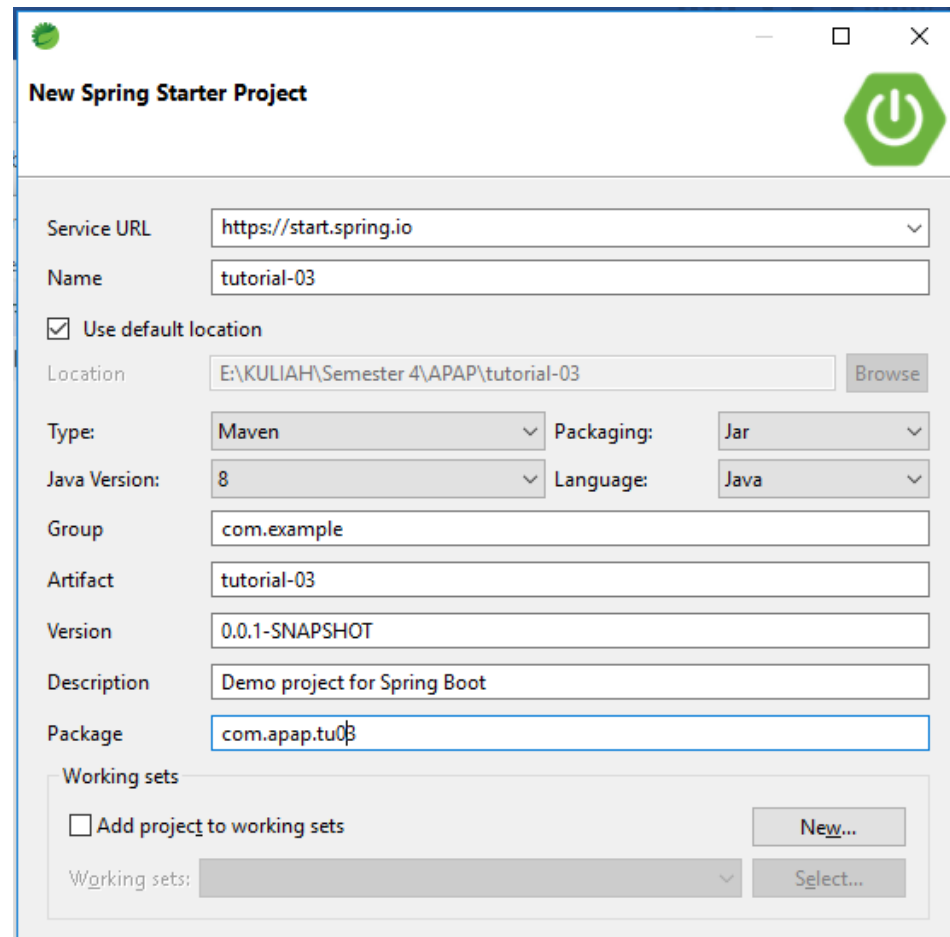


I. Pendahuluan

- Jalankan STS
- Buat project baru dengan nama tutorial-03 dan package com.apap.tu3



New Spring Starter Project

Service URL:

Name:

☒ Use default location

Location:

Type: Packaging:

Java Version: Language:

Group:

Artifact:

Version:

Description:

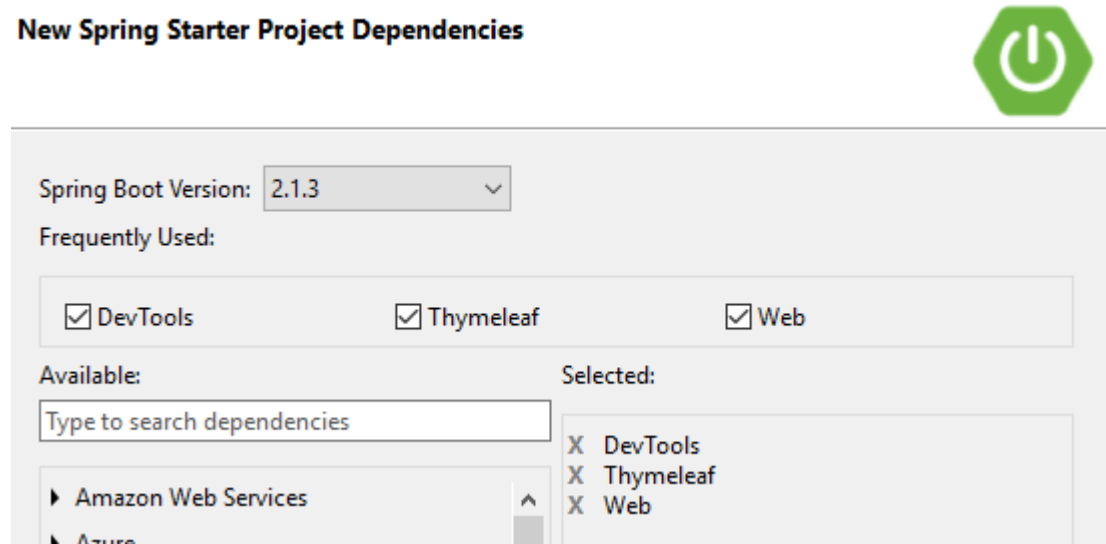
Package:

Working sets

☐ Add project to working sets

Working sets:

- Pilih dependency DevTools, Thymeleaf, dan Web



New Spring Starter Project Dependencies

Spring Boot Version:

Frequently Used:

☒ DevTools ☒ Thymeleaf ☒ Web

Available:

Type to search dependencies

Amazon Web Services

Azure

Selected:

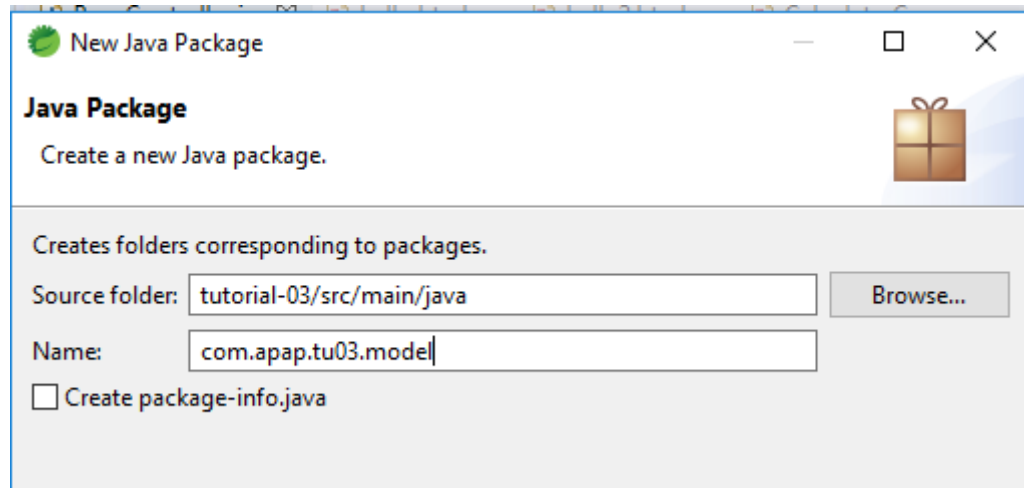
X DevTools

X Thymeleaf

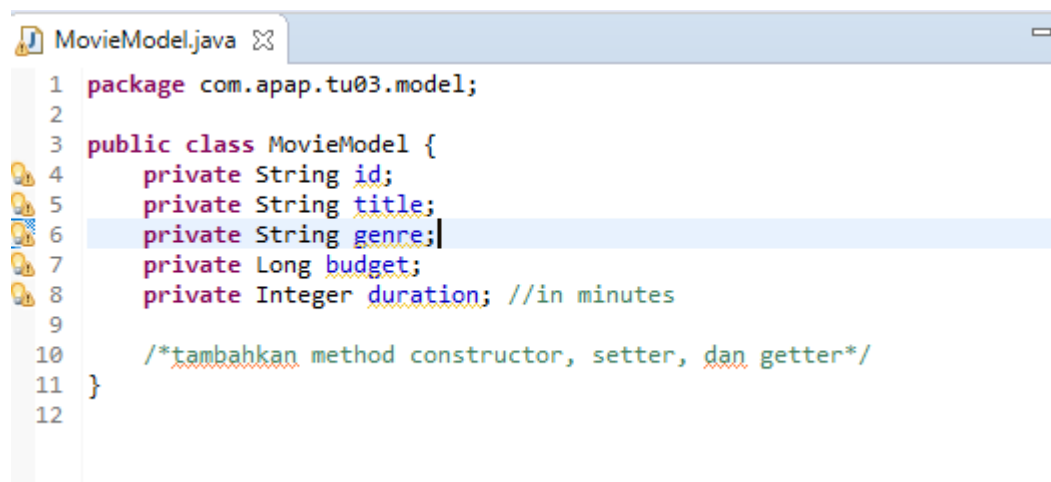
X Web

II. Membuat Model

- Buat package dengan nama com.apap.tu3.model

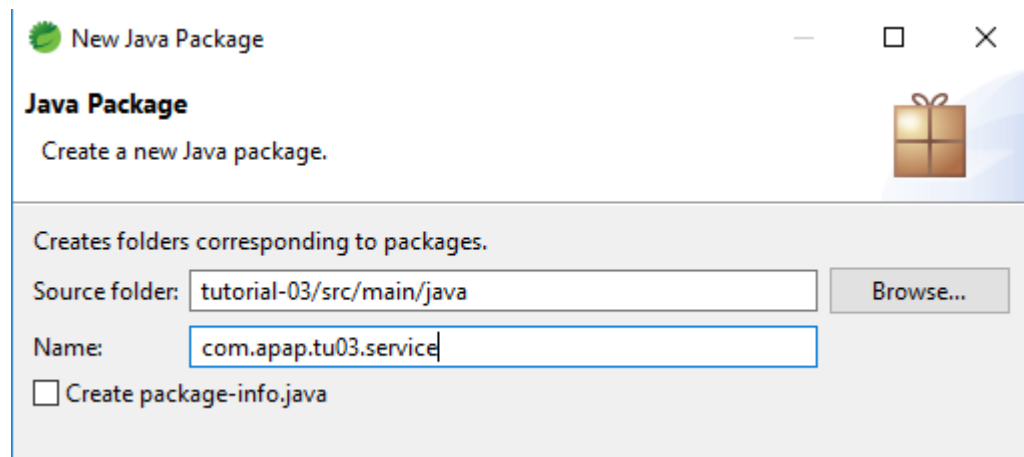


- Buat class movie

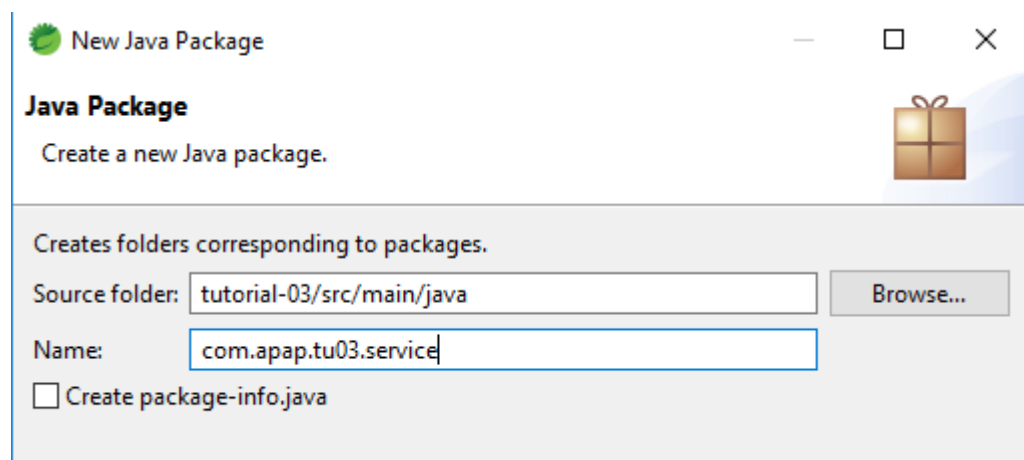


III. Membuat Service

- Buat package baru dengan nama com.apap.tu3.service



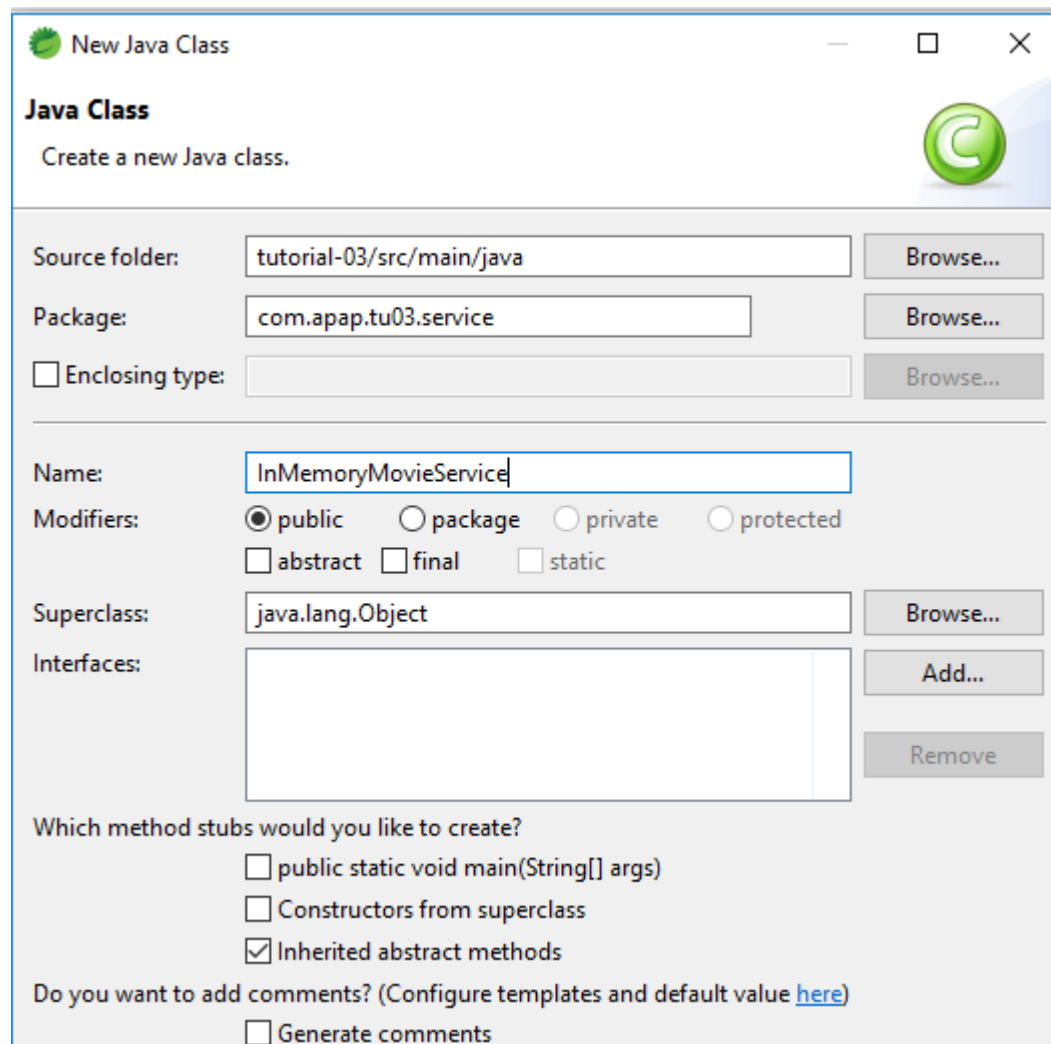
- Buat interface dengan nama com.apap.tu3.service



Dengan spesifikasi sbb:

```
1 package com.apap.tu03.service;
2
3 import java.util.List;
4 import com.apap.tu03.model.MovieModel;
5
6 public interface MovieService {
7     void addMovie (MovieModel movie);
8     List<MovieModel> getMovieList();
9     MovieModel getMovieDetail(String id);
10 }
```

- Pada package yang sama buat class dengan nama InMemoryMovieService



New Java Class

Java Class
Create a new Java class.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

- Dengan Spesifikasi sbb:

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

import java.util.*;
import org.springframework.stereotype.Service;
import com.apap.tu03.model.*;

@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;
    }
}
```

- Tambahkan method `getMovieDetail`. Method ini menerima id movie dan mengembalikan object movie dengan id tersebut. Return null jika tidak ditemukan.

```
@Override
public MovieModel getMovieDetail(String id) {
    for(MovieModel movie : archiveMovie) {
        if(movie.getId().equalsIgnoreCase(id)) {
            return movie;
        }
    }
    return null;
}
```

IV. Membuat Controller dan Method Add

- Buat package baru dengan nama `com.apap.tu3.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;

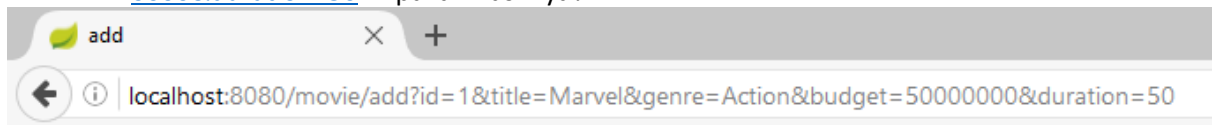
    @RequestMapping("/movie/add")
    public String add(@RequestParam(value = "id", required = true) String id,
        @RequestParam(value = "title", required = true) String title,
        @RequestParam(value = "genre", required = true) String genre,
        @RequestParam(value = "budget", required = true)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

- Buat view `add.html` dengan isi sbb:

```
<!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> . Apakah hasilnya?



Data Berhasil Ditambahkan!

Data berhasil ditambahkan karena untuk menambah movie akan memanggil path/movie/add , lalu akan otomatis menjalankan method Add() dengan memberikan syarat parameter yang harus ada yaitu id, title, genre, budget, dan duration. Jika operasi berhasil maka akan mereturn view add.html yang akan menampilkan “Data Berhasil Ditambahkan!”.

- <http://localhost:8080/movie/add?id=1&title=Marvel&genre=Action&budget=50000000> . Apakah hasilnya?



Whitelabel Error Page

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

Sun Mar 17 15:27:33 ICT 2019

There was an unexpected error (type=Bad Request, status=400).

Required Integer parameter 'duration' is not present

org.springframework.web.bind.MissingServletRequestParameterException: Required Integer parameter 'duration' is not present

```
at org.springframework.web.method.annotation.RequestParamMethodArgumentResolver.handleMissingValue(RequestParamMethodArgumentResolver.java:202)
at org.springframework.web.method.annotation.AbstractNamedValueMethodArgumentResolver.resolveArgument(AbstractNamedValueMethodArgumentResolver.java:113)
at org.springframework.web.method.support.HandlerMethodArgumentResolverComposite.resolveArgument(HandlerMethodArgumentResolverComposite.java:126)
at org.springframework.web.method.support.InvocableHandlerMethod.getMethodArgumentValues(InvocableHandlerMethod.java:166)
at org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest(InvocableHandlerMethod.java:134)
at org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocableHandlerMethod.java:102)
at org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invokeHandlerMethod(RequestMappingHandlerAdapter.java:895)
at org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.handleInternal(RequestMappingHandlerAdapter.java:800)
```

Hasilnya error karena parameter duration tidak ada.

V. Membuat Method view by ID

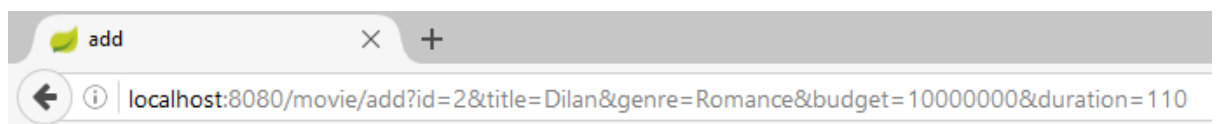
- Pada class MovieController tambahkan method berikut:

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

- Buat view view-movie.html dengan isi sbb:

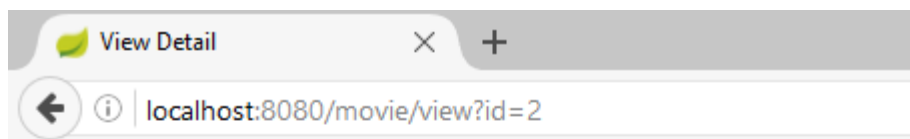
```
<!DOCTYPE html>
<html xmlns="http://www.thymeleaf.org">
  <html>
    <head>
      <title>View Detail</title>
    </head>
    <body>
      <h2>Data</h2>
      <h3 th:text="${movie.id}"></h3>
      <h3 th:text="${movie.title}"></h3>
      <h3 th:text="${movie.genre}"></h3>
      <h3 th:text="${movie.budget}">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

- Run project, lalu akses:
 - <http://localhost:8080/movie/add?id=2&title=Dilan&genre=Romance&budget=10000000&duration=110> . Apa yang terjadi?



Data Berhasil Ditambahkan!

- <http://localhost:8080/movie/view?id=2> . Apa yang terjadi? Jelaskan.



Data

2

Dilan

Romance

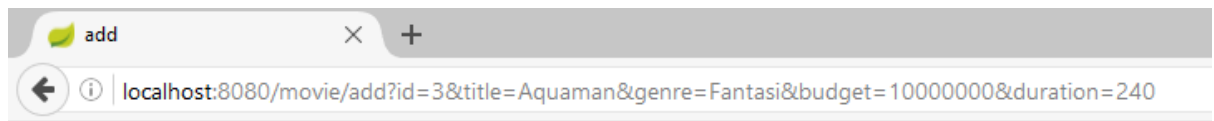
10000000

110

Data movie muncul karena inputan yang berupa id = 2 memang tersimpan dan dapat diakses, jika id movie memang ada dan ditemukan maka akan mereturn view-movie.html, nampun a[abila id tidak ada atau tidak ditemukan akan keluar error pada browser.

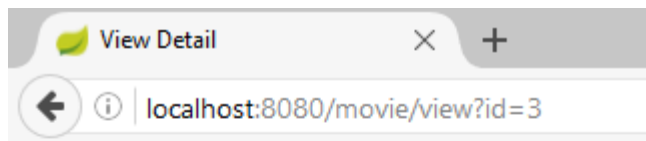
- Tambahkan data movie lainnya dengan id berbeda.

<http://localhost:8080/movie/add?id=3&title=Aquaman&genre=Fantasi&budget=10000000&duration=240>



Data Berhasil Ditambahkan!

Jika view:



Data

3

Aquaman

Fantasi

10000000

240

VI. Membuat Method View All

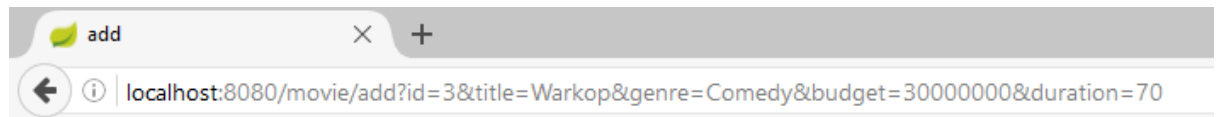
- Pada Class MovieController tambahkan method :

```
@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 sbb:


```
<!DOCTYPE html>
<html xmlns:th="http://www.thymleaf.org">
  <head>
    <title>View All</title>
  </head>
  <body>
    <h1>Data</h1>
    <div th:each="movie : ${movies}">
      <h2 th:text="${movie.id}"></h3>
      <h2 th:text="${movie.title}"></h3>
      <h2 th:text="${movie.genre}"></h3>
      <h2 th:text="${movie.budget}"></h3>
      <h2 th:text="${movie.duration}"></h3>
    </div>
  </body>
</html>
```

- Run project, lalu akses:
 - <http://localhost:8080/movie/add?id=4&title=Warkop&genre=Comedy&budget=300000000&duration=70>



Data Berhasil Ditambahkan!

- <http://localhost:8080/movie/viewall> . Apa yang terjadi? Jelaskan?



Data

1

Marvel

Action

50000000

50

2

Dilan

Romance

10000000

110

3

Aquaman

Fantasi

10000000

240

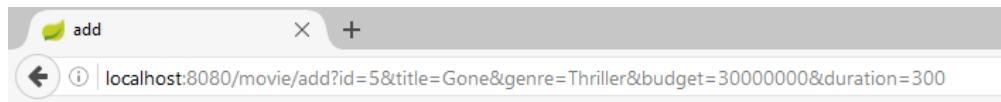
4

Warkop

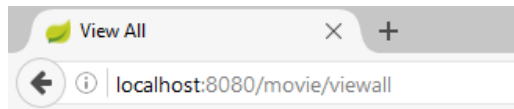
Comedy

Data movie muncul, karena semua parameter yang diperlukan sudah terpenuhi dan data yang tersimpan akan ditampilkan dengan me-return view viewall-movie.html

- Tambahkan data movie dengan id yang berbeda. lalu akses <http://localhost:8080/movie/viewall> . Apakah semua data movie muncul? Jelaskan yang terjadi.



Data Berhasil Ditambahkan!



Data

1
Marvel
Action
50000000
50
2
Dilan
Romance
10000000
110
3
Aquaman
Fantasi
10000000
240
4
Warkop
Comedy
30000000
70
5
Gone
Thriller
30000000
300

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.

Jawaban:

Pada MovieController tambahkan sbb:

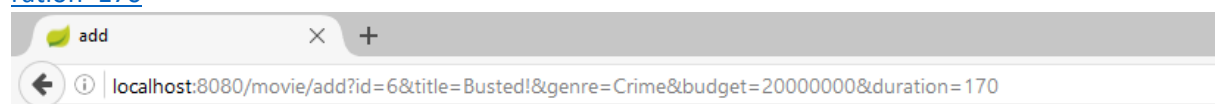
```
@RequestMapping("/movie/view/{id}")
public String viewPath(@PathVariable String id, Model model) {
    MovieModel archive = movieService.getMovieDetail(id);
    if (archive != null) {
        model.addAttribute("movie", archive);
        return "view-movie";
    } else {
        return "error";
    }
}
```

Buat file error.html dengan spesifikasi sbb:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>ERROR!</title>
</head>
<body>
    <h4>Nomor ID kosong atau tidak ditemukan!</h4>
</body>
</html>
```

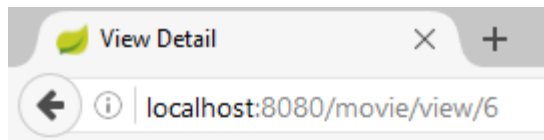
Menambahkan movie :

<http://localhost:8080/movie/add?id=6&title=Busted!&genre=Crime&budget=20000000&duration=170>



Data Berhasil Ditambahkan!

Lihat Movie, jika tersedia maka data movie akan muncul.



Data

6

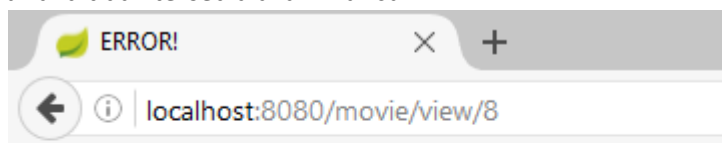
Busted!

Crime

20000000

170

Jika id tidak tersedia akan muncul :



Nomor ID kosong atau tidak ditemukan!

2. 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.

Jawaban:

Tambahkan code berikut pada MovieService :

```
void updateMovie(String id, Integer newDuration);
package com.apap.tu03.service;

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

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

Tambahkan methode MovieController dengan code sbb:

```

    @RequestMapping("/movie/update/{id}/duration/{newDuration}")
    public String update(@PathVariable String id, @PathVariable
Integer newDuration, Model model) {
        MovieModel archive = movieService.getMovieDetail(id);
        if(archive != null) {
            movieService.updateMovie(id, newDuration);
            return "update";
        }else {
            return "error2";
        }
    }
}

```

Buat file update.html dengan kode sbb:

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>Update Duration</title>
</head>
<body>
    <h3>Duration Successfully Changed!</h3>
</body>
</html>

```

Buat file error2.html dengan kode sbb:

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>ERROR!</title>
</head>
<body>
    <h4>Nomor ID kosong atau tidak ditemukan dan proses update/delete dibatalkan</h4>
</body>
</html>

```

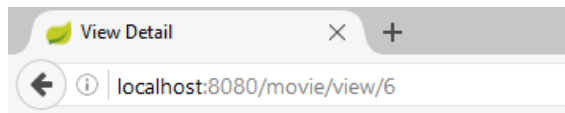
Pada InMemoryMovieService tambahkan kode sbb:

```

@Override
public void updateMovie(String id, Integer newDuration) {
    for(int i = 0; i < archiveMovie.size(); i++) {
        if(archiveMovie.get(i).getId().equals(id)) {
            archiveMovie.get(i).setDuration(newDuration);
            archiveMovie.set(i, archiveMovie.get(i));
        }
    }
}

```

Run dan lalu akses <http://localhost:8080/movie/view/6> untuk melihat data sebelum di update.



Data

6

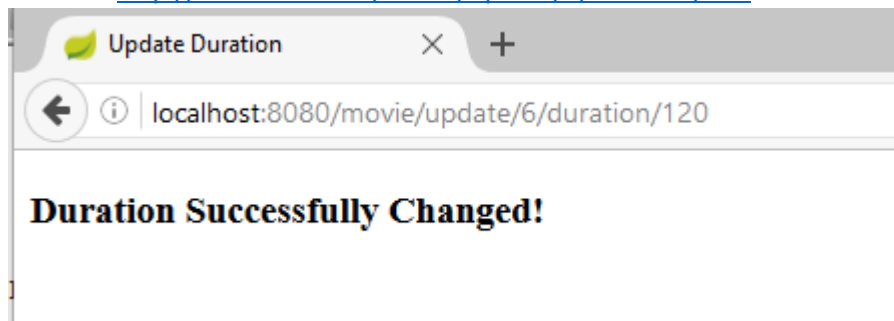
Busted!

Crime

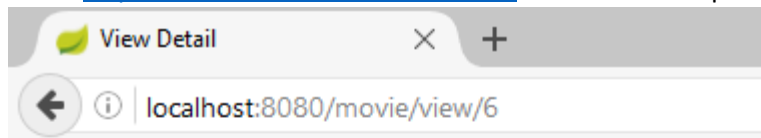
20000000

170

Jalankan <http://localhost:8080/movie/update/6/duration/120>



Akses <http://localhost:8080/movie/view/6> untuk melihat apakah duration sudah terupdate:



Data

6

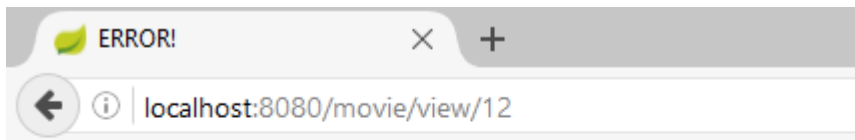
Busted!

Crime

20000000

120

Jika mengupdate dengan id yang tidak ada akan muncul sbb:



Nomor ID kosong atau tidak ditemukan!

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.

Jawaban :

Pada MovieService tambahkan kode `void deleteMovie(MovieModel movie);`
`package com.apap.tu03.service;`

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

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

Pada InMemoryMovieService tambahkan kode:

```
@Override
public void deleteMovie(MovieModel movie) {
    archiveMovie.remove(movie);
}
```

Pada MovieController tambahkan kode:

```
@RequestMapping("/movie/delete/{id}")
public String delete(@PathVariable String id, Model model) {
    MovieModel archive = movieService.getMovieDetail(id);
    if(archive != null) {
        movieService.deleteMovie(archive);
        return "delete";
    }else {
        return "error2";
    }
}
```

Buat file delete.html dengan kode sbb:

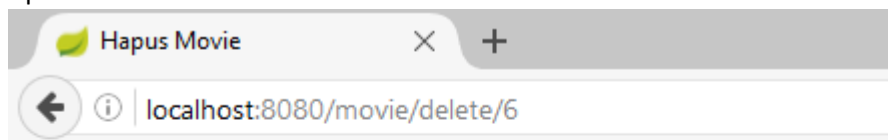

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <title>Hapus Movie</title>
</head>
<body>
  <h3>Data Movie berhasil dihapus!</h3>
</body>
</html>
```

Apabila id movie tidak ada:



Nomor ID kosong atau tidak ditemukan dan proses update/delete dibatalkan

Apabila id movie ada



Data Movie berhasil dihapus!