

SpringMVC 終了後テスト

テストの注意点

- ・所要時間は40分です。問題3問、ボーナス問題1問です。
- ・IntelliJを使用してください。
- ・SpringMVCとThymeleafが使用できる「1-1.test-spring-mvc」というプロジェクトを「[Spring Initializr](#)」を使用して作成してください。
- ・テスト中は**以下のサイトのみ閲覧可能**です。
 - <https://docs.spring.io/spring/docs/current/spring-framework-reference/>
 - <http://docs.spring.io/spring/docs/current/javadoc-api/>
 - <http://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
 - <https://www.thymeleaf.org/apidocs/thymeleaf/3.0.11.RELEASE/>
 - https://www.thymeleaf.org/doc/tutorials/2.1/usingthymeleaf_ja.html
- ・問題の意図、意味がわからなければ講師に質問してください。
- ・途中プロジェクトがうまく動かない＆何やっても404エラーから直せないような場合は講師を呼んでください

Ver.202504

作成者：伊賀 将之

前提

- 今回のテストは以下のThymeleafのテンプレートを使用します

```
<!DOCTYPE html>
<html lang="ja" xmlns:th="http://www.thymeleaf.org">
<head>
<!-- ヘッド要素 -->
<meta charset="UTF-8">
<title>タイトル</title>
</head>
<body>
<!-- ボディ要素 -->
</body>
</html>
```

第1問：(配点20点) フォワード

- [exam-01.html]という名前のViewを適切な場所に作成してください。このHTMLファイルには、自分の名前と趣味をブラウザに表示させる内容を記述してください(HTMLに直接書いてしまってOKです)
- [Exam01Controller.java]という名前のコントローラを作成してください。このクラスに先ほど作成した[exam-01.html]に**フォワード**させる `showSelfIntroduction()`メソッドを作成してください
- また、動作確認用のURLは以下になるように@RequestMappingや@GetMappingを指定してください
<http://localhost:8080/exam01/show-self-introduction>
- IntelliJから起動し、Webブラウザを立ち上げて動作確認をしてください
- 自分の名前と趣味が表示されることを確認してください

表示結果例

名前：山田太郎
趣味：サイクリング

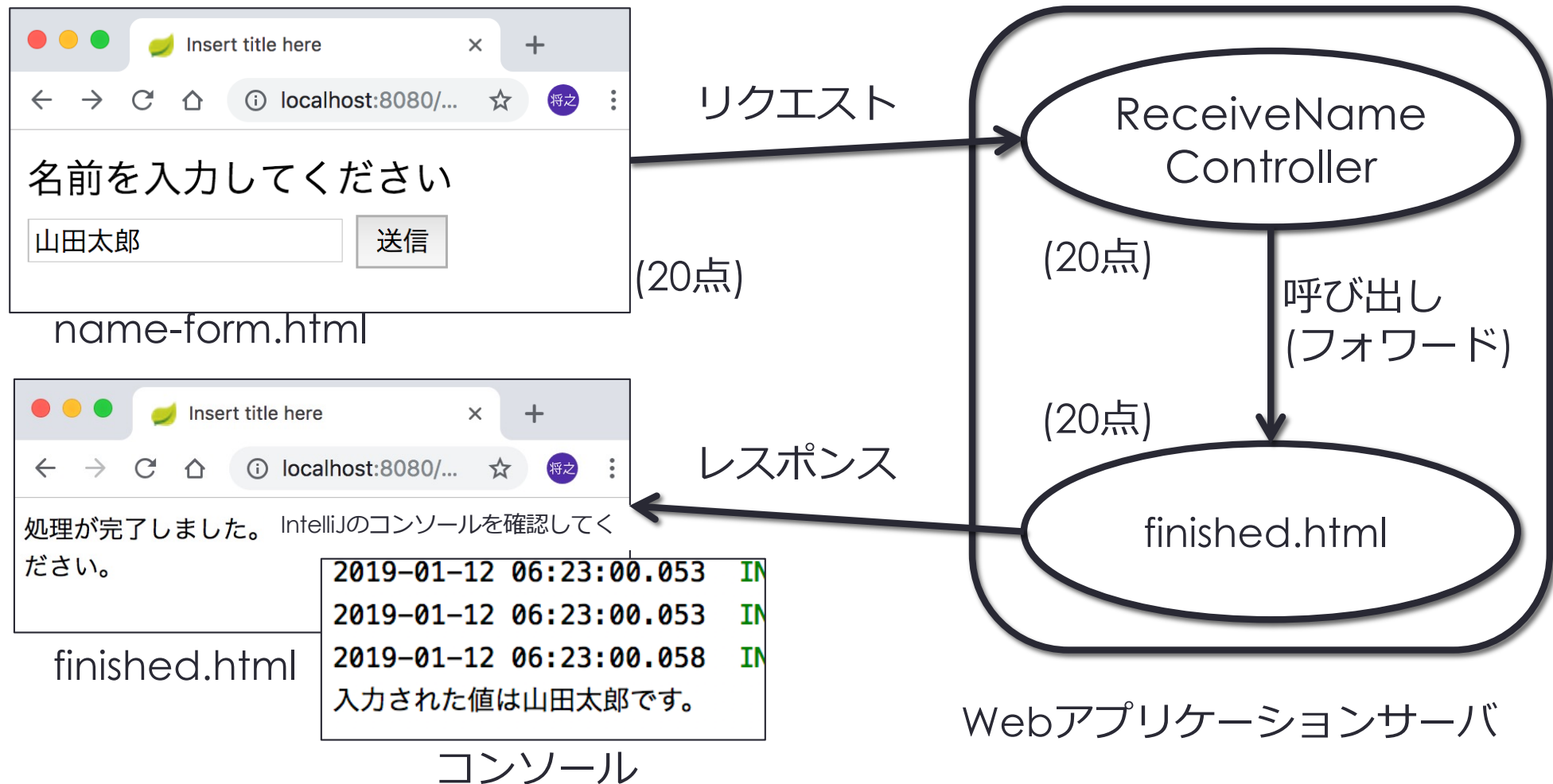
第2問：(配点20点)リダイレクト

- [exam-02.html]という名前のViewを適切な場所に作成してください。このHTMLファイルには、自分の所属している会社名と部署をブラウザに表示させる内容を記述してください (HTMLに直接書いてしまってOKです)
- [Exam02Controller.java]という名前のコントローラを作成してください
- 今作成したクラスに先ほど作成した[exam-02.html]に**リダイレクト**させる showCompanyIntroduction()メソッドを作成してください。この時、必要であれば showCompanyIntroduction2()メソッドも合わせて作成してください
- また、動作確認用のURLは以下になるように@RequestMappingや@GetMappingを指定してください
<http://localhost:8080/exam02/show-company-introduction>
- IntelliJから起動し、Webブラウザを立ち上げて動作確認をしてください 表示結果例
- 会社の名前と部署が表示されることを確認してください

会社名：株式会社〇〇 部署：開発部

第3問：(配点60点)リクエストパラメータ

- 以下のアプリケーションを作成してください。この際使用するURL(パス)は任意で構いません



ボーナス問題：(配点20点)フォームクラス

- 以下の手順で、問3のリクエストパラメータをフォームで受け取るようにします
 - リクエストパラメータ受け取り用クラスであるReceiveNameFormクラスを作成してください
 - ReceiveNameControllerを、Formを使用するように書き換えてください(この時、問3で作成したメソッドはそのままにして、Formを受け取る新しいメソッドを作成してください)
 - name-form.htmlのth:action属性は今回作成したメソッドを呼ぶように変更して動作確認を行ってください
 - ※コントローラのindex()メソッドに引数を受け取る方法、ビューのth:object,th:fieldは書いても書かなくてもどちらでも良いです(今回は動作すればOK)