここでしょうか。

```
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;
    }
}
```

クッキー、謎に直面する

クッキー、単体テストを実装する

参考文献

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