

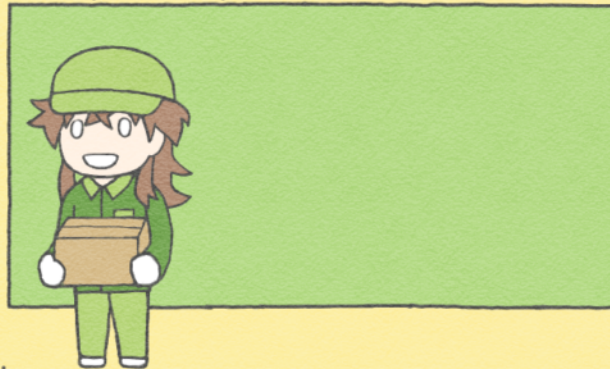
# クッキーと素敵な Spring Boot

## 未完成ドラフト

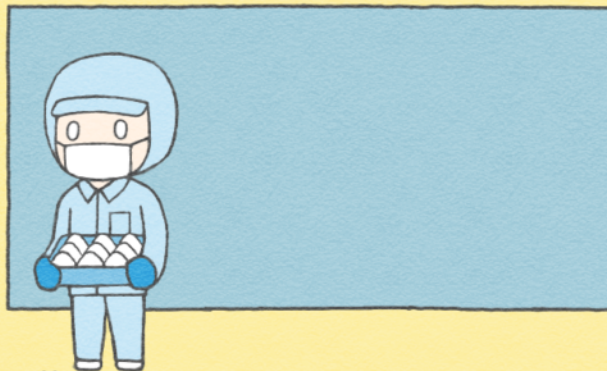
クッキー

この本は未完成ドラフトです。  
目次の内容をかくことを目指しましたが  
全然完成しませんでした。

@Repository



@Service



@Controller



## 目次

まえがき	1
第 1 話 クッキー、Spring Boot 開発に参加する	2
第 2 話 クッキー、最大の謎に直面する	3
第 3 話 クッキー、Controller の単体テストで行き詰まる	4
第 4 話 クッキー、RestTemplate の利用に行き詰まる	4
第 5 話 クッキー、Service の単体テストで行き詰まる	4
第 6 話 クッキー、サーキットブレーカーに嫌われる	5
第 7 話 クッキー、接続プールが不足する	5
第 8 話 クッキー、メトリクスを隠される	5
第 9 話 クッキーと素敵な Spring Boot	5
参考文献	6

## 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

本書の内容についてお気付きの点がありましたら、大変お手数ですが、この本のコードか原稿のリポジトリの Issues、または著者ブログのコメント欄までお知らせください。著者ブログへのコメントはただちには公開されません。非公開希望の方はその旨をお知らせください。非公開希望であって返信が必要な場合はご連絡先の明記をお願いいたします。

コードリポジトリ <https://github.com/CookieBox26/bentou-application>

原稿リポジトリ <https://github.com/CookieBox26/notes/>

- この本のディレクトリは 20211226\_spring\_boot です。

著者ブログ <https://cookie-box.hatenablog.com/>

### 登場人物 (の代理) 紹介



この人はベイズ統計部の部長ですが、この本ではクッキーの代理として登場します。この人がベイズ統計部の部長として出てくるお話は他の本を参照してください。

## 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。



……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——。



とに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 onigiriService;

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

    @GetMapping("/bentou")
    public Bentou provide() {
        Bentou bentou = new Bentou();
        bentou.onigiri = this.onigiriService.provideOnigiri();
        return bentou;
    }
}
```

`@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——



調べてみたところ、Spring Boot プロジェクトをビルドして実行すると、@SpringBootApplication 付きのクラスに対して、以下の 1. 2. のオブジェクトが勝手に生成されるのですね [3]。

1. そのクラス自身 or そのパッケージ以下で @Configuration 付きで定義されたクラスがもつ @Bean 付きのメソッドが返すオブジェクト。
2. そのパッケージ以下で @RestController, @Controller, @Service, @Repository, @Component 付きで定義されたクラスのインスタンス。
  - ・ そもそも Spring には @Component しかなく、Spring Boot ではそのクラスの役割がリクエスト窓口 / ロジック / データ操作かに応じて @Controller / @Service / @Repository を使い分けるがクラスの役割の明確化以上の意味はないらしい [1] [2]。

1. 2. から生成されたオブジェクトを **Bean** とよぶのですね。自作クラスを Bean とするときは 2. の方法、サードパーティのクラスを Bean とするときは 1. の方法をとることになるようです [4]。

### 第 3 話 クッキー、Controller の単体テストで行き詰まる



MockMvc を用いるとリクエストに対するレスポンスが期待通りかのテストを実装できそうです [5]。



しかし、このとき Bean を本番用のものと置き換えたいですね [6]。

### 第 4 話 クッキー、RestTemplate の利用に行き詰まる



外部 API のレスポンス形式が JSON のとき、XML のときにレスポンスをパースするにはどうすればいいのでしょうか。

### 第 5 話 クッキー、Service の単体テストで行き詰まる



外部 API をモックするには MockRestServiceServer が使えそうです。



## 第 6 話 クッキー、サーキットブレーカーに嫌われる



サーキットブレーカーとは、外部へのアクセスが失敗し続けている状況のときに遮断できる仕組みなのですね [7]。交通整理員さんのようなものでしょうか？ 遮断している状態は電気回路でいうと途切れてしまっている状態（解放）ですから、「オープン」なのですね。

## 第 7 話 クッキー、接続プールが不足する



要求仕様通りに実装できたと思いきや、リクエスト数を捌いてくれませんか。いったいなぜ——



`org.apache.http.impl.conn.PoolingHttpClientConnectionManager` が接続プール数をつかさどっているのですね。予め上限をゆるめた `PoolingHttpClientConnectionManager` を用意する必要があります。

## 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



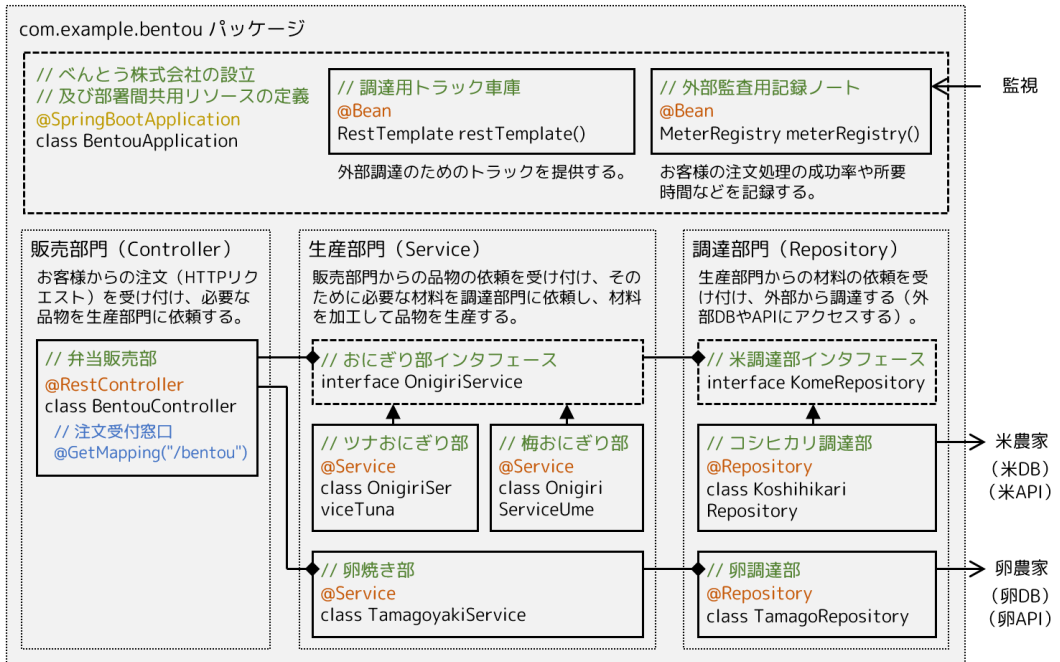
Spring Boot プロジェクトの開発とは、必要な Bean たちを用意し、適宜 Bean に他の Bean を持たせることだと思いました。



Spring Boot プロジェクトを会社に、各 Bean を部署に喩えるなら以下です。

## ★ Spring Boot 開発でやること

- まず会社を設立しよう！ = `@SpringBootApplication` 付きのクラスを定義しよう！
- 部署を組織しよう！ = 会社を設立したパッケージ下に `@(Rest)Controller/Service/Repository` 付きのクラスを定義しよう！
  - 部署は販売部門、生産部門、調達部門に整理するのがマナー！ Ex. 販売部門ならメンバにもつ部は生産部門のみにしよう！
- 全社共用リソースが必要なら会社とその生成メソッドを `@Bean` 付きで定義するか、`@Component` 付きのクラスを定義しよう！
- Spring Boot は会社/パッケージ下の `@(Rest)Controller/Service/Repository/Component` 付きのクラスのインスタンスや会社の `@Bean` メソッドの返り値を勝手に生成するので、`@Autowired` を駆使してそれらの間に依存関係 (has-a 関係) をもたせよう！



## 参考文献

- [1] Spring Boot アプリケーションのコードレビューポイント - Qiita, <https://qiita.com/cross-xross/items/144f8bde2ef6fa4b379f>, 参照日：2022 年 1 月 30 日.
- [2] 【Spring】@Autowired と @Component を使用した DI の基本 - 山崎屋の技術メモ, <https://www.shookuro.com/entry/2016/08/09/175801>, 参照日：2022 年 1 月 30 日.
- [3] Spring Boot での開発 - リファレンスドキュメント #6. @SpringBootApplication アノテーションの使用, <https://spring.pleiades.io/spring-boot/docs/current/reference/html/using.html#using.using-the-springbootapplication-annotation>, 参照日：2022 年 1 月 30 日.
- [4] Spring Boot の @Component と @Bean の違い - grep Tips \*, <https://www.greptips.com/posts/1318/>, 参照日：2022 年 1 月 30 日.
- [5] Spring Boot MockMvc と @MockBean で Web レイヤーテスト - 公式サンプルコード, <https://spring.pleiades.io/guides/gs/testing-web/>, 参照日：2022 年 1 月 30 日.
- [6] 雑記：テスト時に Bean を別の Bean に置き換えたい話 - クッキーの日記, <https://cookie-box.hatenablog.com/entry/2021/12/05/124520>, 参照日：2022 年 1 月 30 日.

[7] 雑記 : Circuit Breaker をオープンさせるだけ (resilience4j-spring-boot2) - クッキーの日記, <https://cookie-box.hatenablog.com/entry/2021/12/13/040728>, 参照日 : 2022 年 1 月 30 日.



## クッキーと素敵な Spring Boot

未完成ドラフト

---

2021 年 1 月 30 日 初版発行

著 者 クッキー

発行者 クッキーの日記

<https://cookie-box.hatenablog.com/>

---