



# Spring 補足資料

隠しコントロールによる値渡し

# 目次

第1章 隠しコントロールによる値渡し .....	2
登録・更新・削除と3画面構成 .....	3

## 第1章 隠しコントロールによる値渡し

本資料では、Spring の補助資料として、登録、更新、削除用の画面作成にて役立つ、隠しコントロール (hidden) による値渡しの方法について紹介します。  
開発演習を進めるにあたっての参考資料としてご活用ください。

なお、本資料用にサンプルコード「shop\_advance」プロジェクトも用意しました。併せてご利用ください。

## 登録・更新・削除と3画面構成

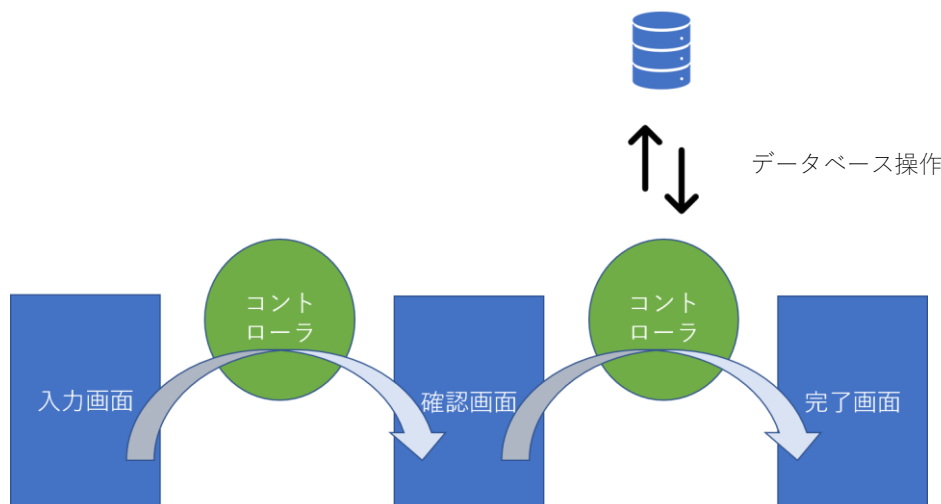
### 1. 隠しコントロール (hidden) による値渡し

講義では、レコードの登録・更新・削除などの更新系処理を実現するにあたり、「入力画面」や「完了画面」を作成しました。

しかし、実務レベルの Web アプリケーションでは、ユーザーの「1 クリック」でレコードの更新系処理を完了させる画面構成は避けられることが多いです。理由としては、ユーザーの誤入力があった際に取り消しができないためです。

そこでよく利用されるのが、更新系処理を実行する前に入力内容を確認するための「確認画面」です。更新系処理では、「入力→確認→完了」の3ステップで画面遷移するように実装することで、ユーザーによる誤入力の頻度を抑えられます。

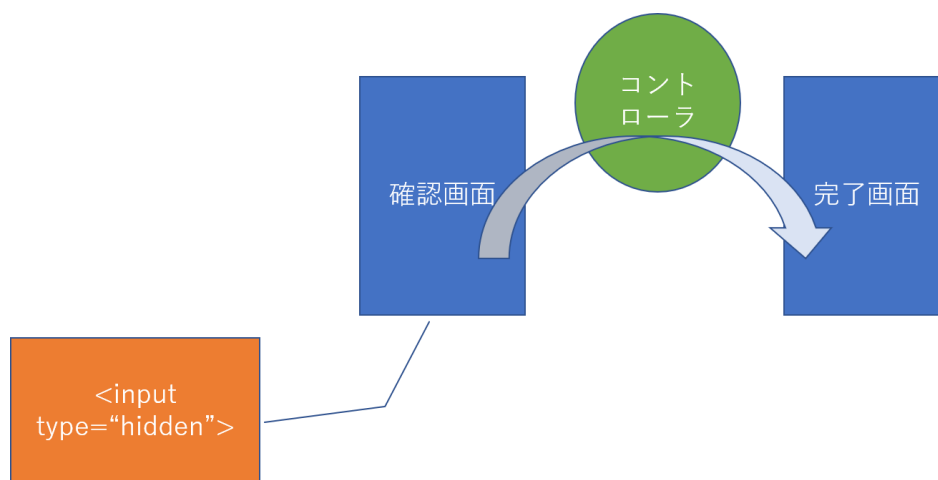
3ステップの画面遷移を実現する際には、下図のように3画面を介した処理を実装します（削除の場合、入力画面がないケースがあります。）。



データベースへの更新系処理を実行するタイミングは、左から2番目のコントローラを動かす際です。

しかし、「入力画面」で入力した値を、2番目のコントローラまで渡すには、少なくとも2回のリクエストを介さなければなりません。しかし、リクエストスコープに格納されているパラメータは1回目の画面遷移の時点では消えてしまうため使用できません。

そこで確認画面の<input>タグ内で「hidden」属性を利用し値を送信するという手法を取ります。hidden 属性が付与されたパラメータは、「隠しパラメータ」として値を送ることができます。



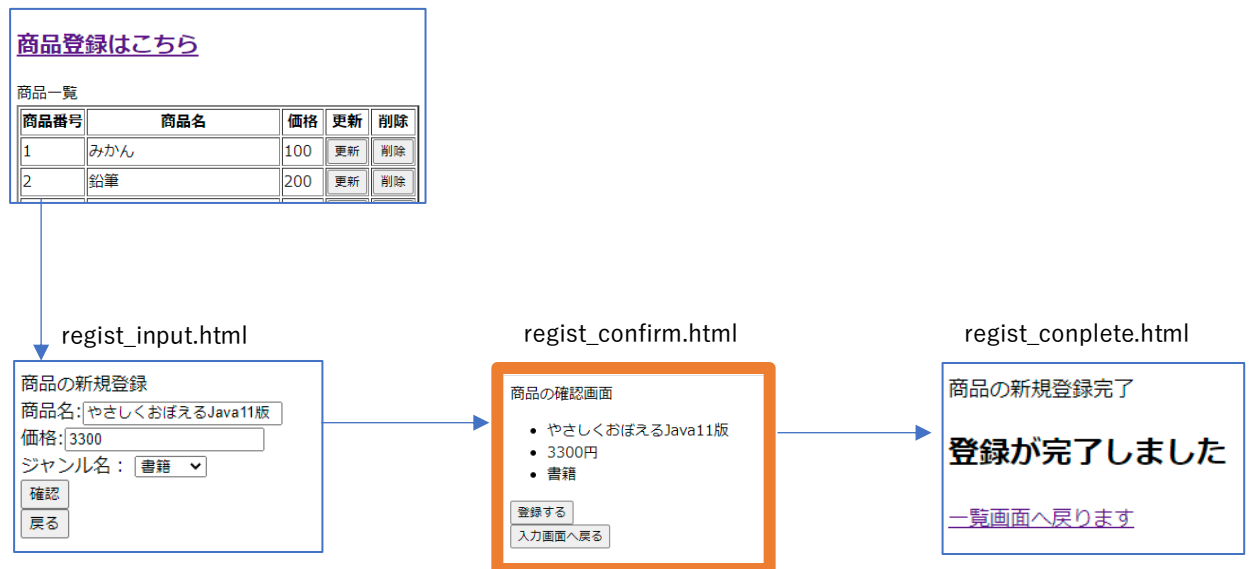
hidden とともに使用されることの多い属性

属性名	説明
value	送信するパラメータを指定する
name	送信するパラメータ名 コントローラ側は ・対応する変数をハンドラメソッドの引数に指定 ・対応するプロパティを持つ Form クラスをハンドラメソッドの引数に指定 のいずれかでパラメータを受け取ることができる

## 2. 【サンプルコード】登録機能

※「shop\_advance」起動後、「http://localhost:7770/shop/items/findAll」にアクセス

【画面構成】



【regist\_confir.html】

```
<body>
  <div>商品の確認画面</div>
  <div th:object="${itemForm}">

    ~~~中略~~~

    <form method="post" th:action="@{/regist/create-item}" onClick="disableBtn(this);">
      <!-- hidden で値を送信する。itemForm の各フィールドに値がバインドされる -->
      <input type="hidden" name="name" th:value="${name}" />
      <input type="hidden" name="price" th:value="${price}" />
      <input type="hidden" name="categoryId" th:value="${categoryId}" />
      <input type="submit" value="登録する" />
    </form>

    ~~~中略~~~

  </div>
</body>
```

## 【ItemRegistController】

```
@Controller
public class ItemRegistController {

    ~~~中略~~~

    //確認画面表示
    @RequestMapping("/regist/confirm")
    public String confirmItem(@Valid @ModelAttribute ItemForm form, BindingResult result) {
        if (result.hasErrors()) {
            System.out.println("back:" + form);
            return inputItem(form); //入力エラー時
        } else {
            System.out.println("check:" + form);
            return "items/regist_confirm"; //確認画面遷移
        }
    }

    //登録処理
    //二重登録防止のため PRG パターンでの処理
    @RequestMapping(path = "/regist/create-item", method = RequestMethod.POST)
    public String exeRegistItem(ItemForm form) {

        //エンティティ生成
        Item item = new Item();
        item.setName(form.getName());
        item.setPrice(form.getPrice());
        Category category = cateRepo.getReferenceById(form.getCategoryId());
        item.setCategory(category);

        //登録
        itemRepo.save(item);

        return "redirect:/regist/complete";
    }

    //登録完了画面表示
    @RequestMapping("/regist/complete")
    public String completeRegistItem() {
        return "items/regist_complete";
    }
}
```

hidden で送られたパラメータをバインド

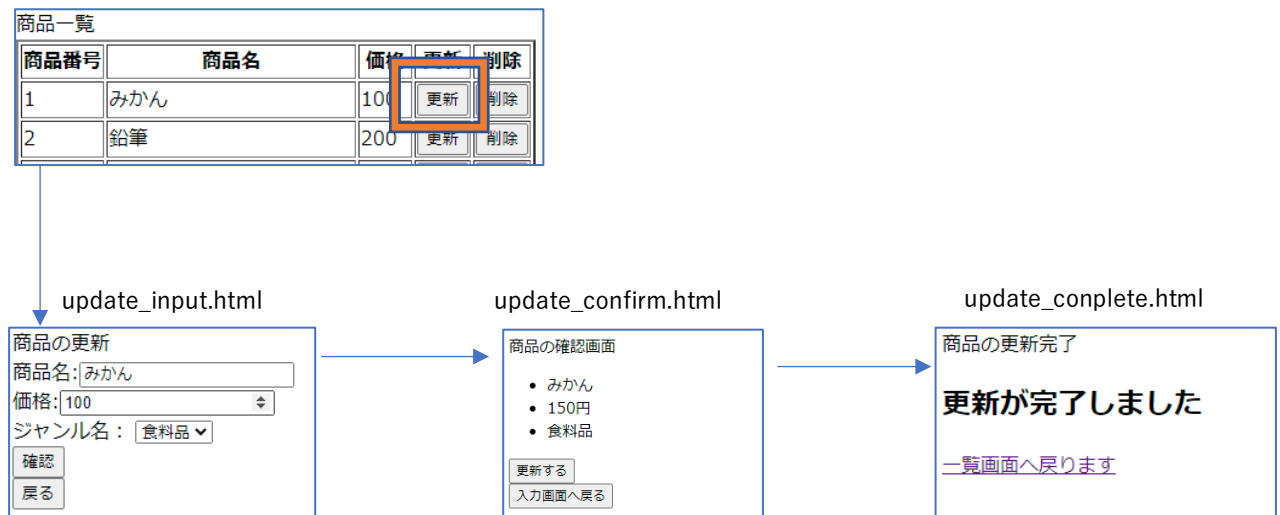


### 3. 【問題】 更新機能・削除機能

以下の要件・画面遷移に沿って実行できるよう実装しましょう

#### 更新機能

##### 【画面遷移】



「shop\_advance」プロジェクト内の以下のファイル（編集対象）を編集し更新機能を実装してください

編集対象のファイル

update\_confirm.html

jp.co.sss.shop.controller.ItemUpdateController.java

定義済みのファイル

update\_input.html

update\_complete.html

削除機能

## 【画面遷移】



「shop\_advance」プロジェクト内の以下のファイル（編集対象）を編集し削除機能を実装してください。

編集対象のファイル

delete\_confirm.html

jp.co.sss.shop.controller.ItemDeleteController.java

定義済みのファイル

delete\_complete.html



Spring 補助資料 第1版

---

2021年06月01日 初版 発行

---

著者 株式会社 SystemShared 東京 IT スクール

---

文中の社名、商品名等は各社の商標または登録商標である場合があります。