Tutorial 03 – Model, Service

Tutorial ini mengasumsikan penggunaan pada *Windows*, tanpa melarang penggunaan OS lain, silakan menyesuaikan.

1. Pendahuluan
   - Jalankan STS
   - Buat *project* baru:
     - Nama : tutorial-03
     - Package: com.apap.tu03
     - *Dependencies*: DevTools, Thymeleaf, dan Web
2. Membuat Model
   Salah satu komponen dalam MVC adalah model
   - Buatlah *package* baru dengan nama **com.apap.tu03.model**
   - Pada *package* model buatlah *class* Movie dengan spesifikasi sbb.:

```
package com.apap.tu03.model;

public class MovieModel {
    private String id;
    private String title;
    private String genre;
    private Long budget;
    private Integer duration; // minutes
}
```

   - Tambahkan *method constructor*, *setter*, dan *getter*.
3. Membuat Service
   Service ini merupakan *interface* yang mendefinisikan *method* apa saja yang dapat dilakukan untuk memanipulasi kelas model.
   - Buatlah *package* baru dengan nama **com.apap.tu03.service**
   - Pada *package* service buatlah *interface* MovieService dengan spesifikasi sbb.:

```
package com.apap.tu03.service;

import java.util.List;
import com.apap.tu03.model.MovieModel;

public interface MovieService {
    void addMovie(MovieModel movie);
    List<MovieModel> getMovieList();
    MovieModel getMovieDetail(String id);
}
```

   - Pada *package* yang sama buatlah *class* InMemoryMovieService dengan spesifikasi sbb.:

```
package com.apap.tu03.service;

import java.util.ArrayList;

@Service
public class InMemoryMovieService implements MovieService {
    private List<MovieModel> archiveMovie;

    public InMemoryMovieService() {
        archiveMovie = new ArrayList<>();
    }

    @Override
    public void addMovie(MovieModel movie) {
        archiveMovie.add(movie);
    }

    @Override
    public List<MovieModel> getMovieList() {
        return archiveMovie;
    }
```

- Implementasikan *method* getMovieDetail. *Method* ini menerima id movie dan mengembalikan *object* Movie dengan id tersebut. *Return* null jika tidak ditemukan.

4. Membuat Controller dan Method Add
   - Buatlah package baru dengan nama **com.apap.tu03.controller**
   - Pada *package* controller buatlah *class* MovieController dengan spesifikasi sbb.:

```
package com.apap.tu03.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;

import com.apap.tu03.model.MovieModel;
import com.apap.tu03.service.MovieService;

@Controller
public class MovieController {
    @Autowired
    private MovieService movieService;

    @RequestMapping("/movie/add")
    public String add(@RequestParam(value = "id", required = true) String id,
            @RequestParam(value = "title", required = true) String title,
            @RequestParam(value = "genre", required = true) String genre,
            @RequestParam(value = "budget", required = true) Long budget,
            @RequestParam(value = "duration", required = true) Integer duration) {
        MovieModel movie = new MovieModel(id, title, genre, budget, duration);
        movieService.addMovie(movie);
        return "add";
    }
}
```

   - Buatlah *view* add.html dengan isi sebagai berikut:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>add</title>
    </head>
    <body>
        <h2>Data berhasil ditambahkan!</h2>
    </body>
</html>
```

   - *Run* project, lalu akses:
     - http://localhost:8080/movie/add?id=1&title=Marvel&genre=Action&budget=50000000&duration=50
       Q1: Apakah hasilnya? Jelaskan apa yang terjadi (lengkapi dengan *screen capture*).
     - http://localhost:8080/movie/add?id=1&title=Marvel&genre=Action&budget=50000000
       Q2: Apakah hasilnya? Jelaskan apa yang terjadi (lengkapi dengan *screen capture*).

5. Membuat Method View by ID
   - Pada *class* MovieController tambahkan *method* berikut:

```
@RequestMapping("/movie/view")
public String view(@RequestParam("id") String id, Model model) {
    MovieModel archive = movieService.getMovieDetail(id);
    model.addAttribute("movie", archive);
    return "view-movie";
}
```

- Buatlah *view* view-movie.html dengan isi sebagai berikut:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>View Detail</title>
    </head>
    <body>
        <h2>Data</h2>
        <h3 th:text="${movie.id}"></h3>
        <h3 th:text="${movie.title}"></h3>
        <h3 th:text="${movie.genre}"></h3>
        <h3 th:text="${movie.budget}"></h3>
        <h3 th:text="${movie.duration}"></h3>
    </body>
</html>
```

- Run project, lalu akses
  - http://localhost:8080/movie/add?id=2&title=Dilan&genre=Romance&budget=10000000&duration=110
  - http://localhost:8080/movie/view?id=2
    Q3: Apakah data movie tersebut muncul? Jelaskan apa yang terjadi (lengkapi dengan *screen capture*).
  - Tambahkan data movie lainnya dengan id yang berbeda.

6. Membuat Method View All
   - Pada *class* MovieController tambahkan *method* berikut:

```
@RequestMapping("/movie/viewall")
public String viewAll(Model model) {
    List<MovieModel> archive = movieService.getMovieList();
    model.addAttribute("movies", archive);
    return "viewall-movie";
}
```

   - Buatlah *view* viewall-movie.html dengan isi sebagai berikut:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>View All</title>
    </head>
    <body>
        <h1>Data</h1>
        <div th:each="movie : ${movies}">
            <h2 th:text="${movie.id}"></h2>
            <h2 th:text="${movie.title}"></h2>
            <h2 th:text="${movie.genre}"></h2>
            <h2 th:text="${movie.budget}"></h2>
            <h2 th:text="${movie.duration}"></h2>
        </div>
    </body>
</html>
```

   - Run project, lalu akses
     - http://localhost:8080/movie/add?id=3&title=Warkop&genre=Comedy&budget=30000000&duration=70
     - http://localhost:8080/movie/viewall
       Q4: Apakah data movie tersebut muncul? Jelaskan apa yang terjadi (lengkapi dengan *screen capture*).
     - Tambahkan data movie lainnya dengan id yang berbeda. Lalu akses http://localhost:8080/movie/viewall

Q5: Apakah semua data Movie muncul? Jelaskan apa yang terjadi (lengkapi dengan screen capture).

Latihan

1. Pada MovieController tambahkan sebuah *method* view Movie dengan menggunakan **PathVariable**. Misalnya, Anda ingin memasukkan data sebuah Movie yang memiliki id 1, untuk melihat data yang baru dimasukkan tersebut dapat mengakses halaman:
   localhost:8080/movie/view/1
   Jika nomor id tidak diberikan atau tidak ditemukan kembalikan halaman *error* yang berisi informasi bahwa nomor id kosong atau tidak ditemukan.
2. Tambahkan fitur untuk melakukan update duration dari Movie berdasarkan id. Misalnya, setelah melakukan add Movie pada soal nomor 1, cobalah untuk mengubah duration objek Movie tersebut menjadi 100 dengan mengakses halaman:
   localhost:8080/movie/update/1/duration/100
   Tampilkan sebuah halaman yang memberikan informasi bahwa data tersebut telah berhasil diubah. Jika nomor id tidak diberikan atau tidak ditemukan kembalikan halaman *error* yang berisi informasi bahwa nomor id kosong atau tidak ditemukan dan proses *update* dibatalkan.
3. Tambahkan fitur untuk melakukan delete Movie berdasarkan id. Misalnya, setelah melakukan perintah nomor 1 dan 2, cobalah untuk melakukan delete data tersebut dengan mengakses halaman:
   localhost:8080/movie/delete/1
   Tampilkan sebuah halaman yang memberikan informasi bahwa data tersebut telah berhasil dihapus. Jika nomor id tidak diberikan atau tidak ditemukan kembalikan halaman error yang berisi informasi bahwa nomor id kosong atau tidak ditemukan dan proses *delete* dibatalkan.

Pengumpulan

1. Jawablah soal yang ditandai dengan Q berwarna hijau.
2. *Screen capture* dari soal latihan baik pada kasus berhasil dan kasus gagal.

Pengumpulan 1 dan 2 dituliskan dalam satu file dengan format **npm_nama-lengkap-anda.pdf** dan unggah ke submission slot yang disediakan di Scele.

3. Folder tutorial-03 dikumpulkan dengan cara push ke GitHub (https://github.com/achmad-f-abka/apap).