0827

오늘 배운거 메뉴얼 꼭 하기

처음엔 그냥따라하고 두번째는 혼자 만들면서 순서 정리하기 + 캡쳐

Spring boot와 Spring

Spring boot setting

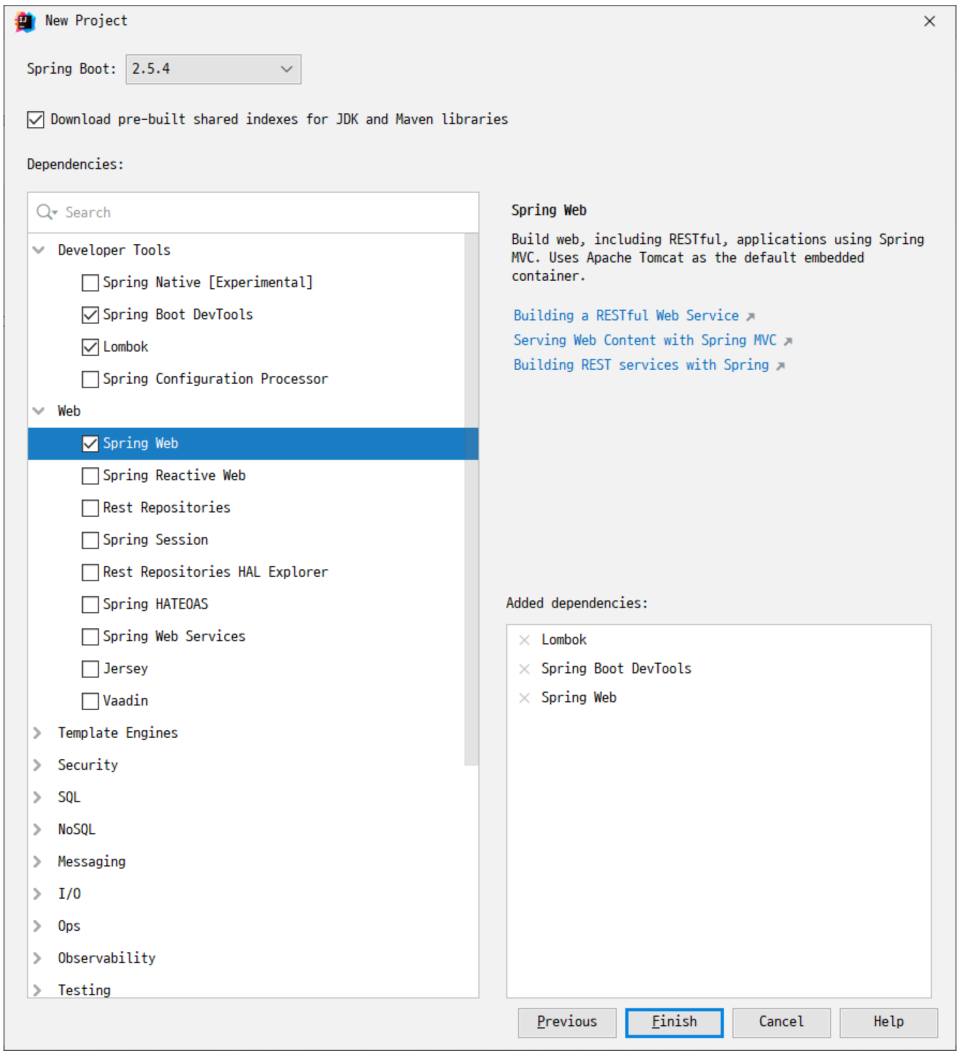
Spring Boot

-Standalone, Spring base app create

-Tomcat 내장

library를 한번에 다운로드해줌

intelliJ에서 boot 사용 시 원하는 library를 선택할 수 있도록 해줌



☑ Download pre-built shared indexes for JDK and Maven libraries

- ☐ Spring Native [Experimental]
- ☑ Spring Boot DevTools
- ☑ Lombok
- ☐ Spring Configuration Processor
- ☑ Spring Web
- ☐ Spring Reactive Web
- ☐ Rest Repositories
- ☐ Spring Session
- ☐ Rest Repositories HAL Explorer
- ☐ Spring HATEOAS
- ☐ Spring Web Services
- ☐ Jersey
- ☐ Vaadin

Spring Web

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Building a RESTful Web Service
Serving Web Content with Spring MVC
Building REST services with Spring

Added dependencies:
Lombok
Spring Boot DevTools
Spring Web

Spring boot 내에는 tomcat이 내장되어있음

파일 실행 시 바로 localhost연결됨

최신버전 자동으로 맞춰줌

Spring Setting 방식 두가지

- xml Setting
- Java Setting

Spring Library Setting

Java EE로 프로젝트 생성

build.gradle setting

Spring Core

```
// https://mvnrepository.com/artifact/org.springframework/spring-core
implementation group: 'org.springframework', name: 'spring-core', version: '5.3.9'
```

Spring Context

```
// https://mvnrepository.com/artifact/org.springframework/spring-context
implementation group: 'org.springframework', name: 'spring-context', version: '5.3.9'
```

web 관련 - Spring Web, Spring Web MVC

Spring Web MVC

Spring Web

```
// https://mvnrepository.com/artifact/org.springframework/spring-web
implementation group: 'org.springframework', name: 'spring-web', version: '5.3.9'
// https://mvnrepository.com/artifact/org.springframework/spring-webmvc
implementation group: 'org.springframework', name: 'spring-webmvc', version: '5.3.9'
```

Test

```
implementation group: 'org.springframework', name: 'spring-test', version: '5.3.9'
```

JSTL » 1.2

```
// https://mvnrepository.com/artifact/javax.servlet.jsp.jstl/jstl
implementation group: 'javax.servlet', name: 'jstl', version: '1.2'
```

XML base Setting

eclipse 에서 만들었던 프로젝트 열어서 기본 세팅파일 등 가져오면 편함

IntelliJ에서 기본으로 만들어주는 Controller 파일과 index.jsp파일 삭제

eclipse의 web.xml 파일 열기

```
     <!-- The definition of the Root Spring Container shared by all Servlets and Filters -->
    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>/WEB-INF/spring/root-context.xml</param-value>
    </context-param>

    <!-- Creates the Spring Container shared by all Servlets and Filters -->
    <listener>
        <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
    </listener>

    <!-- Processes application requests -->
    <servlet>
        <servlet-name>appServlet</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>appServlet</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>
```

Intellij의 xml의 `<web-app>` 내에 붙여넣기

붙여넣기 하면 servlet-context 파일에 에러발생

- WEB-INF밑에 spring 폴더도 붙여넣기
- webapp폴더 밑에 resources폴더 붙여넣기

Lombok setting

build.gradle 에 Project Lombok » 1.18.20 추가

```
 //6. Lombok 추가
// https://mvnrepository.com/artifact/org.projectlombok/lombok
compileOnly group: 'org.projectlombok', name: 'lombok', version: '1.18.20'
testCompileOnly group: 'org.projectlombok', name: 'lombok', version: '1.18.20'
annotationProcessor group: 'org.projectlombok', name: 'lombok', version: '1.18.20'
testAnnotationProcessor group: 'org.projectlombok', name: 'lombok', version: '1.18.20'
```

Log4j2 설정

main 아래의 resources 폴더에 log4j2.xml 파일 삽입

Log4j2.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<configuration status="debug">

    <Appenders>
        <!-- 콘솔 -->
        <Console name="console" target="SYSTEM_OUT">
            <PatternLayout charset="UTF-8" pattern="%d{yyyy-MM-dd hh:mm:ss} %5p [%c] %m%n"/>
        </Console>
    </Appenders>

    <loggers>

        <logger name="jdbc.sqltiming" level="INFO" additivity="false">
            <appender-ref ref="console" />
        </logger>
        <logger name="java.sql.Connection" level="DEBUG" additivity="false">
            <appender-ref ref="console" />
        </logger>
        <logger name="java.sql.Statement" level="DEBUG" additivity="false">
            <appender-ref ref="console" />
        </logger>
        <logger name="java.sql.PreparedStatement" level="DEBUG" additivity="false">
            <appender-ref ref="console" />
        </logger>
        <logger name="java.sql.ResultSet" level="DEBUG" additivity="false">
            <appender-ref ref="console" />
        </logger>
        <logger name="org.zerock.bitboard.dao" level="DEBUG" additivity="false">
            <appender-ref ref="console" />
        </logger>
        <root level="info" additivity="false">
            <AppenderRef ref="console"/>
        </root>

    </loggers>

</configuration>
```

build.gradle 에 log4j2 삽입

```
//7. Log4j2 추가
// https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core
implementation group: 'org.apache.logging.log4j', name: 'log4j-core', version: '2.14.0'
// https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-api
implementation group: 'org.apache.logging.log4j', name: 'log4j-api', version: '2.14.0'
```

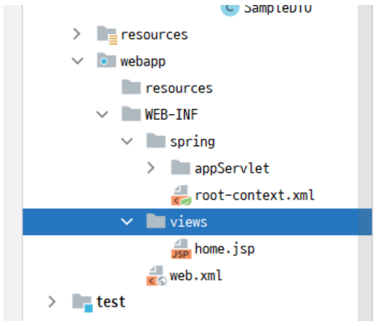
HomeController 붙여넣기

controller 패키지에 eclipse에서 만들었던 controller 파일을 붙여넣음 Logger의 경우에는 여기 없으니 일단 지우고 System.out.println 으로 대체한 후 테스트

```
@Controller
public class HomeController {

    /**
     * Simply selects the home view to render by returning its name.
     */
    @RequestMapping(value = "/", method = RequestMethod.GET)
    public String home(Locale locale, Model model) {
        System.out.println("Welcome home! The client locale is {}." + locale);

        Date date = new Date();
        DateFormat dateFormat = DateFormat.getDateTimeInstance(DateFormat.LONG, DateFormat.LONG, locale);

        String formattedDate = dateFormat.format(date);

        model.addAttribute("serverTime", formattedDate );

        return "home";
    }

}
```

webapp폴더 밑에 views 폴더 붙여넣기

이클립스 폴더 내에 있는 거 그대로 붙여넣기

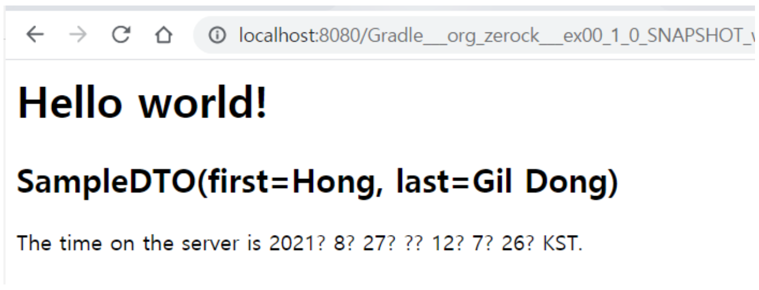


home.jsp 수정

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page session="false" %>
<html>
<head>
    <title>Home</title>
</head>
<body>
<h1>Hello world!</h1>

<h2>${dto}</h2>

<P>  The time on the server is ${serverTime}. </P>
</body>
</html>
```

결과화면

설정 후 아래같이 화면 뜨면 성공한 것



localhost:8080/Gradle__org_zerock__ex00_1_0_SNAPSHOT_

Hello world!

SampleDTO(first=Hong, last=Gil Dong)

The time on the server is 2021? 8? 27? ?? 12? 7? 26? KST.

test폴더에 패키지 추가

com.example.ex00

Spring 연결 test

SampleTests 클래스 생성

```
 //JUnit의 버전이 맞지않아 RunWith 사용 불가 => how?
//@RunWith(SpringJUnitClassRunner.class)
@Log4j2
@ExtendWith(SpringExtension.class)
@ContextConfiguration("file:src/main/webapp/WEB-INF/spring/root-context.xml")
public class SampleTests {

    @Test
    public void test1() {
        log.info("-------------------1");
        log.info("-------------------1");
        log.info("-------------------1");
        log.info("-------------------1");
    }

}
```

@RunWith & @ExtendWith

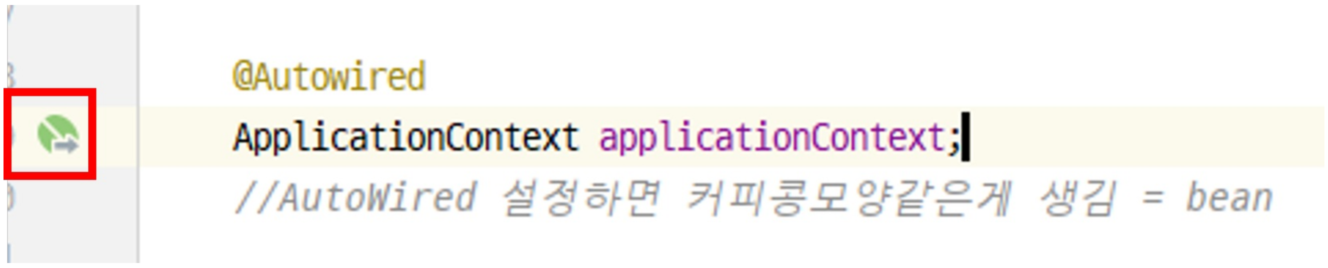
교재에서는 JUnit4버전이어서 RunWith를 사용하지만 내 프로젝트에선 JUnit5버전을 사용하기때문에 대신 다른 것을 사용해줘야함!!

참고 : https://www.baeldung.com/junit-5-runwith

```
 @ExtendWith(SpringExtension.class)
@ContextConfiguration("file:src/main/webapp/WEB-INF/spring/root-context.xml")
```

AppliicationContext 추가

```
@Log4j2
@ExtendWith(SpringExtension.class)
@ContextConfiguration("file:src/main/webapp/WEB-INF/spring/root-context.xml")
public class SampleTests {

    @Autowired
    ApplicationContext applicationContext;
    //AutoWired 설정하면 커피콩모양같은게 생김 = bean

    @Test
    public void test1() {
        log.info("-------------------1");
        log.info("-------------------1");
        log.info(applicationContext);  //null이면 안됨
        //org.springframework.context.support.GenericApplicationContext@1841e6a4, started on Fri Aug 27 12:30:14 KST 2021
        log.info("-------------------1");
        log.info("-------------------1");
        //applicationContext가 null이 아니고 값이 나올 경우 Spring 연결이 잘 되었다는 것!
    }

}
```



```
@Autowired
ApplicationContext applicationContext;
//AutoWired 설정하면 커피콩모양같은게 생김 = bean
```

MySQL에 Spring 용 Schema 생성

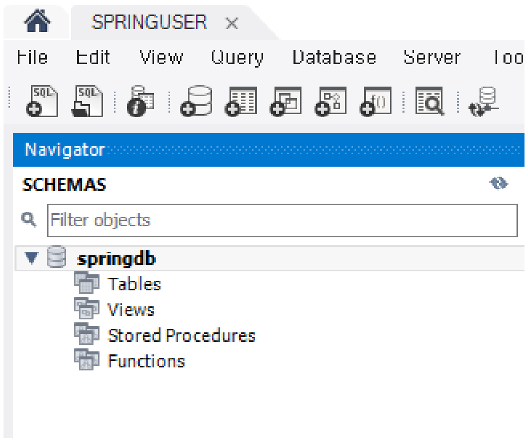
springdb Schema 생성

springuser 계정 생성

연결방식 %, localhost => 각각 springdb 권한 부여

new connection 생성

SPRINGUSER - springuser 연결


SPRINGUSER
File
Edit
View
Query
Database
Server
Too
Navigator
SCHEMAS
Filter objects
springdb
Tables
Views
Stored Procedures
Functions