

SP60 課題

課題内容

次の仕様を満たしたプログラムを作成しなさい

- 商品管理機能

プロジェクト構成

- プロジェクト名：spring-kadai6-jpa-crud
- 依存関係：Spring Web / Thymeleaf / Spring Data JPA / PostgreSQL Driver / Spring Dev Tools

**Step0：（準備）プロジェクトを作成後、「application.properties」
「schema.sql」 「data.sql」 をサンプルプロジェクトの「spring-chap
7-jpa-crud」よりコピーしてきます**

Step1：商品一覧画面を作りなさい（復習）

- 出力結果



商品ID	商品名	値段
1	Javaの基本	2500
2	MLB Fun	980
3	料理BOOK!	1200
4	なつかしのアニメシリーズ	2000
5	The Racer	1000
6	Space Wars 3	1800
7	パズルゲーム	780
8	Invader Fighter	3400
9	Play the BascketBall	2200

- ファイル構成

```
spring-kadai6-jpa-crud
├─src/main/java
│   └─com.example.demo
│       ├──controller
│       │   └─ItemController.java (★新規)
│       ├──entity
│       │   └─Item.java (★新規)
│       └─repository
│           └─ItemRepository.java (★新規)
└─src/main/resources
    ├──templates
    │   └─items.html (★新規)
    ├──application.properties (★spring-chap7-jpa-crudよりコピー)
    ├──data.sql (★spring-chap7-jpa-crudよりコピー)
    └─schema.sql (★spring-chap7-jpa-crudよりコピー)
```

- 仕様

HTTP メソッド	URLパス	パラメータ	メソッド	内容
GET	/items	なし	ItemController.index()	初期表示

Step2：新規登録の処理を追加しなさい

- 出力結果

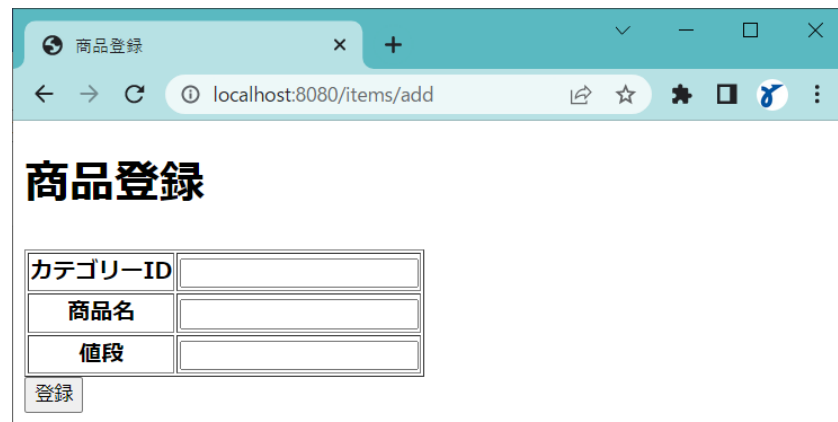
一覧画面に新規登録リンクを追加



商品ID	商品名	値段
1	Javaの基本	2500
2	MLB Fun	980
3	料理BOOK!	1200
4	なつかしのアニメシリーズ	2000
5	The Racer	1000
6	Space Wars 3	1800
7	パズルゲーム	780
8	Invader Fighter	3400
9	Play the BascketBall	2200

[新規登録](#)

新規登録画面の表示（初期）

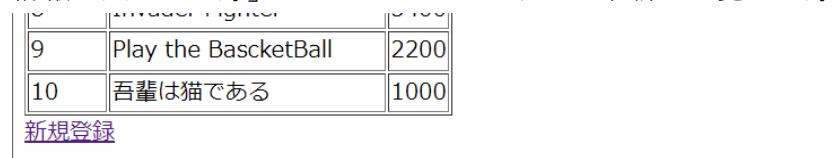


カテゴリID

商品名

値段

情報を入れ「登録」ボタンをクリックすると最新の一覧に登録した商品の情報が追加される



9	Play the BascketBall	2200
10	吾輩は猫である	1000

[新規登録](#)

● ファイル構成

```
spring-kadai6-jpa-crud
├─src/main/java
│   └─com.example.demo
│       └─controller
│           └─ItemController.java
│       └─entity
│           └─Item.java
│       └─repository
│           └─ItemRepository.java
└─src/main/resources
    └─templates
        ├──addItems.html (★新規)
        └─items.html
```

● 仕様

HTTP メソッド	URLパス	パラメータ	メソッド	内容
GET	/items	なし	ItemController.index()	初期表示
GET	/items/add	なし	ItemController.create()	新規登録画面表示
POST	/items/add	categoryId name price	ItemController.store()	新規登録処理（登録ボタンクリック）

- 取得したパラメータでitemsテーブルにデータを登録し、最新の一覧を再描画する (/itemsへのリダイレクト)

Step3：一覧画面に更新ボタンを追加し、更新画面の表示をしてください（更新処理は次のStepで実装します）

- 出力結果

一覧画面（更新ボタンを追加）



更新画面表示（DBの登録情報がテキストボックスに出力される）



商品IDはテキストボックスではありません（編集不可）

このStepでは、更新ボタンを押したときの処理はまだ不要です。

● 仕様

HTTP メソッド	URLパス	パラメータ	メソッド	内容
GET	/items	なし	ItemController.index()	初期表示
GET	/items/add	なし	ItemController.create()	新規登録画面表示
POST	/items/add	categoryId name price	ItemController.store()	新規登録処理（登録ボタンクリック）
GET	/items/{id}/edit	id (URL)	ItemController.edit()	指定した商品IDの更新画面を表示

- 一覧画面に追加する更新ボタンのリクエスト先
 - ☆ [GET] /items/{id}/edit (※idは商品ID)
- 更新画面の更新ボタンのリクエスト先
 - ☆ [POST] /items/{id}/edit

Step4：更新画面にて、情報を変更し更新ボタンをクリックしたとき、データベースの内容を更新し、最新の情報で一覧画面を表示しなさい

● 出力結果

更新画面（情報を変更）

商品ID	10
カテゴリーID	2
商品名	吾輩は犬である
値段	2000

更新

更新ボタンをクリック

商品ID	商品名	値段	更新
1	Javaの基本	2500	更新
2	MLB Fun	980	更新
3	料理BOOK!	1200	更新
4	なつかしのアニメシリーズ	2000	更新
5	The Racer	1000	更新
6	Space Wars 3	1800	更新
7	パズルゲーム	780	更新
8	Invader Fighter	3400	更新
9	Play the BasketBall	2200	更新
10	吾輩は犬である	2000	更新

[新規登録](#)

更新された最新の情報で一覧画面が表示される（リダイレクト）

● 仕様

HTTP メソッド	URLパス	パラメータ	メソッド	内容
GET	/items	なし	ItemController.index()	初期表示
GET	/items/add	なし	ItemController.create()	新規登録画面表示
POST	/items/add	categoryId name price	ItemController.store()	新規登録処理（登録ボタンクリック）
GET	/items/{id}/edit	id (URL)	ItemController.edit()	指定した商品IDの更新画面を表示
POST	/items/{id}/edit	id (URL) categoryId name price	ItemController.update()	更新処理（更新画面で更新ボタンをクリック）

➤ 更新画面の更新ボタンのリクエスト先

✧ [POST] /items/{id}/edit

Step5：削除機能を追加してください

- 出力結果

一覧画面に削除ボタンを追加



商品ID	商品名	値段	更新	削除
1	Javaの基本	2500	更新	削除
2	MLB Fun	980	更新	削除
3	料理BOOK!	1200	更新	削除
4	なつかしのアニメシリーズ	2000	更新	削除
5	The Racer	1000	更新	削除
6	Space Wars 3	1800	更新	削除
7	パズルゲーム	780	更新	削除
8	Invader Fighter	3400	更新	削除
9	Play the BascketBall	2200	更新	削除
10	吾輩は犬である	2000	更新	削除

[新規登録](#)

削除ボタンをクリックすると対象の商品が削除され一覧が更新される（商品ID：10を削除）



商品ID	商品名	値段	更新	削除
1	Javaの基本	2500	更新	削除
2	MLB Fun	980	更新	削除
3	料理BOOK!	1200	更新	削除
4	なつかしのアニメシリーズ	2000	更新	削除
5	The Racer	1000	更新	削除
6	Space Wars 3	1800	更新	削除
7	パズルゲーム	780	更新	削除
8	Invader Fighter	3400	更新	削除
9	Play the BascketBall	2200	更新	削除

[新規登録](#)

● 仕様

HTTP メソッド	URLパス	パラメータ	メソッド	内容
GET	/items	なし	ItemController.index()	初期表示
GET	/items/add	なし	ItemController.create()	新規登録画面表示
POST	/items/add	categoryId name price	ItemController.store()	新規登録処理 (登録ボタンクリック)
GET	/items/{id}/edit	id (URL)	ItemController.edit()	指定した商品ID の更新画面を表示
POST	/items/{id}/edit	id (URL) categoryId name price	ItemController.update()	更新処理 (更新 画面で更新ボタン をクリック)
POST	/items/{id}/delete	id (URL)	ItemController.delete()	削除処理 (削除 ボタンをクリック)

- 削除ボタンのリクエスト先
 - ✧ [POST] /items/{id}/delete

評価

- 実施できたStep数を自己申告してください