

Day4：勤怠一覧画面とDB確認

JPAクエリメソッドとH2コンソール

新人エンジニア研修 講師

本日のゴール

- **履歴の可視化**

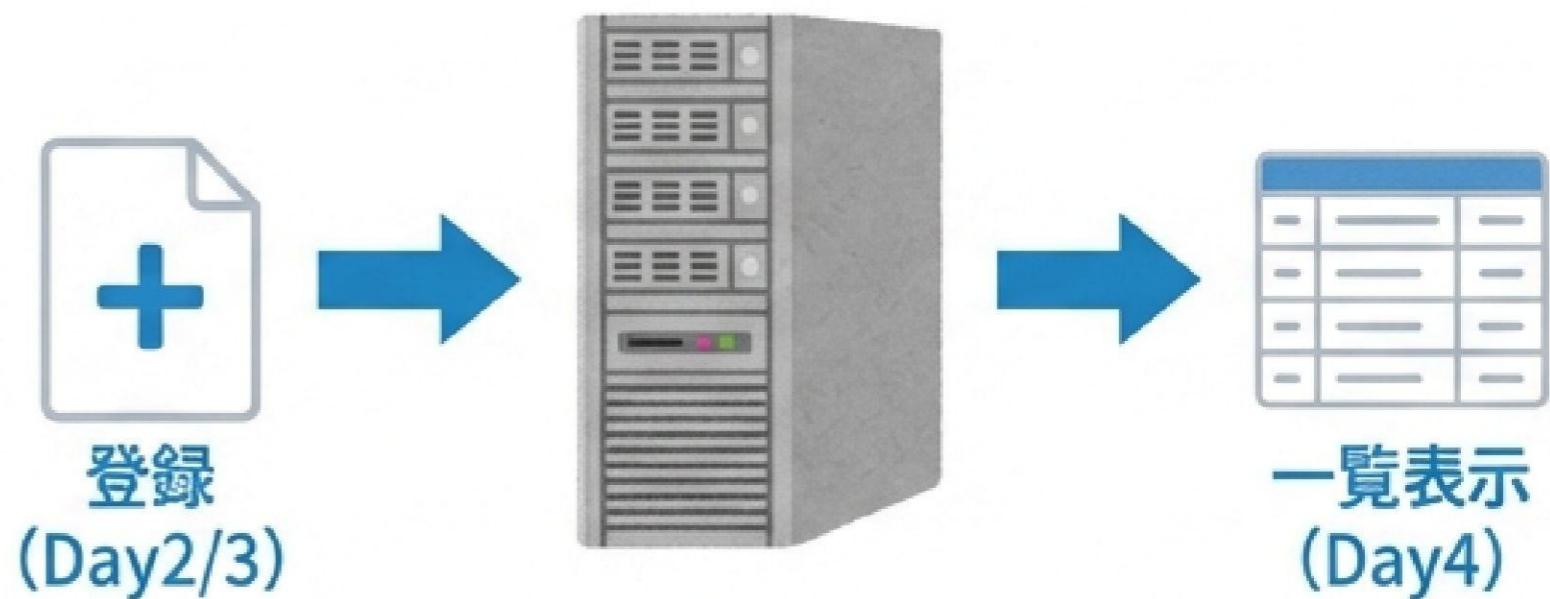
登録した勤怠データを一覧ページで表示する（降順）。

- **データフローの実装**

Repository → Service → Controller → View のデータ取得フローを構築する。

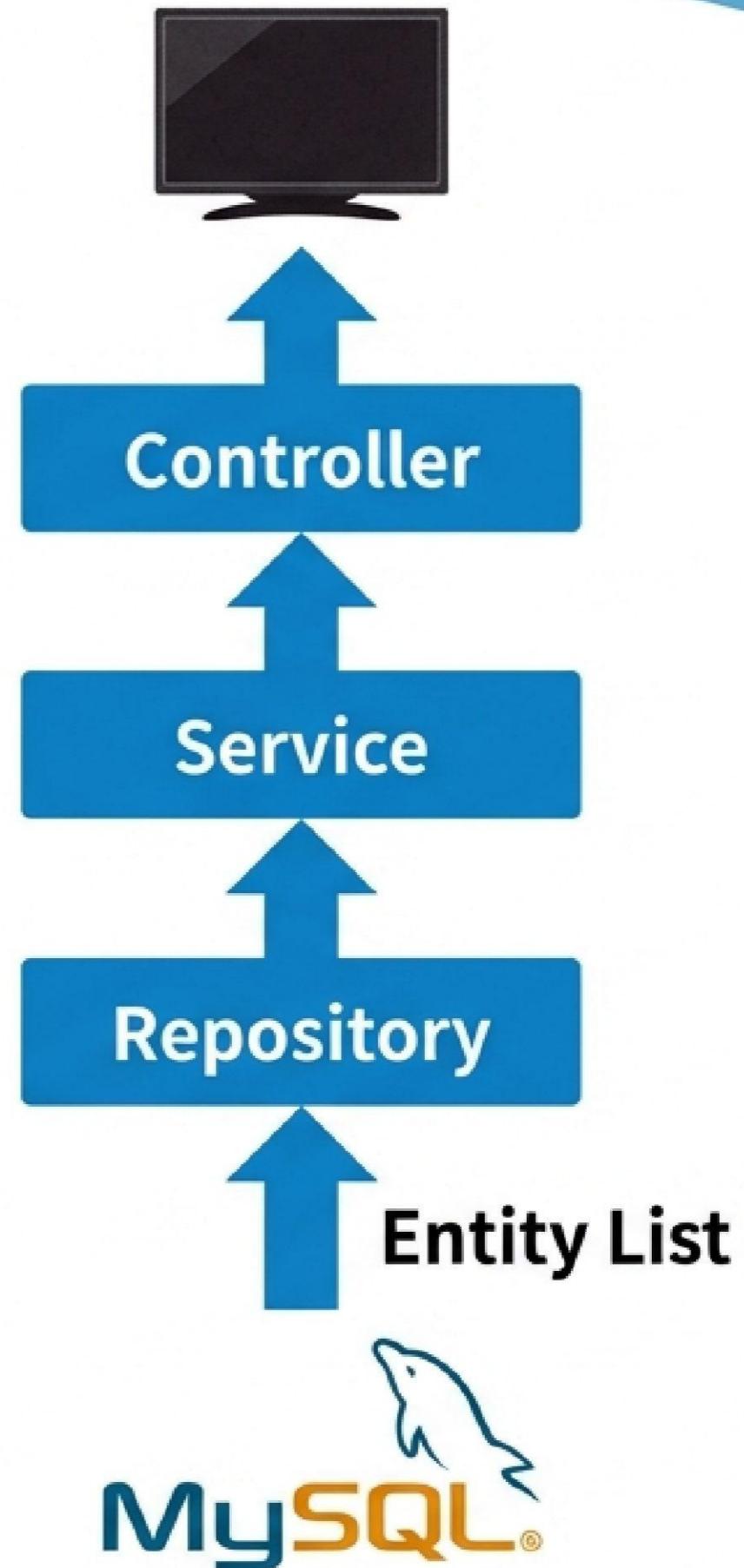
- **DBの裏側**

H2コンソールを使って、アプリケーションの裏側にあるDBデータを直接SQLで確認する。



データ取得のフロー

- **Repository**
DBから条件に合うデータを検索・取得。
- **Service**
業務ロジックの適用（今回はパスルー）。
- **Controller**
画面表示用にデータをModelへ詰める。
- **View**
ユーザーにHTMLとしてデータを表示。



Repository - クエリメソッド

Query Method

メソッド名からSQLが自動生成される仕組み。

命名規則

findBy + 検索条件 + OrderBy + ソート順

今回の要件

ユーザーIDで絞り込み、日付と時刻で「降順
(新しい順)」に並べる。

AttendanceRepository.java

```
// ユーザーIDで検索し、日付・開始時刻の降順で取得
List<Attendance> findByUserIdOrderByWorkDateDescStartTimeDesc(
    int userId
);
```

降順 (Descending)

Service - 一覧取得ロジック

- **役割**

Controllerから呼び出され、Repositoryを実行する。

- **処理内容**

今回はデータの加工がないため、Repositoryの結果をそのまま返す（パススルー）。

- **拡張性**

将来的にデータの加工が必要になればここに記述する。

AttendanceService.java

```
public List<Attendance> getAttendanceList(User user) {  
    // Repositoryのメソッドを呼び出すだけ  
    return attendanceRepository  
        .findByUserIdOrderByWorkDateDescStartTimeDesc(  
            user.getId()  
        );  
}
```

Controller - 一覧画面用

- **新規作成**

AttendanceController クラスを作成。

- **マッピング**

@GetMapping("/attendances") で URL
を定義。

- **Modelへの登録**

Serviceから取得したリストを
"attendances" というキーで画面に渡
す。

AttendanceController.java

```
@GetMapping("/attendances")
public String index(Model model) {
    User user = ... // ユーザー取得

    // Serviceからリストを取得
    List<Attendance> list = service.getAttendanceList(user);

    // Modelに詰めてHTMLへ渡す
    model.addAttribute("attendances", list);

    return "attendances";
}
```

View - Thymeleafでの繰り返し

- **th:each**

Javaの for-each 文と同じ動きをする属性。

- **仕組み**

リスト (attendances) の件数分、<tr> (表の行) を繰り返し生成する。

- **変数**

\${attendance} にループ中の1件ずつのデータが入る。

attendances.html

```
<!-- リストの件数分、trが繰り返される -->
|
    <td th:text="${attendance.workDate}"></td>
    <td th:text="${attendance.startTime}"></td>
    <td th:text="${attendance.endTime}"></td>
    <td th:text="${attendance.status}"></td>

|  |

```

トップ画面からの遷移

- **導線**
トップページ (index.html) から一覧ページへ行けるようにする。
- **Thymeleaf URL**
@{/attendances} と書く。
- **メリット**
@{...} 記法を使うことで、コンテキストパス（アプリのルートパス）を自動調整してくれる。

index.html

```
<!-- 既存のパネル内に追加 -->

```

動作確認 - 画面

1. トップ画面
(<http://localhost:8080/>) を開く。
2. 「出勤」 → 「退勤」を数回行い、
データを増やす。
3. 追加したリンク「勤怠一覧を見る」
をクリック。
4. 確認：新しい日付・時刻が表の上に
来ているか（降順）確認する。



動作確認 - H2コンソール

- **H2コンソール**

メモリ内データベースの中身を直接
SQLで確認できる管理画面。

- **Access**

<http://localhost:8080/h2-console>

- **設定 (重要)**

JDBC URL: jdbc:h2:mem:attendance

User Name: sa

Password: (空欄)

H2 Database Login

Saved Settings: - Saved Settings - Save It

Setting Name:

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:attendance

User Name: sa

Password:

Connect Saved Settings

SELECT result

ID	WORK_DATE	START_TIME	END_TIME	STATUS
3	2023-07-14	09:00:00	18:00:00	退勤済
2	2023-07-13	09:05:30	18:15:00	退勤済
1	2023-07-12	08:55:10	17:50:00	退勤済

よくあるエラー

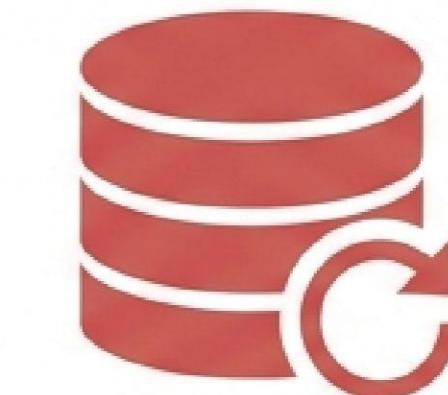
TemplateInputException

原因: `attendances.html` の配置場所ミス、またはControllerでのファイル名指定ミス。



一覧が空 (Empty List)

原因: アプリ再起動でH2メモリDBがリセットされている。先にトップ画面で「出勤/退勤」を行う必要がある。



H2接続エラー

原因: JDBC URLのスペルミス (`test` ではなく `attendance` になっているか確認)。



まとめ

- **Read実装**

データを保存するだけでなく、取り出して表示できた。

- **CRUDの基礎**

Create (登録) と Read (参照) が揃い、アプリらしくなった。

- **Next Step**

明日以降は、より複雑な機能やインフラ (Docker) へ。

研修お疲れ様でした！



Day 4 Complete