

Spring Boot を使った Web アプリケーション開発

福井清悟

工学系技術支援室 情報通信技術系

はじめに

現在情報通信技術系で管理しているサーバの中には Web アプリケーションが稼働しているものが多く存在している。業務依頼としては不具合の原因調査やプログラムの修正等既存アプリケーションに対するものが多く、それらに加えて新規開発依頼への対応も求められている。これらの案件に対して情報通信技術系では個々に対応することが多く、アプリケーションの作り方が人によって異なっていた。そこで、開発手法の統一の観点から Java のフレームワークである Spring Boot を用いた Web アプリケーション開発技術の習得を目的として本個別研修を実施した。本報告書では Spring Boot について説明した後、実際に作成した機能を紹介し、最後にまとめを報告する。

1. Spring Boot について

1.1 言語とフレームワークの選択について

本研修では Web アプリケーションを開発するにあたって、プログラミング言語として Java、フレームワークとして Spring Boot を用いた。プログラミング言語には数多くの種類があり、Web アプリケーション開発でよく使われるものとしては Java、PHP、Ruby などがある。また、フレームワークについても言語ごとに多くのものがある。言語とフレームワークの組み合わせは膨大になり、開発者はそれぞれ好きな言語・フレームワークを使って開発を行うことができる。しかし、複数人体制で行うシステムの管理・運用においては、言語・フレームワークの選択を個人の好みに任せていると、長期的な保守性の観点で問題が発生する。例えばある開発者が、組織内でその人しか使っていない言語・フレームワークで開発した場合、そのアプリケーションを引き継ぐためには一からその言語・フレームワークを学習しなければならない。こういった事態を避けるために言語・フレームワーク選択で大切なのは、「組織で統一する」ことである。この考えに基づいた上で、情報通信技術系では今後、開発言語は Java、フレームワークは Spring Boot (Spring Framework) で統一するという方針なので、今回はそれらを使った開発方法について研鑽した。

1.2 フレームワークについて

フレームワークとは、「アプリケーションを開発するために必須なものではないが、使うと開発が楽に早く進められるようになるもの」である。また、framework を日本語にすると「枠組み」という単語になるように、フレームワークを使うということはある程度決まったルールに従ってプログラムを書く、ということにもなる。したがって、フレームワークを使うメリットとして、開発者以外がソースコードを見たときに「このフレームワークを使っているからあの書き方だろう」という目処が立ちやすくなるという点がある。これは業務引き継ぎが必要になったときに、引き継いだ者にかかる負担が少なくなることにつながると思われる。一方で、

デメリットとしては、フレームワークの使い方を学習するために時間がかかるということ、小規模なアプリケーションであればフレームワークを使わない方が簡単に開発できる場合もあることなどが挙げられる。このようにフレームワークを使う場合は、本当にフレームワークが必要か考慮する必要もある。

1.3 Spring Boot とは

Spring Boot とは Web アプリケーション開発のための Java のフレームワークの一つである。Spring Boot よりも以前に Spring Framework (以下、Spring) というフレームワークがあり、Spring Boot は Spring を使用したアプリケーション開発を容易にするために登場した。Spring Boot の基盤には Spring があるため、Spring Boot を使う本研修では Spring についての学習から始めた。また、Spring を使えば Web アプリケーションは開発でき、必ずしも Spring Boot をセットで使わなければならないわけではない。しかし、Spring 単体使用と比較した場合の Spring Boot のメリットとして、XML を書かなくてよい、依存モジュールの管理がしやすい、単体実行可能な jar ファイルが作成できる、などのメリットがある。そのため、Spring を使うならば Spring Boot も用いるのがよいと考える。

2. 実装機能

本研修ではログイン認証、アクセス制限、フォーム入力による情報登録、操作ログの記録、例外処理、定期実行処理、メール送信などの機能を実装した。これらの機能は一般的な Web アプリケーションとしてはごく基本的な機能であり、通常はこういった機能に加えて各アプリケーションの目的に特化した機能を加えることになる。

具体例としてフォーム入力による情報登録機能について説明する。「お知らせ登録」画面でお知らせ情報を登録し、「確認画面」で内容を確認するという単純な機能である(図2、図3)。

お知らせ登録

タイトル [必須]
技術部研修会

登録者氏名 [必須]
山田

内容 [必須]
平成28年度工学部技術部研修会を以下の通り開催します。
日時：12月13日(火)
場所：VBL3階

添付ファイル
サイズは1MB以下のみ
[ファイルを選択] 選択されていません

掲載終了日 [必須]
半角でyyyy-MM-ddという形式で入力してください。
例：2016年12月1日 → 2016-12-01
2016-12-13

登録

図 2. お知らせ登録画面

登録内容確認

以下の内容でお知らせを登録しました。

登録日	2016年11月17日
タイトル	技術部研修会
内容	平成28年度工学部技術部研修会を以下の通り開催します。 日時：12月13日(火) 場所：VBL3階
添付ファイル	
登録者	山田
掲載終了日	2016年12月13日

お知らせ一覧 メニュー ホーム

図 3. 確認画面

Spring Boot を使った Web アプリケーションは Spring MVC というデザインパターンに従うため、ここでもモデル、ビュー、コントローラーを作ることになる。「お知らせ登録機能」に関する処理の流れのイメージを図4に示す。

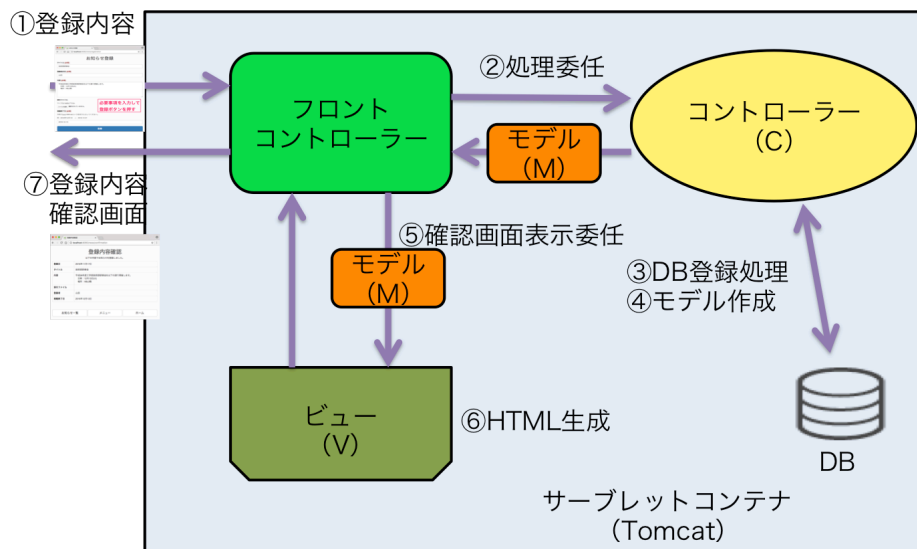


図 4. お知らせ登録処理

実際に「お知らせ登録機能」を実装するためには以下の手順が必要になる。

- 1) お知らせ情報保存用データベースを準備する
- 2) テーブルデータをマッピングしたクラス（モデル）を作る
- 3) POSTされたデータを処理するコントローラーを作る
- 4) 確認画面を出力するビューを作る

本報告書では上記手順 1)と 2)については省略し、3)と 4)について説明する。具体的には NewsController.java というコントローラー、confirm.html というビューを作成した。コントローラーとビューのソースコードの抜粋を図 1、図 2 に示す。

```

1 @Controller
2 public class NewsController {
3     @RequestMapping(value="/registration", method=RequestMethod.POST)
4     public String newsAdd (@ModelAttribute News news) {
5         /* DB への登録処理 */
6         return "confirm";
7     }
8 }
  
```

図 1. コントローラーのソースコード（抜粋）

```

1 <!DOCTYPE html>
2 <html lang="ja" xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>登録内容確認</title>
  
```

5	</head>
6	<body>
7	<h1 class="text-center">登録内容確認</h1>
8	<p class="text-center">以下の内容でお知らせを登録しました。</p>
9	<table class="table table-bordered">
10	<tr>
11	<th>タイトル</th>
12	<td th:text="{news.title}">title</td>
13	</tr>
14	</table>
15	</body>
16	</html>

図 2. ビューのソースコード（抜粋）

まず、コントローラーについて、図 1 で「@～」という記述が 3 箇所あるが、これが **Spring Boot** の特徴的な書き方である。**Spring** などと同じ実装をする場合 XML ファイルを別途作成する必要がある。大規模な Web アプリケーションになると XML ファイルの作成・管理が煩雑になり、初学者には一つの壁となる。**Spring Boot** では基本的に XML ファイルを記述する必要はない。

続いて、図 s2 ではコントローラーから「news」という名前のモデルを受け取り、その内容を出力している。例えば 12 行目はお知らせのタイトルを出力する行であり、「th:text="{news.title}"」の部分が「『news』というモデルの『title』という属性に代入されている値を表示する」という意味になっている。このようにしてモデルの中身に応じて動的に HTML を生成している。

3. まとめ

本研修を通じて、**Spring Boot** を使用すると簡単に Web アプリケーションが作成できることが分かった。今回実装した機能は Web アプリケーションとしては基本的なものだが、多くのアプリケーションで共通して使われるものでもあるため、一度実装したという経験ができたのは重要であった。一方で、実運用を考えると機能的には乏しい点が多い。例えば、本報告書で紹介した「お知らせ登録機能」については、確認画面で入力内容の修正ができるようにした方がよい。このように、今回実装した機能にはまだまだ改善の余地が残されている。

Spring Boot を使用すると開発が簡単になる一方で、フレームワーク内部の仕組みを理解することが必要であると感じた。フレームワークは非常に便利なものであり、また、インターネットで検索すればコードの書き方などはすぐに見つかる。しかし、フレームワーク自体について理解していないと、不具合が生じた時に原因が見つけられないという事態に陥る危険性がある。また、そもそも Web アプリケーションについて、Java サーブレットについて、HTTP プロトコルについてなど、基本要素からの理解も必要になると感じた。

今後とも研鑽を続け、今回感じた課題を解決し、最終的には業務に活かせる技術を習得したいと考えている。