Praktikum 1

Aplikasi Web Spring MVC

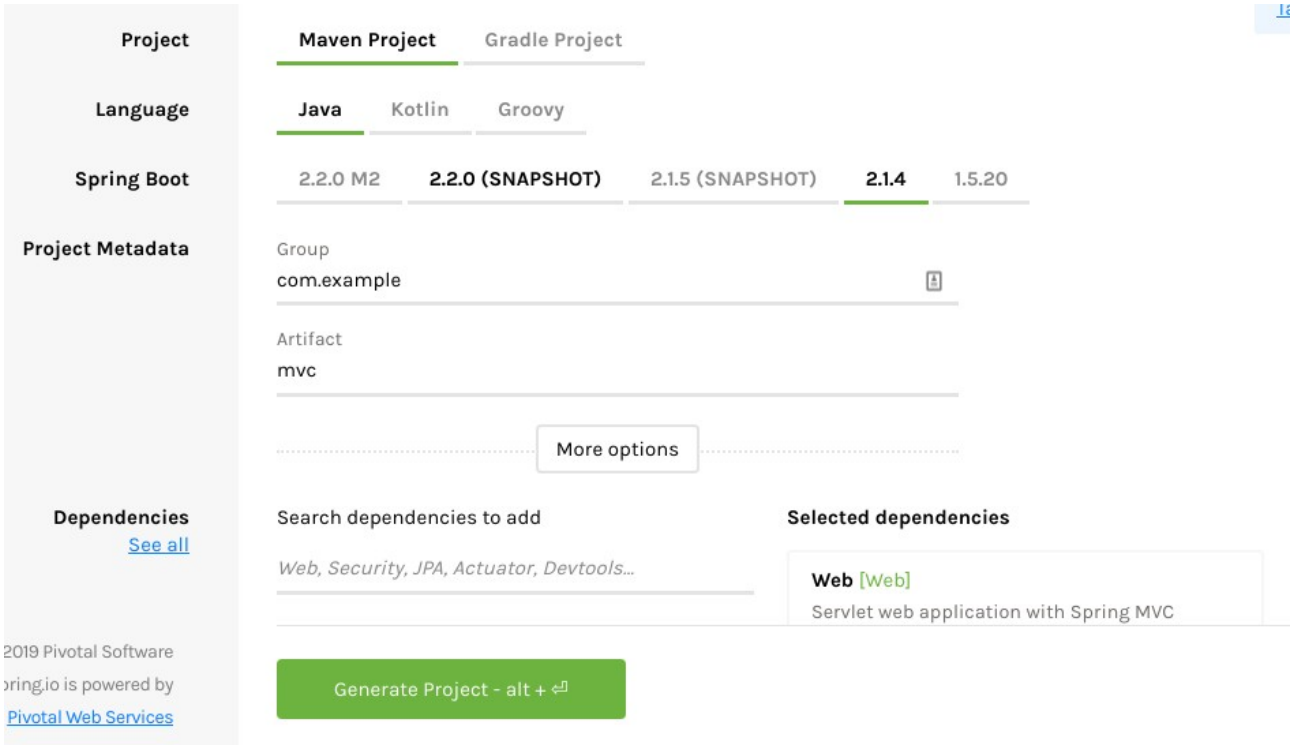
1. Pengantar

- Spring Framework adalah framework yang digunakan untuk membangun aplikasi berbasis web dan Java. Dengan menggunakan Spring Framework, mahasiswa dapat membangun aplikasi dengan mudah dan cepat
- Dalam proses pembuatan projek, semua dilakukan menggunakan Netbeans IDE yang terkenal akan keramahan tampilan dan penggunaannya. Nantinya mahasiswa dapat memulai mengerjakan kode-kode aplikasi tersebut
- Praktikum ini akan membantu mahasiswa untuk mempelajari dan menggunakan Spring MVC menggunakan Spring initializer untuk mempermudah pembuatan projek

2. Pembahasan

Praktikum ini dimulai dengan pembuatan projek pertama dengan menggunakan Spring Init dan kemudian file awal ini akan dimasukkan ke dalam NetBeans. Lalu mahasiswa dapat memulai praktikumnya.

1. Bukalah web https://start.spring.io untuk memulai projek yang diinginkan
2. Masukkan nama projek serta bahasa dan versi Spring yang akan digunakan



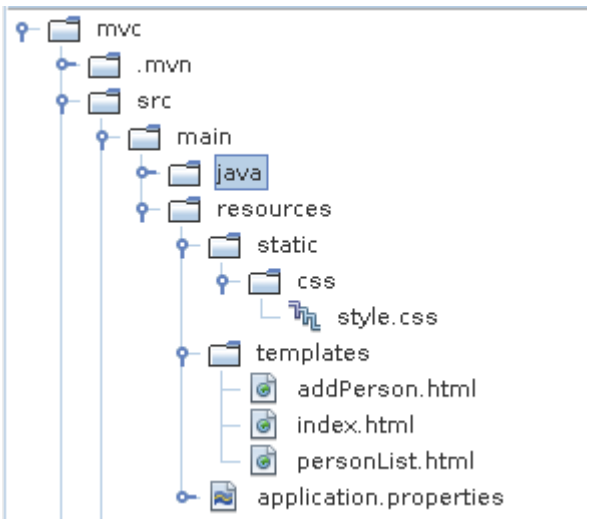
3. Di bagian dependency masukkan dependency Web dan Thymeleaf

4. Lalu kemudian klik Generate Project, yang kemudian akan mendapatkan sebuah file zip. Setelah itu buka NetBeans

5. Dengan NetBeans klik File > Import Project > From ZIP. Lalu pilih file tadi lalu import ke dalam NetBeans.

6. File pom.xml tidak perlu dimodifikasi karena sudah diadaptasi oleh website tersebut, yang kita cukup lakukan adalah mengatur Thymeleaf dan membuat Model serta Controller.

7. Ubah ke tab File lalu buatlah struktur file sesuai gambar berikut ini (folder static, templates di dalam resources):



8. Buka file index.html lalu ketikkan kode berikut ini:

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <meta charset="UTF-8" />
        <title>Welcome</title>
        <link rel="stylesheet" type="text/css" h:href="@{/css/style.css}"/>
    </head>
    <body>
        <h1>Welcome</h1>
        <h2 th:utext="${message}">..!..</h2>
        <a th:href="@{/personList}">Person List</a>
    </body>
</html>
```

9. Buka file personList.html lalu masukkan kode berikut ini:

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <meta charset="UTF-8" />
        <title>Person List</title>
        <link rel="stylesheet" type="text/css" th:href="@{/css/style.css}"/>
    </head>
    <body>
        <h1>Person List</h1>
        <a href="addPerson">Add Person</a>
        <br/><br/>
        <div>
            <table border="1">
                <tr>
                    <th>First Name</th>
```

```
                <th>Last Name</th>
            </tr>
            <tr th:each ="person : ${persons}">
                <td th:utext="${person.firstName}">...</td>
                <td th:utext="${person.lastName}">...</td>
            </tr>
        </table>
    </div>
</body>
</html>
```

10. Buka file addPerson.html lalu masukkan kode berikut ini:

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <meta charset="UTF-8" />
        <title>Add Person</title>
        <link rel="stylesheet" type="text/css" th:href="@{/css/style.css}"/>
    </head>
    <body>
        <h1>Create a Person:</h1>
        <form th:action="@{/addPerson}"
            th:object="${personForm}" method="POST">
            First Name:
            <input type="text" th:field="*{firstName}" />
            <br/>
            Last Name:
            <input type="text" th:field="*{lastName}" />
            <br/>
            <input type="submit" value="Create" />
        </form>
        <br/>
        <div th:if="${errorMessage}" th:utext="${errorMessage}"
            style="color:red;font-style:italic;">
        </div>
    </body>
</html>
```

11. Buka file style.css yang telah dibuat, lalu masukkan kode berikut:

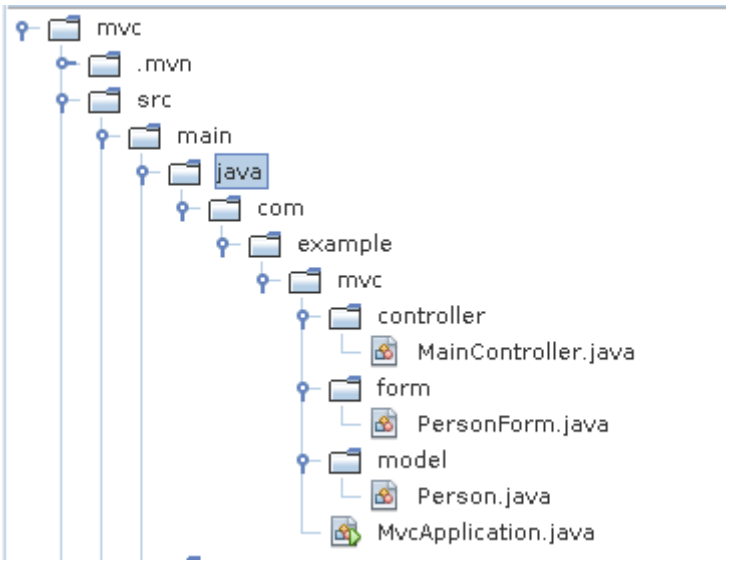
```
h1 {
    color:#0000FF;
}

h2 {
    color:#FF0000;
```

```
}

table {
	border-collapse: collapse;
}

table th, table td {
	padding: 5px;
}
```

12. Buka file application.properties lalu masukkan kode berikut ini:

```
spring.thymeleaf.cache=false

welcome.message=Hello Thymeleaf
error.message=First Name & Last Name is required!
```

13. View dengan Thymeleaf sudah selesai, berikutnya membuat model dan controller. Buatlah struktur file sebagai berikut:



14. Buka Person.java lalu masukkan kode berikut ini:

```
public class Person {

	private String firstName;
	private String lastName;

	public Person() {

	}
```

```
	public Person(String firstName, String lastName) {
		this.firstName = firstName;
		this.lastName = lastName;
	}

	public String getFirstName() {
		return firstName;
	}

	public void setFirstName(String firstName) {
		this.firstName = firstName;
	}

	public String getLastName() {
		return lastName;
	}

	public void setLastName(String lastName) {
		this.lastName = lastName;
	}

}
```

15. Kemudian buka PersonForm lalu masukkan kode berikut ini:

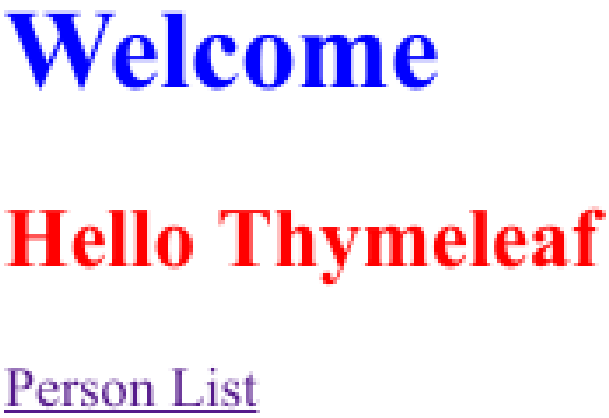
```
public class PersonForm {

	private String firstName;
	private String lastName;

	public String getFirstName() {
		return firstName;
	}

	public void setFirstName(String firstName) {
		this.firstName = firstName;
	}

	public String getLastName() {
		return lastName;
	}

	public void setLastName(String lastName) {
		this.lastName = lastName;
	}
}
```

16. Bukalah file MainController.java dan masukkan kode berikut ini:

```
import java.util.ArrayList;
import java.util.List;

import com.example.mvc.form.PersonForm;
import com.example.mvc.model.Person;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

@Controller
public class MainController {

	private static List<Person> persons = new ArrayList<Person>();

	static {
		persons.add(new Person("Bill", "Gates"));
		persons.add(new Person("Steve", "Jobs"));
	}

	// Inject via application.properties
	@Value("${welcome.message}")
	private String message;

	@Value("${error.message}")
	private String errorMessage;

	@RequestMapping(value = { "/", "/index" }, method =
RequestMethod.GET)
	public String index(Model model) {

		model.addAttribute("message", message);

		return "index";
	}

	@RequestMapping(value = { "/personList" }, method =
RequestMethod.GET)
	public String personList(Model model) {

		model.addAttribute("persons", persons);

		return "personList";
	}

	@RequestMapping(value = { "/addPerson" }, method =
```

```
RequestMethod.GET)
	public String showAddPersonPage(Model model) {

		PersonForm personForm = new PersonForm();
		model.addAttribute("personForm", personForm);

		return "addPerson";
	}

	@RequestMapping(value = { "/addPerson" }, method =
RequestMethod.POST)
	public String savePerson(Model model, //
			@ModelAttribute("personForm") PersonForm personForm) {

		String firstName = personForm.getFirstName();
		String lastName = personForm.getLastName();

		if (firstName != null && firstName.length() > 0 //
				&& lastName != null && lastName.length() > 0) {
			Person newPerson = new Person(firstName, lastName);
			persons.add(newPerson);

			return "redirect:/personList";
		}

		model.addAttribute("errorMessage", errorMessage);
		return "addPerson";
	}
}
```

17. Setelah itu program siap dijalankan layaknya aplikasi web pada biasanya dengan url: localhost:8080



Create a Person:

First Name:
Last Name:
Create

Person List

Add Person

First Name	Last Name
Bill	Gates
Steve	Jobs
NIna	Maria