

データベース連携 in Spring Boot

Introduction

JpaRepositoryを用いて、Javaオブジェクトとデータベースを連携させるための資料である。

JpaRepositoryを用いることにより、開発者はSQL文を記述せずに、データベースとの連携を行うことが出来る。

JpaRepositoryはSQL文を自動生成する機能が存在する。

注意：

本資料は、Macで作成している。

そのためコマンドに関しては、必要に応じてWindowsやLinuxに読み替えて欲しい。

目次

1, DB環境構築

2, DBから情報取得

3, DBにデータ追加

4, DBのデータ変更

5, DBのデータ削除

1, DB環境構築

1, DB環境構築

1-1, MySQLのインストール

Windowsの場合

1, インストーラーをダウンロード

<https://dev.mysql.com/downloads/installer/>

※64 bit環境でも上記のインストーラーで問題ない。

2, インストーラーの実行

※パスワード設定を求められた場合は、「pwd」を設定。

3, 環境変数を設定

Macの場合

下記のコマンドでインストール

```
brew install mysql
```

Linuxの場合

下記のコマンドでインストール

```
sudo apt install mysql-server
```

※パスワード設定を求められた場合は、「pwd」を設定。

1, DB環境構築

1-2, MySQL環境構築

MySQL_env.txtを参考に、下記のコマンドを用いてDB環境構築を行う。

※緑字は通常のコマンド。青字はMySQL内でのコマンド。

1, MySQLを起動する。

`MySQL.server start`

2, MySQLにログインする。

`mysql -uroot`

3, パスワードを「pwd」に変更する。

`SET PASSWORD FOR 'root'@'localhost' = 'pwd';`

4, testdbデータベースを作成する。

`CREATE DATABASE testdb;`

5, testdbデータベースに切り替える。

`USE testdb;`

1, DB環境構築

1-2, MySQL環境構築

6, cuntomer_tableテーブルを作成する。

```
CREATE TABLE `customer_table` (  
  `cus_Num` bigint(20) NOT NULL AUTO_INCREMENT,  
  `cus_Name` varchar(255) DEFAULT NULL,  
  `cus_Address` varchar(255) DEFAULT NULL,  
  `cus_Telno` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`Cus_Num`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
```

※コマンドは上記をコピー&ペーストする。

1, DB環境構築

1-2, MySQL環境構築

7, customer_tableテーブルに情報をインサートする。

```
INSERT INTO customer_table  
  (`cus_Name`, `cus_Address`, `cus_Telno`)  
VALUES  
  ('Alex', 'California', 00000),  
  ('Dana', 'Florida', 11111),  
  ('Jamie', 'Iowa', 22222),  
  ('Morgan', 'Kansas', 33333),  
  ('Robin', 'Idaho', 44444),  
  ('Terry', 'Alaska', 55555),  
  ('Chris', 'Texas', 66666),  
  ('Pat', 'Vermont', 77777),  
  ('Tracey', 'Wisconsin', 88888),  
  ('Ronnie', 'Utah', 99999);
```

※コマンドは上記をコピー&ペーストする。

1, DB環境構築

1-2, MySQL環境構築

8, customer_tableテーブル情報を確認のため表示する。

`SELECT * FROM customer_table;`

```
mysql> SELECT * FROM customer_table;
+-----+-----+-----+-----+
| cus_Num | cus_Name | cus_Address | cus_Telno |
+-----+-----+-----+-----+
| 1 | Alex | California | 0 |
| 2 | Dana | Florida | 11111 |
| 3 | Jamie | Iowa | 22222 |
| 4 | Morgan | Kansas | 33333 |
| 5 | Robin | Idaho | 44444 |
| 6 | Terry | Alaska | 55555 |
| 7 | Chris | Texas | 66666 |
| 8 | Pat | Vermont | 77777 |
| 9 | Tracey | Wisconsin | 88888 |
| 10 | Ronnie | Utah | 99999 |
+-----+-----+-----+-----+
```

9, MySQLからログアウトする。

`EXIT`

1, DB環境構築

1-3, MySQLログイン確認

MySQL_Login.txtを参考に、下記のコマンドを用いてログイン確認を行う。
変更したパスワード「pwd」が、正しく反映されているかを確認する。

1, MySQLにパスワードありでログイン。

`mysql -uroot -p`

2, パスワードを入力する。

`pwd`

3, MySQLにログインしたことを確認する。

4, MySQLからログアウトする。

`EXIT`

1, DB環境構築

1-4, MySQL停止処理

MySQL_StartStop.txtを参考に、下記コマンドでMySQLを停止させる。

◇MySQL停止コマンド

MySQL.server stop

MySQLは起動/停止時に、PIDファイルへの書き込みを行なっている。

停止処理を行わずにPCをシャットダウンすると、PIDファイルへの適切な書き込みが行われない。

そうすると、**MySQLの環境が壊れてしまい、起動出来なくなる可能性がある。**

※「ERROR! MySQL server PID file could not be found!」が出力された場合は、MySQLの再インストールをおすすめする。

2, DBから情報取得

2, DBから情報取得

2-1, プロジェクト作成

1, Spring スターター・プロジェクト作成

名前: SpringSampleDBInfo

型: Gradle (Buildship 3.x)

Java バージョン: 11

2, 新規 Spring スターター・プロジェクト依存関係

下記の項目をチェック。

- Lombok
- Spring Boot DevTools
- Spring Web
- Spring Data JPA
- Thymeleaf
- MySQL Driver

※画面上部に表示されていない場合は、検索欄を使用。

新規 Spring スターター・プロジェクト

サービス URL: <https://start.spring.io>

名前: SpringSampleDBInfo

☒ デフォルト・ロケーションを使用

ロケーション: /Applications/Eclipse_2019-12.app/Contents/workspace/SpringSampleDBInfo

型: Gradle (Buildship 3.x) パッケージング: Jar

Java バージョン: 11 言語: Java

グループ: com.example

成果物: SpringSampleDBInfo

バージョン: 0.0.1-SNAPSHOT

説明: Demo project for Spring Boot

パッケージ: com.example.demo

ワーキング・セット

☐ ワーキング・セットにプロジェクトを追加

ワーキング・セット:

< 戻る 次へ キャンセル 完了

新規 Spring スターター・プロジェクト依存関係

Spring Boot バージョン: 2.3.1

使用頻度高:

☒ Lombok ☐ MyBatis Framework ☒ MySQL Driver

☒ Spring Boot DevTools ☒ Spring Data JPA ☐ Spring Security

☒ Spring Web ☒ Thymeleaf

使用可能:

検索する依存関係を入力

Alibaba

Amazon Web サービス

開発ツール

Google Cloud Platform

I/O

メッセージング

Microsoft Azure

NoSQL

選択済み:

X Spring Boot DevTools

X Lombok

X Spring Data JPA

X MySQL Driver

X Thymeleaf

X Spring Web

< 戻る 次へ キャンセル 完了

2, DBから情報取得

2-1, プロジェクト作成

3, 追加ファイル

Javaクラスファイル

Customer.java

DBController.java

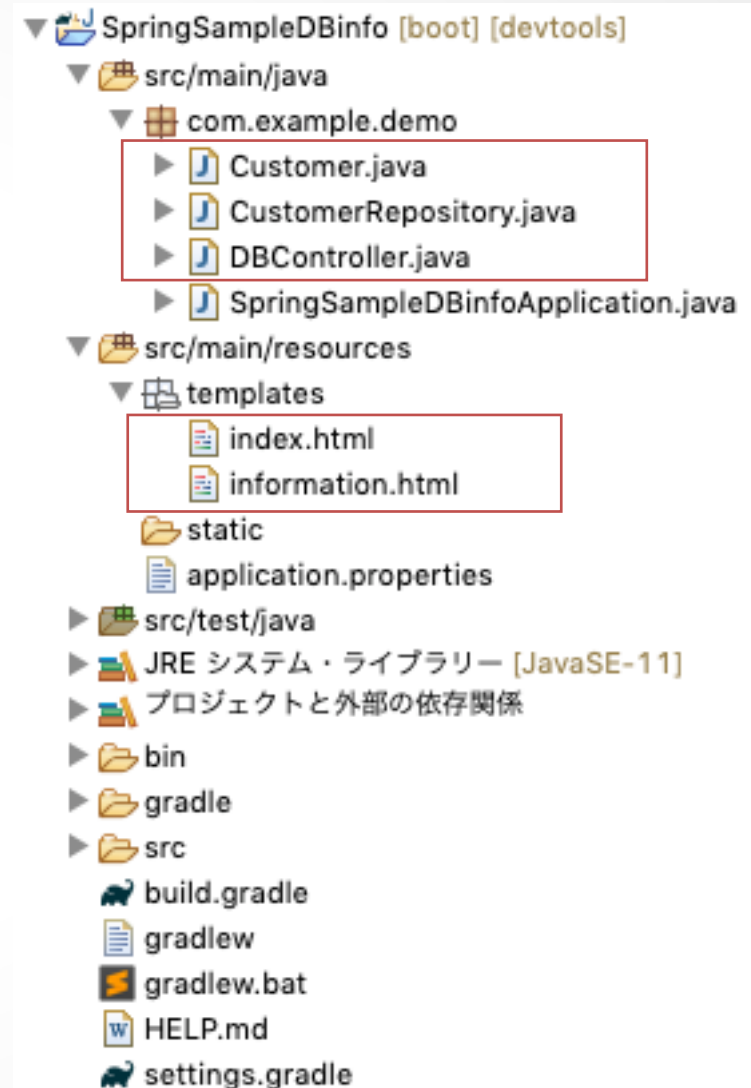
Javaインターフェースファイル

CustomerRepository.java

HTMLファイル

index.html

information.html



※ファイル内容は、サンプルファイルよりコピー&ペースト。

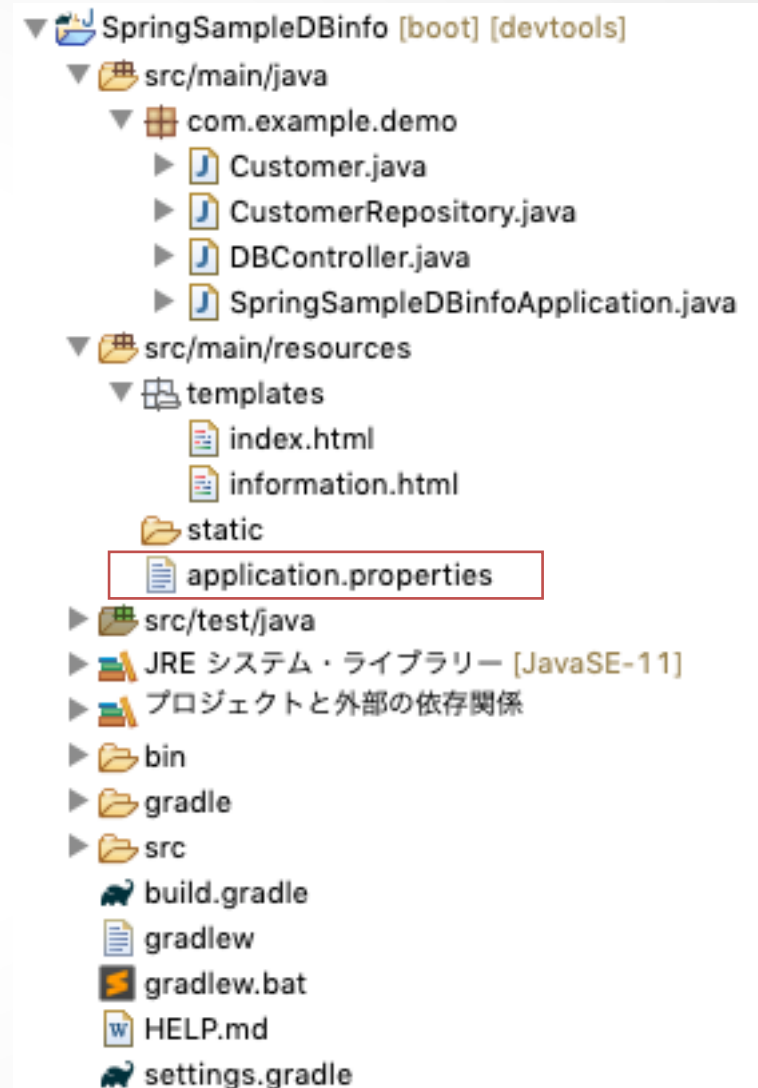
2, DBから情報取得

2-1, プロジェクト作成

4, application.propertiesへの追記

サンプルファイルを参考に、以下を追記する。

```
spring.datasource.url=jdbc:mysql://localhost:3306/testdb
spring.datasource.username=root
# Set your MySQL password.
spring.datasource.password=pwd
spring.jpa.database=MYSQL
```



2, DBから情報取得

2-2, 学習ポイント

CustomerRepository.java

JpaRepositoryにより、SQLのクエリを自動生成することが出来る。

@Repository

public **interface CustomerRepository** extends **JpaRepository**<Customer, Long> {}



JavaオブジェクトとDBを繋ぐインターフェース

2, DBから情報取得

2-2, 学習ポイント

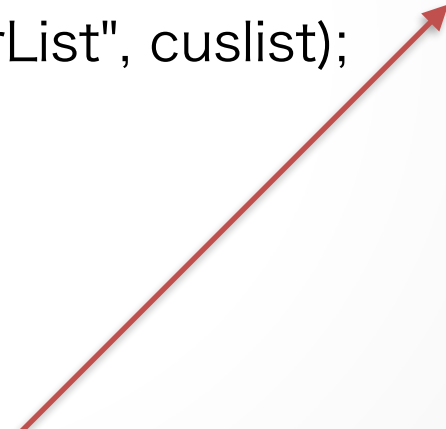
DBController.java

```
@RequestMapping(value = "/get")
public String Information(Model model) {

    List<Customer> cuslist = customerRepository.findAll();
    model.addAttribute("customerList", cuslist);

    return "information";
}
```

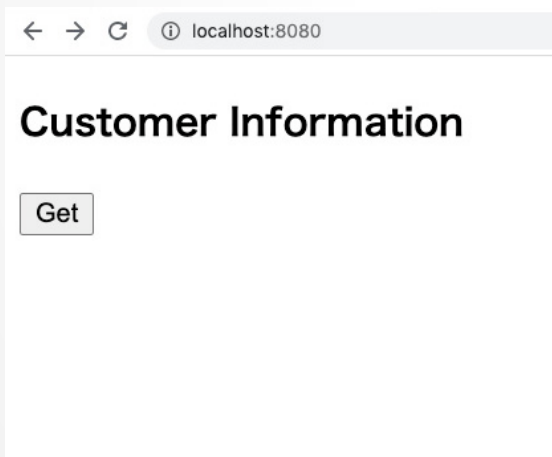
JpaRepositoryのFindAll関数にて、DB情報を全て取得している。



2, DBから情報取得

2-4, プロジェクト実行

- 1, MySQLサーバーを起動
- 2, SpringSampleDBinfoを実行
- 3, <http://localhost:8080/> にアクセス
- 4, Getボタンを押下
- 5, DBから取得した情報が表示される



A screenshot of a web browser window. The address bar shows 'localhost:8080/get?'. The page title is 'Customer Information'. Below the title, there is a table with customer information.

Name	Address	Telno
Alex	California	0
Dana	Florida	11111
Jamie	Iowa	22222
Morgan	Kansas	33333
Robin	Idaho	44444
Terry	Alaska	55555
Chris	Texas	66666
Pat	Vermont	77777
Tracey	Wisconsin	88888
Ronnie	Utah	99999

3, DBにデータ追加

3, DBにデータ追加

3-1, プロジェクト作成

1, Spring スターター・プロジェクト作成

名前: SpringSampleDBadd

型: Gradle (Buildship 3.x)

Java バージョン: 11

2, 新規 Spring スターター・プロジェクト依存関係

下記の項目をチェック。

- Lombok
- Spring Boot DevTools
- Spring Web
- Spring Data JPA
- Thymeleaf
- MySQL Driver

※画面上部に表示されていない場合は、検索欄を使用。

新規 Spring スターター・プロジェクト

サービス URL: <https://start.spring.io>

名前: SpringSampleDBadd

☒ デフォルト・ロケーションを使用

ロケーション: /Applications/Eclipse_2019-12.app/Contents/workspace/SpringSampleDBadd

型: Gradle (Buildship 3.x) | パッケージング: Jar

Java バージョン: 11 | 言語: Java

グループ: com.example

成果物: SpringSampleDBadd

バージョン: 0.0.1-SNAPSHOT

説明: Demo project for Spring Boot

パッケージ: com.example.demo

ワーキング・セット

☐ ワーキング・セットにプロジェクトを追加

ワーキング・セット:

新規 Spring スターター・プロジェクト依存関係

Spring Boot バージョン: 2.3.1

使用頻度高:

☒ Lombok | ☐ MyBatis Framework | ☒ MySQL Driver

☒ Spring Boot DevTools | ☒ Spring Data JPA | ☐ Spring Security

☒ Spring Web | ☒ Thymeleaf

使用可能:

検索する依存関係を入力

選択済み:

- X Spring Boot DevTools
- X Lombok
- X Spring Data JPA
- X MySQL Driver
- X Thymeleaf
- X Spring Web

Alibaba

Amazon Web サービス

開発ツール

Google Cloud Platform

I/O

メッセージング

Microsoft Azure

NoSQL

3, DBにデータ追加

3-1, プロジェクト作成

3, 追加ファイル

Javaクラスファイル

Customer.java

DBController.java

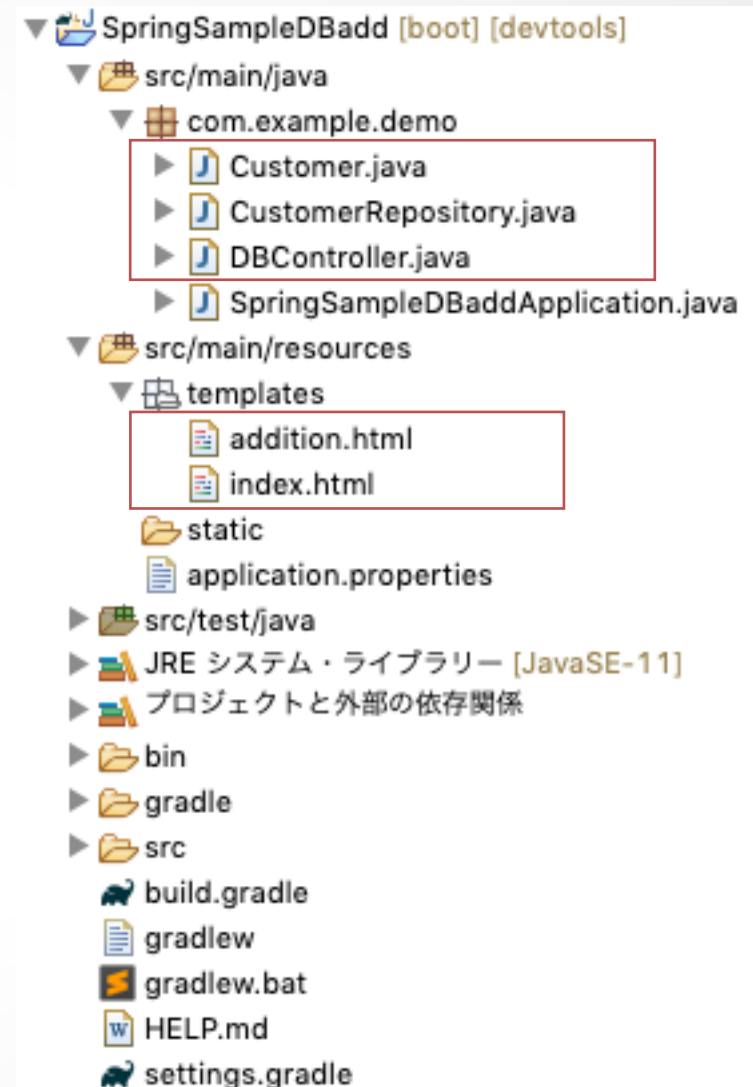
Javaインターフェースファイル

CustomerRepository.java

HTMLファイル

addition.html

index.html



※ファイル内容は、サンプルファイルよりコピー&ペースト。

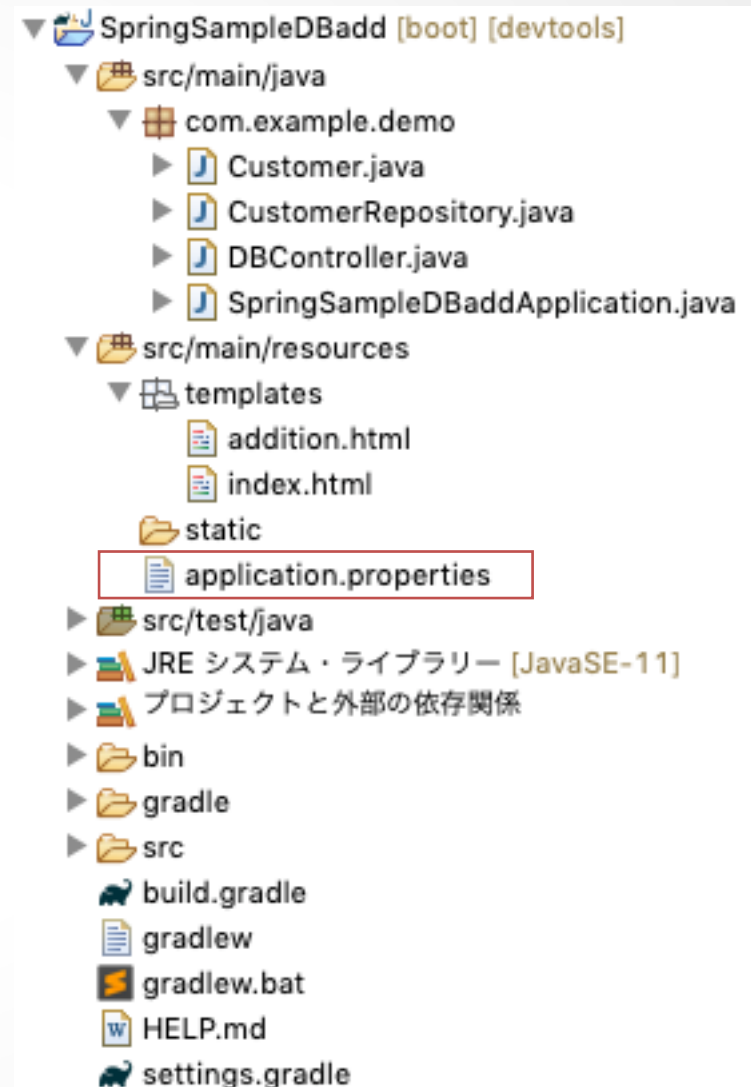
3, DBにデータ追加

3-1, プロジェクト作成

4, application.propertiesへの追記

サンプルファイルを参考に、以下を追記する。

```
spring.datasource.url=jdbc:mysql://localhost:3306/testdb
spring.datasource.username=root
# Set your MySQL password.
spring.datasource.password=pwd
spring.jpa.database=MYSQL
```



3, DBにデータ追加

3-2, 学習ポイント

DBController.java

```
@RequestMapping(value="/insert", method=RequestMethod.POST)
public String insert(@ModelAttribute Customer cusInfo) {

    customerRepository.save(cusInfo);

    return "redirect:/";
}
```

JpaRepositoryのsave関数にて、DBに情報を保存している。

3, DBにデータ追加

3-3, プロジェクト実行

- 1, MySQLサーバーを起動
- 2, SpringSampleDBaddを実行
- 3, <http://localhost:8080/> にアクセス
- 4, Additionボタンを押下
- 5, 追加する情報を入力し、Sign Upボタンを押下
- 6, 追加情報を含む、DB情報が取得される

← → ↻ localhost:8080

Customer Information

Name	Address	Telno
Alex	California	0
Dana	Florida	11111
Jamie	Iowa	22222
Morgan	Kansas	33333
Robin	Idaho	44444
Terry	Alaska	55555
Chris	Texas	66666
Pat	Vermont	77777
Tracey	Wisconsin	88888
Ronnie	Utah	99999

Addition

← → ↻ localhost:8080/addition?

Customer Information Registration

Name	Address	Telno
<input type="text" value="Brenda"/>	<input type="text" value="Colorado"/>	<input type="text" value="10101"/>

Sign Up

← → ↻ localhost:8080

Customer Information

Name	Address	Telno
Alex	California	0
Dana	Florida	11111
Jamie	Iowa	22222
Morgan	Kansas	33333
Robin	Idaho	44444
Terry	Alaska	55555
Chris	Texas	66666
Pat	Vermont	77777
Tracey	Wisconsin	88888
Ronnie	Utah	99999
Brenda	Colorado	10101

Addition

4, DBのデータ変更

4, DBのデータ変更

4-1, プロジェクト作成

1, Spring スターター・プロジェクト作成

名前: SpringSampleDBchange

型: Gradle (Buildship 3.x)

Java バージョン: 11

新規 Spring スターター・プロジェクト

サービス URL: <https://start.spring.io>

名前: SpringSampleDBchange

☒ デフォルト・ロケーションを使用

ロケーション: /Applications/Eclipse_2019-12.app/Contents/workspace/SpringSampleDBchange

型: Gradle (Buildship 3.x) パッケージング: Jar

Java バージョン: 11 言語: Java

グループ: com.example

成果物: SpringSampleDBchange

バージョン: 0.0.1-SNAPSHOT

説明: Demo project for Spring Boot

パッケージ: com.example.demo

ワーキング・セット

☐ ワーキング・セットにプロジェクトを追加

ワーキング・セット:

< 戻る 次へ > キャンセル 完了

2, 新規 Spring スターター・プロジェクト依存関係

下記の項目をチェック。

- Lombok
- Spring Boot DevTools
- Spring Web
- Spring Data JPA
- Thymeleaf
- MySQL Driver

※画面上部に表示されていない場合は、検索欄を使用。

新規 Spring スターター・プロジェクト依存関係

Spring Boot バージョン: 2.3.1

使用頻度高:

☒ Lombok ☐ MyBatis Framework ☒ MySQL Driver

☒ Spring Boot DevTools ☒ Spring Data JPA ☐ Spring Security

☒ Spring Web ☒ Thymeleaf

使用可能:

検索する依存関係を入力

Alibaba Amazon Web サービス 開発ツール Google Cloud Platform I/O メッセージング Microsoft Azure NoSQL

選択済み:

X Spring Boot DevTools
X Lombok
X Spring Data JPA
X MySQL Driver
X Thymeleaf
X Spring Web

< 戻る 次へ > キャンセル 完了

4, DBのデータ変更

4-1, プロジェクト作成

3, 追加ファイル

Javaクラスファイル

Customer.java

DBController.java

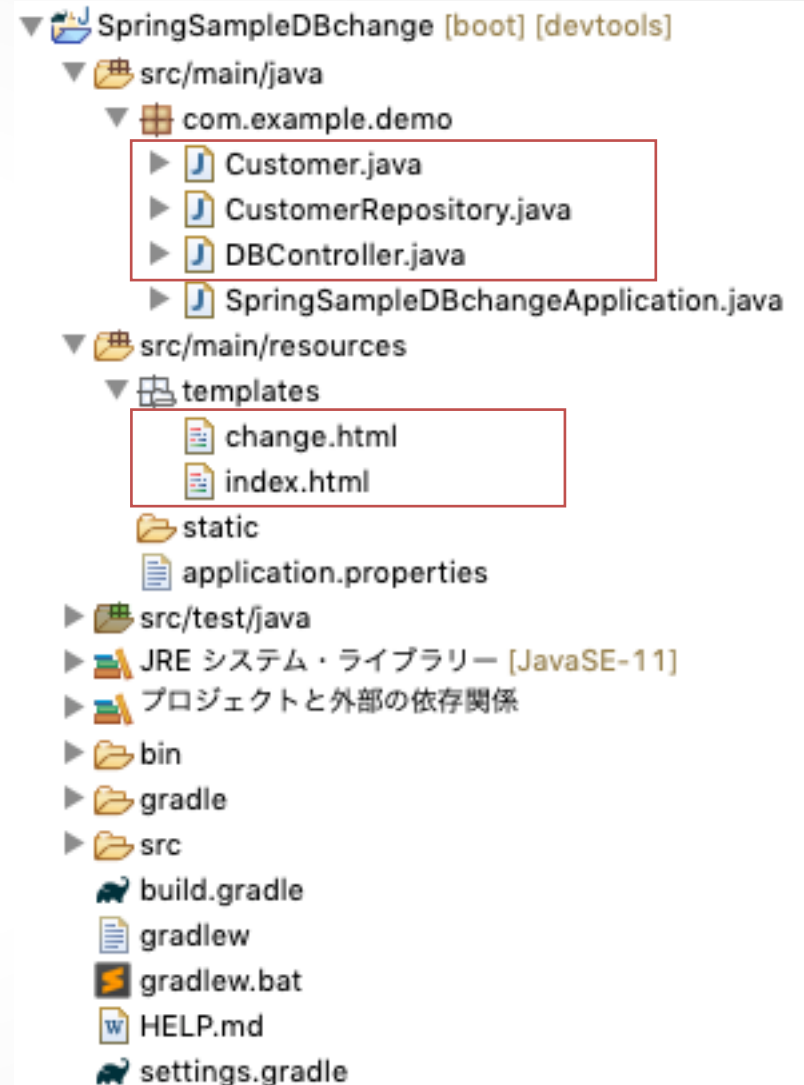
Javaインターフェースファイル

CustomerRepository.java

HTMLファイル

change.html

index.html



※ファイル内容は、サンプルファイルよりコピー&ペースト。

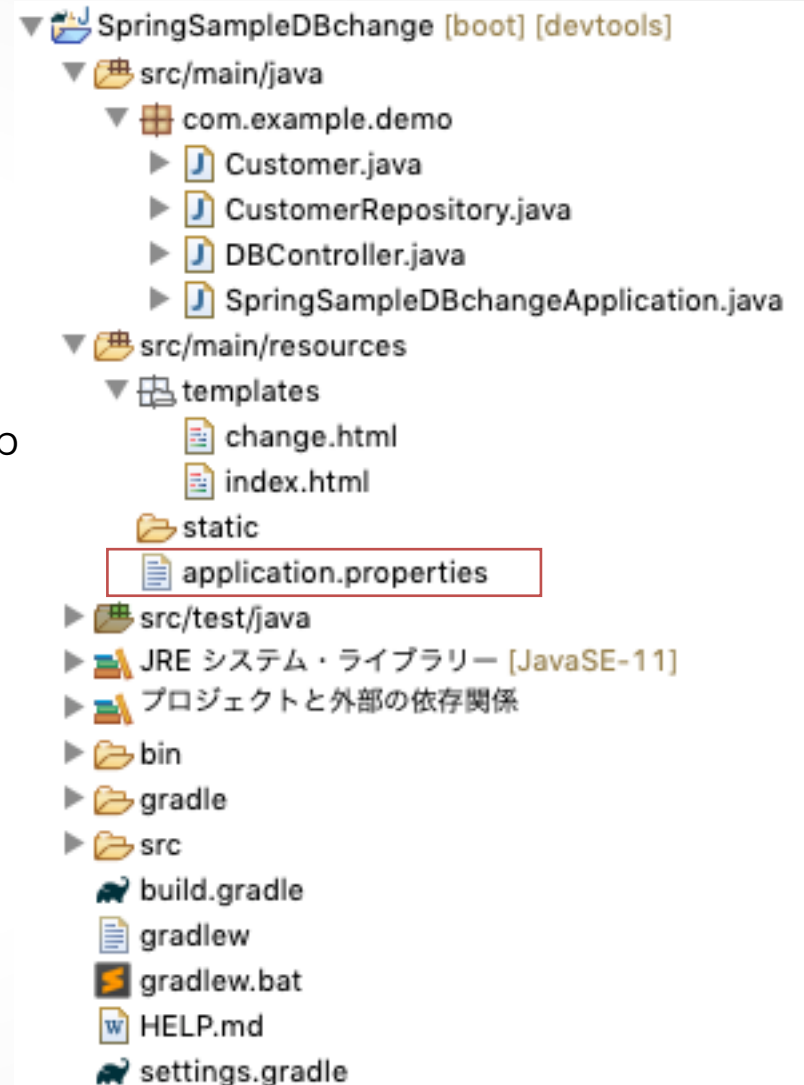
4, DBのデータ変更

4-1, プロジェクト作成

4, application.propertiesへの追記

サンプルファイルを参考に、以下を追記する。

```
spring.datasource.url=jdbc:mysql://localhost:3306/testdb
spring.datasource.username=root
# Set your MySQL password.
spring.datasource.password=pwd
spring.jpa.database=MYSQL
```



4, DBのデータ変更

4-2, 学習ポイント

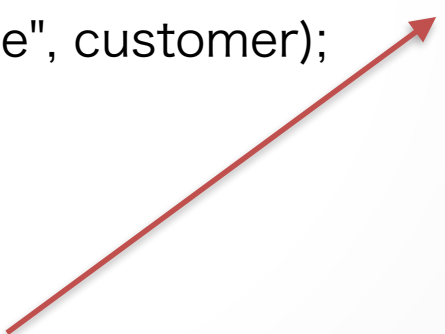
DBController.java

```
@RequestMapping(value = "/{num}/change", method = RequestMethod.GET)
public String changeValue(@PathVariable Long num, Model model) {

    Customer customer = customerRepository.findByld(num).get();
    model.addAttribute("customerUpdate", customer);

    return "change";
}
```

JpaRepositoryのfindByld関数にて、変更対象の情報を取得している。



4, DBのデータ変更

4-3, プロジェクト実行

- 1, MySQLサーバーを起動
- 2, SpringSampleDBchangeを実行
- 3, <http://localhost:8080/> にアクセス
- 4, 変更対象の◆を押下
- 5, 変更する情報を入力し、Saveボタンを押下
- 6, 変更情報を含む、DB情報が取得される

← → ↻ ⓘ localhost:8080

Customer Information Change

Name	Address	Telno
◆ Alex	California	0
◆ Dana	Florida	11111
◆ Jamie	Iowa	22222
◆ Morgan	Kansas	33333
◆ Robin	Idaho	44444
◆ Terry	Alaska	55555
◆ Chris	Texas	66666
◆ Pat	Vermont	77777
◆ Tracey	Wisconsin	88888
◆ Ronnie	Utah	99999
◆ Brenda	Colorado	10101

← → ↻ ⓘ localhost:8080/1/change

Customer Information Change

Name	Address	Telno
<input type="text" value="Alex"/>	<input type="text" value="California"/>	<input type="text" value="00000"/>
<input type="button" value="Save"/>		
<input type="button" value="Retuen"/>		

← → ↻ ⓘ localhost:8080

Customer Information Change

Name	Address	Telno
◆ Alex	California	00000
◆ Dana	Florida	11111
◆ Jamie	Iowa	22222
◆ Morgan	Kansas	33333
◆ Robin	Idaho	44444
◆ Terry	Alaska	55555
◆ Chris	Texas	66666
◆ Pat	Vermont	77777
◆ Tracey	Wisconsin	88888
◆ Ronnie	Utah	99999
◆ Brenda	Colorado	10101

5, DBのデータ削除

5, DBのデータ削除

5-1, プロジェクト作成

1, Spring スターター・プロジェクト作成

名前: SpringSampleDBdelete

型: Gradle (Buildship 3.x)

Java バージョン: 11

新規 Spring スターター・プロジェクト

サービス URL: <https://start.spring.io>

名前: SpringSampleDBdelete

☒ デフォルト・ロケーションを使用

ロケーション: /Applications/Eclipse_2019-12.app/Contents/workspace/SpringSampleDBdelete

型: Gradle (Buildship 3.x) パッケージング: Jar

Java バージョン: 11 言語: Java

グループ: com.example

成果物: SpringSampleDBdelete

バージョン: 0.0.1-SNAPSHOT

説明: Demo project for Spring Boot

パッケージ: com.example.demo

ワーキング・セット

☐ ワーキング・セットにプロジェクトを追加

ワーキング・セット:

< 戻る 次へ > キャンセル 完了

2, 新規 Spring スターター・プロジェクト依存関係

下記の項目をチェック。

- Lombok
- Spring Boot DevTools
- Spring Web
- Spring Data JPA
- Thymeleaf
- MySQL Driver

※画面上部に表示されていない場合は、検索欄を使用。

新規 Spring スターター・プロジェクト依存関係

Spring Boot バージョン: 2.3.1

使用頻度高:

☒ Lombok ☐ MyBatis Framework ☒ MySQL Driver

☒ Spring Boot DevTools ☒ Spring Data JPA ☐ Spring Security

☒ Spring Web ☒ Thymeleaf

使用可能:

検索する依存関係を入力

Alibaba

Amazon Web サービス

開発ツール

Google Cloud Platform

I/O

メッセージング

Microsoft Azure

NoSQL

選択済み:

X Spring Boot DevTools

X Lombok

X Spring Data JPA

X MySQL Driver

X Thymeleaf

X Spring Web

< 戻る 次へ > キャンセル 完了

5, DBのデータ削除

5-1, プロジェクト作成

3, 追加ファイル

Javaクラスファイル

Customer.java

DBController.java

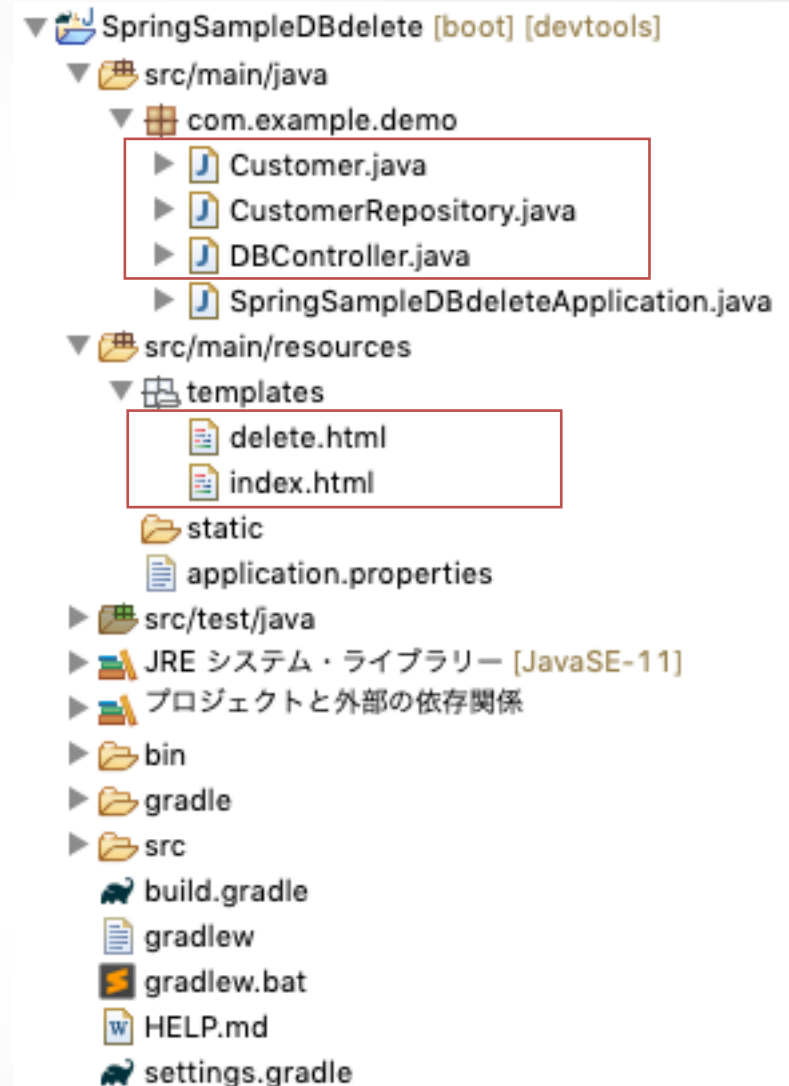
Javaインターフェースファイル

CustomerRepository.java

HTMLファイル

delete.html

index.html



※ファイル内容は、サンプルファイルよりコピー&ペースト。

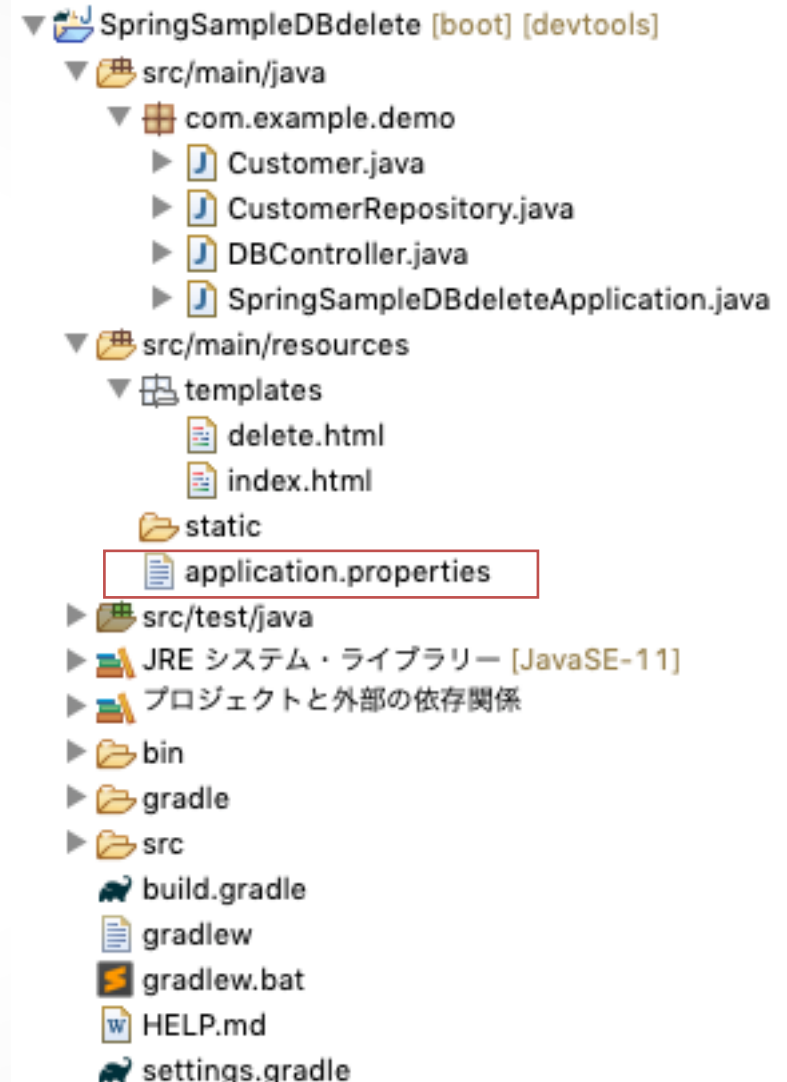
5, DBのデータ削除

5-1, プロジェクト作成

4, application.propertiesへの追記

サンプルファイルを参考に、以下を追記する。

```
spring.datasource.url=jdbc:mysql://localhost:3306/testdb
spring.datasource.username=root
# Set your MySQL password.
spring.datasource.password=pwd
spring.jpa.database=MYSQL
```



5, DBのデータ削除

5-2, 学習ポイント


DBController.java

```
@RequestMapping(value="/delete", method=RequestMethod.POST)
public String delete(@ModelAttribute Customer customerDelete) {

    customerRepository.delete(customerDelete);

    return "redirect:/";
}
```

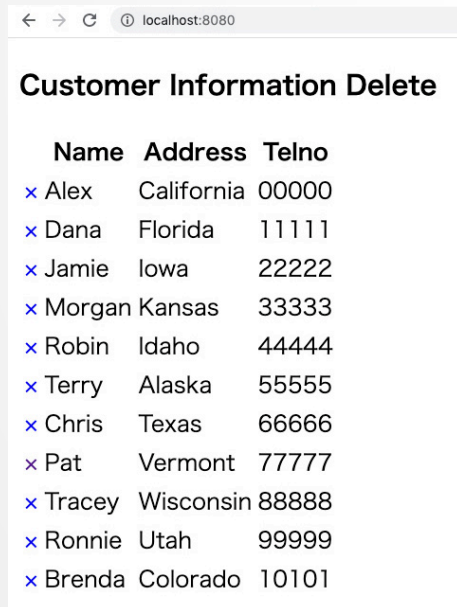
JpaRepositoryのdelete関数にて、対象情報を削除している。



5, DBのデータ削除

5-3, プロジェクト実行

- 1, MySQLサーバーを起動
- 2, SpringSampleDBchangeを実行
- 3, <http://localhost:8080/> にアクセス
- 4, 削除対象の×を押下
- 5, 削除対象を確認し、Deleteボタンを押下
- 6, 削除後の、DB情報が取得される

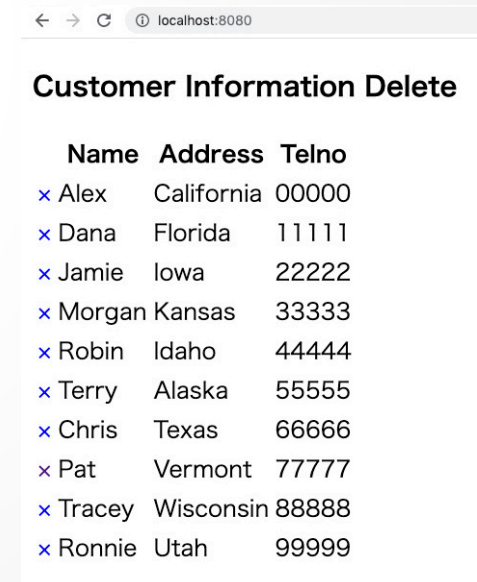


Name	Address	Telno
× Alex	California	00000
× Dana	Florida	11111
× Jamie	Iowa	22222
× Morgan	Kansas	33333
× Robin	Idaho	44444
× Terry	Alaska	55555
× Chris	Texas	66666
× Pat	Vermont	77777
× Tracey	Wisconsin	88888
× Ronnie	Utah	99999
× Brenda	Colorado	10101



Name	Address	Telno
Brenda	Colorado	10101

Delete
Retuen



Name	Address	Telno
× Alex	California	00000
× Dana	Florida	11111
× Jamie	Iowa	22222
× Morgan	Kansas	33333
× Robin	Idaho	44444
× Terry	Alaska	55555
× Chris	Texas	66666
× Pat	Vermont	77777
× Tracey	Wisconsin	88888
× Ronnie	Utah	99999

データベース連携 in Spring Boot

Fin.