

초보자를 위한



~



Spring



강태경 지음

머리말

이 책의 특징을 보면...

1

따라 하기 예제를 통해 기본 IT 기술을 따라 해 봅니다.
기술을 코딩과 결과를 이해해 보는 단계입니다.

따라 하기 예제

2

바로 그 뒤에 퀴즈를 풀면서 이해를 좀 더 높입니다.
배운 기술은 이 단계에서 좀 더 확실히 이해할 수 있습니다.

단계적 퀴즈 풀기

3

각 챕터가 끝나면 마무리 퀴즈를 풀어봅니다.
마무리 퀴즈를 통해 응용력을 높일 수 있습니다.

마무리 퀴즈 제공

처음에 세상에 나온 아기는 엄마/아빠의 행동을 보며 그대로 따라 합니다.
어느 정도 익숙해지면 자기만의 주도적인 행동을 시작합니다.

마찬가지입니다.

원가를 배우기 위해서는 좋은 참고 대상이 필요합니다.

무에서 유를 창조하는 것이 아니라,

단지 좋은 예제를 조금씩 변경해 가면서 재창조하는 것입니다.

그런 사례는 현대 실무에서 많이 이루어집니다.

일본 소니 제품의 디자인에 영감을 받아 만든 애플의 iPod 도 마찬가지입니다.

코딩을 익히는 가장 빠른 길은 쉬운 예제를 따라서 코딩하고(클론코딩) 조금씩 변경해 보는 것입니다.

이 책에서는 예제를 따라 코딩하고 조금씩 변경할 수 있도록 연습문제가 있습니다.

“**따라 하기**”로 좋은 코딩 습관을 흡수할 수 있습니다.
또한 “**변경하기**”로 응용력을 증가시킬 수 있습니다.

예제만 간단히 눈으로 보고 넘어갈 수 있습니다.
그 순간에는 이해되더라도 실제로 코딩할 때 1 개도 기억나지 않을 수 있습니다.
변경하기를 통해 자연스럽게 기술을 습득해야 합니다.

다양하게 변경해 가면서 기술과 응용력을 습득합니다.
반드시 퀴즈/마무리 퀴즈를 직접 코딩해보시기 바랍니다.

저자 강태경

초보자를 위한
Spring

저자 | 강태경

본 책 내용은 대한민국 저작권법의 보호를 받습니다.
작성된 모든 내용의 권리는 작성자에게 있으며,
작성자의 동의 없는 사용이 금지됩니다.

목차

<제 목 차 례>

Spring 개발 환경 설정하기	7
Spring 프레임워크	8
Spring 프레임워크가 대체 뭔가요?	8
Spring을 설치하기 위해 무엇을 해야 하나요?	9
자바 & 이클립스 & 톰캣 설치	10
자바 설치	10
이클립스 with 전자정부프레임워크 설치하기	11
톰캣 설치하기	12
이클립스에 톰캣 설정	15
이클립스에 톰캣 설정하기	15
스프링 프로젝트 시작하기	18
스프링 프로젝트 빠르게 만들기	19
전자정부프레임워크 기본 환경에서 빠르게 만들어보기	19
Hello World 화면에 표시하기	27
web.xml 호환성 설정	27
샘플을 이용해서 Hello World를 화면에 표시합니다.	27
컨트롤러 배우기	35

@GetMapping, Model 배우기	36
컨트롤러에 사용하는 기본 어노테이션을 배워봅시다.	36
따라 하기 예제로 알아봅시다.	37
@RequestParam 배우기	59
컨트롤러에 사용하는 어노테이션을 배워봅시다.	59
따라 하기 예제로 알아봅시다.	59
@RequestParam 배우기 2	65
컨트롤러에 사용하는 어노테이션을 배워봅시다.	65
따라 하기 예제로 알아봅시다.	65
@ModelAttribute	90
컨트롤러에 사용하는 어노테이션을 배워봅시다.	90
따라 하기 예제로 알아봅시다.	90
AJAX 사용하기	95
AJAX를 배워봅시다.	95
따라 하기 예제로 알아봅시다.	96
JSP 추가 사용법 알아보기	104
AJAX 사용하기	104
JSP 추가 예제를 배워봅시다.	104
따라 하기 예제로 알아봅시다.	104

Chapter

01

Spring
개발 환경
설정하기

SECTION

01

Spring 프레임워크

01

Spring 프레임워크가 대체 뭔가요?

실무에서 가장 인기있는 자바 개발 환경은 **Spring 프레임워크** 입니다.
또한 실무 웹 개발을 진행할 때 가장 많이 등장하는 기본 형태가 웹 게시판 만들기입니다.
대략 웹 서비스의 70%가 기초 게시판 기술로 구현 가능합니다.
기초 게시판부터 좀 더 어려운 예제로 진행해 보겠습니다.

Spring 프레임워크는 복잡한 환경설정으로 인해 초급자를 좌절에 빠뜨리곤 합니다.
그래서 환경 설정은 세부 설명은 생략한 채 간략하고 빠르게 진행합니다.

핵심기술은 웹 게시판 코딩이며 그 이후에 환경 및 기타 기술에 주목하는 것이 좋습니다.
어떤 기술을 배우고자 할 때는 핵심 사항을 먼저 파악한 후 그 외 기술을 익히는 것이
효율적입니다.

웹 개발자의 기본 기술은 "**다양한 웹 게시판을 만들 수 있는가**" 입니다.
복잡한 기술 설명을 간단, 쉽게 비유해서 설명하고 최소한의 기술을 사용해서 웹 게시판
개발을 해봅니다.
최소한의 기술로도 보편적인 게시판을 만들 수 있다는 놀라움을 선사해 드리겠습니다.

복잡한 환경설정과 JSP , 스프링의 철학에 깊은 관심이 있으신 분들은 부록 편의 자세한
설명을 참고하시기 바랍니다

02

Spring 을 설치하기 위해 무엇을 해야 하나요?

스프링 환경 구축은 자바 및 이클립스(with 전자정부프레임워크), 톰캣(Tomcat) 설치부터 시작됩니다.

이클립스(with 전자정부프레임워크)는 과거에 구축된 jsp/spring 사이트로 인해 아직도 실무에서 많이 사용되며, 전자정부프레임워크의 미리 제작된 기능(페이징, 유효성체크)을 사용하기 위해 활용합니다.

참고로 vue/react/angular.js 연동 springboot 사이트는 주로 인텔리제이를 사용합니다.

이 서적에서는 "jsp/spring 게시판 만들기" 이므로 이클립스(with 전자정부프레임워크)를 사용해서 진행합니다

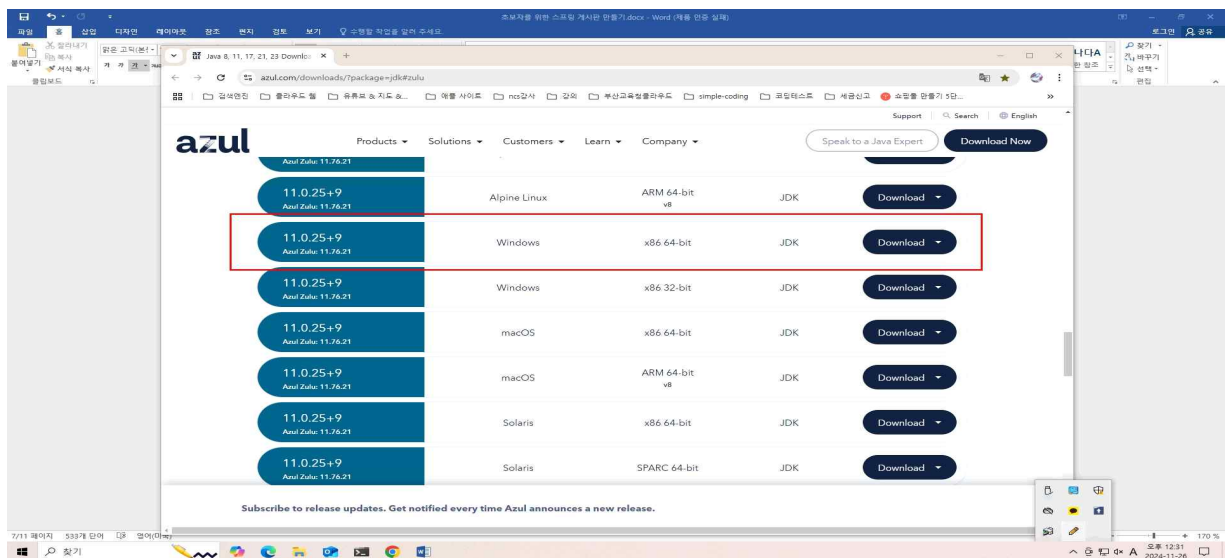
SECTION
02

자바 & 이클립스 & 톰캣 설치

01 자바 설치

- Azul 회사의 제품을 내려받습니다.
단, 설치했으면 넘어갑니다.

① 다운로드 사이트: <https://www.azul.com/downloads/?package=jdk#zulu>



② 종료 시까지 Next 버튼 클릭

02

이클립스 with 전자정부프레임워크 설치하기

- 전자정부프레임워크가 포함된 이클립스를 설치합니다.

단, 설치했으면 넘어갑니다.

- ① 다운로드 사이트:

<https://www.egovframe.go.kr/home/ntt/nttList.do?searchKey=&searchValue=&menuNo=94&bbsId=204&pageOffset=20>

다운로드

실명환경 | **개발환경** | 운영환경 | 공통컴포넌트 | 모바일 실행환경 | 모바일 공통컴포넌트 | 모바일 디바이스 API

Release Notes | **4.x 다운로드** | 3.x 다운로드 | 2.x 다운로드 | 1.0 다운로드

개발환경 - 4.x 다운로드

4.x 다운로드

제목+내용 검색

4.1.0	구현도구(Implementation Tool) Version 4.1.0	릴리즈 일자 - 2023-05-19	▼
4.1.0	구현도구(Implementation Tool) 소스코드 Version 4.1.0	릴리즈 일자 - 2023-05-19	▼
4.0.0	개발자용 개발환경 for Windows 64bit (Implementation Tool) Version 4.0.0	릴리즈 일자 - 2022-03-02	▼
4.0.0	개발자용 개발환경 for macOS 64bit (Implementation Tool) Version 4.0.0	릴리즈 일자 - 2022-03-02	▼
4.0.0	개발자용 개발환경 for Linux 64bit (Implementation Tool) Version 4.0.0	릴리즈 일자 - 2022-03-02	▼
4.0.0	구현도구(Implementation Tool) Version 4.0.0	릴리즈 일자 - 2022-03-02	▼

Windows 제품 인증

- ② 개발자용 개발환경 for Windows 64bit (Implementation Tool) Version 4.0.0
내려받기

eGovFrameDev-4.0.0-64bit.zip

03 톰캣 설치하기

WAS 는 자바를 웹 브라우저에 게시할 수 있게 해주는 도구입니다.

제품 중에 무료로 사용 가능한 톰캣(Tomcat)으로 진행합니다.

상용 제품으로는 WebLogic 등이 있습니다.

자바 11 버전과 호환되는 톰캣 (Tomcat) 9 버전을 설치합니다.

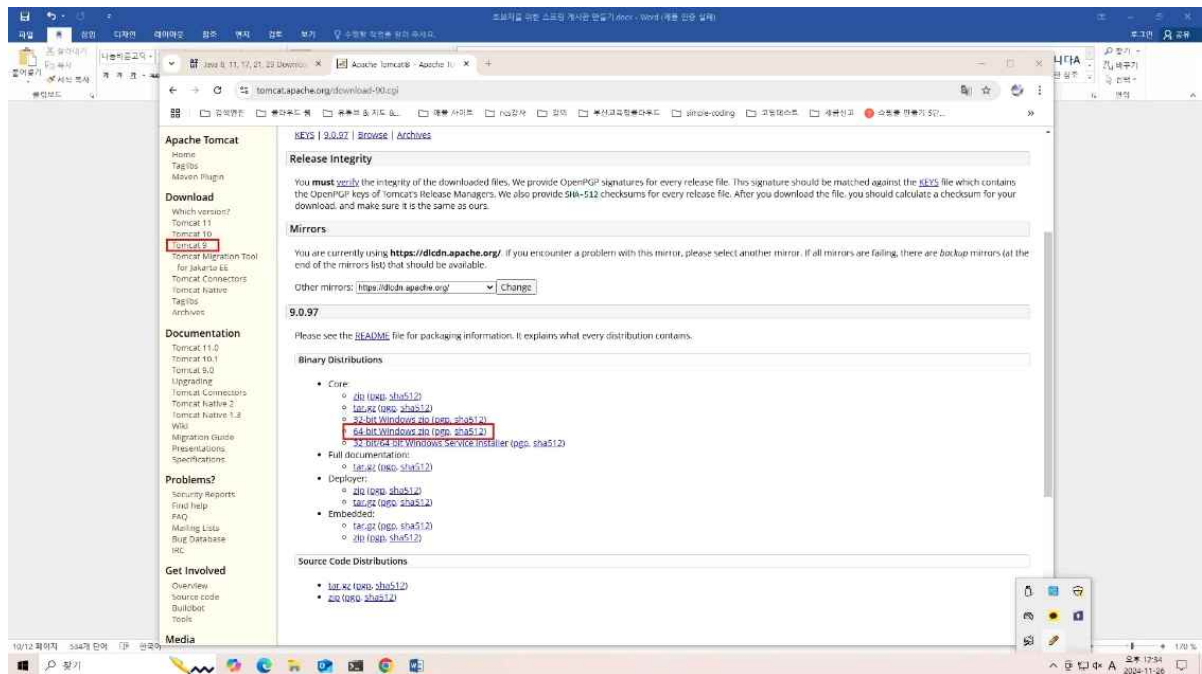
● 톰캣 9 버전을 내려받습니다.

① 다운로드 사이트:

<https://www.egovframe.go.kr/home/ntt/nttList.do?searchKey=&searchValue=&menuNo=94&bbsId=204&pageOffset=20>

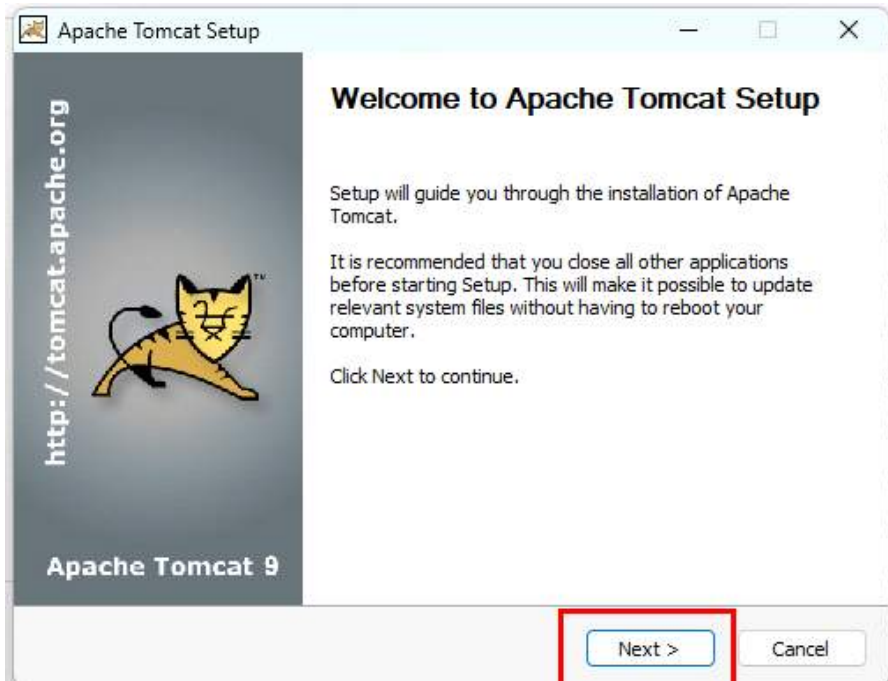
② Tomcat 9 다운로드 (윈도우 환경)

64-bit Windows zip (pgp, sha512) 클릭 다운로드



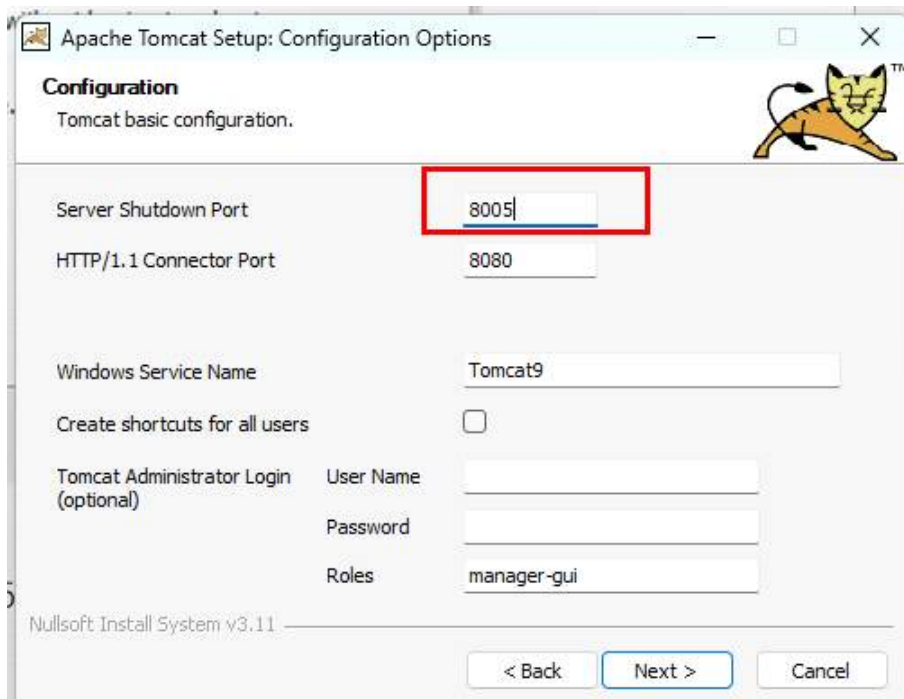
● 톰캣을 설치합니다.

① 톰캣 첫 화면입니다.



② 설치 중간에 아래 정보를 입력합니다.

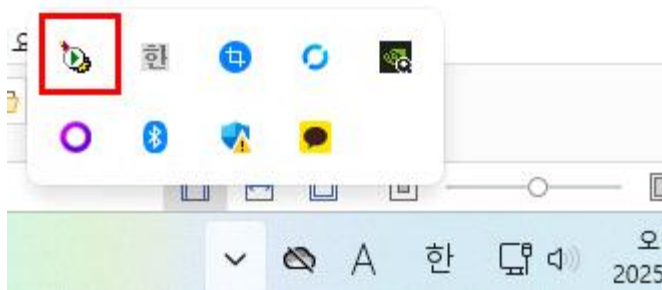
- Server Shutdown Port: -1 → 8005



초보자를 위한 SQL

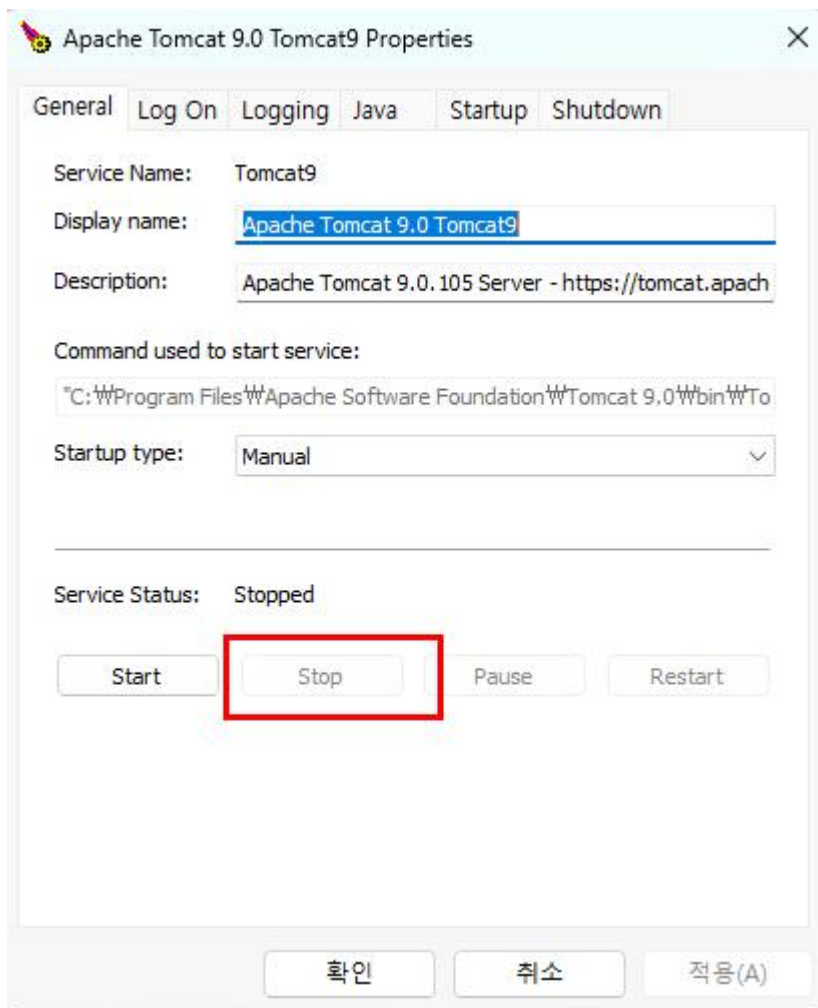
③ 톰캣이 마지막에 윈도우 서비스에 등록됩니다.

아래 아이콘을 바탕화면 작업표시줄에서 찾아서 클릭합니다.



④ STOP 버튼을 클릭합니다.

- 톰캣 서비스가 중지됩니다.
- 서비스 중지 이유는 **이클립스에서 톰캣 시작/중지**하기 위해서입니다.

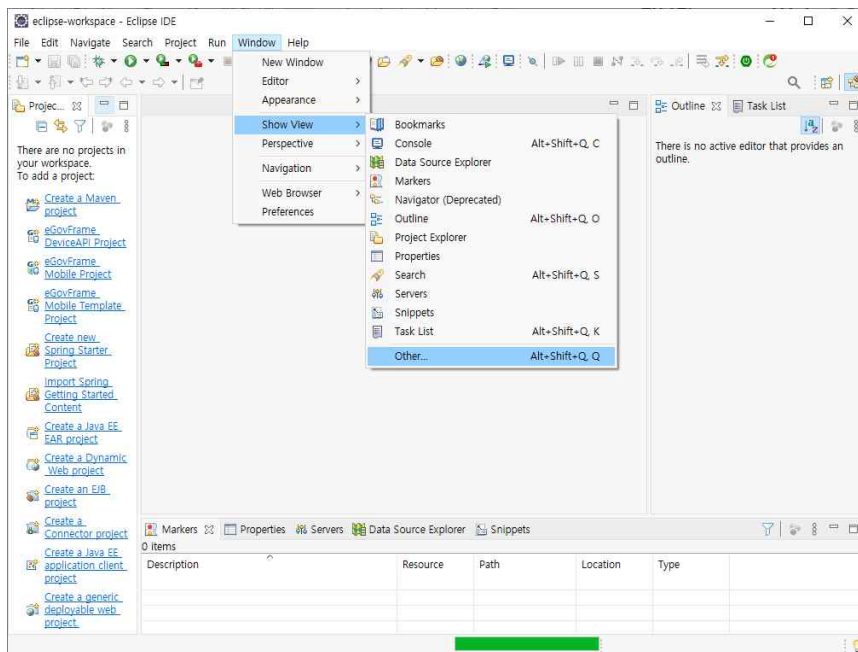


SECTION
03

이클립스에 톨킷 설정

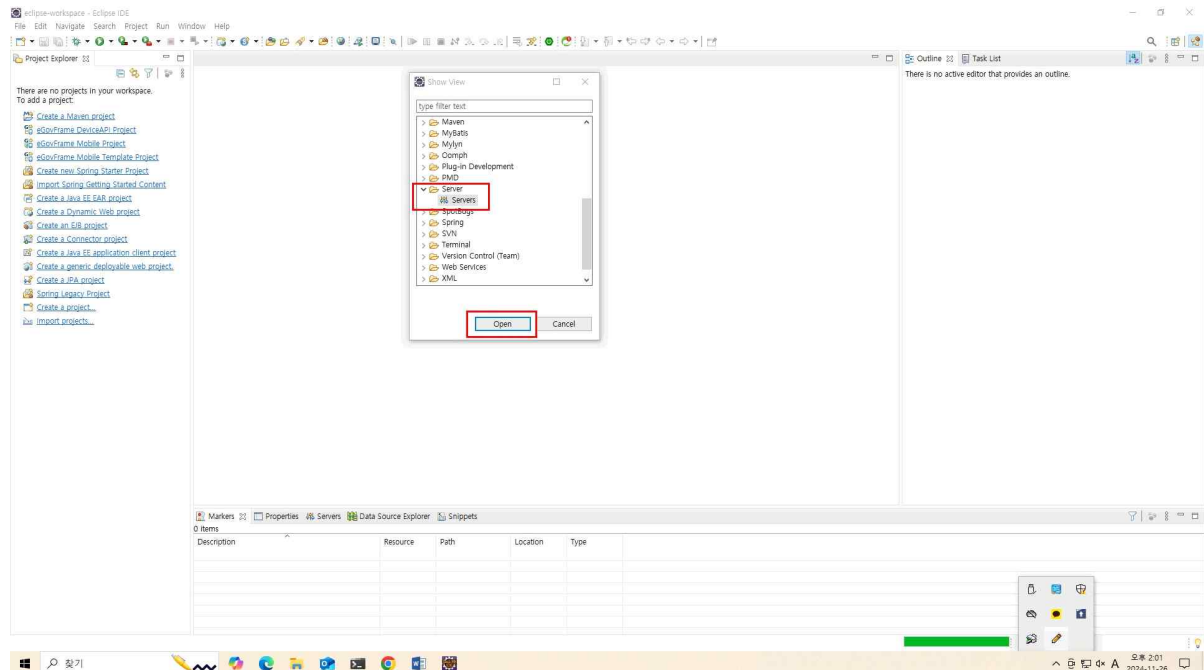
01 이클립스에 톨킷 설정하기

- 이클립스 메뉴에서 아래 그림을 따라하시면 됩니다.
단, 톨킷 9, 자바 11 버전이 사전에 설치되어야 합니다.
- ① Window – Show View – Other... 클릭



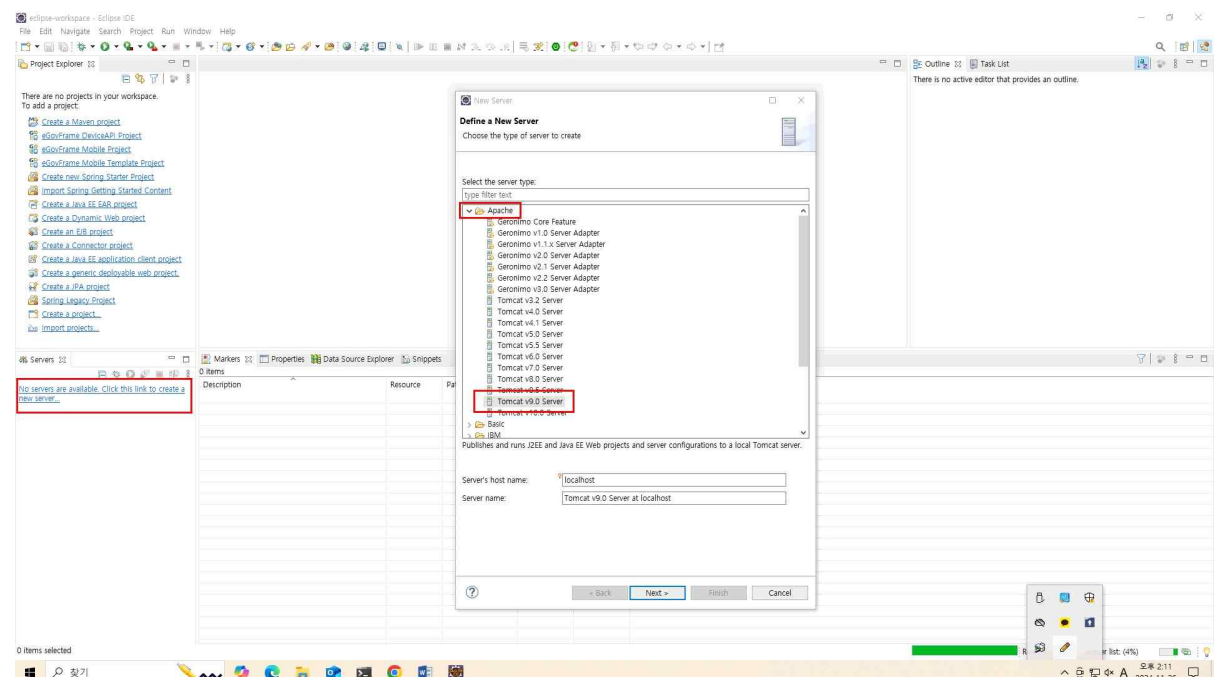
초보자를 위한 SQL

② Show View 창: Server – Servers 선택 Open 버튼 클릭



③ Servers 탭:

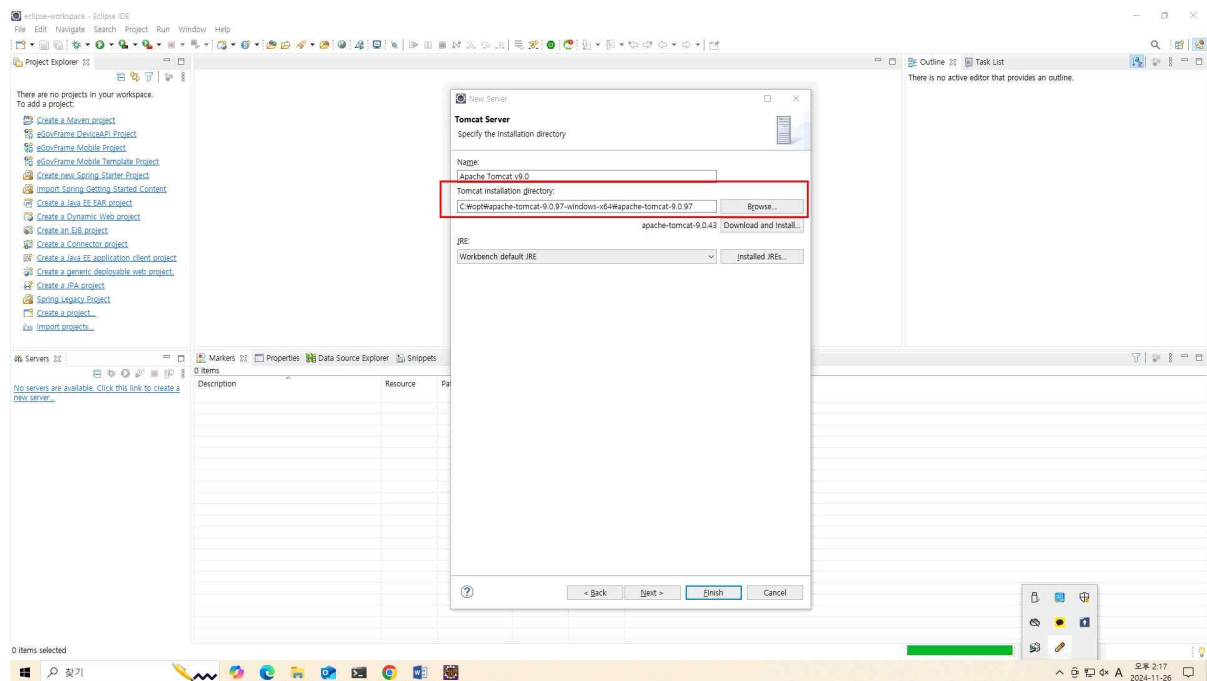
- No servers are... 링크 클릭
- Define a New Server 창: Apache ? Tomcat v9.0 Server



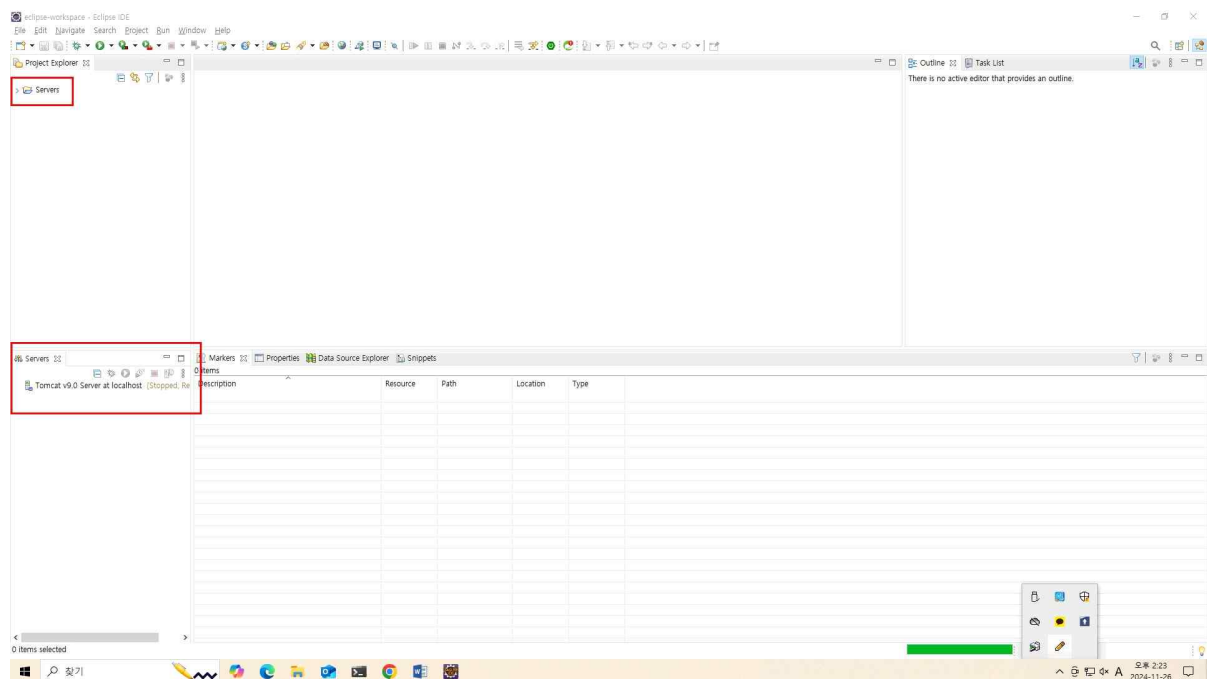
④ Next 버튼 클릭

⑤ 다운로드 받은 Tomcat 9 경로 선택

Chapter 01 Spring 개발 환경 설정하기



⑥ 설정 완료 화면: 아래 화면처럼 Servers 탭에 Tomcat v9.0 Server... 있으면 성공



Chapter

02

**스프링
프로젝트
시작하기**

SECTION
01

스프링 프로젝트 빠르게 만들기

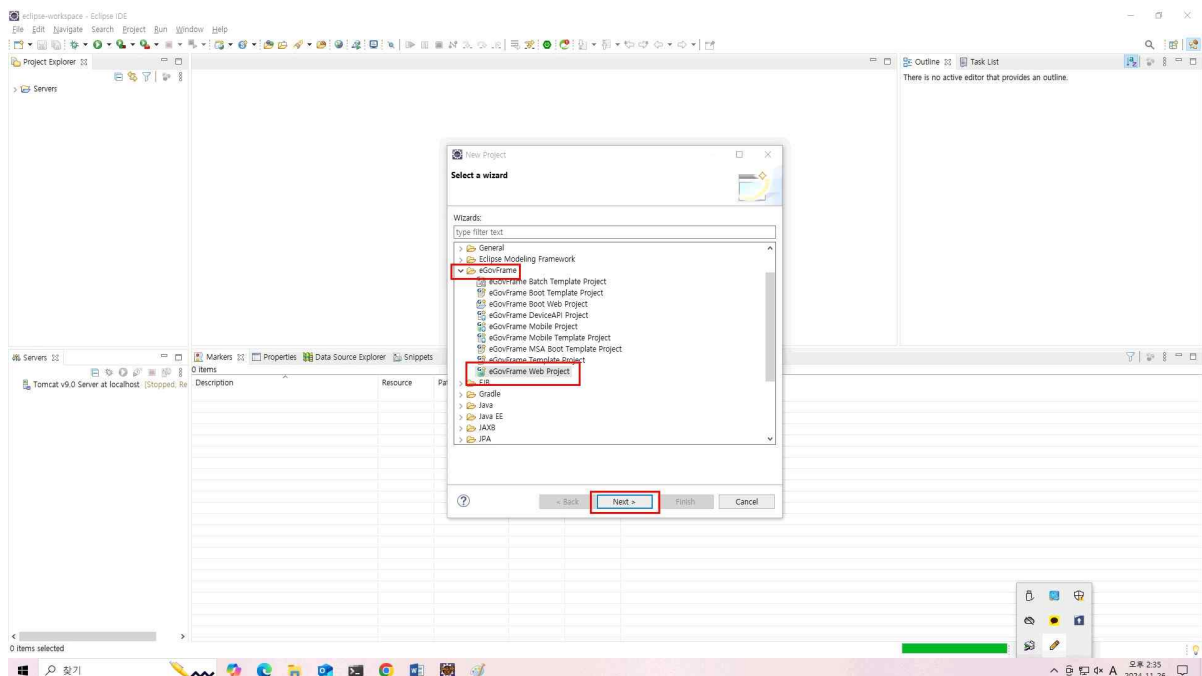
01

전자정부프레임워크 기본 환경에서 빠르게 만들어보기

전자정부프레임워크 기본 게시판 템플릿을 이용해서 스프링 환경설정을 완료해 봅니다.
전자정부프레임워크에는 게시판을 만들기 위한 다양한 템플릿이 있습니다.
이 책에서는 제공 기능 중 유효성 체크 기능만 사용합니다.

● 전자정부프레임워크 템플릿 사용하기

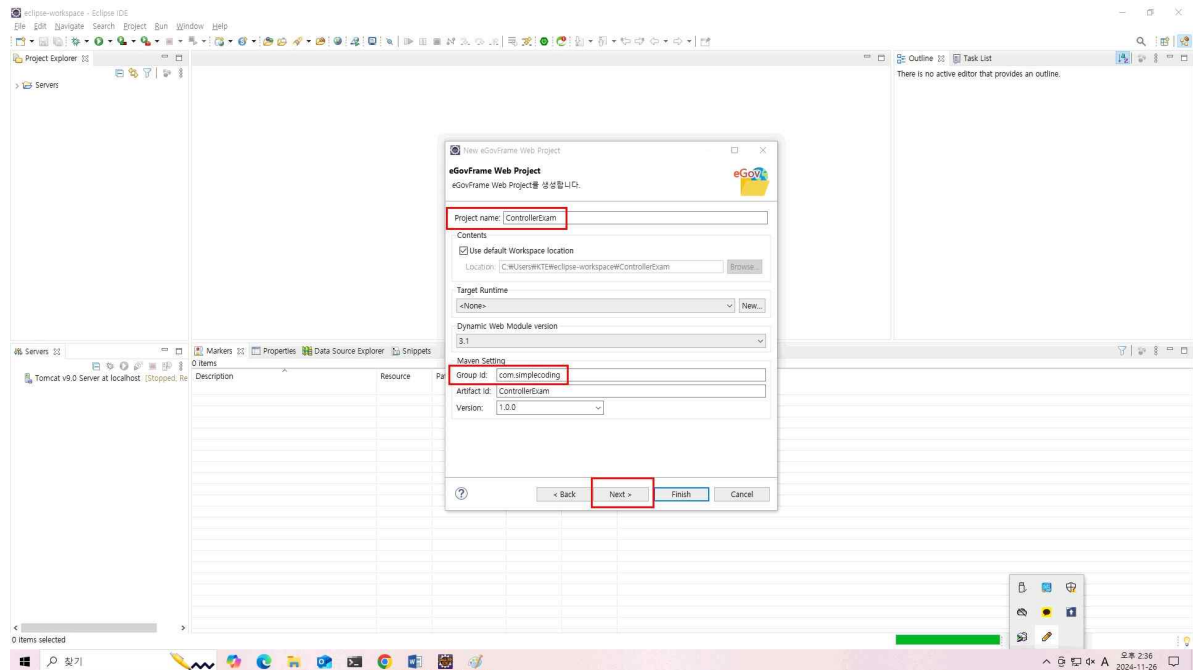
- ① 이클립스 메뉴: File – New – Project...
- ② eGovFrame: eGovFrame Web Project 선택



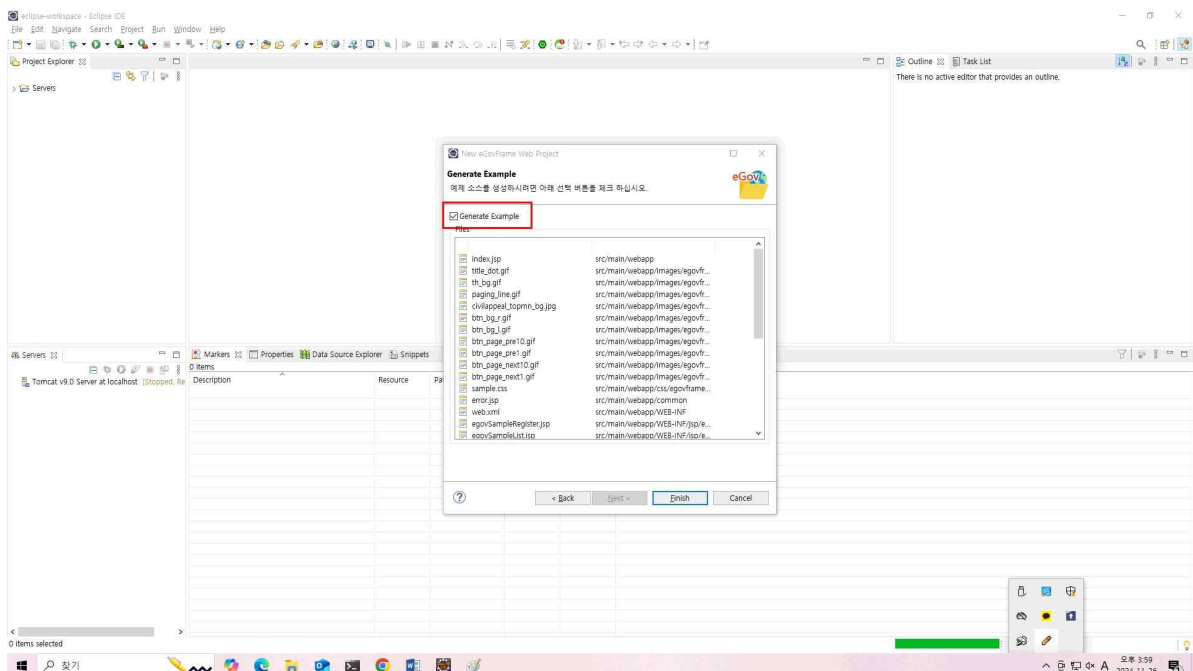
초보자를 위한 SQL

③ eGovFrame Web Project 창:

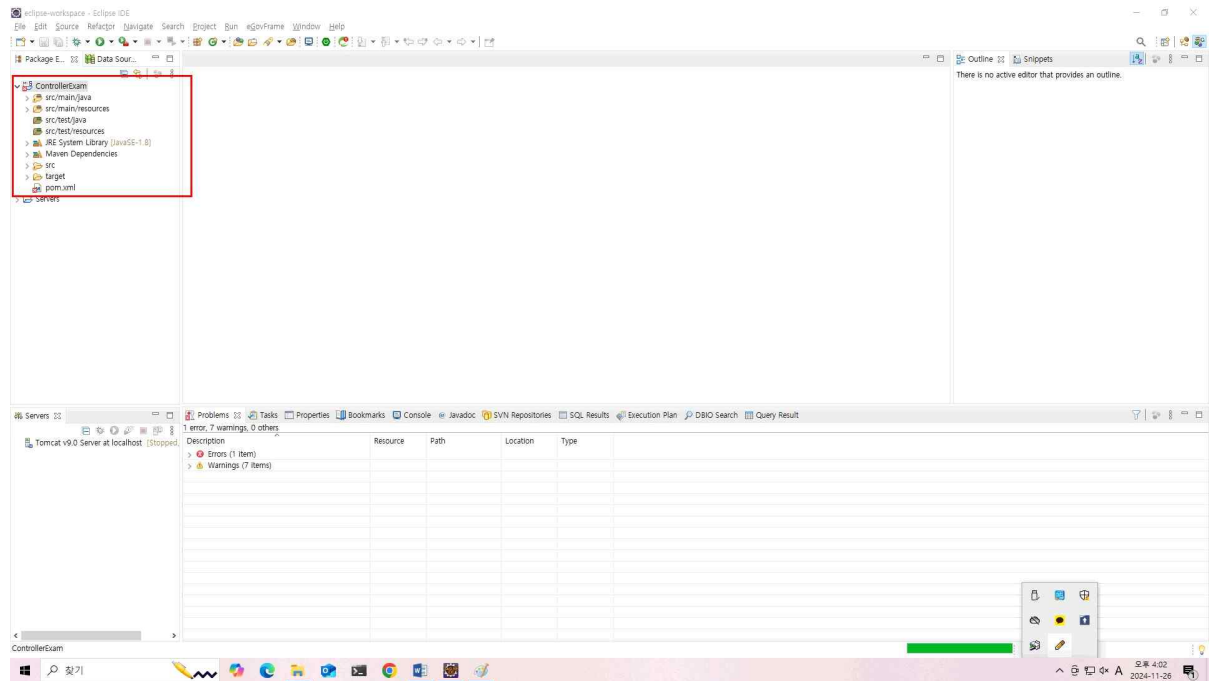
- Project name: ControllerExam
- Group Id: com.simple-coding



④ Next 버튼 클릭: Generate Example 선택, Finish 클릭



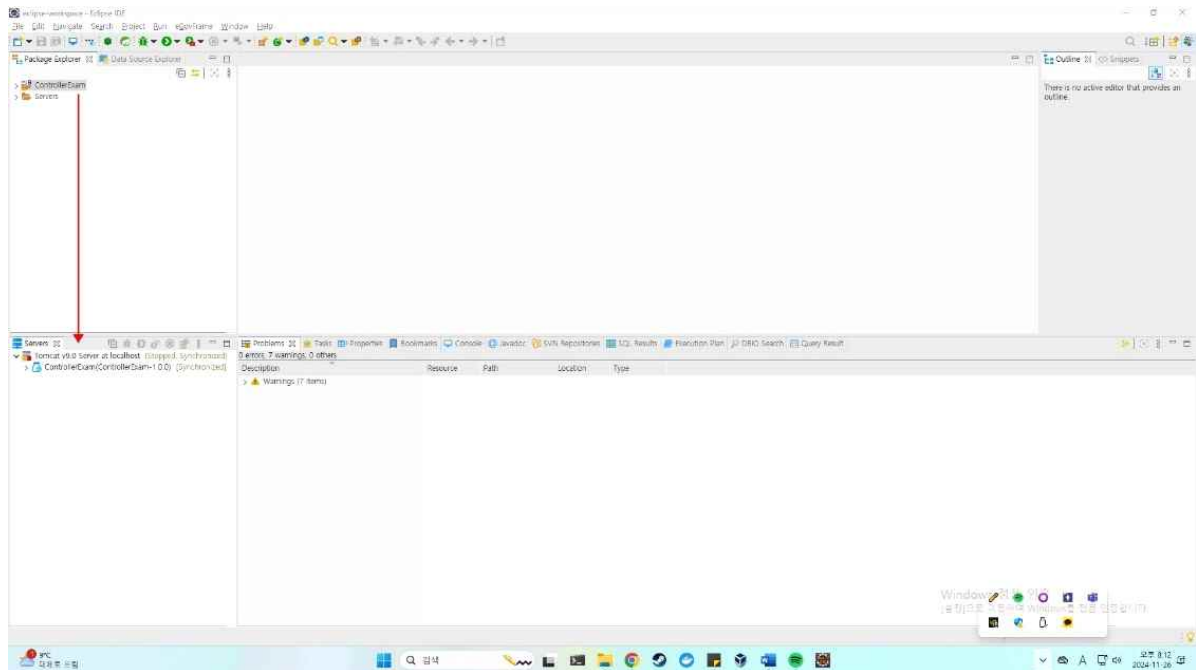
⑤ Open Perspective 버튼 클릭, 아래와 같이 보이면 성공



초보자를 위한 SQL

● ControllerExam 프로젝트 톰캣(Tomcat)에 배포하기

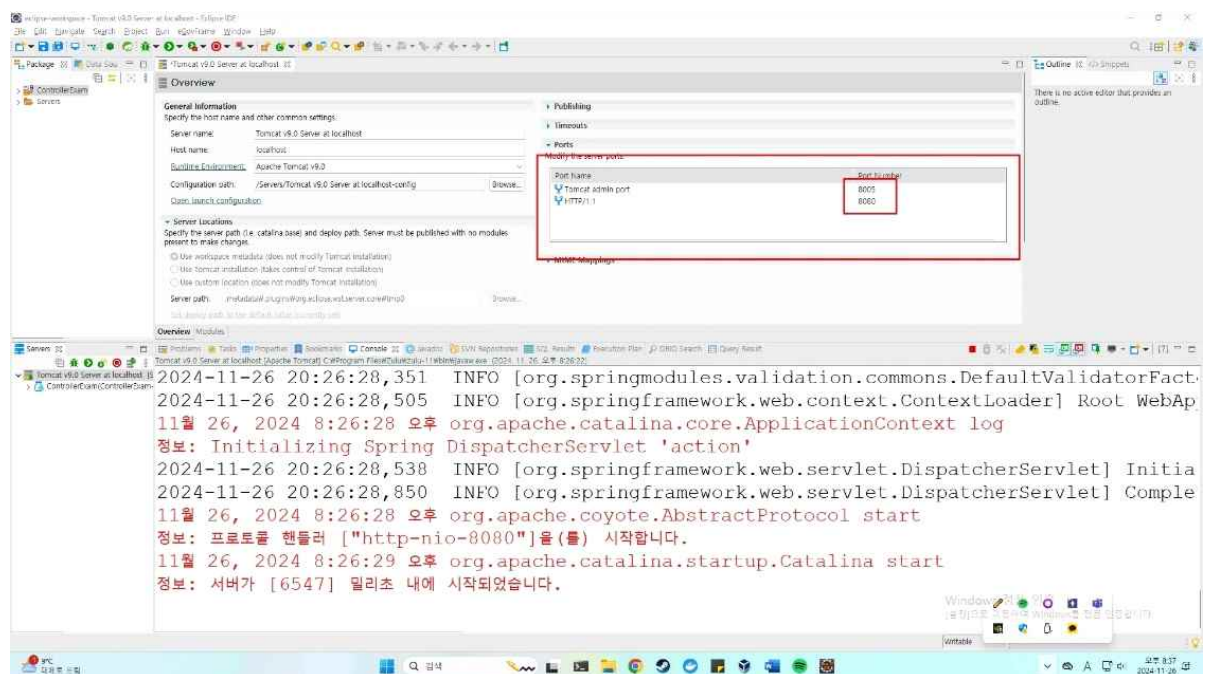
① ControllerExam 폴더를 Servers 탭의 Tomcat v9.0... 에 드래그하세요



② Tomcat v9.0... 더블 클릭: 창 보임

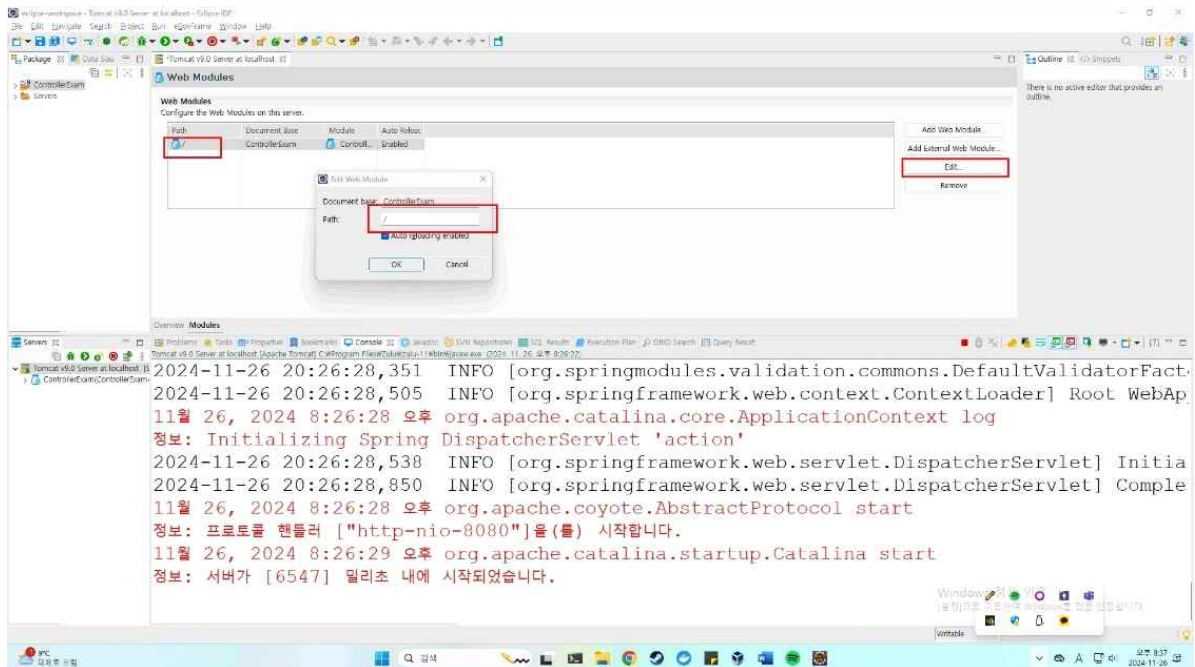
③ Overview 탭

- Tomcat admin port: 8005 확인
- HTTP/1.1 : 8080 확인



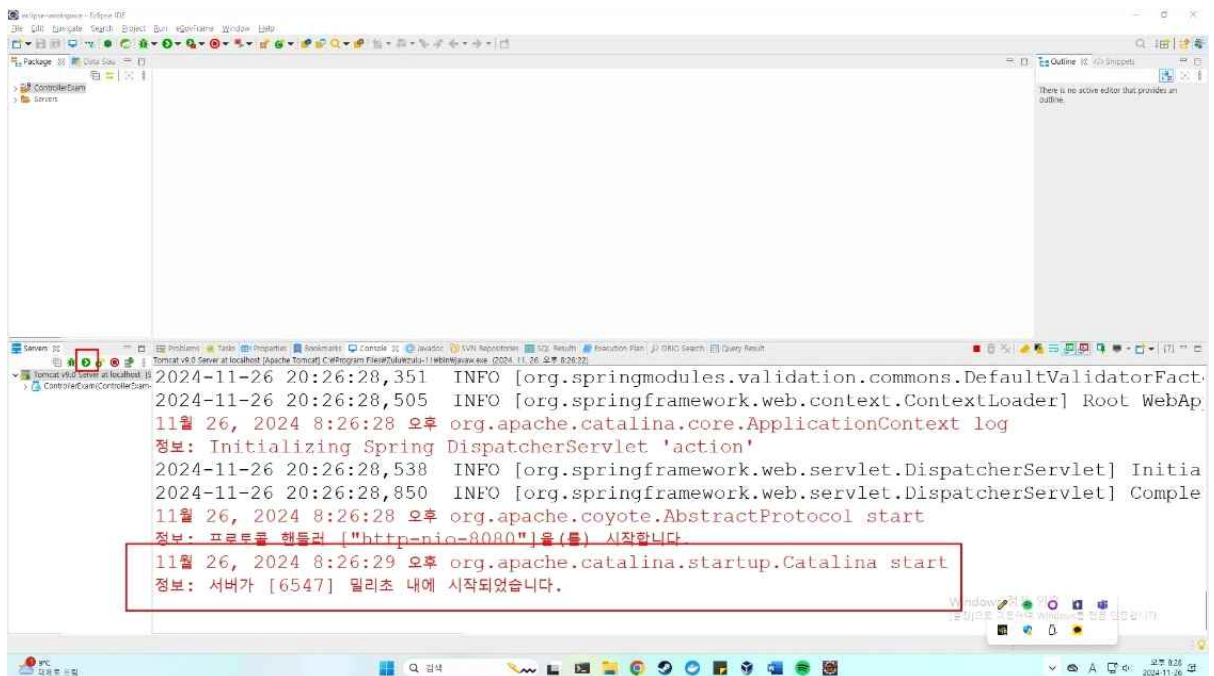
④ Modules 탭

- Edit 버튼 클릭
- Path: /ControllerExam → / 변경 (접속 URL 주소 변경)



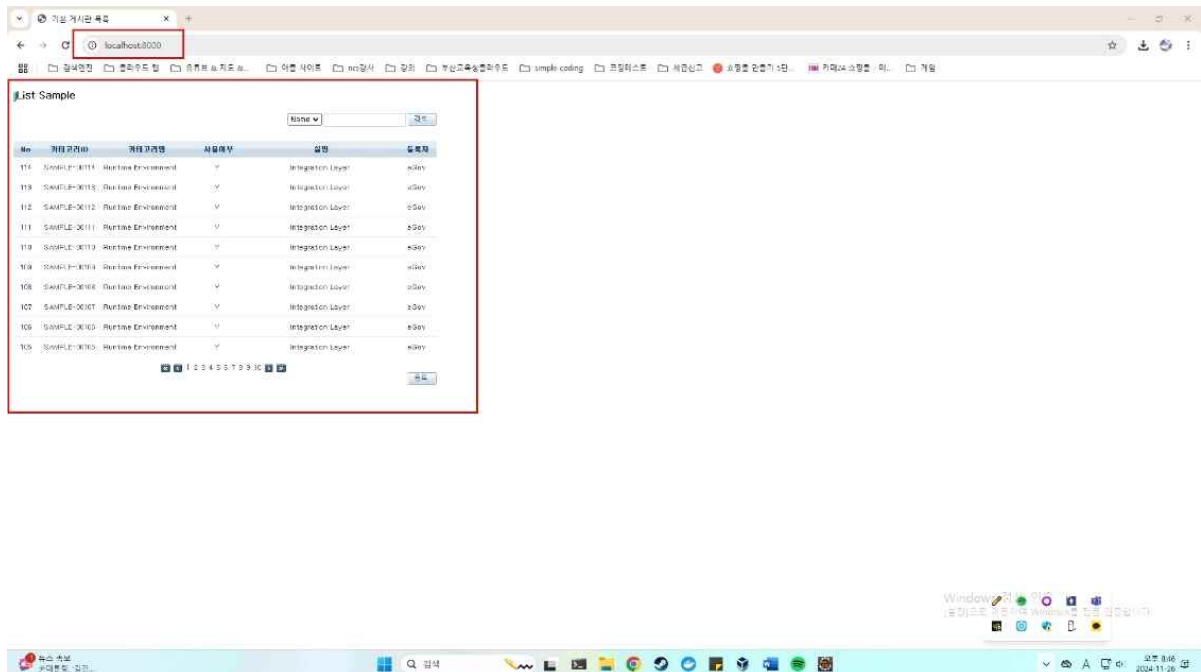
⑤ 설정 저장: Ctrl + s 타이핑

- Servers 탭: 시작 버튼 클릭하면 톰캣(Tomcat) 서버가 실행됩니다.



초보자를 위한 SQL

⑥ 웹브라우저 확인: http://localhost:8000



새로 나온 기술 알아보기

URL

http://localhost:8080/exam00/hello.do
위와 같은 인터넷주소를 말합니다.

URL 구조는 아래와 같습니다.
프로토콜://도메인:포트/경로/이름

- http: 프로토콜, 인터넷 통신 약속(규칙)입니다.
- localhost: 도메인이라고 합니다.
포트 번호와 경로를 제외한 인터넷주소입니다.

각 컴퓨터에서 기본 제공됩니다.

오직 자신만 볼 수 있고, 다른 사람은 볼 수 없습니다.
외부에 서비스하려면 주로 돈을 지불하고 도메인을 구매합니다.

- 8080: 포트 번호입니다. 도메인과 함께 사용합니다.
유일해야 합니다.

SPRING에서는 **8080 포트 번호**를 사용합니다.

	<ul style="list-style-type: none"> exam00/hello.do: 경로와 이름입니다. 이름은 유일해야 합니다. 각 경로/이름과 JSP 파일이 1:1 로 연결되어 관리됩니다.
WAS	<p>자바를 웹브라우저에서 서비스할 수 있게 해주는 프로그램입니다.</p> <p>대표적으로 무료인 톰캣이 있습니다.</p> <p>웹브라우저에서 실행되는 자바를 서블릿이라고 합니다.</p> <p>상용 제품으로는 웹로직이 유명합니다.</p>
서블릿	<p>자바에서 HTML, CSS, JS 를 섞어서 사용할 수 있게 하는 기술입니다.</p> <p>WAS 는 서블릿을 실행해서 결과를 웹브라우저에 표시합니다.</p> <p>코딩하기 복잡합니다.</p> <p>예) <code>out.println("<html>");</code></p>
JSP	<p>서블릿의 단점을 개선했습니다.</p> <p>HTML 처럼 코딩할 수 있습니다.</p> <p>또한 백엔드 결과를 전달받을 수 있습니다.</p> <p>화면 입력을 백엔드를 보낼 수도 있습니다.</p> <p>스프링에서 화면을 표시하는 기술로 사용합니다.</p> <p>MVC 디자인 패턴에서 뷰에 해당합니다.</p> <p>자동으로 서블릿으로 변경 코딩됩니다.</p> <p>서블릿의 복잡성과 가독성 저하로 JSP 가 등장했습니다.</p> <p>JSP 모델 1, 모델 2, SPRING MVC 코딩법이 있습니다.</p>
JSP 모델 1	<p>JSP 에서 HTML 과 자바 코딩을 동시에 하는 코딩법입니다.</p> <p>2 개가 섞여 있어서 가독성이 떨어집니다.</p> <p>소스가 길어질수록 수정/추가가 복잡합니다.</p>

	요즘은 잘 사용하지 않습니다.
JSP 모델 2	<p>JSP에서는 HTML, CSS 같은 화면에 관련된 코딩만 합니다.</p> <p>일부 백엔드로 입력값을 보내거나 받는 코딩이 포함될 수 있습니다.</p> <p>화면과 관계없는 코딩은 자바로 하는 코딩법입니다.</p> <p>예를 들면 URL로 데이터를 요청했을 때 DB에 조회하는 코딩은 자바에서 합니다.</p> <p>스프링 프레임워크는 JSP 모델 2를 사용합니다.</p>
SPRING 프레임워크	<p>JSP 모델 2가 내장되어 있습니다.</p> <p>또한 MVC 디자인 패턴에 따라 코딩합니다.</p> <p>과거에 유명한 IT 전문가 4명이 공동 집필한 디자인 패턴 책에서 MVC 패턴이 소개되었습니다.</p> <p>다음 섹션에서 상세하게 소개합니다.</p>

SECTION
02

Hello World 화면에 표시하기

01 web.xml 호환성 설정

web.xml 은 전자정부프레임워크에서 만들면 3.1 버전으로 만들어 집니다.
 실행에는 문제가 없으나 버전으로 인한 에러가 발생할 수 있습니다.
 버전을 아래와 같이 지우고 사용하실 것을 추천합니다.

2 번째 라인을 아래와 같이 수정함(버전 표시 없음)

web.xml

```
<web-app id="WebApp_ID" xmlns="http://~ /web-app.xsd">
```

02 샘플을 이용해서 Hello World 를 화면에 표시합니다.

전자정부프레임워크 기본 게시판 템플릿을 이용해서 스프링 환경 설정을 일부 변경하고
 Hello World 를 화면에 표시해 봅시다.

또한 Spring 은 다양한 **어노테이션을 이용한 코딩**을 합니다.
 어노테이션은 메소드처럼 간단한 기능이 내장되어 있습니다.
 클래스, 메소드, 필드에 붙여 새로운 추가 기능을 부여할 수 있습니다.
 장점은 코딩이 간단해지고 가독성이 높아집니다.

어노테이션을 이용한 코딩도 같이 알아보시다.

● 전자정부프레임워크에 간단한 예제를 만들어봅시다.

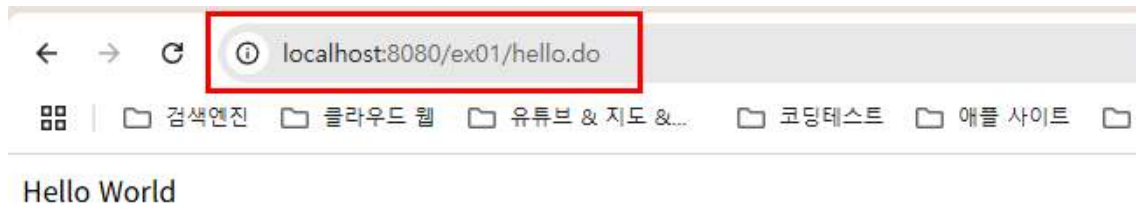
단, sample 패키지를 참고하세요

① 예제) Hello World 를 화면에 표시하세요.

아래 그림처럼 컨트롤러와 JSP 를 코딩하세요

초보자를 위한 SQL

[결과]



[JAVA/JSP]



요건

테스트 URL	http://localhost:8080/ex01/hello.do
패키지	egovframework.example.controllerexam.web.ex01
자바	BasicController.java
메소드	hello()

JSP	ex01/hello
-----	------------

BasicController.java

```
/**
 *
 */
package egovframework.example.controllerexam.web.ex01;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.RequestMapping;
import
org.springframework.web.bind.annotation.RequestMethod;
/**
 * @author simple-coding
 *
 */
@Controller
public class BasicController{
    @RequestMapping(value ="/ex01/hello.do", method
=RequestMethod.GET)
    public String hello() {
        return "/ex01/hello";
    }
}
```

hello.jsp

```

<%@ page language="java" contentType="text/html;
charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
Hello World
</body>
</html>

```

새로 나온 기술 알아보기

<p>SPRING MVC 패턴</p>	<p>JSP 모델 2 가 포함되어 있습니다.</p> <p>JSP 는 화면 코딩을 담당합니다. 자바는 화면과 관계없는 코딩을 담당합니다.</p> <p>좀 더 세부적으로 디자인 패턴 책에 있는 MVC 패턴 코딩법을 따릅니다. 장점은 폴더 관리와 유사합니다. 목적에 따라 분리된 폴더는 수정/추가가 쉽습니다.</p> <p>SPRING M(MODEL)/V(VIEW)/C(CONTROLLER)로 자바 소스를 나누어 관리합니다. 3 가지는 아래와 같이 목적에 따른 코딩을 합니다.</p>
<p>모델 (MODEL)</p>	<p>M(MODEL) 데이터에 관련된 코딩을 합니다. 예를 들면 DB 를 말합니다.</p>
<p>뷰 (VIEW)</p>	<p>V(VIEW) 화면에 관련된 코딩을 합니다. 예를 들면 JSP 를 말합니다.</p>

컨트롤러 (CONTROLLER)	C(CONTROLLER) 는 URL 과 화면을 연결해주고 MODEL 에서 전달받은 결과를 보내주기도 합니다.
디자인 패턴	<p>과거에 4 명의 유명한 IT 전문가가 모여 실무에서 쓸만한 23 가지 코딩 패턴을 책으로 출판했습니다. 디자인 패턴에 따라 코딩하면 미래에 기능 추가/변경이 발생할 때 코딩이 편리해집니다.</p> <p>무분별한 적용은 오히려 코딩 복잡도를 증가시킵니다. 미래에 추가/변경이 많이 일어날 것 같은 필수적인 곳에서만 사용하는 것이 좋습니다.</p> <p>SPRING 은 일부 디자인 패턴을 사용하고 있습니다. 그중에서 SPRING MVC 디자인 패턴이 가장 유명합니다. 실무에서 많이 활용됩니다.</p>
GET	<p>웹페이지에서 통신하는 방식입니다. 몇 가지가 약속되어 있습니다. 예를 들면, 편지도 긴급, 일반 등 목적에 따라 구분합니다. 마찬가지로 인터넷도 목적에 따라 통신하고 있습니다. GET 방식(METHOD)은 백엔드에 결과를 요청합니다.</p>
POST	POST 방식은 화면에서 입력된 글자를 백엔드에 저장할 것을 요청합니다.
@Controller	<p>클래스에 컨트롤러 기능을 부여합니다.</p> <p>각 메소드는 이제 URL 경로를 붙일 수 있고 결과를 return 예약어로 JSP 화면으로 내보내기 합니다. 각 메소드의 URL 경로는 유일해야 합니다.</p>
@RequestMapping	메소드에 해당 URL 경로에 따른 결과를 JSP 화면에 전달하는 기능을 제공합니다.

	<p>어노테이션의 () 안에는 특별한 속성과 값이 포함될 수 있습니다.</p> <ul style="list-style-type: none"> ▪ value="URL" ▪ method = RequestMethod.GET ▪ return jsp 이름; <p>해석: 웹브라우저에서 URL 경로를 입력하면 해당 URL 경로가 붙은 메소드가 실행됩니다. GET 방식 요청입니다.</p> <p>메소드 안에서 결과를 return 예약어로 JSP 로 전달합니다.</p>
<%@ ~ %>	<p>JSP 의 페이지 지시자입니다. 문서 종류, 언어셋 등을 넣을 수 있습니다.</p> <p>기본적으로 JSP 첫줄에 작성합니다.</p>
사용법	<pre>@Controller public class 클래스{ @RequestMapping(value ="url 경로", method =RequestMethod.GET) public String 메소드() { return "폴더/jsp 명"; } }</pre>
JSP 사용법	<pre><%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%> <!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <title>Insert title here</title> </head> <body></pre>

초보자를 위한 SQL

```
Hello World  
</body>  
</html>
```

Chapter

03

컨트롤러

배우기

SECTION
01

@GetMapping, Model 배우기

01

컨트롤러에 사용하는 기본 어노테이션을 배워봅시다.

전자정부프레임워크 기본 게시판 템플릿을 이용해서 컨트롤러 코딩을 해봅시다.
컨트롤러는 URL 로 요청하면 특정 결과를 JSP 로 내보내기를 하는 메소드를 작성할 수 있습니다.
또한 컨트롤러에 사용하는 기본 어노테이션들이 있습니다.

실습을 위한 Lombok 과 Ajax 라이브러리를 설치합니다.

pom.xml 파일에 **Lombok, Ajax** 추가합니다.

dependencies 태그 안에 추가합니다.

제일 아래쪽을 추가해 주세요

pom.xml

```
...  
  
    <!-- Lombok 추가 -->  
    <dependency>  
        <groupId>org.projectlombok</groupId>  
        <artifactId>lombok</artifactId>  
        <version>1.18.34</version>  
        <scope>provided</scope>  
    </dependency>  
  
    <!--ajax-->  
    <dependency>  
        <groupId>com.fasterxml.jackson.core</groupId>
```

```

<artifactId>jackson-core</artifactId>
<version>2.9.4</version>
</dependency>

<dependency>
<groupId>com.fasterxml.jackson.core</groupId>
<artifactId>jackson-databind</artifactId>
<version>2.9.4</version>
</dependency>
</dependencies>

```

dispatcher-servlet.xml 에 아래 태그 추가합니다.

ajax 실행을 위한 태그입니다.

context:component-scan 태그 아래 적당한 위치에 추가하면 됩니다.

dispatcher-servlet.xml

```

<!-- ajax -->
<mvc:annotation-driven />

```

02 따라 하기 예제로 알아봅시다.

● @GetMapping 사용하기

① 예제) 안녕 JSP & Spring 을 화면에 표시하세요.

아래 그림처럼 컨트롤러와 JSP 를 코딩하세요

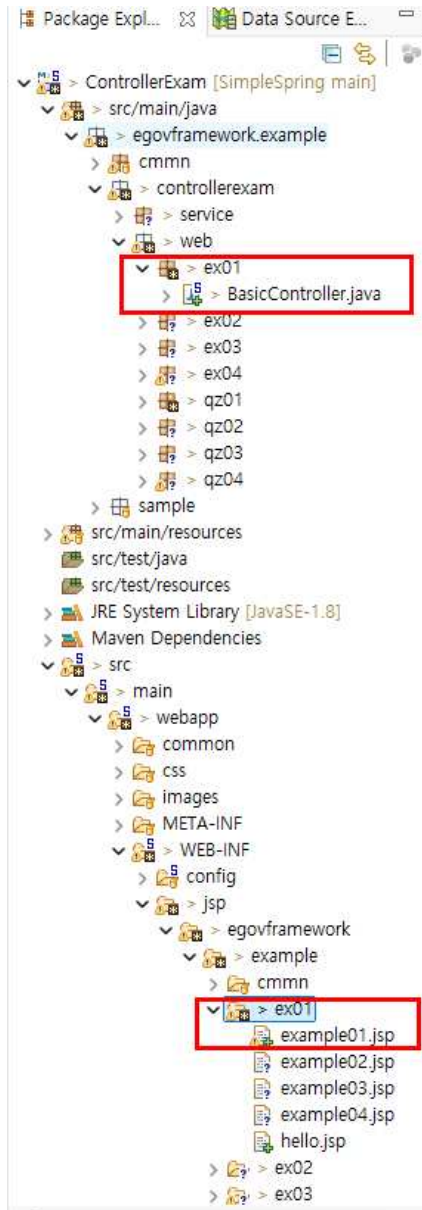
단, @GetMapping 을 사용하세요

초보자를 위한 SQL

[결과]



[JAVA/JSP]



요건

테스트 URL	http://localhost:8080/ex01/example01.do
패키지	egovframework.example.controllerexam.web.ex01
자바	BasicController.java
메소드	example01
매개변수	없음

리턴 자료형	String
JSP	ex01/example01

BasicController.java

```
/**
 *
 */
package egovframework.example.controllerexam.web.ex01;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.RequestMapping;
import
org.springframework.web.bind.annotation.RequestMethod;
/**
 * @author simple-coding
 *
 */
@Controller
public class BasicController{
    @RequestMapping(value = "/ex01/hello.do", method
    =RequestMethod.GET)
    public String hello() {
        return "ex01/hello";
    }
    // 따라하기 예제 1
    @GetMapping("/ex01/example01.do")
    public String example01() {
        return "ex01/example01";
    }
}
```


}

example01.jsp

```

<%@ page language="java" contentType="text/html;
charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
안녕 JSP & Spring
</body>
</html>

```

새로 나온 기술 알아보기

@GetMapping	<p>메소드에 해당 URL 경로에 따른 결과를 JSP 화면에 전달하는 기능을 제공합니다.</p> <p>GET 방식 요청입니다.</p> <p>@RequestMapping 어노테이션과 같습니다.</p> <p>좀더 간단하게 사용할 수 있습니다.</p>
사용법	<pre> @Controller public class 클래스{ @GetMapping("url 경로") public String 메소드() { return "폴더/jsp 명"; } } </pre>

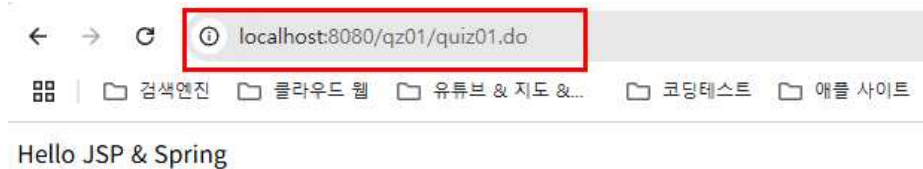
초보자를 위한 SQL

② 퀴즈) Hello JSP & Spring 을 화면에 표시하세요.

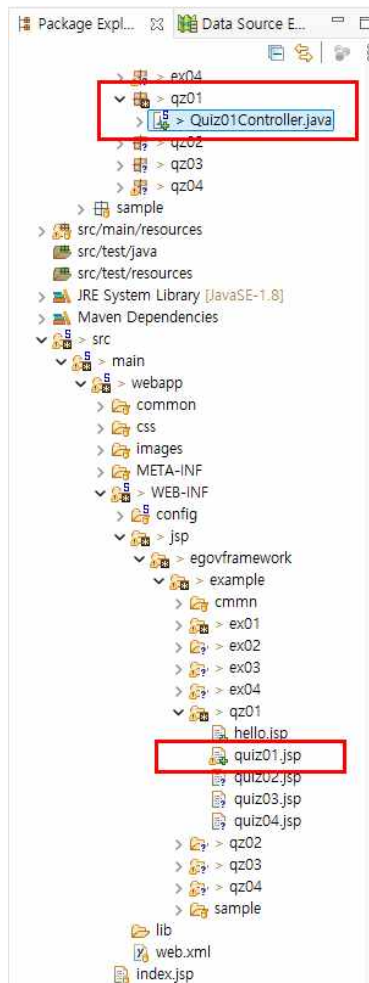
아래 그림처럼 컨트롤러와 JSP 를 코딩하세요

단, @GetMapping 을 사용하세요

[결과]



[JAVA/JSP]



요건

테스트 URL	http://localhost:8080/qz01/quiz01.do
패키지	egovframework.example.controllerexam.web.qz01
자바	Quiz01Controller.java
메소드	quiz01
매개변수	없음
리턴 자료형	String
JSP	qz01/quiz01

해답 : <https://github.com/simple-coding-forbob/SimpleSpring>

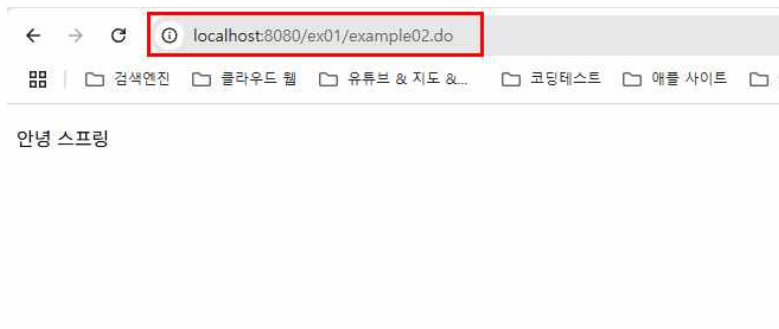
초보자를 위한 SQL

③ 예제) 안녕 JSP & Spring 을 화면에 표시하세요.

아래 그림처럼 컨트롤러와 JSP 를 코딩하세요

단, @GetMapping, Model 을 사용하세요

[결과]



요건

테스트 URL	http://localhost:8080/ex01/example02.do
패키지	egovframework.example.controllerexam.web.ex01
자바	BasicController.java
메소드	example02
매개변수	Model model
리턴 자료형	String
Model (키, 값)	키: "greeting", 값: "안녕 스프링"
JSP	ex01/example02

BasicController.java

```
/**
 *
 */
package egovframework.example.controllerexam.web.ex01;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
```

```
import
org.springframework.web.bind.annotation.RequestMapping;
import
org.springframework.web.bind.annotation.RequestMethod;
/**
 * @author simple-coding
 *
 */
@Controller
public class BasicController{
@RequestMapping(value = "/ex01/hello.do", method
=RequestMethod.GET)
    public String hello() {
        return "ex01/hello";
    }
// 따라하기 예제 1
@GetMapping("/ex01/example01.do")
    public String example01() {
        return "ex01/example01";
    }

// 따라하기 예제 3
@GetMapping("/ex01/example02.do")
    public String example02(Model model) {
        model.addAttribute("greeting", "안녕 스프링");
        return "ex01/example02";
    }
}
```

초보자를 위한 SQL

example02.jsp

```
<%@ page language="java" contentType="text/html;
charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <p>${greeting}</p>
</body>
</html>
```

새로 나온 기술 알아보기

Model	컨트롤러에서 값을 JSP 로 전달해주는 객체입니다. JSP 는 Model 의 키로 전달받고, 값을 화면에 표시합니다. Map 자료구조와 비슷합니다.
{키이름}	JSP 의 EL 표현식입니다. 자바에서 보내준 정보를 html 화면에 표현할 수 있습니다. 위 소스는 컨트롤러에서 보내준 모델 키에 해당 되는 값을 화면에 표시합니다.
자바 사용법	<pre>@Controller public class 클래스{ @GetMapping("url 경로") public String 메소드(Model model) { model.addAttribute("키", "값"); } }</pre>

	<pre> return "폴더/jsp명"; } }</pre>
JSP 사용법	<pre>\${키}</pre>

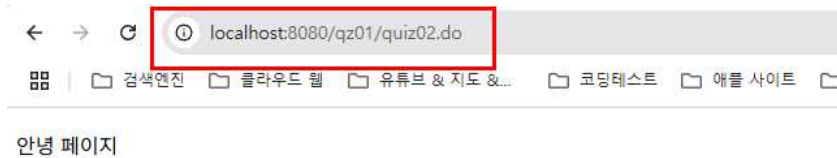
초보자를 위한 SQL

④ 퀴즈) 안녕 페이지를 화면에 표시하세요.

아래 그림처럼 컨트롤러와 JSP 를 코딩하세요

단, @GetMapping, Model 을 사용하세요

[결과]

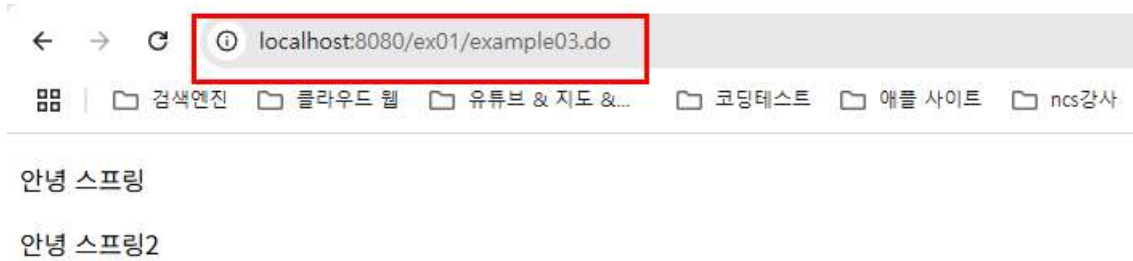


요건

테스트 URL	http://localhost:8080/qz01/quiz01.do
패키지	egovframework.example.controllerexam.web.qz01
자바	Quiz01Controller.java
메소드	quiz02
매개변수	Model model
리턴 자료형	String
Model (키, 값)	키: "quiz", 값: "안녕 페이지"
JSP	qz01/quiz02

해답 : <https://github.com/simple-coding-forbob/SimpleSpring>

- ⑤ 예제) 아래 그림처럼 컨트롤러와 JSP 를 코딩하세요
단, @GetMapping, Model 은 2 개 사용하세요



요건

테스트 URL	http://localhost:8080/ex01/example03.do
패키지	egovframework.example.controllerexam.web.ex01
자바	BasicController.java
메소드	example03
매개변수	Model model
리턴 자료형	String
Model (키, 값)	키: "greeting", 값: "안녕 스프링" 키: "greeting2", 값: "안녕 스프링 2"
JSP	ex01/example03

BasicController.java

```
/**
 *
 */
package egovframework.example.controllerexam.web.ex01;
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.RequestMapping;
import
org.springframework.web.bind.annotation.RequestMethod;
/**
 * @author simple-coding
 *
 */
@Controller
public class BasicController{
    @RequestMapping(value = "/ex01/hello.do", method
    =RequestMethod.GET)
        public String hello() {
            return "/ex01/hello";
        }
    // 따라하기 예제 1
    @GetMapping("/ex01/example01.do")
        public String example01() {
            return "ex01/example01";
        }

    // 따라하기 예제 3
    @GetMapping("/ex01/example02.do")
        public String example02(Model model) {
            model.addAttribute("greeting", "안녕 스프링");
            return "ex01/example02";
        }

    // 따라하기 예제 5
    @GetMapping("/ex01/example03.do")
        public String example03(Model model) {
            model.addAttribute("greeting", "안녕 스프링");
```

```

        model.addAttribute("greeting2", "안녕 스프링 2");
        return "ex01/example03";
    }
}

```

example03.jsp

```

<%@ page language="java" contentType="text/html;
charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <p>${greeting}</p>
    <p>${greeting2}</p>
</body>
</html>

```

새로 나온 기술 알아보기

자바 사용법

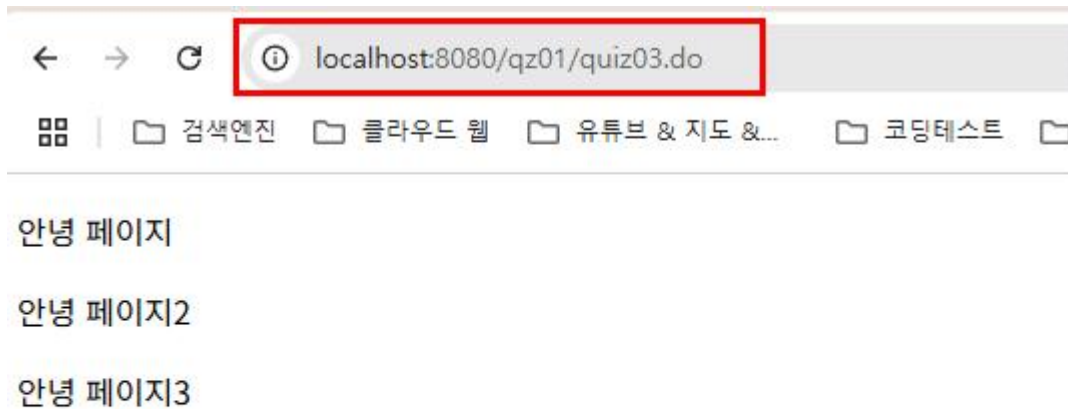
```

@Controller
public class 클래스{
    @GetMapping("url 경로")
    public String 메소드(Model model) {
        model.addAttribute("키", "값");
        model.addAttribute("키 2", "값 2");
        ...
        return "폴더/jsp 명";
    }
}

```

JSP 사용법	<code>\${키}</code> <code>\${키 2}</code>
---------	--

- ⑥ 퀴즈) 아래 그림처럼 컨트롤러와 JSP 를 코딩하세요
단, @GetMapping, Model 은 3 개 사용하세요

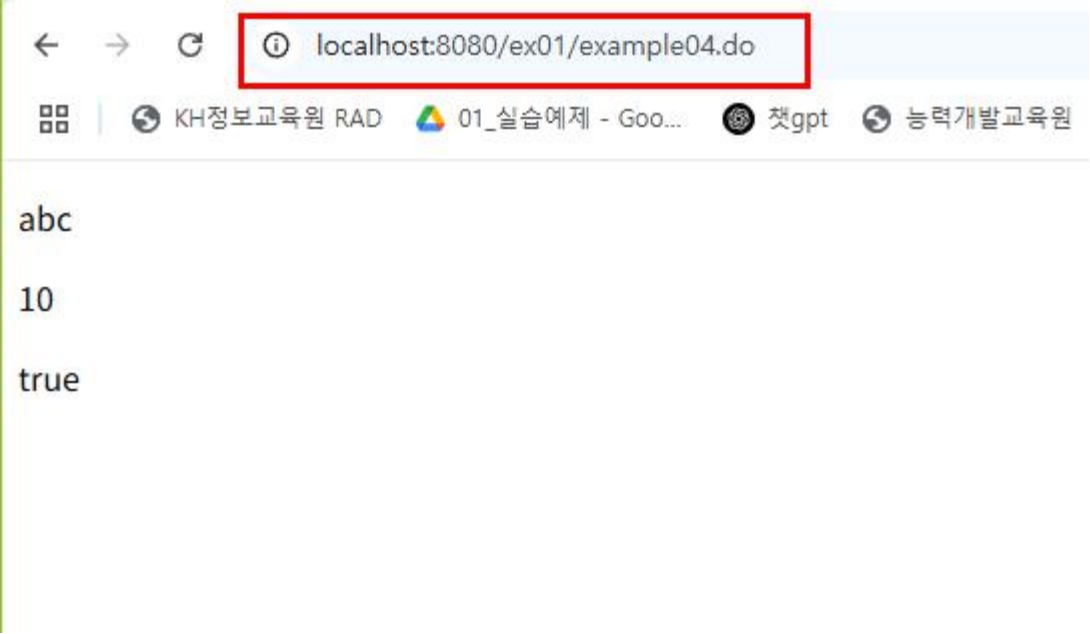


요건

테스트 URL	http://localhost:8080/qz01/quiz01.do
패키지	egovframework.example.controllerexam.web.qz01
자바	Quiz01Controller.java
메소드	quiz03
매개변수	Model model
리턴 자료형	String
Model (키, 값)	키: "quiz", 값: "안녕 페이지" 키: "quiz2", 값: "안녕 페이지 2" 키: "quiz3", 값: "안녕 페이지 3"
JSP	qz01/quiz03

해답 : <https://github.com/simple-coding-forbob/SimpleSpring>

- ⑦ 예제) 아래 그림처럼 컨트롤러와 JSP 를 코딩하세요
단, @GetMapping, Model 은 3 개 사용하고 키는 자유롭게 코딩하세요
단, 변수를 이용해서 Model 값을 넣고 사용하세요



요건

테스트 URL	http://localhost:8080/ex01/example04.do
패키지	egovframework.example.controllerexam.web.ex01
자바	BasicController.java
메소드	example04
매개변수	Model model
리턴 자료형	String
Model (키, 값)	키: “str”, 값: “abc” 키: “num”, 값: 10 키: “value”, 값: true
JSP	ex01/example04

BasicController.java

```
/**
 *
 */
```

```
*/
package egovframework.example.controllerexam.web.ex01;

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

/**
 * @author simple-coding
 *
 */
@Controller
public class BasicController {
    @RequestMapping(value = "/ex01/hello.do", method =
RequestMethod.GET)
    public String hello() {
        return "/ex01/hello";
    }
// 따라하기 예제 1
    @GetMapping("/ex01/example01.do")
    public String example01() {
        return "ex01/example01";
    }

// 따라하기 예제 3
    @GetMapping("/ex01/example02.do")
    public String example02(Model model) {
        model.addAttribute("greeting", "안녕 스프링");
        return "ex01/example02";
    }
}
```

```
// 따라하기 예제 5
@GetMapping("/ex01/example03.do")
public String example03(Model model) {
    model.addAttribute("greeting", "안녕 스프링");
    model.addAttribute("greeting2", "안녕 스프링 2");
    return "ex01/example03";
}

// 따라하기 예제 7
@GetMapping("/ex01/example04.do")
public String example04(Model model) {

    String str = "abc";
    int num = 10;
    boolean value = true;

    model.addAttribute("str", str);
    model.addAttribute("num", num);
    model.addAttribute("value", value);
    return "ex01/example04";
}
}
```


example04.jsp

```

<%@ page language="java" contentType="text/html;
charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <p>${str}</p>
    <p>${num}</p>
    <p>${value}</p>
</body>
</html>

```

새로 나온 기술 알아보기

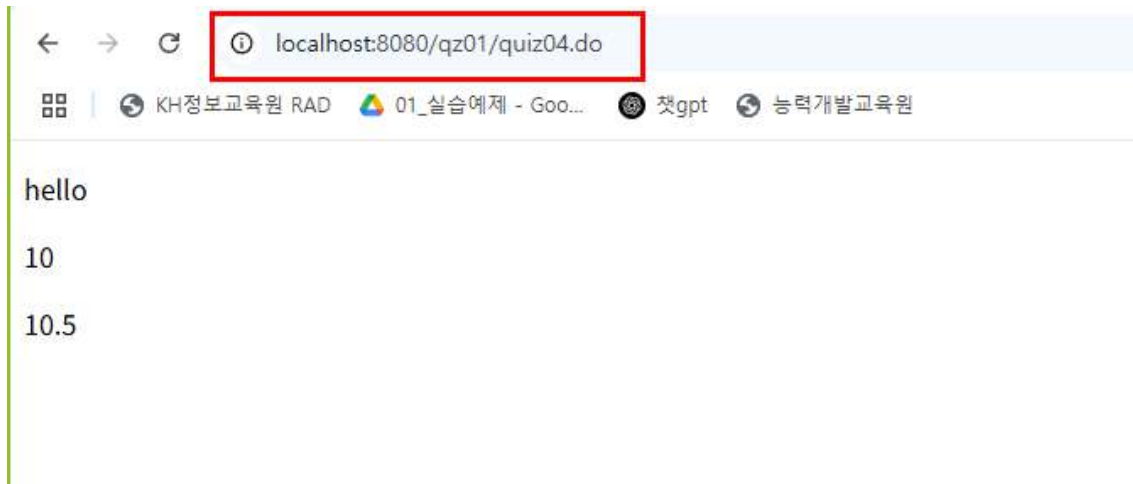
자바 사용법	<pre> @Controller public class 클래스{ @GetMapping("url 경로") public String 메소드(Model model) { String 변수 = "값"; int 변수 2 = 값; model.addAttribute("키", 변수); model.addAttribute("키 2", 변수 2); ... return "폴더/jsp 명"; } } </pre>
JSP 사용법	<pre> \${키} \${키 2} </pre>

초보자를 위한 SQL

⑧ 퀴즈) 아래 그림처럼 컨트롤러와 JSP 를 코딩하세요

단, @GetMapping, Model 은 3 개 사용하고 키는 자유롭게 코딩하세요

단, 변수를 이용해서 Model 값을 넣고 사용하세요



요건

테스트 URL	http://localhost:8080/qz01/quiz04.do
패키지	egovframework.example.controllerexam.web.qz01
자바	Quiz01Controller.java
메소드	quiz04
매개변수	Model model
리턴 자료형	String
Model (키, 값)	키: "str", 값: "hello" 키: "num", 값: 10 키: "value", 값: 10.5
JSP	qz01/quiz04

해답 : <https://github.com/simple-coding-forbob/SimpleSpring>

SECTION
02

@RequestParam 배우기

01 컨트롤러에 사용하는 어노테이션을 배워봅시다.

기본 어노테이션을 앞 예제에서 알아보았습니다.

컨트롤러에 사용하는

@RequestParam **어노테이션**을 알아보고 어떤 기능을 하는지 알아보시다.

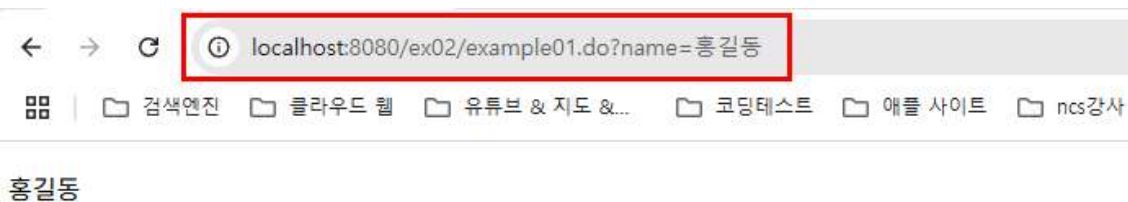
또한, 사용법은 어떻게 되는지 알아보시다.

따라하기 예제를 확인해 보시죠

02 따라 하기 예제로 알아보시다.

● @RequestParam 사용하기

- ① 예제) 아래 요건에 따라 controller 함수 작성하고 JSP 화면에 결과 출력하세요
단, 쿼리스트링(query string) 방식을 이용하세요



초보자를 위한 SQL

요건

테스트 URL	http://localhost:8080/ex02/example01.do?name=홍길 동
패키지	egovframework.example.controllerexam.web.ex02
자바	ParamController.java
메소드	example01
매개변수	Model model @RequestParam(defaultValue = "") String name
리턴 자료형	String
Model (키, 값)	키: "name", 값: name
JSP	ex02/example01

ParamController.java

```
package egovframework.example.controllerexam.web.ex02;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.RequestParam;
@Controller
public class ParamController{

// 따라하기 예제 1
    @GetMapping("/ex02/example01.do")
    public String example01(Model model,
        @RequestParam(defaultValue = "") String name) {
        model.addAttribute("name", name);
        return "ex02/example01";
    }
}
```

```
}
}
```

example01.jsp

```
<%@ page language="java" contentType="text/html;
charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <p>${name}</p>
</body>
</html>
```

새로 나온 기술 알아보기

@RequestParam	<p>쿼리스트링 방식으로 전송된 값을 컨트롤러 메소드의 매개변수로 전달받습니다.</p> <p>단, 서로 변수 이름이 같아야 합니다.</p> <ul style="list-style-type: none"> ▪ http://도메인:포트번호/경로?name=값 ▪ @RequestParam(defaultValue = "") String name ▪ 위 name 이 같아야 값이 메소드로 전달됩니다. <p>() 통해 추가적인 옵션 기능을 사용/생략 할 수 있습니다.</p>
쿼리 스트링	<p>웹 브라우저 주소 창에서 ?매개변수=값 형태로 값을 전달할 수 있는 기능입니다.</p> <p>주로 현재 페이지에서 다른 페이지로 값을 전달할 때 많이 사용됩니다.</p>

	예) url?name=홍길동
웹브라우저 사용법	http://localhost:8080/경로/이름?변수=값
사용법	<pre> @Controller public class 클래스{ @GetMapping("url 경로") public String 메소드(Model model, @RequestParam(defaultValue = "") String 변수) { model.addAttribute("키", 변수); return "jsp 명"; } } </pre>

- ② 퀴즈) 아래 요건에 따라 controller 함수 작성하고 JSP 화면에 결과 출력하세요
단, 쿼리스트링(query string) 방식을 이용하세요

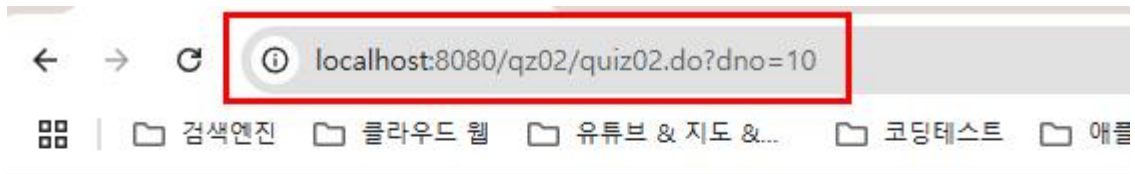


요건

테스트 URL	http://localhost:8080/qz02/quiz01.do?dname=Sales
패키지	egovframework.example.controllerexam.web.qz02
자바	Quiz02Controller.java
메소드	quiz01
매개변수	Model model @RequestParam(defaultValue = "") String dname
리턴 자료형	String
Model (키, 값)	키: "dname", 값: dname
JSP	qz02/quiz01

해답 : <https://github.com/simple-coding-forbob/SimpleSpring>

- ③ 퀴즈) 아래 요건에 따라 controller 함수 작성하고 JSP 화면에 결과 출력하세요
단, 쿼리스트링(query string) 방식을 이용하세요
매개변수 값은 정수 자료형으로 코딩하세요



10

요건

테스트 URL	http://localhost:8080/qz02/quiz02.do?dno=10
패키지	egovframework.example.controllerexam.web.qz02
자바	Quiz02Controller.java
메소드	quiz02
매개변수	Model model @RequestParam(defaultValue = "") int dno
리턴 자료형	String
Model (키, 값)	키: "dno", 값: dno
JSP	qz02/quiz02

해답 : <https://github.com/simple-coding-forbob/SimpleSpring>

SECTION
03

@RequestParam 배우기 2

01 컨트롤러에 사용하는 어노테이션을 배워봅시다.

기본 어노테이션을 앞 예제에서 알아보았습니다.

컨트롤러에 사용하는

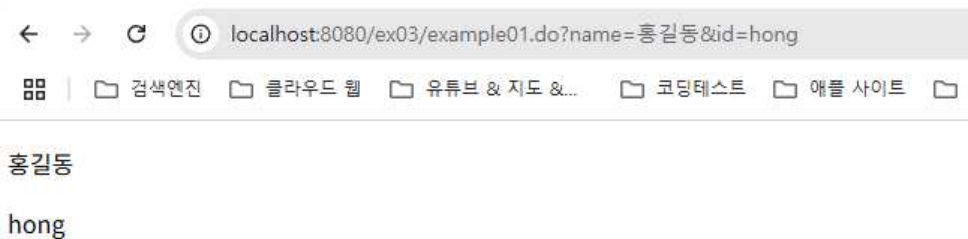
@RequestParam **어노테이션**을 사용해서 여러 가지로 응용해 봅시다.

따라하기 예제를 확인해 보시죠

02 따라 하기 예제로 알아봅시다.

● @RequestParam 여러 개 사용하기

- ① 예제) 아래 요건에 따라 controller 함수 작성하고 JSP 화면에 결과 출력하세요
단, 쿼리스트링(query string) 방식을 이용하세요



초보자를 위한 SQL

요건

테스트 URL	http://localhost:8080/ex03/example01.do?name=홍길 동&id=hong
패키지	egovframework.example.controllerexam.web.ex03
자바	MultiParamController.java
메소드	example01
매개변수	Model model @RequestParam(defaultValue = "") String name, @RequestParam(defaultValue = "") String id
리턴 자료형	String
Model (키, 값)	키: "name", 값: name
JSP	ex03/example01

MultiParamController.java

```
package egovframework.example.controllerexam.web.ex03;
import java.util.ArrayList;
import java.util.List;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.RequestParam;
@Controller
public class MultiParamController{
// 따라하기 예제 1
    @GetMapping("/ex03/example01.do")
    public String example01(Model model,
@RequestParam(defaultValue = "") String name,
@RequestParam(defaultValue = "") String id) {
```

```

        model.addAttribute("name", name);
        model.addAttribute("id", id);
        return "ex03/example01";
    }
}

```

example01.jsp

```

<%@ page language="java" contentType="text/html;
charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <p>${name}</p>
    <p>${id}</p>
</body>
</html>

```

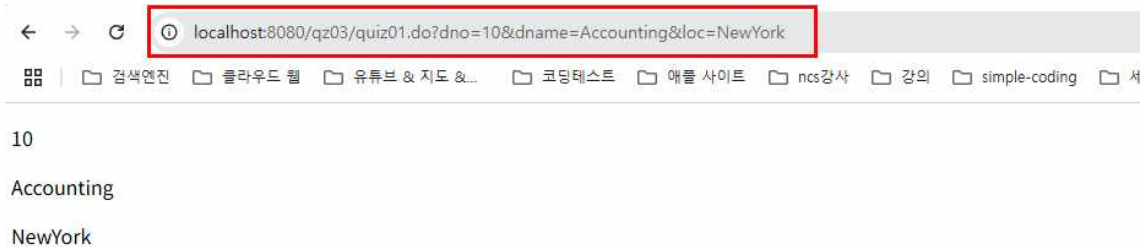
새로 나온 기술 알아보기

웹브라우저 사용법	http://localhost:8080/경로/이름?변수 1=값&변수 2=값 2...
사용법	<pre> @Controller public class 클래스{ @GetMapping("url 경로") public String 메소드(Model model, @RequestParam(defaultValue = "") String 변수 1), @RequestParam(defaultValue = "") String 변수 2 { </pre>

초보자를 위한 SQL

```
model.addAttribute("키 1", 변수 1);  
model.addAttribute("키 2", 변수 2);  
return "jsp 명";  
  
}  
}
```

- ② 퀴즈) 아래 요건에 따라 controller 함수 작성하고 JSP 화면에 결과 출력하세요
단, 쿼리스트링(query string) 방식을 이용하세요



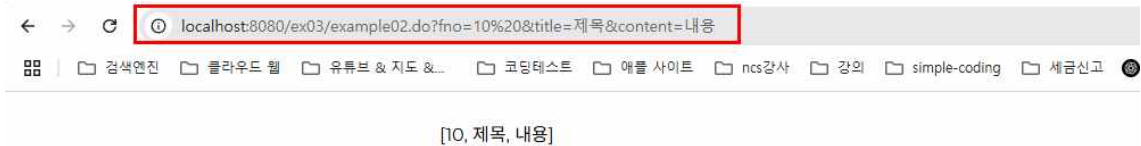
요건

테스트 URL	http://localhost:8080/qz03/quiz01.do?dno=10&dname=Accounting&loc=NewYork
패키지	egovframework.example.controllerexam.web.qz03
자바	Quiz03Controller.java
메소드	quiz01
매개변수	Model model @RequestParam(defaultValue = "") String dno @RequestParam(defaultValue = "") String dname @RequestParam(defaultValue = "") String loc
리턴 자료형	String
Model (키, 값)	키: dno 값: dno 키: dname 값: dname 키: loc 값: loc
JSP	qz03/quiz01

해답 : <https://github.com/simple-coding-forbob/SimpleSpring>

초보자를 위한 SQL

- ③ 예제) 아래 요건에 따라 controller 함수 작성하고 JSP 화면에 결과 출력하세요
단, 쿼리스트링(query string) 방식을 이용하세요
단, css 를 이용해서 중앙 정렬하고 margin-top 을 적당히 적용하세요



요건

테스트 URL	http://localhost:8080/ex03/example02.do?fno=10 &title=제목&content=내용
패키지	egovframework.example.controllerexam.web.ex03
자바	MultiParamController.java
메소드	example02
매개변수	Model model @RequestParam(defaultValue = "") String fno, @RequestParam(defaultValue = "") String title @RequestParam(defaultValue = "") String content
리턴 자료형	String
Model (키, 값)	키: list, 값: 배열변수
JSP	ex03/example02
CSS	style.css main.css

MultiParamController.java

```
package egovframework.example.controllerexam.web.ex03;  
import java.util.ArrayList;  
import java.util.List;  
import org.springframework.stereotype.Controller;
```

```

import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.RequestParam;
@Controller
public class MultiParamController{
// 따라하기 예제 1
    @GetMapping("/ex03/example01.do")
    public String example01(Model model,
@RequestParam(defaultValue = "") String name,
@RequestParam(defaultValue = "") String id) {
        model.addAttribute("name", name);
        model.addAttribute("id", id);
        return "ex03/example01";
    }
// 따라하기 예제 3
    @GetMapping("/ex03/example02.do")
    public String example02(Model model,
@RequestParam(defaultValue = "") String fno,
@RequestParam(defaultValue = "") String title,
@RequestParam(defaultValue = "") String content) {
        List<String> list =new ArrayList<String>();
        list.add(fno);
        list.add(title);
        list.add(content);
        model.addAttribute("list", list);
        return "ex03/example02";
    }
}

```

main.css

```

/* 추가할 디자인 요소 */

```

style.css

```
/* 공통 css: 수정하지 마시오 */
@import
url('https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100..900;1,100..900&family=Noto+Sans+KR:wght@100..900&display=swap');
:root{
    --black: black;
    --gray: gray;
}
*{margin: 0;padding: 0;box-sizing: border-box;}
body{font-family: "Montserrat","Noto Sans KR",
sans-serif;}
li{list-style-type: none;}
a{text-decoration: none;color: var(--black);}
/* 공통 여백 정하기 */
/* 마진 위 5개 */
.mt0{margin-top:0.25rem !important;}
.mt1{margin-top:0.5rem !important;}
.mt2{margin-top:0.75rem !important;}
.mt3{margin-top:1rem !important;}
.mt4{margin-top:1.25rem !important;}
.mt5{margin-top:1.5rem !important;}
/* 마진 오른쪽 5개 */
.mr0{margin-right:0.25rem !important;}
.mr1{margin-right:0.5rem !important;}
.mr2{margin-right:0.75rem !important;}
.mr3{margin-right:1rem !important;}
.mr4{margin-right:1.25rem !important;}
.mr5{margin-right:1.5rem !important;}
/* 마진 하 5개 */
.mb0{margin-bottom:0.25rem !important;}
.mb1{margin-bottom:0.5rem !important;}
.mb2{margin-bottom:0.75rem !important;}
```



```
.mb3{margin-bottom:1rem !important;}
.mb4{margin-bottom:1.25rem !important;}
.mb5{margin-bottom:1.5rem !important;}
/* 마진 좌 5개 */
.ml0{margin-left:0.25rem !important;}
.ml1{margin-left:0.5rem !important;}
.ml2{margin-left:0.75rem !important;}
.ml3{margin-left:1rem !important;}
.ml4{margin-left:1.25rem !important;}
.ml5{margin-left:1.5rem !important;}
/* 패딩 위 5개 */
.pt0{padding-top:0.25rem !important;}
.pt1{padding-top:0.5rem !important;}
.pt2{padding-top:0.75rem !important;}
.pt3{padding-top:1rem !important;}
.pt4{padding-top:1.25rem !important;}
.pt5{padding-top:1.5rem !important;}
/* 패딩 오른쪽 5개 */
.pr0{padding-right:0.25rem !important;}
.pr1{padding-right:0.5rem !important;}
.pr2{padding-right:0.75rem !important;}
.pr3{padding-right:1rem !important;}
.pr4{padding-right:1.25rem !important;}
.pr5{padding-right:1.5rem !important;}
/* 패딩 아래 5개 */
.pb0{padding-bottom:0.25rem !important;}
.pb1{padding-bottom:0.5rem !important;}
.pb2{padding-bottom:0.75rem !important;}
.pb3{padding-bottom:1rem !important;}
.pb4{padding-bottom:1.25rem !important;}
.pb5{padding-bottom:1.5rem !important;}
/* 패딩 왼쪽 5개 */
.pl0{padding-left:0.25rem !important;}
.pl1{padding-left:0.5rem !important;}
```

```
.pl2{padding-left:0.75rem !important;}
.pl3{padding-left:1rem !important;}
.pl4{padding-left:1.25rem !important;}
.pl5{padding-left:1.5rem !important;}
/* 글자 중앙정렬 */
.tcenter{text-align: center !important;}
.page{width: 99vw; margin: 0 auto}
/* TODO: 태블릿 */
@media screen and(min-width: 768px){
    .page{width: 90vw;}
}
/* TODO: pc */
@media screen and(min-width: 1024px){
    .page{width: 60vw;}
}
```

example02.jsp

```

<%@ page language="java" contentType="text/html;
charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<link rel="stylesheet"href="/css/style.css">
<link rel="stylesheet"href="/css/main.css">
</head>
<body>
    <div class="page mt5">
        <p>${list}</p>
    </div>
</body>
</html>

```

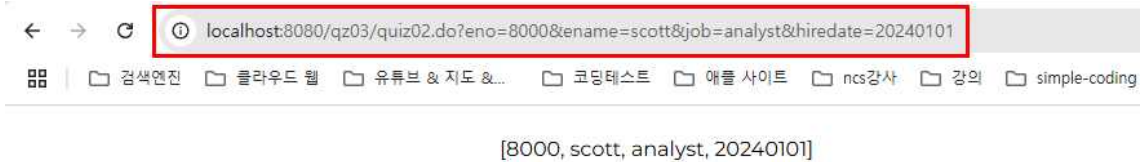
새로 나온 기술 알아보기

웹브라우저 사용법	http://localhost:8080/경로/이름?변수=값, ..
사용법	<pre> @Controller public class 클래스{ @GetMapping("url 경로") public String 메소드(Model model, @RequestParam(defaultValue = "") String 변수,...) { List<String> 배열변수 =new ArrayList<String>(); 배열변수.add(변수); ... model.addAttribute("키 1", 배열변수); return "jsp명"; } } </pre>

초보자를 위한 SQL

	}
	}

- ④ 퀴즈) 아래 요건에 따라 controller 함수 작성하고 JSP 화면에 결과 출력하세요
 단, 쿼리스트링(query string) 방식을 이용하세요
 단, css 를 이용해서 중앙 정렬하고 margin-top 을 적당히 적용하세요



요건

테스트 URL	http://localhost:8080/qz03/quiz02.do?eno=8000&ename=scott&job=analyst&hiredate=20240101
패키지	egovframework.example.controllerexam.web.qz03
자바	Quiz03Controller.java
메소드	quiz02
매개변수	Model model @RequestParam(defaultValue = "") String eno @RequestParam(defaultValue = "") String ename @RequestParam(defaultValue = "") String job @RequestParam(defaultValue = "") String hiredate
리턴 자료형	String
Model (키, 값)	키: list, 값: 배열변수
JSP	qz03/quiz02
CSS	style.css main.css

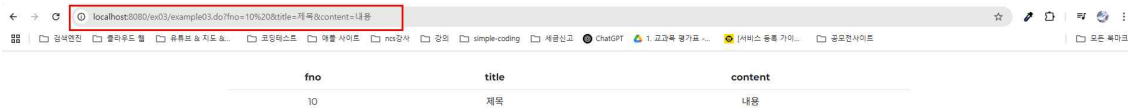
해답 : <https://github.com/simple-coding-forbob/SimpleSpring>

초보자를 위한 SQL

⑤ 예제) 이전 예제와 비슷하게 컨트롤러 메소드를 작성하고 배열값을 JSP 반복문을 이용해서 화면에 표시하세요

단, 부트스트랩 테이블을 이용하세요

단, css 를 이용해서 중앙 정렬하고 margin-top 을 적당히 적용하세요



fno	title	content
10	제목	내용

요건

테스트 URL	http://localhost:8080/ex03/example03.do?fno=10 &title=제목&content=내용
패키지	egovframework.example.controllerexam.web.ex03
자바	MultiParamController.java
메소드	example03
매개변수	Model model @RequestParam(defaultValue = "") String fno, @RequestParam(defaultValue = "") String title @RequestParam(defaultValue = "") String content
리턴 자료형	String
Model (키, 값)	키: list, 값: 배열변수
JSP	ex03/example03
CSS	style.css main.css

MultiParamController.java

```
package egovframework.example.controllerexam.web.ex03;
import java.util.ArrayList;
import java.util.List;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.RequestParam;
@Controller
public class MultiParamController{
// 따라하기 예제 1
    @GetMapping("/ex03/example01.do")
    public String example01(Model model,
@RequestParam(defaultValue = "") String name,
@RequestParam(defaultValue = "") String id) {
        model.addAttribute("name", name);
        model.addAttribute("id", id);
        return "ex03/example01";
    }
// 따라하기 예제 3
    @GetMapping("/ex03/example02.do")
    public String example02(Model model,
@RequestParam(defaultValue = "") String fno,
@RequestParam(defaultValue = "") String title,
@RequestParam(defaultValue = "") String content) {
        List<String> list =new ArrayList<String>();
        list.add(fno);
        list.add(title);
        list.add(content);
        model.addAttribute("list", list);
        return "ex03/example02";
    }
// 따라하기 예제 5
    @GetMapping("/ex03/example03.do")
    public String example03(Model model,
@RequestParam(defaultValue = "") String fno,
@RequestParam(defaultValue = "") String title,
@RequestParam(defaultValue = "") String content) {
        List<String> list =new ArrayList<String>();
```

```
        list.add(fno);
        list.add(title);
        list.add(content);
        model.addAttribute("list", list);
        return "ex03/example03";
    }
}
```

example03.jsp

```
<%@ page language="java" contentType="text/html;
charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core"
prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<!-- bootstrap 5.2 css cdn -->
<link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/cs
s/bootstrap.min.css"
    rel="stylesheet"

integrity="sha384-rbsA2VBKQhggwzxH7pPCaAq046MgnOM80zW1RWuH
61DGLwZJEdK2Kadq2F9CUG65"
    crossorigin="anonymous">
<link rel="stylesheet"href="/css/style.css">
<link rel="stylesheet"href="/css/main.css">
</head>
<body>
    <div class="page mt5">
```



```

<table class="table tcenter">
  <thead>
    <tr>
      <th scope="col">fno</th>
      <th scope="col">title</th>
      <th scope="col">content</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <c:forEach var="data" items="${list}">
        <td>${data}</td>
      </c:forEach>
    </tr>
  </tbody>
</table>
</div>
</body>
</html>

```

새로 나온 기술 알아보기

JSP 반복문/조건문 지시자

```
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c"%>
```

인터넷으로 반복문/조건문이 포함된 라이브러리를
다운로드 받습니다.
JSP 파일 위에 위 코딩을 포함해야 반복문/조건문이
사용 가능합니다.

CSS link 태그나 JAVA import 와 비슷한
기능입니다.

	<p>웹브라우저에 태그를 반복해서 보여줍니다.</p> <p>주로 테이블 행태그 또는 목록 태그를 반복합니다.</p>
웹브라우저 사용법	<pre>http://localhost:8080/경로/이름?변수=값,..</pre>
사용법	<pre>@Controller public class 클래스{ @GetMapping("url 경로") public String 메소드(Model model, @RequestParam(defaultValue = "") String 변수,...) { List<String> 배열변수 =new ArrayList<String>(); 배열변수.add(변수); ... model.addAttribute("키 1", 배열변수); return "jsp 명"; } }</pre>
JSP 반복문 사용법	<pre><%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%> <html> ... <c:forEach var="변수" items="\${배열}"> <td>\${변수}</td> </c:forEach> </html></pre>

⑥ 퀴즈) 이전 퀴즈와 비슷하게 컨트롤러 메소드를 작성하고 배열값을 JSP 반복문을 이용해서 화면에 표시하세요

단, 부트스트랩 테이블을 이용하세요

eno	ename	job	hiredate
8000	scott	analyst	20240101

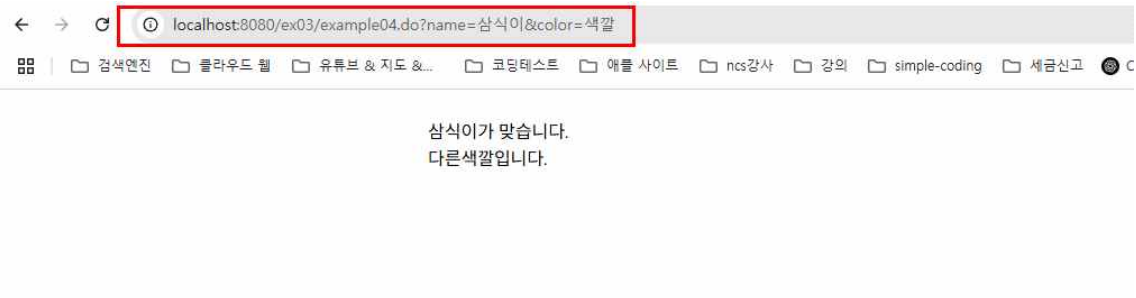
요건

테스트 URL	http://localhost:8080/qz03/quiz03.do?eno=8000&ename=scott&job=analyst&hiredate=20240101
패키지	egovframework.example.controllerexam.web.qz03
자바	Quiz03Controller.java
메소드	quiz03
매개변수	Model model @RequestParam(defaultValue = "") String eno @RequestParam(defaultValue = "") String ename @RequestParam(defaultValue = "") String job @RequestParam(defaultValue = "") String hiredate
리턴 자료형	String
Model (키, 값)	키: list, 값: 배열변수
JSP	qz03/quiz03
CSS	style.css main.css

해답 : <https://github.com/simple-coding-forbob/SimpleSpring>

초보자를 위한 SQL

⑦ 예제) 아래 요건에 따라 controller 함수 작성하고 JSP 조건문을 이용해 화면에 표시하세요



요건

테스트 URL	http://localhost:8080/ex03/example04.do?name=삼식이&color=색깔
패키지	egovframework.example.controllerexam.web.ex03
자바	MultiParamController.java
메소드	example04
매개변수	Model model @RequestParam(defaultValue = "") String fno, @RequestParam(defaultValue = "") String title @RequestParam(defaultValue = "") String content
리턴 자료형	String
Model (키, 값)	키: list, 값: 배열변수
JSP	ex03/example04
CSS	style.css main.css

MultiParamController.java

```
package egovframework.example.controllerexam.web.ex03;
import java.util.ArrayList;
import java.util.List;
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.RequestParam;
@Controller
public class MultiParamController{
// 따라하기 예제 1
    @GetMapping("/ex03/example01.do")
    public String example01(Model model,
@RequestParam(defaultValue = "") String name,
@RequestParam(defaultValue = "") String id) {
        model.addAttribute("name", name);
        model.addAttribute("id", id);
        return "ex03/example01";
    }
// 따라하기 예제 3
    @GetMapping("/ex03/example02.do")
    public String example02(Model model,
@RequestParam(defaultValue = "") String fno,
@RequestParam(defaultValue = "") String title,
@RequestParam(defaultValue = "") String content) {
        List<String> list =new ArrayList<String>();
        list.add(fno);
        list.add(title);
        list.add(content);
        model.addAttribute("list", list);
        return "ex03/example02";
    }
// 따라하기 예제 5
    @GetMapping("/ex03/example03.do")
    public String example03(Model model,
@RequestParam(defaultValue = "") String fno,
@RequestParam(defaultValue = "") String title,
@RequestParam(defaultValue = "") String content) {
```

```
        List<String> list =new ArrayList<String>();
        list.add(fno);
        list.add(title);
        list.add(content);
        model.addAttribute("list", list);
        return "ex03/example03";
    }
    // 따라하기 예제 7
    @GetMapping("/ex03/example04.do")
    public String example04(Model model,
    @RequestParam(defaultValue = "") String name,
    @RequestParam(defaultValue = "") String color) {
        model.addAttribute("name", name);
        model.addAttribute("color", color);
        return "ex03/example04";
    }
}
```

example04.jsp

```
<%@ page language="java" contentType="text/html;
charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core"
prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<link rel="stylesheet"href="/css/style.css">
<link rel="stylesheet"href="/css/main.css">
</head>
<body>
    <div class="page mt5">
```

```

    <c:if test="${name == '삼식이'}">
        <p>삼식이가 맞습니다.</p>
    </c:if>
    <c:choose>
        <c:when test="${color == 'brown'}">
            <p>갈색입니다.</p>
        </c:when>
        <c:when test="${color == 'yellow'}">
            <p>노란색입니다.</p>
        </c:when>
        <c:otherwise>
            <p>다른색깔입니다.</p>
        </c:otherwise>
    </c:choose>
</div>
</body>
</html>

```

새로 나온 기술 알아보기

JSP 조건문

```

<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c"%>

```

인터넷으로 반복문/조건문이 포함된 라이브러리를
다운로드 받습니다.

JSP 파일 위에 위 코딩을 포함해야 반복문/조건문이
사용 가능합니다.

CSS link 태그나 JAVA import 와 비슷한
기능입니다.

조건문은 2 가지가 있습니다.

- <c:if test="\${조건문}">

	<p>조건문이 참이면 자식 태그를 보여줍니다.</p> <ul style="list-style-type: none"> ▪ <code><c:choose> ~</code> <code><c:when test="\${조건문}">글자 1</code> <code><c:when test="\${조건문 2}">글자 2</code> <code><c:otherwise>글자 3</code> 조건문이 참이면 글자 1 을 보여줍니다. 조건문 2 가 참이면 글자 2 를 보여줍니다. 모두 아니면 글자 3 을 보여줍니다.
웹브라우저 사용법	<pre>http://localhost:8080/경로/이름?변수=값,..</pre>
사용법	<pre>@Controller public class 클래스{ @GetMapping("url 경로") public String 메소드(Model model, @RequestParam(defaultValue = "")) String 변수,...) { List<String> 배열변수 =new ArrayList<String>(); 배열변수.add(변수); ... model.addAttribute("키 1", 배열변수); return "jsp 명"; } }</pre>
JSP 조건문 사용법	<pre><!-- 변수=='값' 참이면 p 태그가 화면에 보임 --> <c:if test="\${변수 == '값'}"> <p>글자</p> </c:if> <!-- 변수=='값' 참이면 글자 1 이 보이고 변수 2=='값 2' 참이면 글자 2 가 보이고 모두 거짓이면 글자 3 이 보임 --> <c:choose> <c:when test="\${변수 == '값 1'}"> <p>글자 1</p> </c:when></pre>


```
<c:when test="${변수 2 == '값 2'}">
  <p>글자 2</p>
</c:when>
<c:otherwise>
  <p>글자 3</p>
</c:otherwise>
</c:choose>
```

SECTION
04

@ModelAttribute

01

컨트롤러에 사용하는 어노테이션을 배워봅시다.

기본 어노테이션을 앞 예제에서 알아보았습니다.

컨트롤러에 사용하는

@ModelAttribute **어노테이션**을 사용해서 여러 가지로 응용해 봅시다.

따라하기 예제를 확인해 보시죠

02

따라 하기 예제로 알아봅시다.

● @RequestParam 여러 개 사용하기

① 예제) 아래 요건에 따라 controller 함수 작성하고 객체를 전달해서 JSP 화면에 결과 출력 하세요

단, MemberVO 는 미리 소소가 제공되었다고 가정합니다.



MemberVO(id=hong, name=홍길동)

요건

테스트 URL

http://localhost:8080/ex04/example01.do?id=hong&na

	me=홍길동
패키지	egovframework.example.controllerexam.web.ex04
자바	ObjectController
메소드	example01
매개변수	@ModelAttribute MemberVO memberVO
리턴 자료형	String
JSP	ex04/example01
CSS	style.css main.css

ObjectController.java

```

package egovframework.example.controllerexam.web.ex04;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.ModelAttribute;
import
egovframework.example.controllerexam.service.DeptVO;
import
egovframework.example.controllerexam.service.MemberVO;
@Controller
public class ObjectController{
// 따라하기 예제 1
    @GetMapping("/ex04/example01.do")
    public String example01(
        @ModelAttribute MemberVO memberVO
        return "ex04/example01";
    }
}

```

example01.jsp

```
<%@ page language="java" contentType="text/html;
charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core"
prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<link rel="stylesheet"href="/css/style.css">
<link rel="stylesheet"href="/css/main.css">
</head>
<body>
    <div class="page mt5">
        <p>${memberV0}</p>
    </div>
</body>
</html>
```

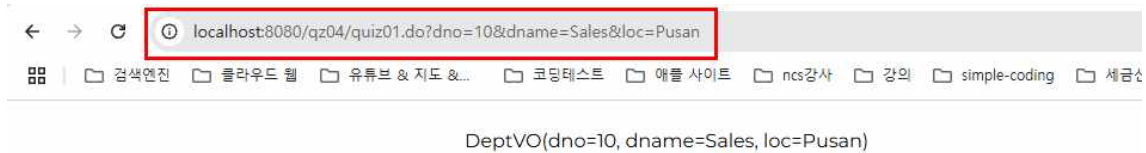
새로 나온 기술 알아보기

<p>@ModelAttribute</p>	<p>쿼리스트링 방식으로 전송된 값을 컨트롤로 함수의 객체로 전달받습니다.</p> <p>단, 웹 매개변수와 객체의 필드명이 같아야 합니다.</p> <p>예를 들면, url/ex04/example01.do?id=값 1&name=값 2 @ModelAttribute MemberV0 memberV0 단, 아래의 이름이 서로 같아야 합니다. 웹 변수의 id == memberV0의 id 필드명 웹 변수의 name == memberV0의 name 필드명</p>
<p>웹브라우저 사용법</p>	<p>http://localhost:8080/경로/이름?변수=값, ..</p>

사용법	<pre> @Controller public class 컨트롤러{ @GetMapping("url 경로") public String 메소드(@ModelAttribute MemberVO VO 명) { return "jsp 명"; } } </pre>
JSP 조건문 사용법	<pre> <p>\${VO 명}</p> </pre>

초보자를 위한 SQL

- ② 퀴즈) 아래 요건에 따라 controller 함수 작성하고 JSP 화면에 결과 출력하세요
단, 쿼리스트링(query string) 방식을 이용하세요



요건

테스트 URL	<code>http://localhost:8080/ex04/quiz01.do?dno=10&dname=Sales&loc=Pusan</code>
패키지	<code>egovframework.example.controllerexam.web.qz04</code>
자바	<code>Quiz04Controller.java</code>
메소드	<code>quiz01</code>
매개변수	<code>@ModelAttribute DeptVO deptVO</code>
리턴 자료형	<code>String</code>
Model (키, 값)	없음
JSP	<code>qz04/quiz01</code>

해답 : <https://github.com/simple-coding-forbob/SimpleSpring>

SECTION
05

AJAX 사용하기

01 AJAX 를 배웁시다.

컨트롤러의 기본 어노테이션을 앞 예제에서 알아보았습니다.

웹페이지는 일부 글자나 태그가 바뀌더라도 **전체 새로고침**이 적용되어 실행됩니다.
만약 이미지, 동영상 등 크기가 큰 파일 등이 있을 경우 매번 새로고침이 일어나면 굉장히
느려집니다.

이것을 개선 하기 위해 나타난 것이 AJAX 입니다.

부분 새로고침이 가능한 기술입니다.

또한 **비동기로 동작**합니다.

비동기 코딩에 대해 알아보시다.

코딩은 주로 라인별로 실행됩니다.

위의 라인이 실행 결과를 기다리지 않고 아래 라인이 먼저 실행되는 방식입니다.

당연히 위의 실행 결과를 기다리지 않고 실행되니 속도가 증가합니다.

단, **위의 라인 실행 결과에 따라 아래 라인이 영향을 받는 코딩일 경우 결과에 오류**
가 있을 수 있습니다.

이것을 **비동기 코딩의 동기화 문제**라고 합니다.

AJAX(Asynchronous JavaScript and XML) 는 부분 새로고침과 비동기 코딩을 지원
합니다.

또한 비동기 코딩의 동기화 문제도 해결했습니다.

요즘 새롭게 등장한 리액트, 뷰는 AJAX 를 기본으로 코딩됩니다.

또한 AJAX 를 업그레이드한 AXIOS 라이브러리를 사용합니다.

초보자를 위한 SQL

JSP 기반 웹페이지는 AJAX 많이 사용합니다.

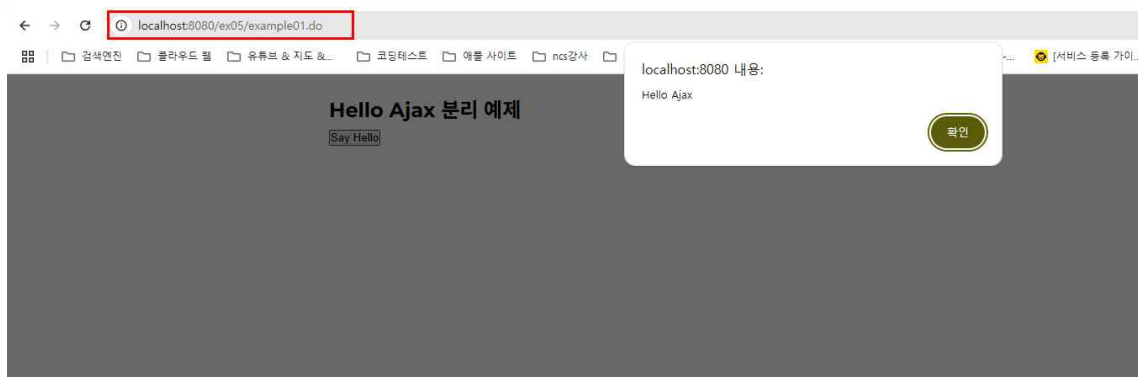
따라하기 예제를 확인해 보시죠

02 따라 하기 예제로 알아봅시다.

● AJAX 사용하기

① 예제) 아래 요건에 따라 controller 함수 작성하고 객체를 전달해서 JSP 화면에 결과 출력 하세요

단, AJAX 를 사용하세요



요건

테스트 URL	http://localhost:8080/ex05/example01.do
패키지	egovframework.example.controllerexam.web.ex05
1번째 컨트롤러	
자바	ButtonController
메소드	example01
매개변수	없음
리턴 자료형	String
JSP	ex05/example01
2번째 컨트롤러	
자바	SelectController
메소드	example01

매개변수	없음
리턴 자료형	String
CSS	style.css main.css

ButtonController.java

```
/**
 *
 */
package egovframework.example.controllerexam.web.ex05;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
/**
 * @author simple-coding
 *
 */
@Controller
public class ButtonController{
    @GetMapping("/ex05/example01.do")
    public String example01() {
        return "ex05/example01";
    }
}
```

SelectController.java

```
/**
 *
 */
package egovframework.example.controllerexam.web.ex05;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.ResponseBody;
import
org.springframework.web.bind.annotation.RestController;
/**
 * @author simple-coding
 *
 */
@RestController
public class SelectController{
    @GetMapping(value="/api/select.do", produces
    ="application/text; charset=UTF-8;")

    public String example01() {
        return "Hello Ajax";
    }
}
```

example01.java

```
<%@ page language="java" contentType="text/html;
charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<link rel="stylesheet"href="/css/style.css">
<link rel="stylesheet"href="/css/main.css">
</head>
<body>
    <div class="page mt5">
        <h2>Hello Ajax 분리 예제</h2>
        <button id="btn">Say Hello</button>
        <p id="result"></p>
    </div>
<script
src="https://code.jquery.com/jquery-3.1.0.js"></script>
<script>
    $(function() {
        $("#btn").click(function() {
            $.ajax({
                url:"/api/select.do",
                type:"GET",
                success:function(data) {
                    console.log(data);
                    alert(data);
                    $("#result").text(data);
                },
                error:function() {
                    console.log("에러 발생");
                }
            });
        });
    });
}
```

초보자를 위한 SQL

```
});  
});  
});  
</script>  
</body>  
</html>
```

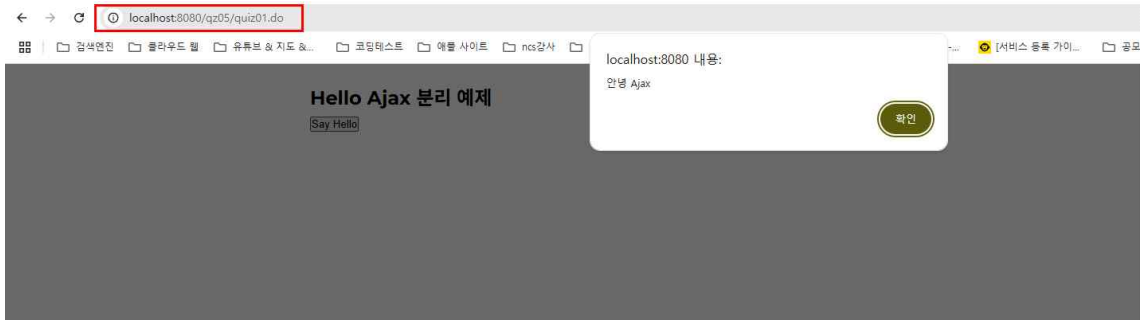
새로 나온 기술 알아보기

@RestController	AJAX 또는 AXIOS 에서 사용하는 컨트롤러입니다. 결과는 JSON(JAVASCRIPT OBJECT NOTATION) 객체로 전달됩니다.
JSON	자바스크립트 객체들을 배열에 담은 형태입니다. 요즘 코딩은 주로 이 객체들을 사용합니다. 프론트와 백엔드는 JSON 객체배열을 주고 받습니다.
AJAX	부분 새로운 기술 입니다. 비동기 코딩을 지원 합니다. 속도가 빠릅니다. 백엔드에서 JSON 데이터를 받습니다. 데이터를 전달받는 2 가지 경우가 있습니다. 받을 경우(성공)와 못받을 경우(실패)에 따라 다른 코딩을 할 수 있습니다. <ul style="list-style-type: none"> 성공: <code>success:function(data){실행문}</code> 실행합니다. 실패: <code>error:function(request){실행문}</code> 실행합니다.
웹브라우저 사용법	<code>http://localhost:8080/경로/이름?변수=값,...</code>
사용법	<pre>@RestController public class 컨트롤러명{ @GetMapping("URL 경로") public 자료형 메소드() { return "글자"; } }</pre>
JSP 에서 AJAX 사용법	<pre><script src="https://code.jquery.com/jquery-3.1.0.js"></script> <script> \$(function() {</pre>

```
$("#선택자").click(function() {  
    $.ajax({  
        url:"컨트롤러 URL 경로",  
        type:"GET",  
        success:function(data) {  
            성공했을 때 실행문;  
        },  
        error:function(request) {  
            실패했을 때 실행문;  
        }  
    });  
});  
});
```

② 퀴즈) 아래 요건에 따라 controller 함수 작성하고 객체를 전달해서 JSP 화면에 결과 출력 하세요

단, AJAX 를 사용하세요



요건

테스트 URL	http://localhost:8080/qz05/quiz01.do
패키지	egovframework.example.controllerexam.web.qz05
1번째 컨트롤러	
자바	Quiz05Controller.java
메소드	quiz01
매개변수	없음
리턴 자료형	String
Model (키, 값)	없음
JSP	qz05/quiz01
2번째 컨트롤러	
자바	Quiz05AjaxController
메소드	quiz01
매개변수	없음
리턴 자료형	String
CSS	style.css main.css

해답 : <https://github.com/simple-coding-forbob/SimpleSpring>

SECTION 05 AJAX 사용하기

01 JSP 추가 예제를 배워봅시다.

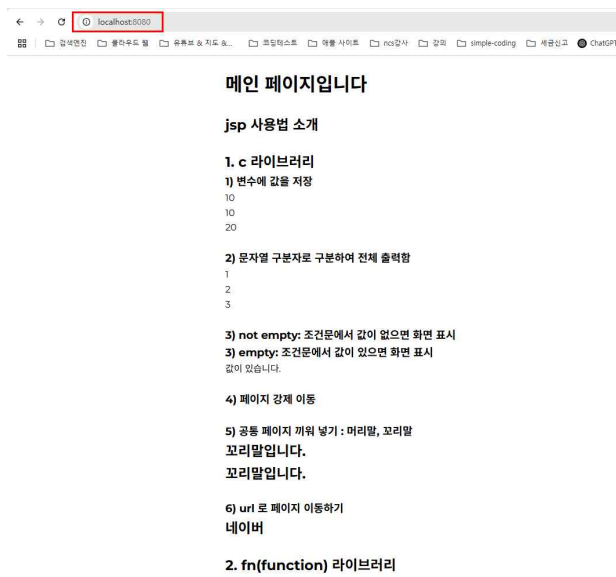
JSP 추가 사용법을 알아보시죠

따라하기 예제를 확인해 보시죠

02 따라 하기 예제로 알아보시다.

● JSP 사용하기

- ① 예제) 아래 요건에 따라 controller 함수 작성하고 객체를 전달해서 JSP 화면에 결과 출력 하세요



요건

테스트 URL	http://localhost:8080
패키지	egovframework.example.common.web
자바	HomeController
메소드	home
매개변수	없음
리턴 자료형	String
JSP	home

HomeController.java

```

/**
 *
 */
package egovframework.example.common.web;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
/**
 * @author simple-coding
 * 메인화면: 첫페이지
 */
@Controller
public class HomeController{
    @GetMapping("/home.do")
    public String home() {
        return "home";
    }
}

```

초보자를 위한 SQL

home.java

```
<%@ page language="java" contentType="text/html;
charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ page import="java.util.Date" %>
<%--TODO: c lib import --%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%--TODO: fn lib import--%>
<%@ taglib prefix="fn"
uri="http://java.sun.com/jsp/jstl/functions"%>
<%--TODO: fmt lib import--%>
<%@ taglib prefix="fmt"
uri="http://java.sun.com/jsp/jstl/fmt" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<link rel="stylesheet"href="/css/style.css">
<link rel="stylesheet"href="/css/main.css">
</head>
<body>
    <div class="page mt5">
        <h1>메인 페이지입니다</h1>
        <br>
        <h2>jsp 사용법 소개</h2>
        <br>
<%--      변수 저장소(사용범위)  --%>
<!--      page: 현재 JSP 페이지 내에서만 유효,  -->
<!--      request: 클라이언트가(화면) 요청을 보내고 서버가(spring
java) 응답할 때까지.  -->
<!--      session: 세션이 유효한 동안 (기본 30 분, 설정에 따라 다름).
예) 로그인  -->
```

```

<!--      application: 모든 사용자와 모든 요청에 대해 전역적으로
공유.  -->
<!--      : page(현재페이지) <request <session
<application(전체프로그램)  --%>
<!--      사용법 : <c:set var="변수명"value="값"/>--%>
<!--      jstl 표현식 --%>
    <h2>1. c 라이브러리</h2>
    <h3>1) 변수에 값을 저장</h3>
    <c:set var="num"value="10"scope="page"/>
    <c:set var="num2"value="10"scope="page"/>
    <c:set var="total"value="${num + num2}"scope="page"/>
<!-- TODO: el 표현식 --%>
    <p>${num}</p>
    <p>${num2}</p>
    <p>${total}</p>
    <br>
<!--      TODO: 2) 문자열 구분자로 구분하여 전체 출력함 --%>
<!--      사용법 : <c:forTokens
items="문자열"delims="구분자"var="변수명">--%>
<!--      실행문      --%>
<!--      </c:forTokens>--%>
    <h3>2) 문자열 구분자로 구분하여 전체 출력함</h3>
    <c:forTokens items="1/2/3"delims="/"var="a">
        <p>${a}</p>
    </c:forTokens>
    <br>
    <h3>3) not empty: 조건문에서 값이 없으면 화면 표시</h3>
    <c:set var="num3"value=""scope="page"/>
    <c:if test="${not empty num3}">
        <p>값이 없습니다.</p>
    </c:if>
    <h3>3) empty: 조건문에서 값이 있으면 화면 표시</h3>
    <c:if test="${empty num3}">
        <p>값이 있습니다.</p>

```

```
</c:if>
<br>

<h3>4) 페이지 강제 이동</h3>
<!--      사용법 : <c:redirect url="이동할_url 주소"/>--%>
<!-- <c:redirect url="/ex01/example01.do"/>--%>
<br>
<h3>5) 공통 페이지 끼워 넣기 : 머리말, 꼬리말</h3>
<!--      TODO: 5) 공통 페이지 끼워 넣기 : 머리말, 꼬리말 --%>
<!--      사용법 : <c:import url="보여줄페이지"/>--%>
<!--      사용법 : <jsp:include
page="보여줄페이지"></jsp:include>--%>
<c:import url="/common/footer.jsp"/>
<jsp:include page="/common/footer.jsp"></jsp:include>
<br>
<!--      TODO: 6) url 로 페이지 이동하기 --%>
<!--      사용법 : <c:url value="url 주소"var="변수명"/>--%>
<!--      사용 : <a href="${변수명}">네이버</a>--%>
<h3>6) url 로 페이지 이동하기</h3>
<c:url value="http://www.naver.com"var="naver"/>
<h2><a href="${naver}">네이버</a></h2>
<br>

<h2>2. fn(function) 라이브러리</h2>
<!--      TODO: 2. fn(function) 라이브러리 --%>
<c:set var="a"value="안녕하세요 홍길동입니다."/>
<c:set var="b"value="홍길동"/>
<!--      1) 사용법 : ${fn:contains(chars, name)} --%>
<!--      chars 문자열에 name 값 있으면 true, 없으면 false 화면
표시 --%>
<h3>1) 사용법 : fn:contains(변수, 값)</h3>
<p>${fn:contains(a, b)}</p>
<br>
<!--      TODO: 2) chars 문자열에 "홍길동" 값이 있는 index 번호
```

```

리턴해서 화면 표시 --%>
    <h3>2) 사용법 : fn:indexOf(변수, 값)</h3>
    <p>${fn:indexOf(a, "홍길동")}</p>
    <br>
<%--    TODO: 3) 전체 길이를 리턴 --%>
    <h3>3) 사용법 : fn:length(변수)</h3>
    <p>${fn:length(a)}</p>
    <br>
<%--    TODO: 4) chars 문자열에서 "홍길동" -> "장길산" 변경해서 화면
표시 --%>
    <h3>4) 사용법 : fn:replace(변수,원본값,바꿀값)</h3>
    <p>replace : ${fn:replace(a, "홍길동", "장길산")}</p>
    <br>
<%--    TODO: 5) chars 에서 " " 로 문자열 자르기해서 화면 표시 :
리턴값 : 배열 --%>
    <h3>5) 사용법 : fn:split(변수,자를글자)</h3>
    <c:set var="b"value='${fn:split(a, " ")}'/>
    <p>${b[0]}</p>
    <p>${b[1]}</p>
    <br>
<%--    TODO: 6) chars 에서 0 인덱스 ~ 6-1 인덱스까지 자르기 --%>
    <h3>6) 사용법 : fn:substring(변수,첫인덱스번호,
끝인덱스번호)</h3>
    <p>${fn:substring(a, 0, 6)}</p>
    <br>
<%--    TODO: 7) 대/소문자 바꾸기 --%>
    <h3>7) 사용법 :
fn:toLowerCase(변수) / fn:toUpperCase(a)</h3>
    <c:set var="a"value='aB'/>
    <p>${fn:toLowerCase(a)}</p>
    <p>${fn:toUpperCase(a)}</p>
    <br>
<%--    jsp 과거 코딩 : 스크립틀릿 -> el, jstl 사용(권고) --%>
    <h2>3. jsp 과거 코딩: 스크립틀릿-> el, jstl 사용(권고)</h2>

```

```
<%
/* 사용법 1: 여기서는 조건문, 반복문, 일반 변수 자바의 모든 것을
코딩가능합니다. */
Date date = new java.util.Date();
<!-- TODO: 변수 화면 표시 : <%=변수명%> --%>
<p><%=date%></p>
<br>

<h2>4. fmt(format) 라이브러리</h2>
<h3>1) 숫자 찍기 : 3 자리마다 쉼표 추가</h3>
<c:set var="a"value="1234567890"/>
<!-- TODO: 1) 숫자 찍기 : 3 자리마다 쉼표 추가 --%>
<p><fmt:formatNumber value="\${a}"/></p>
<br>
<!-- TODO: 2) 통화 찍기 --%>
<h3>2) 통화 찍기</h3>
<p><fmt:formatNumber value="\${a}"type="currency"/></p>
<br>
<!-- TODO: 3) 날짜 포맷 주기--%>
<h3>3) 날짜 포맷 주기</h3>
<c:set var="a"value="<%=new java.util.Date()%>"/>
<p><fmt:formatDate value="\${a}"
                    pattern="yyyy-MM-dd
hh:mm:ss"/></p>

</div>
</body>
</html>
```

새로 나온 기술 알아보기

웹브라우저 사용법	<code>http://localhost:8080/경로/이름?변수=값, ..</code>
사용법	<pre> @RestController public class 컨트롤러명{ @GetMapping("URL 경로") public 자료형 메소드() { return "글자"; } } </pre>
JSP 사용법	<pre> <c:set var="변수명" value="값" /> <c:forTokens items="글자" delims="구분글자" var="변수"> <p>\${변수}</p> </c:forTokens> \${not empty 변수} \${empty 변수} <c:redirect url="이동할_url 주소" /> <c:import url="보여줄페이지" /> <jsp:include page="보여줄페이지"></jsp:include> <c:url value="url 주소" var="변수명" /> \${fn:contains(변수, 값)} \${fn:indexOf(변수, 값)} \${fn:length(변수)} \${fn:replace(변수, 원본값, 바꿀값)} \${fn:split(변수, 자를글자)} \${fn:substring(변수, 첫인덱스번호, 끝인덱스번호)} </pre>

```
${fn:toLowerCase(변수)}  
${fn:toUpperCase(변수)}  
  
<%  
    Date 변수= new 메소드();  
%>  
<%=변수%>  
  
<fmt:formatNumber value="${숫자}"/>  
<fmt:formatNumber  
value="${금액}" type="currency"/>  
  
<fmt:formatDate value="${날짜변수}"  
pattern="yyyy-MM-dd hh:mm:ss"/>
```