

JAVA アプリ SpringBoot 講座

1.Spring Initializr (公式プロジェクト生成ツール)

【STEP 1】 Spring Initializr にアクセス

ブラウザで以下を開く：

The screenshot shows the Spring Initializr web application. On the left, there are sections for Project (Gradle - Groovy, Gradle - Kotlin, Maven), Language (Java, Kotlin, Groovy), and Spring Boot (4.0.2 (SNAPSHOT), 4.0.1, 3.5.10 (SNAPSHOT), 3.5.9). The 'Project Metadata' section includes fields for Group (com.example), Artifact (demo), Name (demo), Description (Demo project for Spring Boot), Package name (com.example.demo), Packaging (Jar), Configuration (Properties), and Java version (25, 21, 17). On the right, there's a 'Dependencies' section with a red box around the 'ADD DEPENDENCIES...' button. Below it, it says 'No dependency selected'. At the bottom, there are 'GENERATE' and 'EXPLORE' buttons.

【STEP 2】 プロジェクト設定を入力

- | | |
|-----------------------|----------------------------------------|
| ① Project | Maven |
| ② Language | Java |
| ③ Spring Boot Version | 3.5.x (最新安定版) |
| ④ Project Metadata | Group : com.example Artifact : demo |
| ⑤ Packaging | Jar |
| ⑥ Java | 17 |

【STEP 3】 Dependencies (必要なライブラリ) を追加

1. ADD DEPENDENCIES をクリックします
2. spring Web を選択します

WEB

Spring Web

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

1. ADD DEPENDENCIES をクリックします
2. Thymeleaf を選択します

TEMPLATE ENGINES

Thymeleaf
A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

The screenshot shows the Spring Initializr interface. On the left, there are sections for Project (Gradle - Groovy, Gradle - Kotlin, Maven), Language (Java, Kotlin, Groovy), Spring Boot (4.0.2 (SNAPSHOT), 4.0.1, 3.5.10 (SNAPSHOT), 3.5.9), Project Metadata (Group: com.example, Artifact: demo, Name: demo, Description: Demo project for Spring Boot, Package name: com.example.demo, Packaging: Jar, Configuration: Properties), and Java version (25, 21, 17). On the right, there is a Dependencies section with a 'Spring Web' entry (WEB) and a 'Thymeleaf' entry (TEMPLATE ENGINES). A red box highlights the 'GENERATE ⌘ + ↵' button at the bottom center. The top right corner has a gear icon.

【STEP 4】プロジェクト生成 (ZIP ダウンロード)

画面右下の GENERATE ボタンをクリック。
 ↓
 自動で ZIP ファイルがダウンロードが開始されます
 ↓
 ダウンロードされたファイルを確認します

demo.zip

【STEP 5】ZIP を展開してフォルダへ移動

ダウンロードした ZIP を右クリック → 展開。
 Mac の場合は右クリックして→開く

例： demo/

【STEP 6】ターミナル（または PowerShell）で起動する

以下のようにフォルダへ移動：

ここ重要です！

```
cd demo
```

① プロジェクト直下 (demo フォルダ内にて) でビルドを実行します（必須）

```
./mvnw clean compile
```

以下が生成されます：

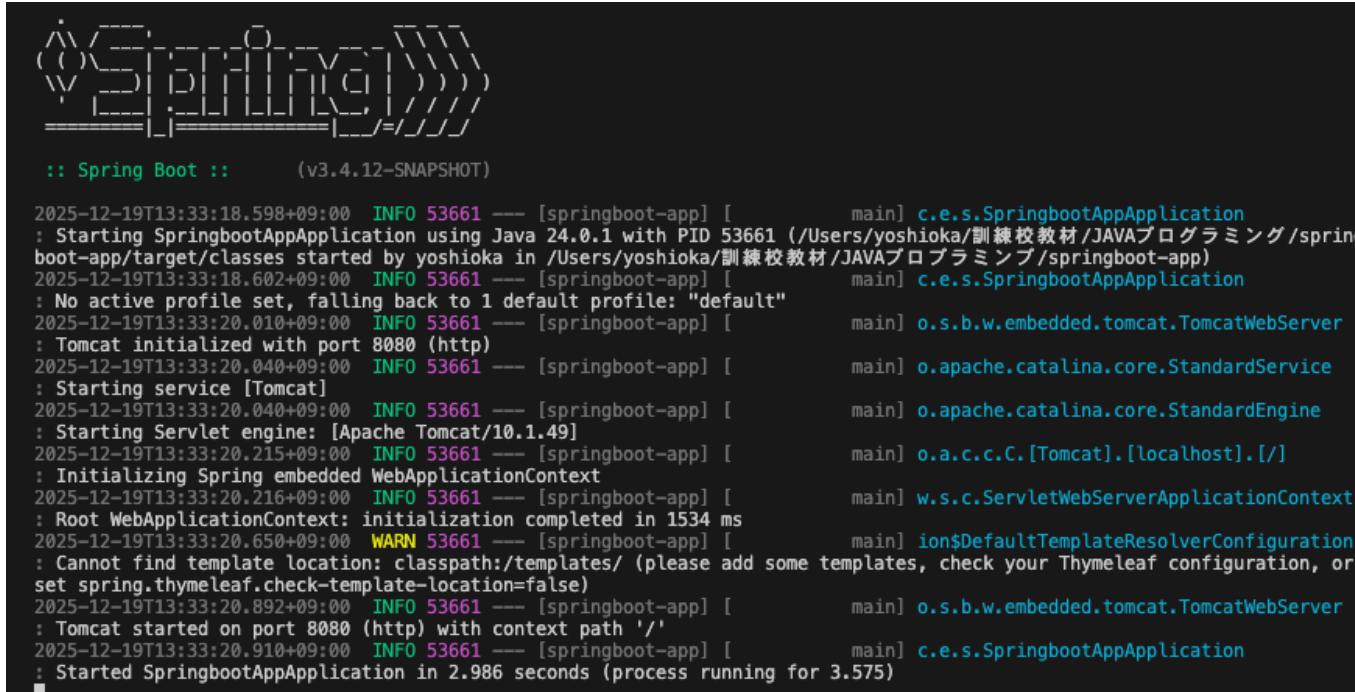
```
target/classes/com/example/demo/DemoApplication.class
```

```
yoshioka@EDIX-P-041 springboot-app % ./mvnw clean compile
WARNING: A terminally deprecated method in sun.misc.Unsafe has been called
WARNING: sun.misc.Unsafe::staticFieldBase has been called by com.google.inject.internal.aop.HiddenClassDefiner (file:/Users/yoshioka/.m2/wrapper/dists/apache-maven-3.9.11/a2d47e15/lib/guice-5.1.0-classes.jar)
WARNING: Please consider reporting this to the maintainers of class com.google.inject.internal.aop.HiddenClassDefiner
WARNING: sun.misc.Unsafe::staticFieldBase will be removed in a future release
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example:springboot-app >-----
[INFO] Building springboot-app 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- clean:3.4.1:clean (default-clean) @ springboot-app ---
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ springboot-app ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO] Copying 0 resource from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ springboot-app ---
[INFO] Recompiling the module because of changed source code.
[INFO] Compiling 1 source file with javac [debug parameters release 17] to target/classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 2.399 s
[INFO] Finished at: 2025-12-19T13:27:20+09:00
[INFO]
yoshioka@EDIX-P-041 springboot-app %
```

続いて以下のコマンドで Spring Boot を起動させましょう

```
./mvnw spring-boot:run
```

```
yoshioka@EDIX-P-041 springboot-app % ./mvnw spring-boot:run
WARNING: A terminally deprecated method in sun.misc.Unsafe has been called
WARNING: sun.misc.Unsafe::staticFieldBase has been called by com.google.inject.internal.aop.HiddenClassDefiner (file:/Users/yoshioka/.m2/wrapper/dists/apache-maven-3.9.11/a2d47e15/lib/guice-5.1.0-classes.jar)
WARNING: Please consider reporting this to the maintainers of class com.google.inject.internal.aop.HiddenClassDefiner
WARNING: sun.misc.Unsafe::staticFieldBase will be removed in a future release
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example:springboot-app >-----
[INFO] Building springboot-app 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] >>> spring-boot:3.4.12-SNAPSHOT:run (default-cli) > test-compile @ springboot-app >>>
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ springboot-app ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO] Copying 0 resource from src/main/resources to target/classes
[INFO]
```



Started DemoApplication in と表示されれば成功です。

【STEP 7】 ブラウザで確認

<http://localhost:8080>



Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Fri Dec 19 13:38:36 JST 2025

There was an unexpected error (type=Not Found, status=404).

上記が表示されれば成功です。

現在は機能を作成していないためページがありませんというエラーが表示されます。

2.Hello Spring Boot! と表示させよう

2-1 メインクラスを確認します（パッケージとアノテーション）

src/main/java/com/example/demo/DemoApplication.java

```
1 package com.example.demo;
```

```
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6 @SpringBootApplication
7 public class DemoApplication {
8
9   public static void main(String[] args) {
10     SpringApplication.run(DemoApplication.class, args);
11   }
12
13 }
```

2-2Controller を作る

以下のディレクトリ

ディレクトリ src/main/java/com/example/demo/

新規ファイルを作成

ファイル名 HelloController.java

```
1 package com.example.demo;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class HelloController {
8
9   @GetMapping("/")
10  public String hello() {
11    return "Hello Spring Boot!";
12  }
13 }
```

サーバを起動

プロジェクトルートのディレクトリにて

| | |
|---------|-------------|
| windows | Ctrl + c |
| mac | control + c |

SpringBoot を一度終了させます。

再度

```
./mvnw spring-boot:run
```

にて起動させます。

ブラウザにて localhost:8080 を入力して表示を確認してください。



Hello Spring Boot! と表示されれば成功です

CRUD システム構築

ここでは以下の 4 つの画面を持つアプリケーションを構築していきます。

```
src/main/resources/templates/
```

に「crudapp」ディレクトリを作成し、以下の 4 つのファイルを作成してください

```
src/main/resources/templates/crudapp
```

- ・ 一覧 ファイル名 index.html NEW
- ・ 登録 ファイル名 input.html NEW
- ・ 確認 ファイル名 confirm.html NEW
- ・ 完了 ファイル名 complete.html NEW

続いて Rest コントローラーを作成していきます。

```
src/main/java/com/example/demo
```

ファイル名 CrudappController.java NEW

各ファイルコーディング

index.html

```
1 <!DOCTYPE html>
```

```
2 <html lang="ja">
3 <head>
4   <meta charset="UTF-8">
5   <title>CRUD App - 一覧 </title>
6 </head>
7 <body>
8 <h1>CRUD App: 一覧 </h1>
9 <p> ここに一覧を表示します。</p>
10 <nav>
11   <ul>
12     <li><a href="/crudapp/input"> 新規登録 </a></li>
13     <li><a href="/crudapp/confirm"> 登録確認 </a></li>
14     <li><a href="/crudapp/complete"> 登録完了 </a></li>
15   </ul>
16   <a href="/crudapp"> 一覧へ </a>
17 </nav>
18 </body>
19 </html>
```

input.html

```
1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <meta charset="UTF-8">
5   <title>CRUD App - 登録 </title>
6 </head>
7 <body>
8 <h1>CRUD App: 登録 </h1>
9 <p> ここに入力フォームを配置します。</p>
10 <nav>
11   <a href="/crudapp"> 一覧へ戻る </a>
12 </nav>
13 </body>
14 </html>
```

confirm.html

```
1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <meta charset="UTF-8">
5   <title>CRUD App - 確認 </title>
6 </head>
7 <body>
8 <h1>CRUD App: 登録確認 </h1>
9 <p> ここに確認内容を表示します。</p>
```

```
10 <nav>
11   <a href="/crudapp/input"> 入力へ戻る </a>
12   <a href="/crudapp/complete"> 登録完了へ </a>
13   <a href="/crudapp"> 一覧へ戻る </a>
14 </nav>
15 </body>
16 </html>
```

complete.html

```
1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <meta charset="UTF-8">
5   <title>CRUD App - 完了 </title>
6 </head>
7 <body>
8 <h1>CRUD App: 登録完了 </h1>
9 <p>登録が完了しました。</p>
10 <nav>
11   <a href="/crudapp"> 一覧へ戻る </a>
12 </nav>
13 </body>
14 </html>
```

CrudappController.java

```
1 package com.example.demo;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.RequestMapping;
6
7 @Controller
8 @RequestMapping("/crudapp")
9 public class CrudappController {
10
11   @GetMapping
12   public String index() {
13     return "crudapp/index";
14   }
15
16   @GetMapping("/input")
17   public String input() {
18     return "crudapp/input";
19   }
20 }
```

```
21 @GetMapping("/confirm")
22 public String confirm() {
23     return "crudapp/confirm";
24 }
25
26 @GetMapping("/complete")
27 public String complete() {
28     return "crudapp/complete";
29 }
30}
```

プログラミングが完了したら SpringBoot を再起動し、動作確認を行いましょう。

プロジェクトルートのディレクトリにて

| | |
|---------|-------------|
| windows | Ctrl + c |
| mac | control + c |

SpringBoot を一度終了させます。

再度

```
./mvnw spring-boot:run
```

にて起動させます。

ブラウザにて localhost:8080/crudapp を入力して表示を確認してください。



CRUD App: 一覧

ここに一覧を表示します。

- [新規登録](#)
- [登録確認](#)
- [登録完了](#)

[一覧へ](#)



CRUD App: 登録

ここに入力フォームを配置します。

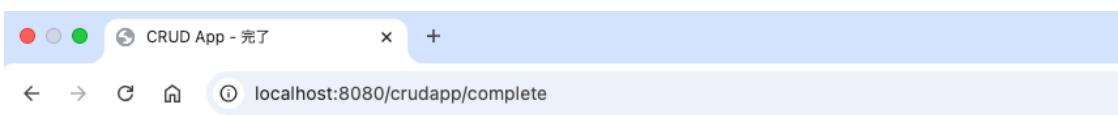
[一覧へ戻る](#)



CRUD App: 登録確認

ここに確認内容を表示します。

[入力へ戻る](#) [登録完了へ](#) [一覧へ戻る](#)



CRUD App: 登録完了

登録が完了しました。

[一覧へ戻る](#)



修正手順

入力フォームを追加し送信先を確認画面に
: src/main/resources/templates/crudapp/input.html

確認画面で送信内容を表示し、戻る／確定するボタンを設置（確定は完了画面へ POST）
: src/main/resources/templates/crudapp/confirm.html

コントローラーに POST ハンドラを追加して確認・完了へ遷移できるように

: src/main/java/com/example/springboot_app/CrudappController.java

完了画面は「登録が完了しました」を表示

: src/main/resources/templates/crudapp/complete.html

起動後、/crudapp/input で入力→送信→確認→確定の流れを確認してください。

ソースコードはグリーンとピンクで追加と削除を判断していきます。

ソースコードにおいてはピンクで表記されている箇所を削除し、緑のコードを追加入力してください

input.html

```
1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <meta charset="UTF-8">
5   <title>CRUD App - 登録 </title>
6 </head>
7 <body>
8 <h1>CRUD App: 登録 </h1>
9 <p> ここに入力フォームを配置します。</p>
9 <form th:action="@{/crudapp/confirm}" method="post">
10  <label for="content"> 内容 </label><br>
11  <textarea id="content" name="content" rows="6" cols="50" required></textarea><br>
12  <button type="submit"> 送信 </button>
13 </form>
14 <nav>
15  <a href="/crudapp"> 一覧へ戻る </a>
16 </nav>
17 </body>
18 </html>
```

confirm.html

```
1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <meta charset="UTF-8">
5   <title>CRUD App - 確認 </title>
6 </head>
7 <body>
8 <h1>CRUD App: 登録確認 </h1>
9 <p> ここに確認内容を表示します。</p>
```

```
9 <section>
10 <h2> 入力内容 </h2>
11 <p th:text="${content}"> 入力内容がここに表示されます。</p>
12 </section>
13 <nav>
14   <a href="/crudapp/input"> 入力へ戻る </a>
15   <a href="/crudapp/complete"> 登録完了へ </a>
16   <form th:action="@{/crudapp/input}" method="get" style="display:inline;">
17     <button type="submit"> 戻る </button>
18   </form>
19   <form th:action="@{/crudapp/complete}" method="post" style="display:inline;">
20     <input type="hidden" name="content" th:value="${content}">
21     <button type="submit"> 確定する </button>
22   </form>
23   <a href="/crudapp"> 一覧へ戻る </a>
24 </nav>
25 </body>
26 </html>
```

CrudappController.java

```
1 package com.example.demo;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PostMapping;
6 import org.springframework.web.bind.annotation.RequestMapping;
7 import org.springframework.web.bind.annotation.RequestParam;
8 import org.springframework.ui.Model;
9
10 @Controller
11 @RequestMapping("/crudapp")
12 public class CrudappController {
13
14   @GetMapping
15   public String index() {
16     return "crudapp/index";
17   }
18
19   @GetMapping("/input")
20   public String input() {
21     return "crudapp/input";
22   }
23
24   @GetMapping("/confirm")
25   public String confirm() {
```

```

26     return "crudapp/confirm";
27 }
28
29 @PostMapping("/confirm")
30 public String confirm(@RequestParam("content") String content, Model model) {
31     model.addAttribute("content", content);
32     return "crudapp/confirm";
33 }
34
35 @GetMapping("/complete")
36 public String complete() {
37     return "crudapp/complete";
38 }
39
40 @PostMapping("/complete")
41 public String completePost() {
42     return "crudapp/complete";
43 }
44 }
```

プログラミングが完了したら SpringBoot を再起動し、動作確認を行いましょう。

プロジェクトルートのディレクトリにて

| | |
|---------|-------------|
| windows | Ctrl + c |
| mac | control + c |

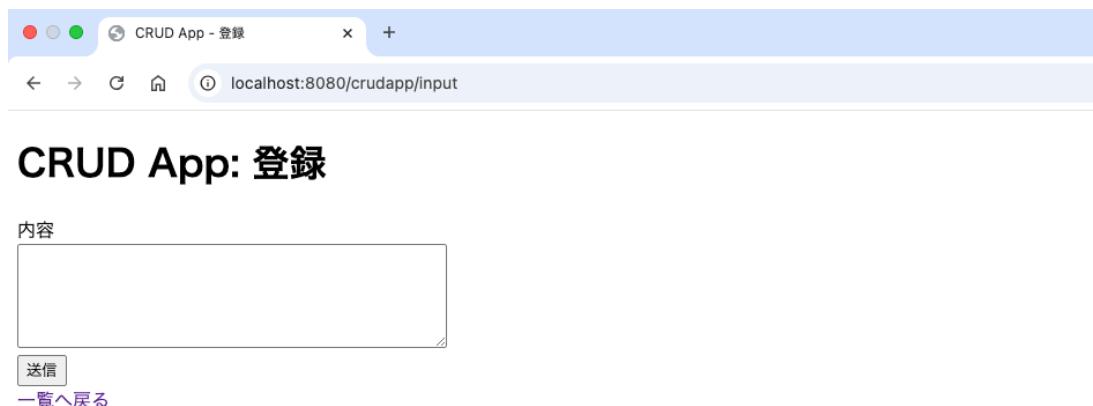
SpringBoot を一度終了させます。

再度

```
./mvnw spring-boot:run
```

にて起動させます。

ブラウザにて localhost:8080/crudapp/input を入力して表示を確認してください。



CRUD App - 登録

localhost:8080/crudapp/input

CRUD App: 登録

内容

CRUDシステムを構築してきましょう

送信

[一覧へ戻る](#)

CRUD App - 確認

localhost:8080/crudapp/confirm

CRUD App: 登録確認

入力内容

CRUDシステムを構築してきましょう

[戻る](#) [確定する](#) [一覧へ戻る](#)

CRUD App - 完了

localhost:8080/crudapp/complete

CRUD App: 登録完了

登録が完了しました。

[一覧へ戻る](#)

データベース構築

現在の状態はデータがどこにも保存されていない状態です。データベースを構築して登録内容を保存しましょう。

本システムでは、データベースの構築およびマイグレーション管理に Flyway を使用します。
データベース環境は Docker を用いて MySQL サーバーを構築します。

プロジェクトルートフォルダに docker-compose.yml を作成します。

docker-compose.yml **NEW**

```
1 version: "3.9"
2 services:
3   mysql:
```

```

4 image: mysql:8.0
5 container_name: crudapp-mysql
6 restart: unless-stopped
7 environment:
8   MYSQL_DATABASE: crudapp_db
9   MYSQL_USER: crudapp_user
10  MYSQL_PASSWORD: crudapp_pass
11  MYSQL_ROOT_PASSWORD: rootpass
12 ports:
13  - "3306:3306"
14 volumes:
15  - db_data:/var/lib/mysql
16 command: >
17   --character-set-server=utf8mb4
18   --collation-server=utf8mb4_unicode_ci
19   --default-authentication-plugin=caching_sha2_password
20 volumes:
21  db_data:

```

pom.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5   <parent>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-starter-parent</artifactId>
8     <version>3.4.1</version>
9     <relativePath/> <!-- lookup parent from repository -->
10  </parent>
11  <groupId>com.example</groupId>
12  <artifactId>springboot-app</artifactId>
13  <version>0.0.1-SNAPSHOT</version>
14  <name>springboot-app</name>
15  <description>demodiscripttions</description>
16  <url/>
17  <licenses>
18    <license/>
19  </licenses>
20  <developers>
21    <developer/>
22  </developers>
23  <scm>
24    <connection/>

```

```
25 <developerConnection/>
26 <tag/>
27 <url/>
28 </scm>
29 <properties>
30   <java.version>17</java.version>
31 </properties>
32 <dependencies>
33   <dependency>
34     <groupId>org.springframework.boot</groupId>
35     <artifactId>spring-boot-starter-jdbc</artifactId>
36   </dependency>
37   <dependency>
38     <groupId>org.springframework.boot</groupId>
39     <artifactId>spring-boot-starter-thymeleaf</artifactId>
40   </dependency>
41   <dependency>
42     <groupId>org.springframework.boot</groupId>
43     <artifactId>spring-boot-starter-web</artifactId>
44   </dependency>
45   <dependency>
46     <groupId>org.flywaydb</groupId>
47     <artifactId>flyway-mysql</artifactId>
48   </dependency>
49   <dependency>
50     <groupId>org.flywaydb</groupId>
51     <artifactId>flyway-core</artifactId>
52   </dependency>
53   <dependency>
54     <groupId>com.mysql</groupId>
55     <artifactId>mysql-connector-j</artifactId>
56   </dependency>
57   <dependency>
58     <groupId>org.springframework.boot</groupId>
59     <artifactId>spring-boot-starter-test</artifactId>
60     <scope>test</scope>
61   </dependency>
62 </dependencies>
63 <build>
64   <plugins>
65     <plugin>
66       <groupId>org.springframework.boot</groupId>
67       <artifactId>spring-boot-maven-plugin</artifactId>
68     </plugin>
69   </plugins>
70 </build>
71 </project>
```

src/main/resources/application.properties

```
1 spring.application.name=demo
2
3 spring.datasource.url=jdbc:mysql://localhost:3306/crudapp_db?useSSL=false&allowPublicKeyRetrieval=true&serverTimezone=UTC
4 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
5 spring.datasource.username=crudapp_user
6 spring.datasource.password=crudapp_pass
7
8 spring.flyway.enabled=true
9
10 # ローカル環境向けの SQL ログ (必要に応じて OFF に)
11 spring.jpa.show-sql=true
```

src/main/resources/ ディレクトリに「db/migration/」を作成

以下のファイルを作成

V1__create_note.sql NEW

```
CREATE TABLE notes (
    id BIGINT NOT NULL AUTO_INCREMENT,
    text VARCHAR(1000) NOT NULL,
    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Docker を起動する

Docker のアプリを起動させましょう。アプリが起動したらターミナルで以下のコマンドを実行します。

docker compose up -d は、

定義された複数のコンテナをバックグラウンドで起動し、サービスを利用可能な状態にするためのコマンドです。

```
docker compose up -d
```

```
● yoshioka@EDIX-P-041 springboot-app % docker compose up -d
WARN[0000] /Users/yoshioka/訓練校教材/JAVAプログラミング/springboot-app/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 1/1
  ✓ mysql Pulled                                         10.6s
[+] Running 3/3
  ✓ Network springboot-app_default   Created          0.1s
  ✓ Volume "springboot-app_db_data" Created          0.0s
  ✓ Container crudapp-mysql        Started          1.2s
○ yoshioka@EDIX-P-041 springboot-app %
```

続いて SpringBoot を再起動しましょう

MySQL にログインをしてテーブルが作成されているか確認します

```
docker exec -it crudapp-mysql mysql -uroot -p
```

問題 1 出力 デバッグ コンソール ターミナル テスト結果 ポート

○ yoshioka@EDIX-P-041 springboot-app % docker exec -it crudapp-mysql mysql -uroot -p
Enter password: [REDACTED]

password は docker-compose.yml に記載されている

```
MYSQL_ROOT_PASSWORD: rootpass
```

○ yoshioka@EDIX-P-041 springboot-app % docker exec -it crudapp-mysql mysql -uroot -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 22
Server version: 8.0.44 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> [REDACTED]

データベースを確認します

```
show databases;
```

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| crudapp_db |  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
5 rows in set (0.01 sec)  
  
mysql> [REDACTED]
```

crudapp_db があることを確認します。

続いて使用するデータベースを選択します。

```
use crudapp_db;
```

```
mysql> use crudapp_db
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
```

作成されているテーブルを確認します。

```
show tables;
```

```
mysql> show tables;
+-----+
| Tables_in_crudapp_db |
+-----+
| flyway_schema_history |
| notes                  |
+-----+
2 rows in set (0.01 sec)
```

notes テーブルが確認できました。

SpringBoot からデータベースに接続する

MyBatis のを利用できるように設定

pom.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6       <groupId>org.springframework.boot</groupId>
7       <artifactId>spring-boot-starter-parent</artifactId>
8       <version>3.4.1</version>
9       <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    <groupId>com.example</groupId>
12    <artifactId>demo</artifactId>
13    <version>0.0.1-SNAPSHOT</version>
14    <name>demo</name>
15    <description>demodiscription</description>
16    <url/>
17    <licenses>
18      <license/>
19    </licenses>
```

```
20 <developers>
21   <developer/>
22 </developers>
23 <scm>
24   <connection/>
25   <developerConnection/>
26   <tag/>
27   <url/>
28 </scm>
29 <properties>
30   <java.version>17</java.version>
31 </properties>
32 <dependencies>
33   <dependency>
34     <groupId>org.springframework.boot</groupId>
35     <artifactId>spring-boot-starter-jdbc</artifactId>
36   </dependency>
37   <dependency>
38     <groupId>org.springframework.boot</groupId>
39     <artifactId>spring-boot-starter-thymeleaf</artifactId>
40   </dependency>
41   <dependency>
42     <groupId>org.springframework.boot</groupId>
43     <artifactId>spring-boot-starter-web</artifactId>
44   </dependency>
45   <dependency>
46     <groupId>org.flywaydb</groupId>
47     <artifactId>flyway-mysql</artifactId>
48   </dependency>
49   <dependency>
50     <groupId>org.flywaydb</groupId>
51     <artifactId>flyway-core</artifactId>
52   </dependency>
53   <dependency>
54     <groupId>com.mysql</groupId>
55     <artifactId>mysql-connector-j</artifactId>
56   </dependency>
57   <dependency>
58     <groupId>org.mybatis.spring.boot</groupId>
59     <artifactId>mybatis-spring-boot-starter</artifactId>
60     <version>3.0.4</version>
61   </dependency>
62   <dependency>
63     <groupId>org.springframework.boot</groupId>
64     <artifactId>spring-boot-starter-test</artifactId>
65     <scope>test</scope>
66   </dependency>
```

```
67 </dependencies>
68 <build>
69   <plugins>
70     <plugin>
71       <groupId>org.springframework.boot</groupId>
72       <artifactId>spring-boot-maven-plugin</artifactId>
73     </plugin>
74   </plugins>
75 </build>
76 </project>
```

application.properties

```
1 spring.application.name=demo
2
3 spring.datasource.url=jdbc:mysql://localhost:3306/crudapp_db?useSSL=false&allowPublicKeyRetrieval=true&se
rverTimezone=UTC
4 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
5 spring.datasource.username=crudapp_user
6 spring.datasource.password=crudapp_pass
7
8 spring.flyway.enabled=true
9
10 # ローカル環境向けの SQL ログ（必要に応じて OFF に）
11 spring.jpa.show-sql=true
12
13 # MyBatis configuration
14 mybatis.mapper-locations=classpath*:mapper/**/*.xml
15 mybatis.type-aliases-package=com.example.demo
16 mybatis.configuration.map-underscore-to-camel-case=true
```

MyBatis 動作確認

- Mapper XML を読み込み
- SQL を実行
- Spring Boot 起動時のエラー確認

③ Mapper XML を 1 つ作る（疎通用）

```
src/main/resources/mapper/NoteMapper.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">
```

```
<mapper namespace="com.example.demo.mapper.NoteMapper">

    <!-- 疎通確認用：件数を取るだけ -->
    <select id="count" resultType="int">
        SELECT COUNT(*) FROM notes
    </select>

</mapper>
```

④ Mapper インターフェースを作る

src/main/java/com/example/demo/mapper/NoteMapper.java

```
package com.example.demo.mapper;

import org.apache.ibatis.annotations.Mapper;

@Mapper
public interface NoteMapper {

    int count();

}
```

起動前に以下を実行します。プロジェクトを一度きれいにしてから、Java コードをコンパイルします。

```
./mvnw clean compile
```

⑤ Spring Boot 起動（疎通チェック）

```
./mvnw spring-boot:run
```

起動エラーが出ない
コンソールに MyBatis / HikariCP / Flyway の通常ログが出る
Mapper XML 読み込みエラーが出ない

通常起動すれば成功です。

⑥ さらに一步確認)

Controller から呼んでみます。

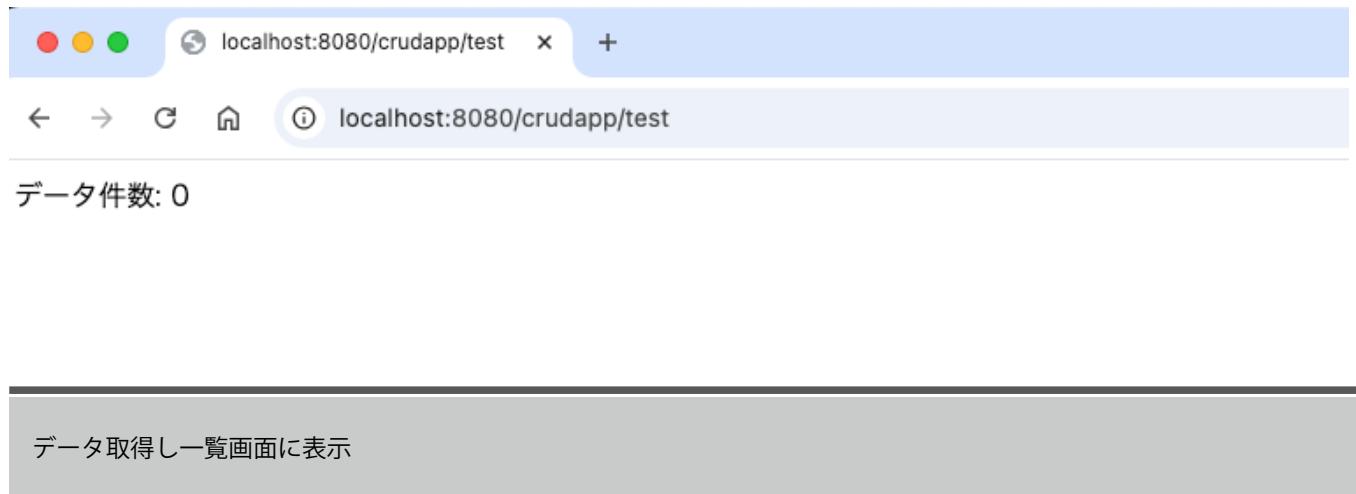
```
1 package com.example.demo;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.GetMapping;
```

```
5 import org.springframework.web.bind.annotation.PostMapping;
6 import org.springframework.web.bind.annotation.RequestMapping;
7 import org.springframework.web.bind.annotation.RequestParam;
8 import org.springframework.web.bind.annotation.ResponseBody;
9
10 import com.example.demo.mapper.NoteMapper;
11
12 import org.springframework.ui.Model;
13
14 @Controller
15 @RequestMapping("/crudapp")
16 public class CrudappController {
17
18     private final NoteMapper noteMapper;
19
20     public CrudappController(NoteMapper noteMapper) {
21         this.noteMapper = noteMapper;
22     }
23
24     @GetMapping("/index")
25     public String index() {
26         return "crudapp/index";
27     }
28
29     @GetMapping("/input")
30     public String input() {
31         return "crudapp/input";
32     }
33
34     @GetMapping("/confirm")
35     public String confirm() {
36         return "crudapp/confirm";
37     }
38
39     @PostMapping("/confirm")
40     public String confirm(@RequestParam("content") String content, Model model) {
41         model.addAttribute("content", content);
42         return "crudapp/confirm";
43     }
44
45     @GetMapping("/complete")
46     public String complete() {
47         return "crudapp/complete";
48     }
49
50     @PostMapping("/complete")
51     public String completePost() {
```

```
52     return "crudapp/complete";
53 }
54
55 @GetMapping("/test")
56 @ResponseBody
57 public String test() {
58     int count = noteMapper.count();
59     return "データ件数：" + count;
60 }
61 }
```

⑦ Spring Boot 起動（疎通チェック）

```
./mvnw spring-boot:run
```



demo/CrudappController.java

```
1 package com.example.demo;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PostMapping;
6 import org.springframework.web.bind.annotation.RequestMapping;
7 import org.springframework.web.bind.annotation.RequestParam;
8 import org.springframework.web.bind.annotation.ResponseBody;
9
10 import com.example.demo.mapper.NoteMapper;
11 import com.example.demo.model.Note;
12
13 import org.springframework.ui.Model;
14 import java.util.List;
15
16 @Controller
17 @RequestMapping("/crudapp")
18 public class CrudappController {
```

```
19
20 private final NoteMapper noteMapper;
21
22 public CrudappController(NoteMapper noteMapper) {
23     this.noteMapper = noteMapper;
24 }
25
26 @GetMapping("/index")
27 public String index() {
28     public String index(Model model) {
29         List<Note> notes = noteMapper.findAll();
30         model.addAttribute("notes", notes);
31         return "crudapp/index";
32     }
33
34 @GetMapping("/input")
35 public String input() {
36     return "crudapp/input";
37 }
38
39 @GetMapping("/confirm")
40 public String confirm() {
41     return "crudapp/confirm";
42 }
43
44 @PostMapping("/confirm")
45 public String confirm(@RequestParam("content") String content, Model model) {
46     model.addAttribute("content", content);
47     return "crudapp/confirm";
48 }
49
50 @GetMapping("/complete")
51 public String complete() {
52     return "crudapp/complete";
53 }
54
55 @PostMapping("/complete")
56 public String completePost() {
57     return "crudapp/complete";
58 }
59
60 @GetMapping("/test")
61 @ResponseBody
62 public String test() {
63     int count = noteMapper.count();
64     return "データ件数：" + count;
65 }
```

```
65 }
```

demo/mapper/NoteMapper.java

```
1 package com.example.springboot_app.mapper;
2
3 import java.util.List;
4
5 import org.apache.ibatis.annotations.Mapper;
6
7 import com.example.demo.model.Note;
8
9 @Mapper
10 public interface NoteMapper {
11
12     int count();
13
14     List<Note> findAll();
15 }
```

demo/model/Note.java NEW

```
package com.example.demo.model;

import java.time.LocalDateTime;

public class Note {
    private Long id;
    private String text;
    private LocalDateTime createdAt;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getText() {
        return text;
    }

    public void setText(String text) {
        this.text = text;
    }
}
```

```
public LocalDateTime getCreatedAt() {
    return createdAt;
}

public void setCreatedAt(LocalDateTime createdAt) {
    this.createdAt = createdAt;
}
}
```

resources/mapper/NoteMapper.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE mapper
3 PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
4 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
5
6 <mapper namespace="com.example.demo.mapper.NoteMapper">
7
8   <!-- 疎通確認用：件数を取るだけ -->
9   <select id="count" resultType="int">
10     SELECT COUNT(*) FROM notes
11   </select>
12
13   <select id="findAll" resultType="com.example.demo.model.Note">
14     SELECT
15       id,
16       text,
17       created_at
18     FROM notes
19     ORDER BY id DESC
20   </select>
21 </mapper>
```

resources/templete/crudapp/index.html

```
1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <meta charset="UTF-8">
5   <title>CRUD App - 一覧 </title>
6 </head>
7 <body>
8 <h1>CRUD App: 一覧 </h1>
9 <p> ここに一覧を表示します。</p>
10<nav>
11  <ul>
```

```

12   <li><a href="/crudapp/input">新規登録 </a></li>
13   <li><a href="/crudapp/confirm">登録確認 </a></li>
14   <li><a href="/crudapp/complete">登録完了 </a></li>
15 </ul>
16 <a href="/crudapp">一覧へ </a>
17</nav>
9 <div>
10 <a href="/crudapp/input">新規登録 </a>
11 </div>
12
13 <div th:if="${#lists.isEmpty(notes)}">
14   <p>データがありません </p>
15 </div>
16
17 <table border="1" cellspacing="0" cellpadding="8" th:if="${!#lists.isEmpty(notes)}">
18   <thead>
19     <tr>
20       <th>ID</th>
21       <th>Text</th>
22       <th>Created At</th>
23     </tr>
24   </thead>
25   <tbody>
26   <tr th:each="note : ${notes}">
27     <td th:text="${note.id}">1</td>
28     <td th:text="${note.text}">サンプル </td>
29     <td th:text="${note.createdAt}">2024-01-01T00:00:00</td>
30   </tr>
31 </tbody>
32 </table>
33 </body>
34 </html>

```

フォームデータを登録

2) NoteMapper に insert を追加

demo/mapper/NoteMapper.java

```

1 package com.example.demo.mapper;
2
3 import java.util.List;
4
5 import org.apache.ibatis.annotations.Mapper;
6

```

```
7 import com.example.demo.model.Note;
8
9 @Mapper
10 public interface NoteMapper {
11
12     int count();
13
14     List<Note> findAll();
15
16     int insert(String text);
17 }
```

3) NoteMapper.xml に INSERT を追加

resources/mapper/NoteMapper.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE mapper
3 PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
4 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
5
6 <mapper namespace="com.example.springboot_app.mapper.NoteMapper">
7
8     <!-- 疎通確認用：件数を取るだけ -->
9     <select id="count" resultType="int">
10        SELECT COUNT(*) FROM notes
11    </select>
12
13     <select id="findAll" resultType="com.example.springboot_app.model.Note">
14        SELECT
15            id,
16            text,
17            created_at
18        FROM notes
19        ORDER BY id DESC
20    </select>
21
22     <insert id="insert" parameterType="string">
23        INSERT INTO notes (text)
24        VALUES (#{$text})
25    </insert>
26 </mapper>
```

4) Controller の completePost で登録する

demo/CrudappController.java

```
1 package com.example.demo;
```

```
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PostMapping;
6 import org.springframework.web.bind.annotation.RequestMapping;
7 import org.springframework.web.bind.annotation.RequestParam;
8 import org.springframework.web.bind.annotation.ResponseBody;
9
10 import com.example.demo.mapper.NoteMapper;
11 import com.example.demo.model.Note;
12
13 import org.springframework.ui.Model;
14 import java.util.List;
15
16 @Controller
17 @RequestMapping("/crudapp")
18 public class CrudappController {
19
20     private final NoteMapper noteMapper;
21
22     public CrudappController(NoteMapper noteMapper) {
23         this.noteMapper = noteMapper;
24     }
25
26     @GetMapping("index")
27     public String index(Model model) {
28         List<Note> notes = noteMapper.findAll();
29         model.addAttribute("notes", notes);
30         return "crudapp/index";
31     }
32
33     @GetMapping("/input")
34     public String input() {
35         return "crudapp/input";
36     }
37
38     @GetMapping("/confirm")
39     public String confirm() {
40         return "crudapp/confirm";
41     }
42
43     @PostMapping("/confirm")
44     public String confirm(@RequestParam("content") String content, Model model) {
45         model.addAttribute("content", content);
46         return "crudapp/confirm";
47     }
48
```

```
49 @GetMapping("/complete")
50 public String complete() {
51     return "crudapp/complete";
52 }
53
54 @PostMapping("/complete")
55 public String completePost() {
56     return "crudapp/complete";
57 }
58 public String completePost(@RequestParam("content") String content) {
59     noteMapper.insert(content);
60     return "redirect:/crudapp";
61 }
62
63 @GetMapping("/test")
64 @ResponseBody
65 public String test() {
66     int count = noteMapper.count();
67     return "データ件数：" + count;
68 }
```

登録されたデータを編集

demo/CrudappController.java

```
1 package com.example.demo;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PostMapping;
6 import org.springframework.web.bind.annotation.RequestMapping;
7 import org.springframework.web.bind.annotation.RequestParam;
8 import org.springframework.web.bind.annotation.ResponseBody;
9
10 import com.example.demo.mapper.NoteMapper;
11 import com.example.demo.model.Note;
12
13 import org.springframework.ui.Model;
14 import java.util.List;
15
16 @Controller
17 @RequestMapping("/crudapp")
18 public class CrudappController {
19
20     private final NoteMapper noteMapper;
21 }
```

```

22 public CrudappController(NoteMapper noteMapper) {
23     this.noteMapper = noteMapper;
24 }
25
26 @GetMapping("/index")
27 public String index(Model model) {
28     List<Note> notes = noteMapper.findAll();
29     model.addAttribute("notes", notes);
30     return "crudapp/index";
31 }
32
33 @GetMapping("/input")
34 public String input() {
35     return "crudapp/input";
36 }
37
38 @GetMapping("/confirm")
39 public String confirm() {
40     return "crudapp/confirm";
41 }
42
43 @PostMapping("/confirm")
44 public String confirm(@RequestParam("content") String content, Model model) {
45     model.addAttribute("content", content);
46     return "crudapp/confirm";
47 }
48
49 @GetMapping("/complete")
50 public String complete() {
51     return "crudapp/complete";
52 }
53
54 @PostMapping("/complete")
55 public String completePost(@RequestParam("content") String content) {
56     noteMapper.insert(content);
57     return "redirect:/crudapp";
58 }
59
60 @GetMapping("/edit")
61 public String edit(@RequestParam("id") Long id, Model model) {
62     Note note = noteMapper.findById(id);
63     model.addAttribute("note", note);
64     return "crudapp/edit";
65 }
66
67 @PostMapping("/edit")
68 public String update(@RequestParam("id") Long id, @RequestParam("content") String content) {

```

```
69 Note note = new Note();
70 note.setId(id);
71 note.setText(content);
72 noteMapper.update(note);
73 return "redirect:/crudapp";
74 }
75
76 @GetMapping("/test")
77 @ResponseBody
78 public String test() {
79     int count = noteMapper.count();
80     return "データ件数：" + count;
81 }
82 }
```

demo/mapper/NoteMapper.java

```
1 package com.example.demo.mapper;
2
3 import java.util.List;
4
5 import org.apache.ibatis.annotations.Mapper;
6
7 import com.example.springboot_app.model.Note;
8
9 @Mapper
10 public interface NoteMapper {
11
12     int count();
13
14     List<Note> findAll();
15
16     int insert(String text);
17
18     Note findById(Long id);
19
20     int update(Note note);
21 }
```

resources/mapper/NoteMapper.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE mapper
3 PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
4 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
5
```

```

6 <mapper namespace="com.example.springboot_app.mapper.NoteMapper">
7
8   <!-- 疎通確認用：件数を取るだけ -->
9   <select id="count" resultType="int">
10    SELECT COUNT(*) FROM notes
11  </select>
12
13 <select id="findAll" resultType="com.example.springboot_app.model.Note">
14   SELECT
15   id,
16   text,
17   created_at
18   FROM notes
19   ORDER BY id DESC
20 </select>
21
22 <insert id="insert" parameterType="string">
23   INSERT INTO notes (text)
24   VALUES (#{{text}})
25 </insert>
26
27 <select id="findById" parameterType="long" resultType="com.example.demo.model.Note">
28   SELECT
29   id,
30   text,
31   created_at
32   FROM notes
33   WHERE id = #{{id}}
34 </select>
35
36 <update id="update" parameterType="com.example.demo.model.Note">
37   UPDATE notes
38   SET text = #{{text}}
39   WHERE id = #{{id}}
40 </update>
41
42 </mapper>

```

resources/templates/crudapp/index.html

```

1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <meta charset="UTF-8">
5   <title>CRUD App - 一覧 </title>
6 </head>

```

```

7 <body>
8 <h1>CRUD App: 一覧 </h1>
9 <div>
10 <a href="/crudapp/input"> 新規登録 </a>
11 </div>
12
13 <div th:if="#lists.isEmpty(notes)">
14 <p> データがありません </p>
15 </div>
16
17 <table border="1" cellspacing="0" cellpadding="8" th:if="#!lists.isEmpty(notes)">
18 <thead>
19 <tr>
20 <th>ID</th>
21 <th>Text</th>
22 <th>Created At</th>
23 </tr>
24 </thead>
25 <tbody>
26 <tr th:each="note : ${notes}">
27 <td th:text="${note.id}">1</td>
28 <td th:text="${note.text}"> サンプル </td>
29 <td th:text="${note.createdAt}">2024-01-01T00:00:00</td>
30 <td>
31 <a th:href="@{/crudapp/edit(id=${note.id})}"> 編集 </a>
32 </td>
33 </tr>
34 </tbody>
35 </table>
36 </body>
37 </html>

```

resources/templates/crudapp/edit.html NEW

```

<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <title>CRUD App - 編集 </title>
</head>
<body>
<h1>CRUD App: 編集 </h1>
<form th:action="@{/crudapp/edit}" method="post">
  <input type="hidden" name="id" th:value="${note.id}">
  <label for="content"> 内容 </label><br>
  <textarea id="content" name="content" rows="6" cols="50" required
    th:text="${note.text}"> 既存の内容 </textarea><br>

```

```
<button type="submit"> 更新する </button>
</form>
<nav>
  <a href="/crudapp">一覧へ戻る </a>
</nav>
</body>
</html>
```

登録されたデータを削除

springboot_app/CrudappController.java

```
1 package com.example.springboot_app;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.PostMapping;
6 import org.springframework.web.bind.annotation.RequestMapping;
7 import org.springframework.web.bind.annotation.RequestParam;
8 import org.springframework.web.bind.annotation.ResponseBody;
9
10 import com.example.springboot_app.mapper.NoteMapper;
11 import com.example.springboot_app.model.Note;
12
13 import org.springframework.ui.Model;
14 import java.util.List;
15
16 @Controller
17 @RequestMapping("/crudapp")
18 public class CrudappController {
19
20   private final NoteMapper noteMapper;
21
22   public CrudappController(NoteMapper noteMapper) {
23     this.noteMapper = noteMapper;
24   }
25
26   @GetMapping("/index")
27   public String index(Model model) {
28     List<Note> notes = noteMapper.findAll();
29     model.addAttribute("notes", notes);
30     return "crudapp/index";
31   }
32
33   @GetMapping("/input")
34   public String input() {
```

```
35     return "crudapp/input";
36 }
37
38 @GetMapping("/confirm")
39 public String confirm() {
40     return "crudapp/confirm";
41 }
42
43 @PostMapping("/confirm")
44 public String confirm(@RequestParam("content") String content, Model model) {
45     model.addAttribute("content", content);
46     return "crudapp/confirm";
47 }
48
49 @GetMapping("/complete")
50 public String complete() {
51     return "crudapp/complete";
52 }
53
54 @PostMapping("/complete")
55 public String completePost(@RequestParam("content") String content) {
56     noteMapper.insert(content);
57     return "redirect:/crudapp";
58 }
59
60 @GetMapping("/edit")
61 public String edit(@RequestParam("id") Long id, Model model) {
62     Note note = noteMapper.findById(id);
63     model.addAttribute("note", note);
64     return "crudapp/edit";
65 }
66
67 @PostMapping("/edit")
68 public String update(@RequestParam("id") Long id, @RequestParam("content") String content) {
69     Note note = new Note();
70     note.setId(id);
71     note.setText(content);
72     noteMapper.update(note);
73     return "redirect:/crudapp";
74 }
75
76 @PostMapping("/delete")
77 public String delete(@RequestParam("id") Long id) {
78     noteMapper.softDelete(id);
79     return "redirect:/crudapp";
80 }
81
```

```
82 @PostMapping("/toggle")
83 public String toggleVisibility(@RequestParam("id") Long id, @RequestParam("isVisible") Integer isVisible) {
84     int newVisible = (isVisible != null && isVisible == 1) ? 0 : 1;
85     noteMapper.toggleVisibility(id, newVisible);
86     return "redirect:/crudapp";
87 }
88
89 @GetMapping("/test")
90 @ResponseBody
91 public String test() {
92     int count = noteMapper.count();
93     return "データ件数：" + count;
94 }
95 }
```

springboot_app/mapper/NoteMapper.java

```
1 package com.example.springboot_app.mapper;
2
3 import java.util.List;
4
5 import org.apache.ibatis.annotations.Mapper;
6 import org.apache.ibatis.annotations.Param;
7
8 import com.example.springboot_app.model.Note;
9
10 @Mapper
11 public interface NoteMapper {
12
13     int count();
14
15     List<Note> findAll();
16
17     int insert(String text);
18
19     Note findById(Long id);
20
21     int update(Note note);
22
23     int softDelete(Long id);
24
25     int toggleVisibility(@Param("id") Long id, @Param("isVisible") Integer isVisible);
26 }
```

demo/model/Note.java

```
1 package com.example.demo.model;
```

```
2
3 import java.time.LocalDateTime;
4
5 public class Note {
6     private Long id;
7     private String text;
8     private LocalDateTime createdAt;
9     private Integer isDeleted;
10    private Integer isVisible;
11
12    public Long getId() {
13        return id;
14    }
15
16    public void setId(Long id) {
17        this.id = id;
18    }
19
20    public String getText() {
21        return text;
22    }
23
24    public void setText(String text) {
25        this.text = text;
26    }
27
28    public LocalDateTime getCreatedAt() {
29        return createdAt;
30    }
31
32    public void setCreatedAt(LocalDateTime createdAt) {
33        this.createdAt = createdAt;
34    }
35
36    public Integer getIsDeleted() {
37        return isDeleted;
38    }
39
40    public void setIsDeleted(Integer isDeleted) {
41        this.isDeleted = isDeleted;
42    }
43
44    public Integer getIsVisible() {
45        return isVisible;
46    }
47
48    public void setIsVisible(Integer isVisible) {
```

```
49     this.isVisible = isVisible;
50 }
51 }
```

resources/db/migration/V2__add_flags_to_notes.sql NEW

```
1 ALTER TABLE notes
2   ADD COLUMN is_deleted TINYINT(1) NOT NULL DEFAULT 0 AFTER created_at,
3   ADD COLUMN is_visible TINYINT(1) NOT NULL DEFAULT 1 AFTER is_deleted;
```

resources/mapper/NoteMapper.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE mapper
3 PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
4 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
5
6 <mapper namespace="com.example.springboot_app.mapper.NoteMapper">
7
8   <!-- 疎通確認用：件数を取るだけ -->
9   <select id="count" resultType="int">
10    SELECT COUNT(*) FROM notes
11   </select>
12
13  <select id="findAll" resultType="com.example.springboot_app.model.Note">
14    SELECT
15      id,
16      text,
17      created_at
17      created_at,
18      is_deleted,
19      is_visible
20    FROM notes
21    WHERE is_deleted = 0
22    AND is_visible = 1
23    ORDER BY id DESC
24   </select>
25
26  <insert id="insert" parameterType="string">
27    INSERT INTO notes (text)
28    VALUES (#{{text}})
29   </insert>
30
31  <select id="findById" parameterType="long" resultType="com.example.springboot_app.model.Note">
32    SELECT
33      id,
```

```

34     text,
35     created_at
35     created_at,
36     is_deleted,
37     is_visible
38   FROM notes
39   WHERE id = #{id}
40 </select>
41
42 <update id="update" parameterType="com.example.springboot_app.model.Note">
43   UPDATE notes
44   SET text = #{text}
45   WHERE id = #{id}
46 </update>
47
48 <update id="softDelete" parameterType="long">
49   UPDATE notes
50   SET is_deleted = 1
51   WHERE id = #{id}
52 </update>
53
54 <update id="toggleVisibility">
55   UPDATE notes
56   SET is_visible = #{isVisible}
57   WHERE id = #{id}
58 </update>
59 </mapper>

```

resources/templates/crudapp/index.html

```

1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <meta charset="UTF-8">
5   <title>CRUD App - 一覧 </title>
6 </head>
7 <body>
8 <h1>CRUD App: 一覧 </h1>
9 <div>
10 <a href="/crudapp/input">新規登録 </a>
11 </div>
12
13 <div th:if="${#lists.isEmpty(notes)}">
14   <p>データがありません </p>
15 </div>
16
17 <table border="1" cellspacing="0" cellpadding="8" th:if="${!#lists.isEmpty(notes)}">

```

```

18 <thead>
19 <tr>
20   <th>ID</th>
21   <th>Text</th>
22   <th>Created At</th>
23   <th> 編集 </th>
24   <th> 削除 </th>
25   <th> 表示切替 </th>
26 </tr>
27 </thead>
28 <tbody>
29 <tr th:each="note : ${notes}">
30   <td th:text="${note.id}">1</td>
31   <td th:text="${note.text}"> サンプル </td>
32   <td th:text="${note.createdAt}">2024-01-01T00:00:00</td>
33   <td><a th:href="@{/crudapp/edit(id=${note.id})}"> 編集 </a></td>
34   <td>
35     <form th:action="@{/crudapp/delete}" method="post">
36       <input type="hidden" name="id" th:value="${note.id}">
37       <button type="submit"> 削除 </button>
38     </form>
39   </td>
40   <td>
41     <a th:href="@{/crudapp/edit(id=${note.id})}"> 編集 </a>
42     <form th:action="@{/crudapp/toggle}" method="post">
43       <input type="hidden" name="id" th:value="${note.id}">
44       <input type="hidden" name="isVisible" th:value="${note.isVisible}">
45       <button type="submit" th:text="${note.isVisible == 1} ? '非表示にする' : '表示する'"> 表示切替 </button>
46   </td>
47 </tr>
48 </tbody>
49 </table>
50 </body>
51 </html>

```