Thymeleaf

안 희 수

Thymeleaf(타임리프)

- Thymeleaf

- **스프링 부트에서는 JSP를 이용해서 view 페이지를 개발하는 방식에서 벗어나 템플릿 기반의 화면 처리를 지원한다.**

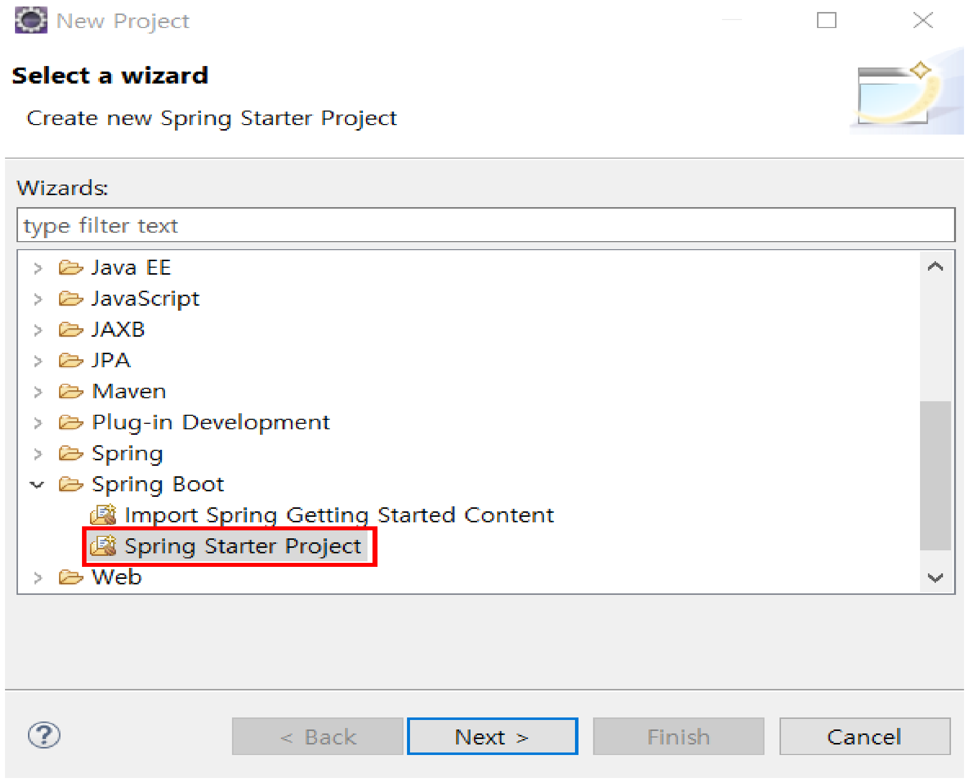
- **FreeMarker, Mustache, Thymeleaf 를 이용해서 확장자가 html 인 페이지를 개발할 수 있다.**

- **Thymeleaf은 기존의 JSP 에서 사용하는 JSTL과 유사한 기능을 제공한다.**

Thymeleaf project 생성

- Thymeleaf **project 생성**

[File] – New - Project



Select a wizard

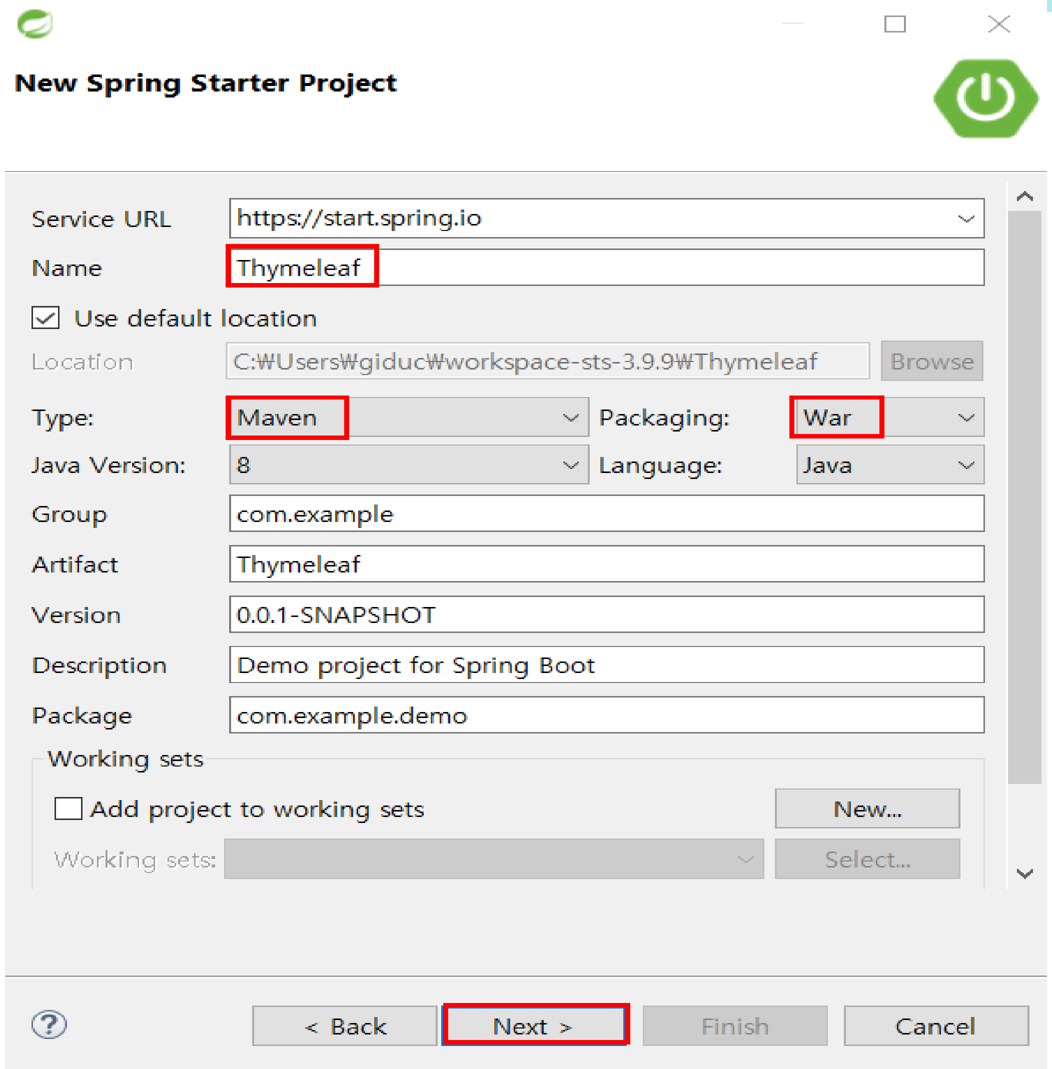
Create new Spring Starter Project

Wizards:

type filter text

Thymeleaf project 생성

- Thymeleaf **project 생성**
  - **Name :** Thymeleaf
  - **Type : Maven, Gradle**
  - **Packaging : War, Jar**



New Spring Starter Project

Service URL: https://start.spring.io

Name: Thymeleaf

☑ Use default location

Location: C:₩Users₩giduc₩workspace-sts-3.9.9₩Thymeleaf

Type: Maven

Packaging: War

Java Version: 8

Language: Java

Group: com.example

Artifact: Thymeleaf

Version: 0.0.1-SNAPSHOT

Description: Demo project for Spring Boot

Package: com.example.demo

Working sets

□ Add project to working sets

Working sets:

Thymeleaf project 생성

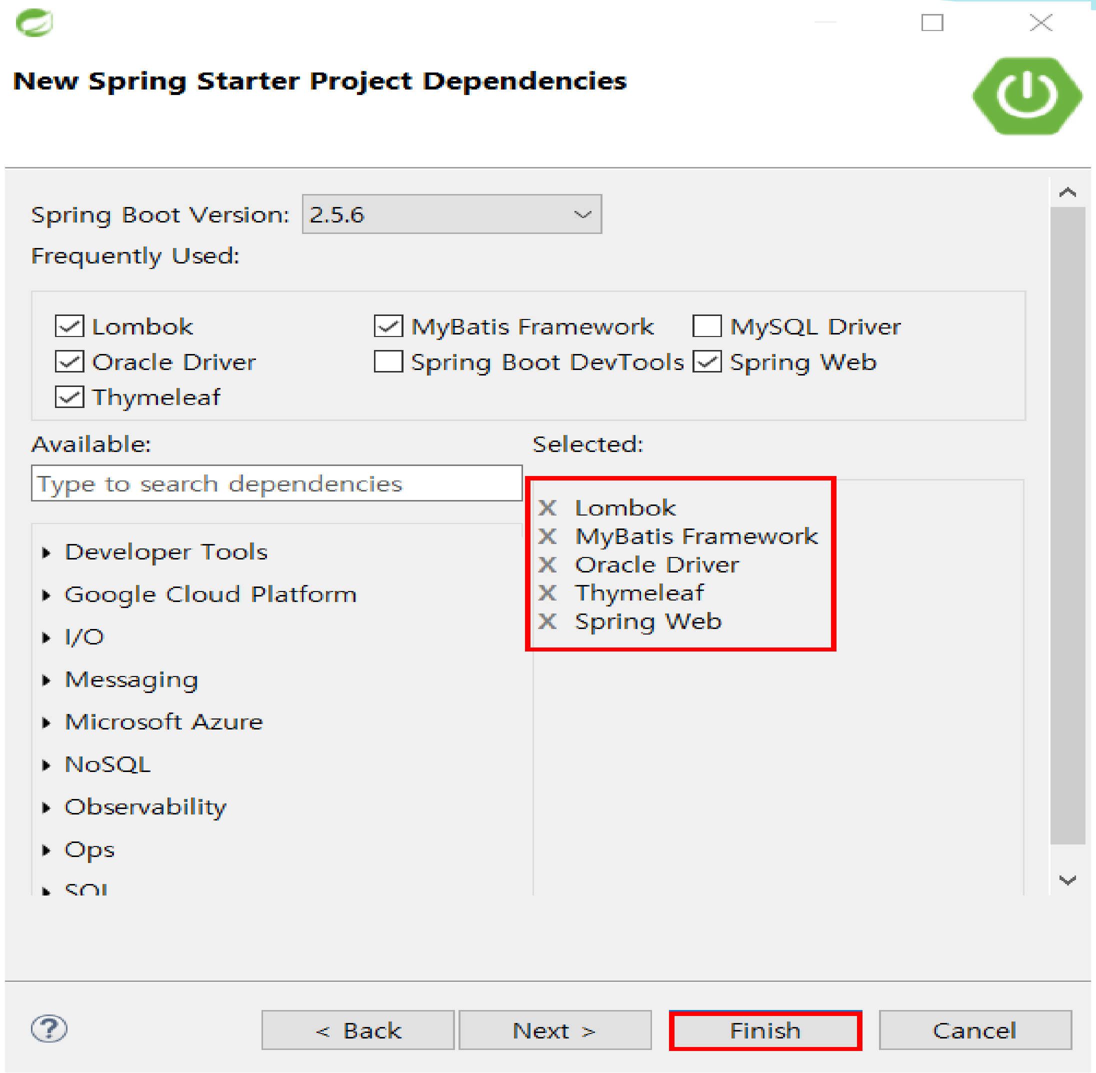
- Thymeleaf **project 생성**

- **Web**
  Spring Web 체크

- **SQL**
  MyBatis Framework 체크
  Oracle Driver 체크

- **Developer Tools**
  Lombok 체크

- **Template Engines**
  Thymeleaf 체크



New Spring Starter Project Dependencies

Spring Boot Version: 2.5.6

Frequently Used:

☑ Lombok ☑ MyBatis Framework □ MySQL Driver
☑ Oracle Driver □ Spring Boot DevTools ☑ Spring Web
☑ Thymeleaf

Available: Type to search dependencies

Selected:
- X Lombok
- X MyBatis Framework
- X Oracle Driver
- X Thymeleaf
- X Spring Web

환경 설정파일 수정

- **환경 설정 파일 수정**

  main / resources / **application.properties**

```
# port
server.port=80
```

DTO 생성(1/2)

- DTO 생성

  main / java / com / example / demo / model – Member.java

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

import java.sql.Timestamp;

public class Member {
        private int no;
        private String id;
        private String pw;
        private String name;
        private Timestamp regdate;

        public Member(int no, String id, String pw, String name, Timestamp regdate) {
                this.no = no;
                this.id = id;
                this.pw = pw;
                this.name = name;
                this.regdate = regdate;
        }
```

DTO 생성(2/2)

- DTO 생성
  main / java / com / example / demo / model – Member.java

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

        public int getNo() {
                return no;
        }
        public void setNo(int no) {
                this.no = no;
        }
        public String getId() {
                return id;
        }
        public void setId(String id) {
                this.id = id;
        }
        public String getPw() {
                return pw;
        }
        public void setPw(String pw) {
                this.pw = pw;
        }
        public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }
        public Timestamp getRegdate() {
                return regdate;
        }
        public void setRegdate(Timestamp regdate) {
                this.regdate = regdate;
        }
}
```

Controller 생성

- Controller 생성

  /main/java/com/example/demo/controller – SampleController.java 생성

```
package com.example.demo.controller;


@Controller
public class SampleController {

        @RequestMapping("sample1")
        public String sample1(Model model) {
//                  model.addAttribute("greeting", "Hello World");
                    model.addAttribute("greeting", "안녕 하세요");
                    return "sample1";
        }
}
```

View – sample1.html

- View 생성

resources / templates – sample1.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

<h1>Thymeleaf Test page</h1>

<!-- th:text : 문자열 출력    -->
<h1 th:text="${greeting}">Thymeleaf Test page</h1>

</body>
</html>
```

Controller 생성

- Controller 생성

/main/java/com/example/demo/controller – SampleController.java 생성

```
package com.example.demo.controller;


@Controller
public class SampleController {

		@RequestMapping("sample2")
		public String sample2(Model model) {
			Member member = new Member(1,"test","1234","홍길동",
									new Timestamp(System.currentTimeMillis()));

			model.addAttribute("member", member);

			return "sample2";
		}}
```

View – sample2.html

- View 생성

resources / templates – sample2.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

<!-- th:text : DTO 객체 출력   -->
<h1 th:text="${member}">Thymeleaf Test page</h1>

<!-- th:text : 문자열(HTML태그) 출력   -->
<div th:text='${"<h3>"+member.no+"</h3>"}'></div>

<!-- th:utext : 데이터 출력   -->
<div th:utext='${"<h3>"+member.no+"</h3>"}'></div>
<div th:utext='${"<h3>"+member.id+"</h3>"}'></div>
<div th:utext='${"<h3>"+member.pw+"</h3>"}'></div>
<div th:utext='${"<h3>"+member.name+"</h3>"}'></div>
<div th:utext='${"<h3>"+member.regdate+"</h3>"}'></div>

</body>
</html>
```

Controller 생성

- Controller 생성

/main/java/com/example/demo/controller – SampleController.java 생성

```
package com.example.demo.controller;


@Controller
public class SampleController {

		@RequestMapping("sample3")
		public String sample3(Model model) {
			List<Member> list = new ArrayList<Member>();

			for(int i=0; i<10; i++) {
				Member member = new Member(1,"test"+i,"1234","홍길동"+i,
											new Timestamp(System.currentTimeMillis()));

				list.add(member);
			}
			model.addAttribute("list", list);

			return "sample3";
		}
}
```

View – sample3.html

- View 생성

resources / templates – sample3.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

<table border="1" align="center" width="300">
        <tr>
                <td>ID</td>
                <td>NAME</td>
                <td>REGDATE</td>
        </tr>
         <!-- th:each="변수 : ${리스트}" -->
        <tr th:each="member : ${list}">
                <td th:text="${member.id}"></td>
                <td th:text="${member.name}"></td>
                <td th:text="${#dates.format(member.regdate,'yyyy-MM-dd HH:mm:ss')}"></td>
        </tr>
</table>
```

Controller 생성

- Controller 생성

/main/java/com/example/demo/controller – SampleController.java 생성

```
package com.example.demo.controller;


@Controller
public class SampleController {

        @RequestMapping("sample4")
        public String sample4(Model model) {
                List<Member> list = new ArrayList<Member>();

                for(int i=0; i<10; i++) {
                        Member member = new Member(i,"u000"+i %3,        // 3으로 나눈 나머지 id
                                                     "p000"+i %3,        // 3으로 나눈 나머지 pw
                                                     "홍길동"+i,
                                                     new Timestamp(System.currentTimeMillis()));

                        list.add(member);
                }
                model.addAttribute("list", list);

                return "sample4";
        }
}
```

View – sample4.html(1/3)

- View 생성

resources / templates – sample4.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

<table border="1" align="center" width="300">
        <tr>
                <td>ID</td>
                <td>NAME</td>
                <td>REGDATE</td>
        </tr>
        <!-- th:each="변수 : ${리스트}" -->
        <tr th:each="member : ${list}">
                <td th:text="${member.id}"></td>
                <td th:text="${member.name}"></td>
                <td th:text="${#dates.format(member.regdate,'yyyy-MM-dd  HH:mm:ss')}"></td>
        </tr>
</table><br>
```

View – sample4.html(2/3)

- View 생성

resources / templates – sample4.html

```
<!-- 변수 정의 -->
<!-- th:with="변수명='값' " -->
<table border="1" align="center" width="300" th:with="target='u0001'">
        <tr>
                <td>ID</td>
                <td>NAME</td>
                <td>REGDATE</td>
        </tr>

        <!-- member.id 값이  target이 회원만 삼항연산자를 이용해서 secret을 출력 -->
        <tr th:each="member : ${list}">
                <td th:text="${member.id == target ? 'secret' : member.id}"></td>
                <td th:text="${member.name}"></td>
                <td th:text="${#dates.format(member.regdate,'yyyy-MM-dd HH:mm:ss')}"></td>
        </tr>
</table><br>
```

View – sample4.html(3/3)

- View 생성

resources / templates – sample4.html

```
<!-- if조건식 -->
<!-- th:if ~ th:unless -->
<table border="1" align="center" width="300" th:with="target='u0001'">
          <tr>
                    <td>ID</td>
                    <td>NAME</td>
                    <td>REGDATE</td>
          </tr>

          <!-- member.id값이 target(u0001)과 같으면 Modify를 출력 하고,
             member.id값이 target(u0001)과 같지 않으면 View를 출력   -->
          <tr th:each="member : ${list}">
                    <td th:if="${member.id}">
                      <a href="/modify" th:if="${member.id == target}">Modify</a>
                      <p th:unless="${member.id == target}">View</p>
                    </td>
                    <td th:text="${member.name}"></td>
                    <td th:text="${#dates.format(member.regdate,'yyyy-MM-dd HH:mm:ss')}"></td>
          </tr>
</table><br>
```

Controller 생성

- Controller 생성

/main/java/com/example/demo/controller – SampleController.java 생성

```
package com.example.demo.controller;


@Controller
public class SampleController {

    @RequestMapping("sample5")
    public String sample5(Model model) {

        String result = "SUCCESS";

        model.addAttribute("result", result);

        return "sample5";
    }
}
```

View – sample5.html

- View 생성

resources / templates – sample5.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

<!-- javascript안에서 Thymeleaf 사용하기 -->
<!-- script태그에  th:inline="javascript"를 추가하면 script안에서 Thymeleaf를 사용할 수 있다. -->
<script th:inline="javascript">
        var result = [[${result}]];         // result = "SUCCESS" 할당됨
        document.write(result);             // SUCCESS  출력
</script>

<script>
        var result = [[${result}]];          // result변수에 값이 할당 안됨
        document.write(result);             // 아무것도 출력 안됨
</script>
</body>
</html>
```

Controller 생성

- Controller 생성

/main/java/com/example/demo/controller – SampleController.java 생성

```
package com.example.demo.controller;

@Controller
public class SampleController {

		@RequestMapping("sample6")
		public String sample6(Model model) {

				model.addAttribute("now", new Date());
				model.addAttribute("price", 123456789);
				model.addAttribute("title", "This is a just sample");
				model.addAttribute("options", Arrays.asList("AAA","BBB","CCC","DDD"));

				return "sample6";
		}
}
```

View – sample6.html(1/2)

- View 생성

resources / templates – sample6.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

<h1 th:text="${now}"></h1>
<h1 th:text="${price}"></h1>
<h1 th:text="${title}"></h1>
<h1 th:text="${options}"></h1>
```

View – sample6.html(2/2)

- View 생성

resources / templates – sample6.html

```
<!--1.날짜 관련 : #dates -->
<h2 th:text="${#dates.format(now, 'yyyy-MM-dd HH:mm:ss')}"></h2>

<!--2.숫자 관련 : #numbers  -->
<h2 th:text="${#numbers.formatInteger(price, 3, 'COMMA')}"></h2>

<!--3.문자 관련 : #strings  -->
<h1 th:text="${title}"></h1>

<!-- replace()함수를 이용해서 s를 진하게 출력 -->
<span th:utext="${#strings.replace(title,'s','<b>s</b>')}"></span>

<!-- listSplit()함수를 이용해서 title변수의 값을 리스트로 처리하고 출력 -->
<ul>
        <li th:each="str:${#strings.listSplit(title,' ')}">[[${str}]]</li>
</ul>
```

Controller 생성

- Controller 생성

/main/java/com/example/demo/controller – SampleController.java 생성

```
package com.example.demo.controller;

@Controller
public class SampleController {

	@RequestMapping("sample7")
	public String sample7(Model model) {

		return "sample7";
	}
}
```

View – sample7.html

- View 생성

resources / templates – sample7.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

<ul>        <!-- 링크 걸기 -->
            <li><a th:href="@{http://localhost:80/sample1}">sample1</a></li>
            <li><a th:href="@{/sample1}">sample1</a></li>
            <li><a th:href="@{~/sample1}">sample1</a></li>

            <!-- 링크를 통해서 값전달 하기 -->
            <!-- http://localhost/Thymeleaf/sample1?p1=aaa&p2=bbb -->
            <li><a th:href="@{/sample1(p1='aaa', p2='bbb')}">sample1</a></li>
</ul>
```