

Spring Boot 시작하기

2023-08-30

이승진

목차

1. hello 프로젝트.....	2
1) 프로젝트 생성.....	2
2) pom.xml 파일 오류.....	5
3) 주의 사항	5
4) application.properties.....	6
5) HomeController.java.....	7
6) index.html	9
7) 실행	10
2. hello 프로젝트 수정.....	11
1) HomeController.java 수정.....	11
2) index.html 수정.....	11
3) 실행	12
3. Spring Web MVC.....	13
1) 컨트롤러 클래스.....	13
2) 뷰(view)	15
3) HomeController.java.....	16
4) index.html	16
5) 서버 종료하기.....	17
4. 연습 문제	18
1) 404 Not Found 에러의 원인은?.....	18
2) model attribute 이름.....	18

1. hello 프로젝트

1) 프로젝트 생성

STS의 메뉴: File - New - Spring Starter Project

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:

Name: hello

Packaging: War

Java Version: 17 (Spring Boot 3은 Java 17 이상을 요구한다)

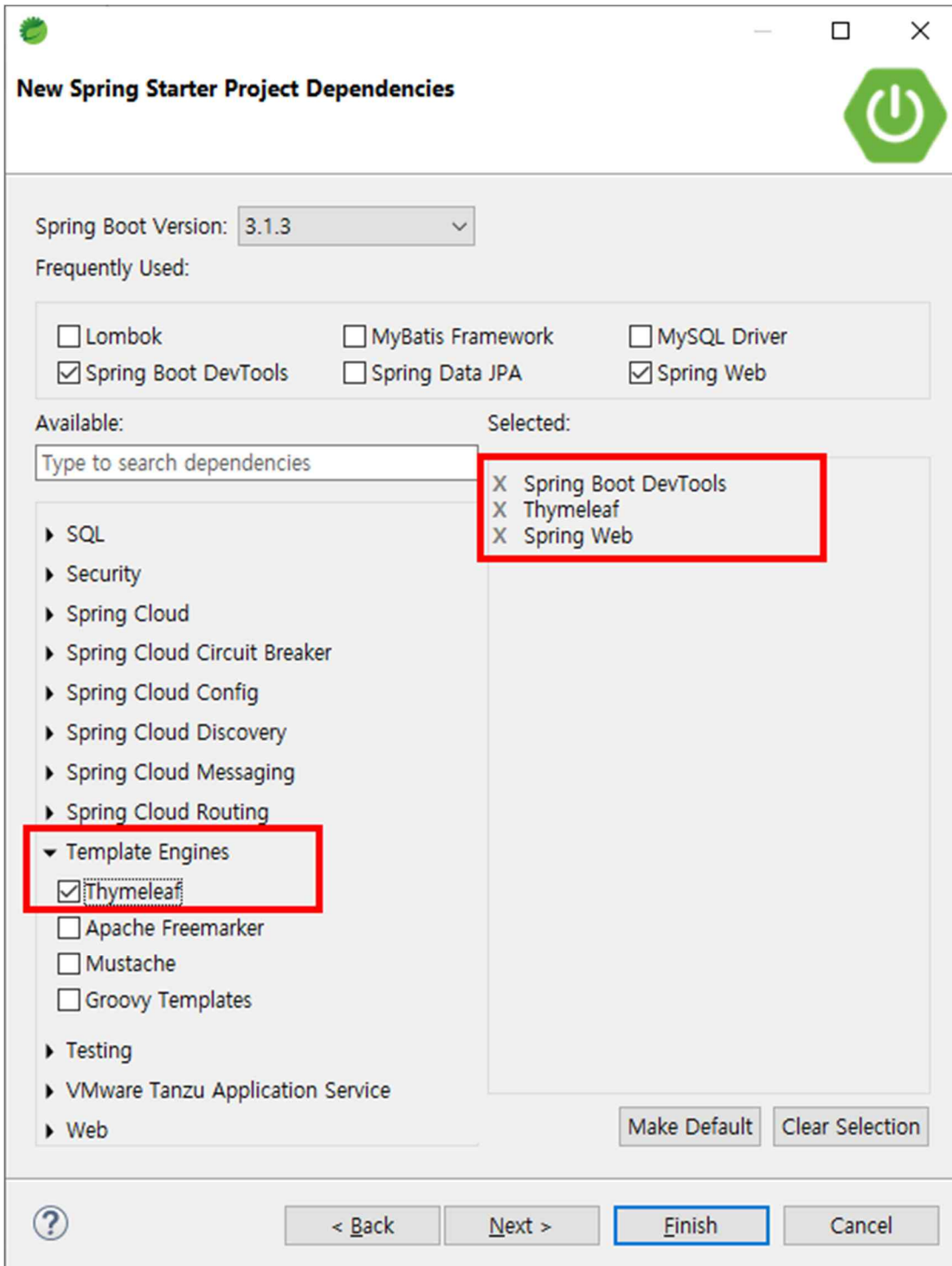
Package: net.skhu

주의

package 항목을 정확하게 입력해야 한다.

이것은 프로젝트에서 사용할 java package 이다.

이 package 아래에 만든 클래스만 spring framework이 인식할 수 있다.



New Spring Starter Project Dependencies

Spring Boot Version: 3.1.3

Frequently Used:

- ☐ Lombok
- ☐ MyBatis Framework
- ☐ MySQL Driver
- ☒ Spring Boot DevTools
- ☐ Spring Data JPA
- ☒ Spring Web

Available:

Type to search dependencies

- SQL
- Security
- Spring Cloud
- Spring Cloud Circuit Breaker
- Spring Cloud Config
- Spring Cloud Discovery
- Spring Cloud Messaging
- Spring Cloud Routing
- Template Engines**
 - ☒ Thymeleaf
 - ☐ Apache Freemarker
 - ☐ Mustache
 - ☐ Groovy Templates
- Testing
- VMware Tanzu Application Service
- Web

Selected:

- X Spring Boot DevTools
- X Thymeleaf
- X Spring Web

Make Default Clear Selection

< Back Next > Finish Cancel

Spring Web

Spring Web MVC 라이브러리

Spring Boot DevTools

프로젝트를 실행할 때, 프로젝트를 재시작하지 않아도 수정된 소스 코드가 실행에 즉시 반영될 수 있도록 해주는 개발 도구

Thymeleaf

뷰(view) 템플릿 엔진

주의

이 예제에서는 데이터베이스 연결을 하지 않을 것이기 때문에,
데이터베이스 항목들을 체크하지 않았다.

데이터베이스 연결을 하지 않으면서 데이터베이스 관련 항목을 체크하면
실행할 때 에러가 발생한다.

Finish 버튼을 클릭하자.

2) pom.xml 파일 오류

프로젝트를 생성했더니 pom.xml 파일 아이콘에 빨간색 에러가 표시될 수도 있다.
만약 빨간색 에러가 뜨지 않았다면, 아래 내용을 무시하자.

해결 방법

pom.xml 파일 아이콘에 빨간색 에러가 표시된다면
Run - Run As - 9 Maven install 메뉴를 실행하자.

그리고 한참을 기다리면 콘솔창에 다음과 같은 성공 메시지가 출력될 것이다.

```
[INFO] Installing C:\PJ\STS\spring\hello\pom.xml to C:\Users\Seungjin\.m2\repository\net\skhu\hello
[INFO] Installing C:\PJ\STS\spring\hello\target\hello-0.0.1-SNAPSHOT.war to C:\Users\Seungjin\.m2\re
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 12.146 s
[INFO] Finished at: 2022-12-25T13:10:45+09:00
[INFO] -----
[WARNING] The requested profile "pom.xml" could not be activated because it does not exist.
```

그 다음

Project - Update Maven Project 메뉴를 실행하자.

3) 주의 사항

강의 노트의 소스코드를 copy & paste 할 때,
소스코드를 웹브라우저 창에서 copy 하면, 공백처럼 보이는 이상한 문자가 포함되어,
실행 오류가 발생할 수도 있다. (언제나 발생하는 것은 아니다)

강의노트를 다운로드하여, MS-Word 앱에서 열어서 copy & past 해야 한다.

이렇게 경고해도 듣지 않고,
이 오류 때문에 구현 시험을 망치는 학생이 매 학기마다 몇 명 있다.
이번 학기의 그 바보는 누구일까?

4) application.properties

src/main/resources/application.properties

```
server.port=8088
```

스프링 부트의 설정 파일

프로젝트에 내장된 톰캣 서버가 사용할 디폴트 포트는 8080인데,
8080 포트를 사용하는 다른 서버와 충돌이 발생할 확률이 높다.
8088 포트로 변경하자.

server.port = 8088 -> 톰캣 서버가 사용할 포트 번호

포트 충돌 에러

포트 번호 변경하지 않고 hello 프로젝트를 실행하는 경우에,
8080 포트를 이미 다른 서버가 사용하고 있다면, 다음과 같은 포트 충돌 에러가 발생한다.



```
<terminated> hello1 - Hello1Application [Spring Boot App] C:\Program Files\Java\jdk1.8.0_181\bin\javaw.exe (2018. 11. 6. 오후 8:01:52)
2018-11-06 20:01:55.260 INFO 14316 --- [ restartedMain] o.apache.catalina.core.StandardService : Stopping service [Tomcat]
2018-11-06 20:01:55.260 INFO 14316 --- [ restartedMain] ConditionEvaluationReportLoggingListener :
Error starting ApplicationContext. To display the conditions report re-run your application with 'debug' enabled.
2018-11-06 20:01:55.260 ERROR 14316 --- [ restartedMain] o.s.b.d.LoggingFailureAnalysisReporter :

*****
APPLICATION FAILED TO START
*****

Description:

The Tomcat connector configured to listen on port 8080 failed to start. The port may already be in use or the connector may be misconfigured.

Action:

Verify the connector's configuration, identify and stop any process that's listening on port 8080, or configure this application to use a different port.

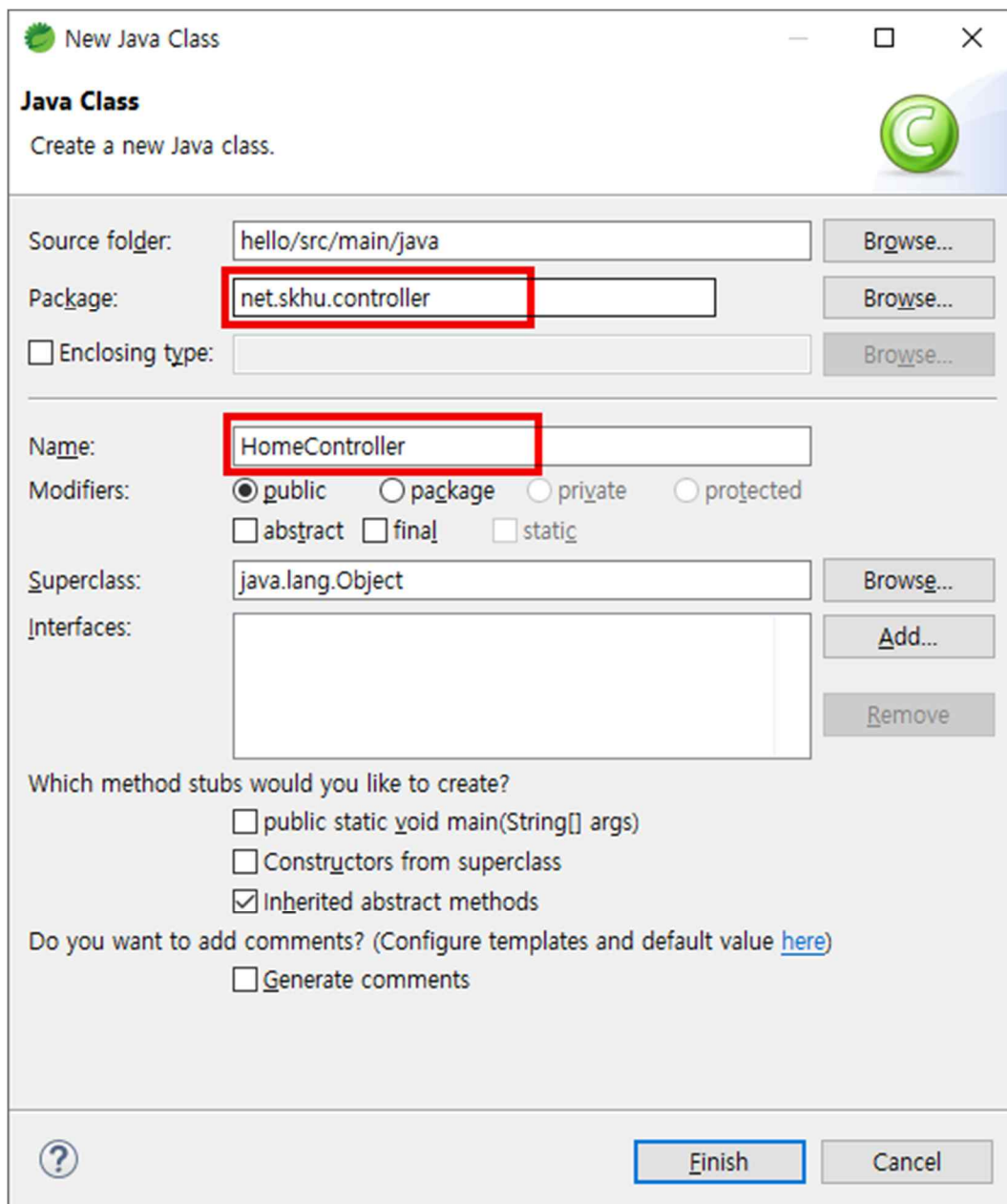
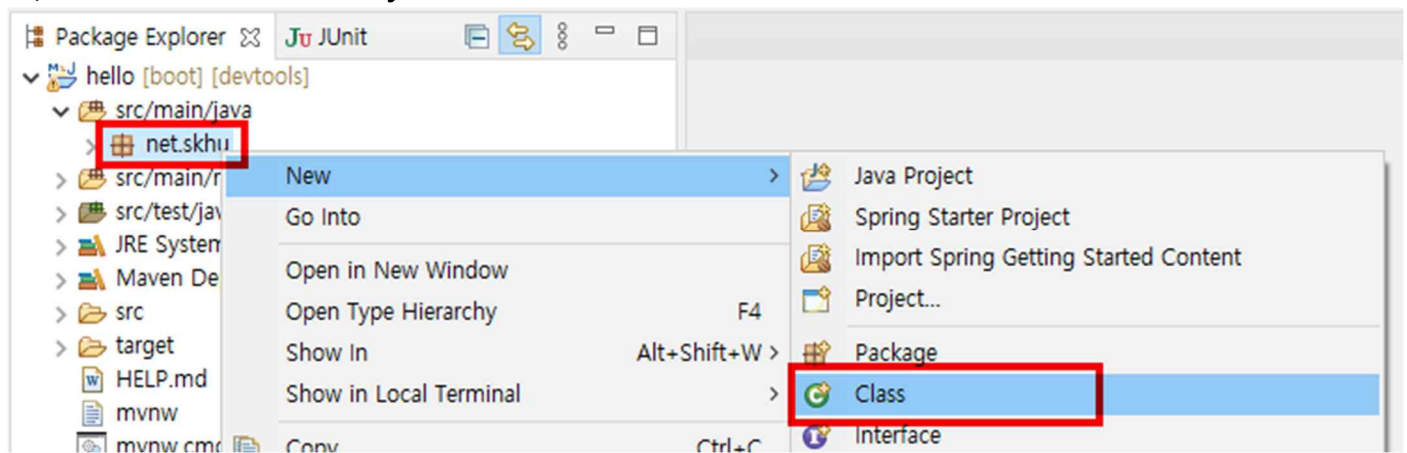
2018-11-06 20:01:55.276 INFO 14316 --- [ restartedMain] ConfigServletWebServerApplicationContext : Closing org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicationContext@1a2b3c4d: started on 2018-11-06 20:01:55.260
2018-11-06 20:01:55.276 INFO 14316 --- [ restartedMain] o.s.j.e.a.AnnotationMBeanExporter : Unregistering JMX-exposed beans on shutdown
```

8080 포트가 이미 사용중이라는 에러 발생.

주의 사항

application.properties 파일의 각 줄 끝에 공백 문자가 있으면 안된다.
실행 오류가 발생한다.

5) HomeController.java



src/main/java/net/skhu/controller/HomeController.java

```
1 package net.skhu.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.Model;
5 import org.springframework.web.bind.annotation.GetMapping;
6
7 @Controller
8 public class HomeController {
9
10     @GetMapping("/")
11     public String index(Model model) {
12         model.addAttribute("message", "좋은 아침");
13         return "index";
14     }
15 }
```

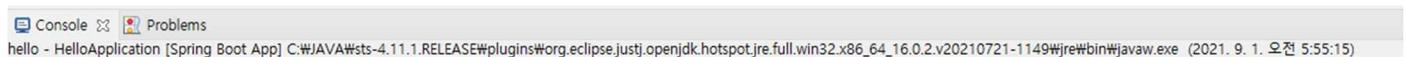
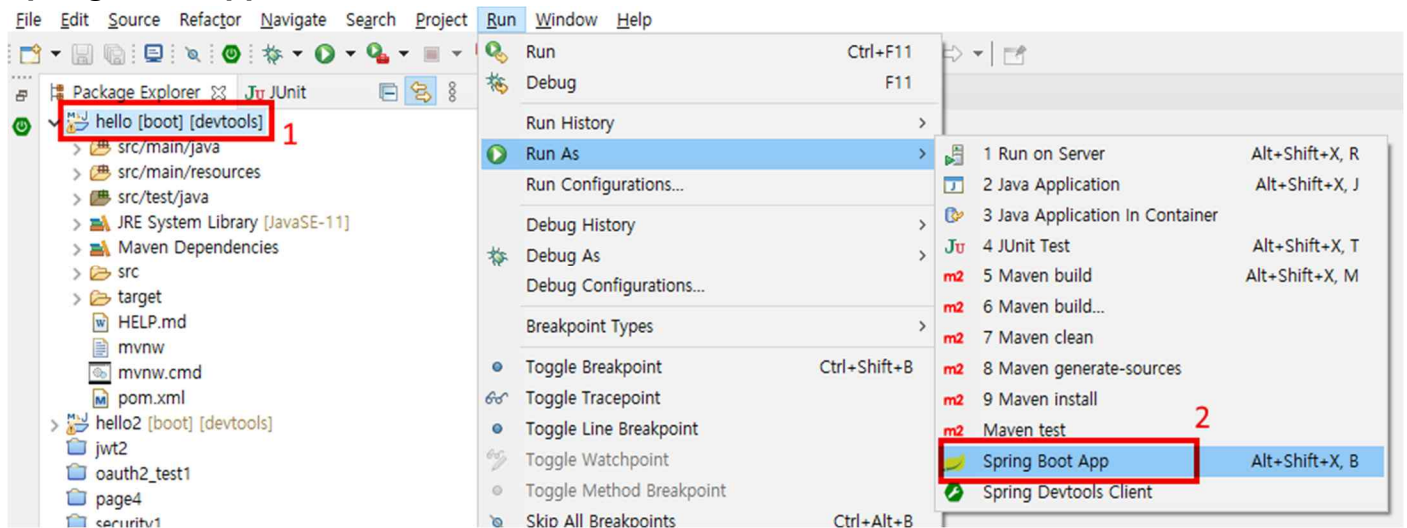

6) index.html

src/main/resources/templates/index.html

```
1 <!DOCTYPE html>
2 <html lang="ko" xmlns:th="http://www.thymeleaf.org">
3 <head>
4   <meta charset="utf-8">
5   <title>hello world</title>
6 </head>
7 <body>
8   <h1>Hello world</h1>
9   <p th:text="${message}">기본 메시지</p>
10 </body>
11 </html>
```

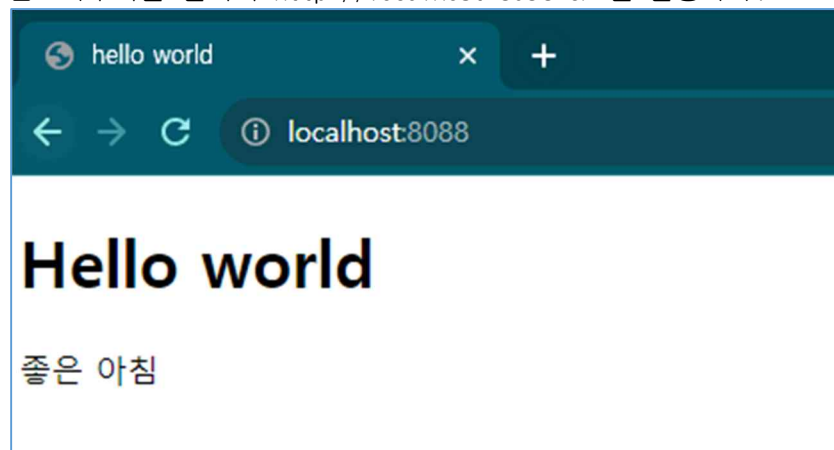
7) 실행

Spring Boot App 실행



콘솔 창의 출력이 위와 같다면, Spring boot app이 잘 실행된 것이다.

웹 브라우저를 열어서 <http://localhost:8088> URL을 실행하자.



hello 프로젝트에는 tomcat 서버가 내장되어 있고,
이 내장된 서버가 실행중이다.

2. hello 프로젝트 수정

수정된 소스코드가 얼마나 빨리 실행되는지 확인해 보자.

1) HelloController.java 수정

src/main/java/net/skhu/controller/HelloController.java

```
1 package net.skhu.controller;
2
3 import java.util.Date;
4
5 import org.springframework.stereotype.Controller;
6 import org.springframework.ui.Model;
7 import org.springframework.web.bind.annotation.GetMapping;
8
9 @Controller
10 public class HomeController {
11
12     @GetMapping("/")
13     public String index(Model model) {
14         model.addAttribute("message", "좋은 아침");
15         model.addAttribute("now", new Date());
16         return "index";
17     }
18 }
```

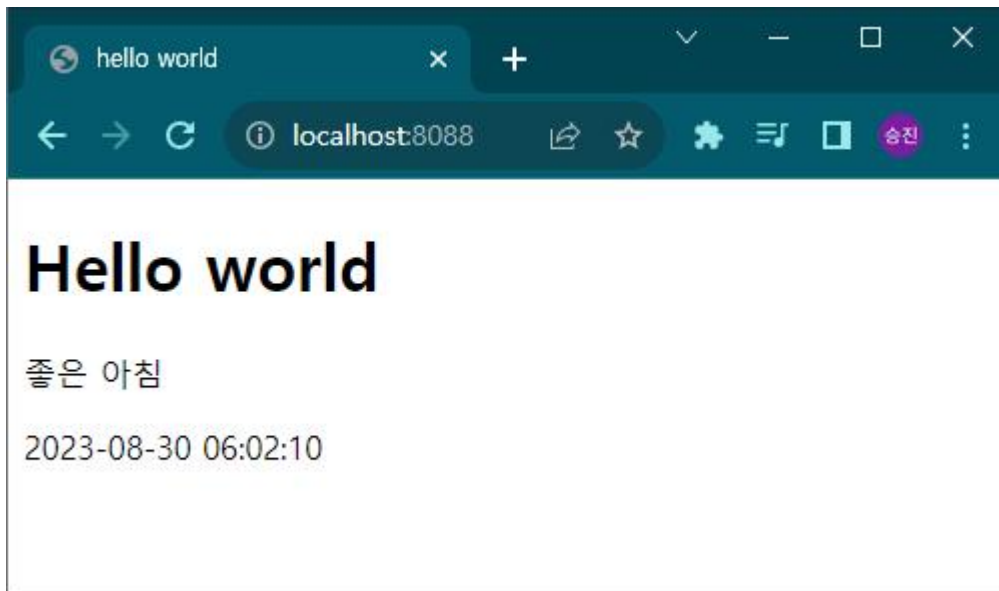
2) index.html 수정

src/main/resources/templates/index.html

```
1 <!DOCTYPE html>
2 <html lang="kr" xmlns:th="http://www.thymeleaf.org">
3 <head>
4     <meta charset="utf-8">
5     <title>hello world</title>
6 </head>
7 <body>
8     <h1>Hello world</h1>
9     <p th:text="${message}">기본 메시지</p>
10    <p th:text="${#dates.format(now, 'yyyy-MM-dd HH:mm:ss')}">기본 날짜</p>
11 </body>
12 </html>
```

3) 실행

소스코드를 수정한 다음에, 웹브라우저 창에서 새로고침하면, 수정된 부분이 반영되어 다시 실행된다.
웹브라우저를 새로고침하자.



수정된 내용이 꽤 빨리 반영된다.

3. Spring Web MVC

1) 컨트롤러 클래스

HomeController.java

```
package net.skhu.controller;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class HomeController {

    @GetMapping("/")
    public String index(Model model) {
        model.addAttribute("message", "좋은 아침");
        return "index";
    }
}
```

컨트롤러 클래스는 웹브라우저의 URL 요청을 받아서, 웹서버에서 실행되는 자바 클래스이다. 웹브라우저가 웹서버에 어떤 URL을 요청을 하면, 그 URL에 해당하는 컨트롤러의 메소드가 자동으로 호출되어 실행된다.

@Controller

컨트롤러 클래스에 이 어노테이션(annotation)을 붙여주어야 한다. 이것을 붙여주지 않으면 컨트롤러 클래스가 실행되지 않는다. 404 Not Found 에러가 발생할 것이다.

액션 메소드

웹브라우저가 어떤 URL을 웹서버에 요청하면, 그 요청된 URL에 해당하는 컨트롤러의 어떤 메소드가 자동으로 호출된다. 이렇게 웹브라우저의 요청에 의해서 자동으로 호출되는 컨트롤러의 메소드를 액션 메소드라고 부른다.

@GetMapping 어노테이션과 액션 메소드 URL

액션 메소드에 붙은 @RequestMapping("/") 어노테이션의 "/" 부분은 그 액션 메소드를 호출하기 위한 URL 이다.

http://localhost:8080/ URL을 웹브라우저가 웹서버에 요청하면, 웹서버에서 HomeController 컨트롤러 클래스의 index 액션 메소드가 자동으로 호출되어 실행된다.

자동으로 호출할 액션 메소드를 찾을 때 컨트롤러 클래스 이름이나 액션 메소드의 이름은 상관 없고, @GetMapping 어노테이션에 등록된 URL만 일치하면 된다. 그리고 HTTP request method는 GET 이어야 한다. 만약 요청된 URL과 일치하는 액션 메소드가 없다면, 404 Not Found 에러가 발생할 것이다.

컨트롤러 액션 메소드의 리턴 값

컨트롤러의 액션 메소드는 문자열을 리턴한다. 컨트롤러의 액션 메소드가 리턴하는 문자열은, 뷰(View) 파일의 이름이다. 컨트롤러의 액션 메소드가 리턴된 후, 뷰 파일이 실행된다. 액션 메소드가 리턴한 이름의 뷰 파일이 실행된다.

Model 객체

```
public String index(Model model) {
```

컨트롤러의 액션 메소드는 Spring Web MVC 엔진에 의해서 호출된다.

대부분의 액션 메소드의 파라미터에 Model model 객체가 포함된다.

컨트롤러의 액션 메소드가 뷰 파일에 전달할 데이터를, Model 객체에 넣어서 전달한다.

즉 Model 객체는, 데이터 전달 상자라고 보면 된다.

model attribute

```
model.addAttribute("message", "좋은 아침");
```

뷰 파일에 전달할 데이터를 Model 객체에 넣는 코드이다.

전달되는 데이터의 이름은 "message" 이고, 값은 "좋은 아침" 문자열이다.

이렇게 모델에 담겨서 뷰에 전달되는 데이터를 model attribute 이라고 한다.

뷰에서 model attribute를 출력하는 코드는 아래와 같다.

```
${ message }
```

model attribute 이름이 정확히 일치해야 한다.

2) 뷰(view)

index.html

```
<!DOCTYPE html>
<html lang="ko" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="utf-8">
  <title>hello world</title>
</head>
<body>
  <h1>Hello world</h1>
  <p th:text="${message}">기본 메시지</p>
</body>
</html>
```

웹브라우저의 요청(http request)에 대한 응답(http response)으로,
웹서버에서 웹브라우저로 전송되는 것은 대부분 html 태그이다.

실행 순서

웹브라우저에서 웹서버에 요청(http request)이 전달되면,
요청된 URL과 일치하는, 컨트롤러의 액션 메소드가 실행된다.

그리고 액션 메소드의 뒤를 이어서 뷰(view) 파일이 실행된다.

뷰 파일의 실행 결과 출력된 html 태그들이 웹브라우저로 전송된다.

```
<p th:text="${message}">기본 메시지</p>
```

뷰에 전달된 model attribute 데이터 중에서, 이름이 "message"인 데이터가 이 태그 사이 텍스트로 출력된다.

3) HelloController.java

src/main/java/net/skhu/HelloController.java

```
package net.skhu.controller;

import java.util.Date;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class HomeController {

    @RequestMapping("/")
    public String index(Model model) {
        model.addAttribute("message", "좋은 아침");
        model.addAttribute("now", new Date());
        return "index";
    }
}
```

```
model.addAttribute("now", new Date());
```

뷰 파일에 전달할 데이터를 Model 객체에 넣는 코드이다.

전달되는 데이터의 이름은 "now" 이고, 값은 현재 시각 Date 객체이다.

4) index.html

```
<!DOCTYPE html>
<html lang="ko" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="utf-8">
  <title>hello world</title>
</head>
<body>
  <h1>Hello world</h1>
  <p th:text="${message}">기본 메시지</p>
  <p th:text="${#dates.format(now, 'yyyy-MM-dd HH:mm:ss')}">기본 날짜</p>
</body>
</html>
```

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

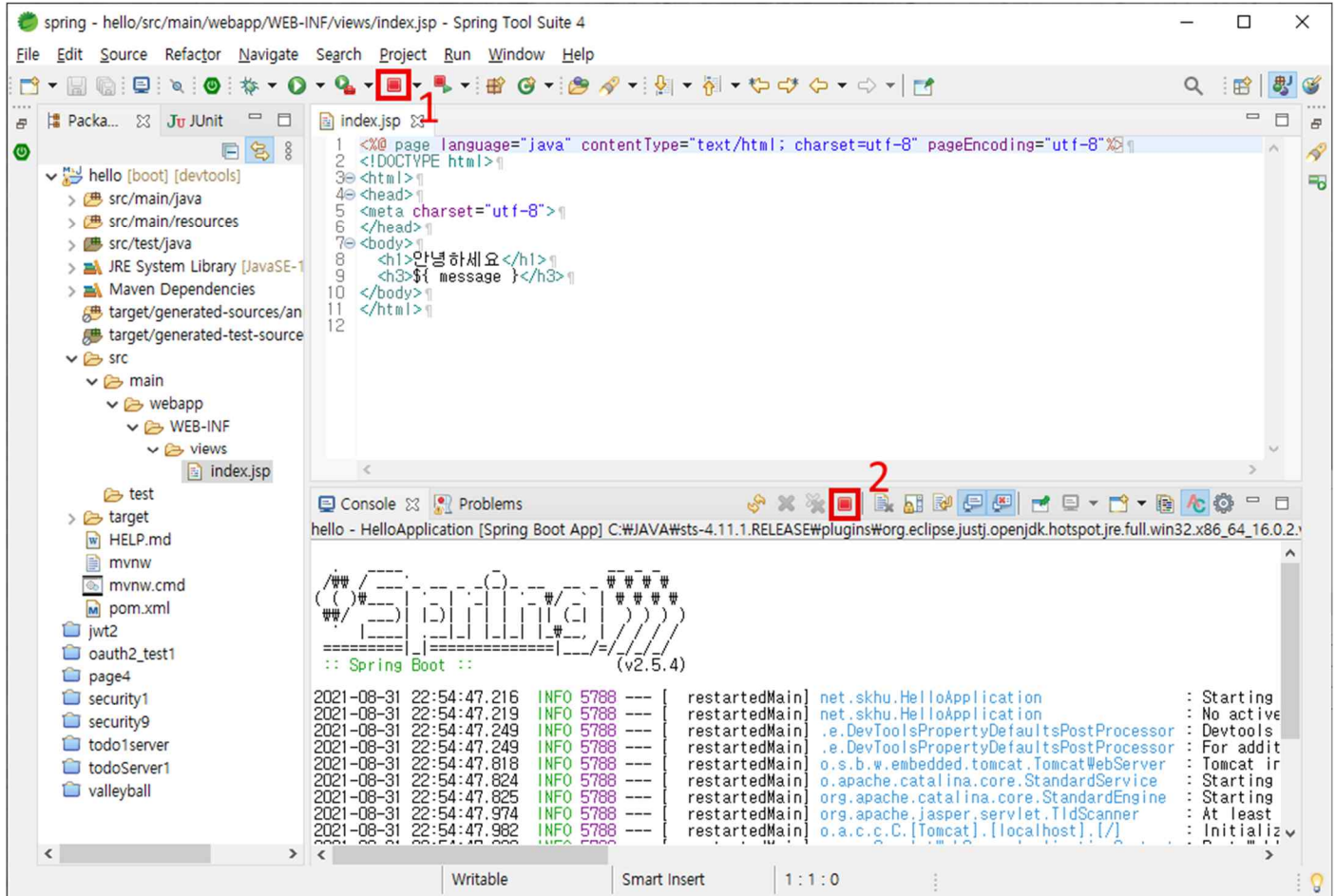
이름이 "now" 인 model attribute는 Date 타입의 객체이다.

이 객체의 값이 'yyyy-MM-dd HH:mm:ss' 포맷의 문자열로 변환되어

<p> 태그의 텍스트로 출력된다.

5) 서버 종료하기

서버를 종료하자.



1번이나 2번 버튼이 활성화되어 있다면, 서버가 실행 상태라는 것을 의미한다.

버튼을 클릭하여 서버를 종료하자.

4. 연습 문제

1) 404 Not Found 에러의 원인은?

프로젝트를 실행하고, 웹브라우저 창에서 URL을 입력했더니, 404 Not Found 에러가 발생했다.
이 에러의 원인을 찾아서 고치기 위해 확인해야 할 부분을 모두 나열하라.

2) model attribute 이름

index.html 에서 model attribute 이름을 일부러 틀리게 수정하고 실행해 보자.

`${ message }`

위 부분을 다음과 같이 수정하자.

`${ msg }`

model attribuet 이름이 틀렸을 때, 어떤 증상이 발생하는가?