

서버 프로그래밍 (스프링 프레임 워크)

임 광 수

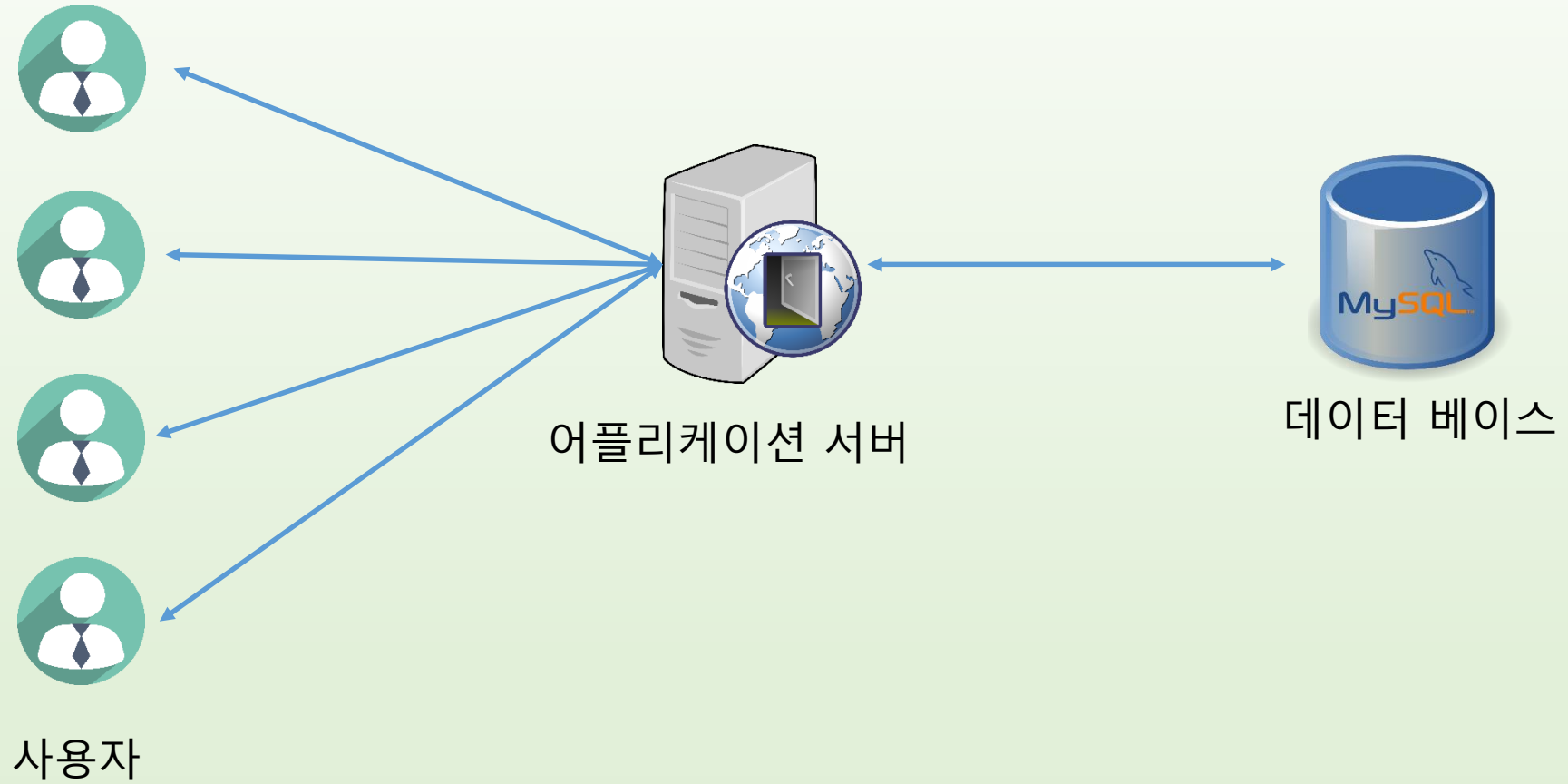
목차

1. Why Java?
2. 서버프로그래밍?
3. EJB?
4. MVC1
5. MVC2
6. 어노테이션
7. 개발 환경 구성하기

왜 JAVA를 사용할까?

- 운영체제에 독립적(JVM)
- 객체 지향언어 (추상화, 상속, 캡슐화, 다형성 등)
- Garbage Collection
- 멀티 쓰레드 지원
- 동적 프로그래밍 가능(필요한 클래스만 동적으로 로딩 가능)

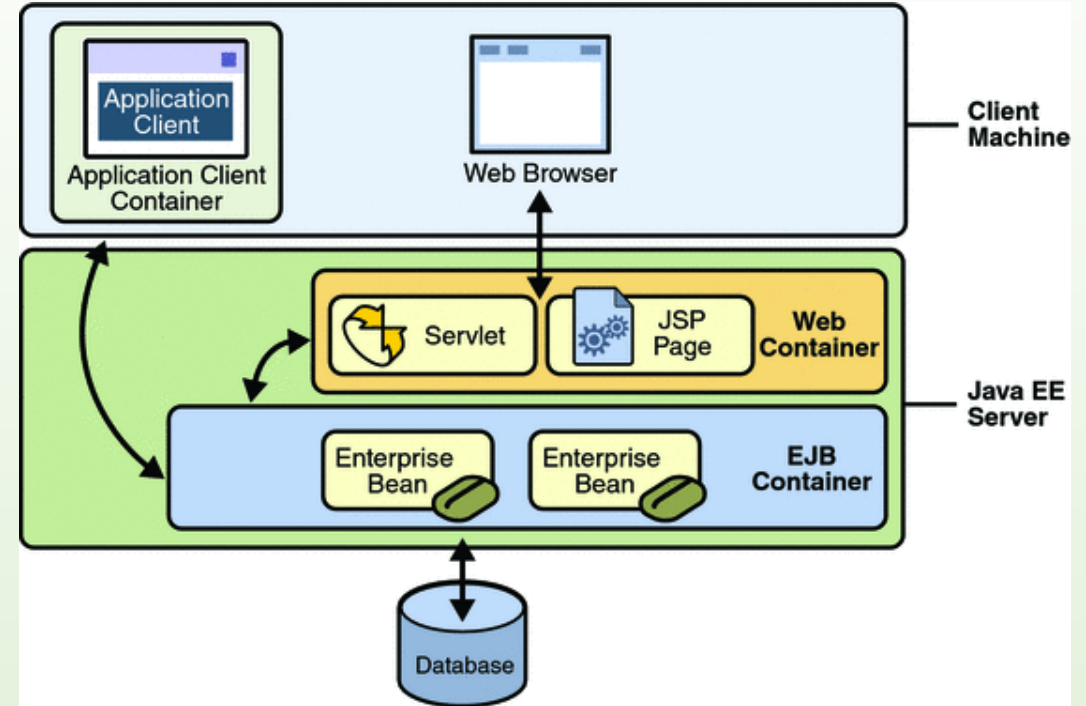
서버 프로그래밍?



EJB (Enterprise JavaBeans)

- Java EE를 위한 서버사이드 컴포넌트 아키텍처

- 개발 생산성 저하
- 유지 보수성 저하
- 테스트 용이성이 떨어짐
- 배포의 어려움

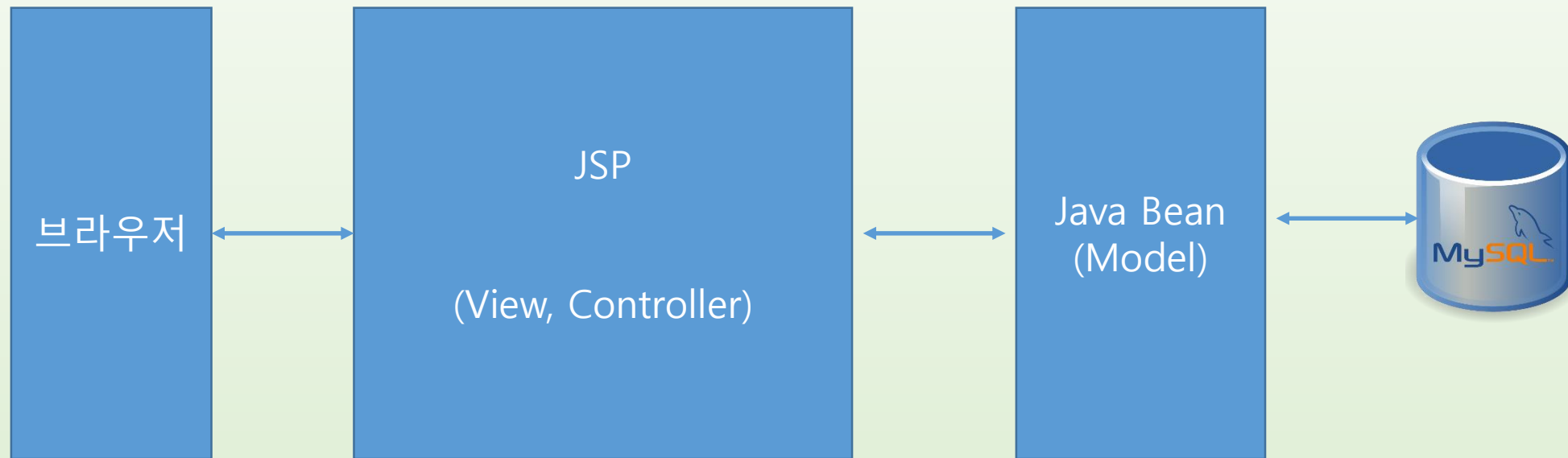


- Plain Old Java Object

- 오래된 방식의 간단한 자바 오브젝트
- Java EE 등의 중량 프레임워크들을 사용하게 되면서 해당 프레임워크에 종속된 "무거운" 객체를 만들게 된 것에 반발해서 사용되게 된 용어

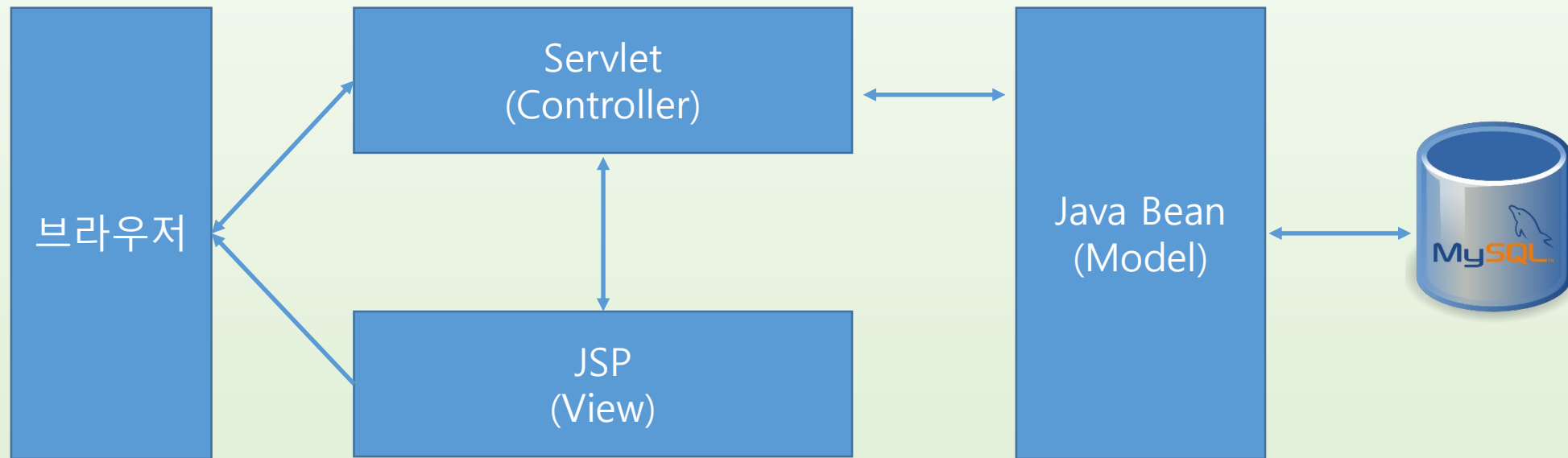
MVC1

- **Model** : 프로그램의 상태를 의미 (데이터)
- **View** : 사용자 인터페이스를 의미
- **Controller** : 데이터와 비즈니스로직 간의 제어를 의미



MVC2

- Servlet : 웹기반의 요청에서 동적인 처리가 가능하도록 해주는 일종의 클래스



스프링 프레임워크

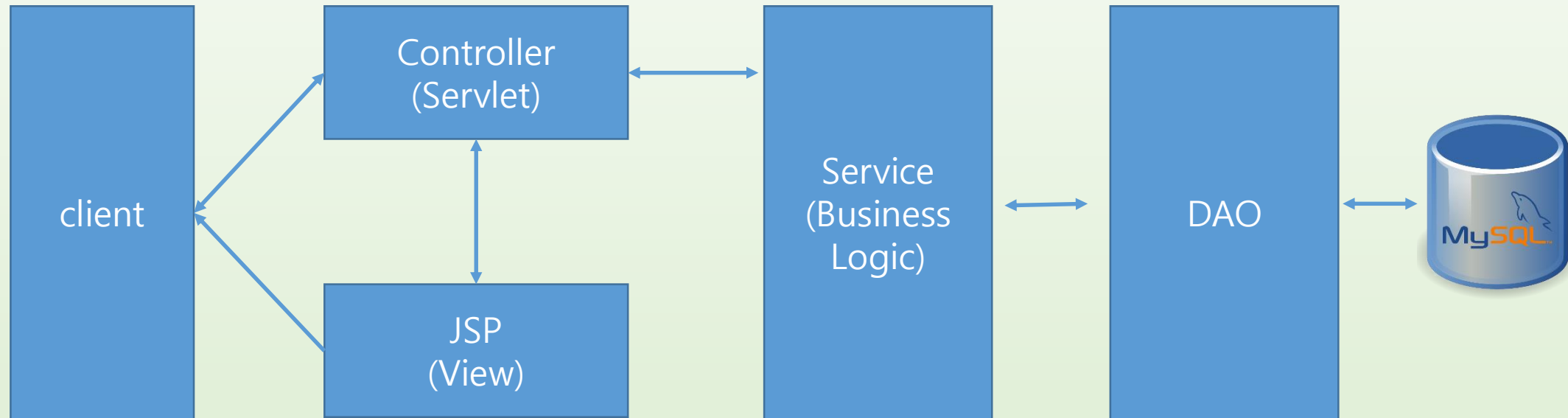
- 프레임 워크 : IoC와 DI의 개념이 포함된 컨테이너
- IoC(Inversion of Control) : 객체의 생성부터 생명주기의 관리까지 모든 객체의 제어권을 가진 일종의 컨테이너
- 의존성 주입 : 프로그래밍에서 구성요소관 의존 관계가 소스코드 내부가 아닌 외부 설정파일 등을 통해 정의하게 되는 디자인 패턴
- 경량 : 스프링의 크기는 작다?
- 관점지향(APO) : 애플리케이션 비즈니스 로직을 시스템의 트랜잭션 관리등의 서비스에서 분리
- 컨테이너 : 독자적으로 운영 가능한 것
- 로드 존슨이 2002년 10월에 출판한 자신의 저서인 Expert One-on-One J2EE Design and Development 에 작성한 코드를 기반으로 최초 버전을 만들게 됨.

스프링 프레임워크

- 활용하기 위해 사전 지식이 필요
- 라이브러리들을 추가하다보면 프로그램이 무거워 짐
- 설정 파일이 너무 복잡함
- 결론 : 어렵다 (T.T)

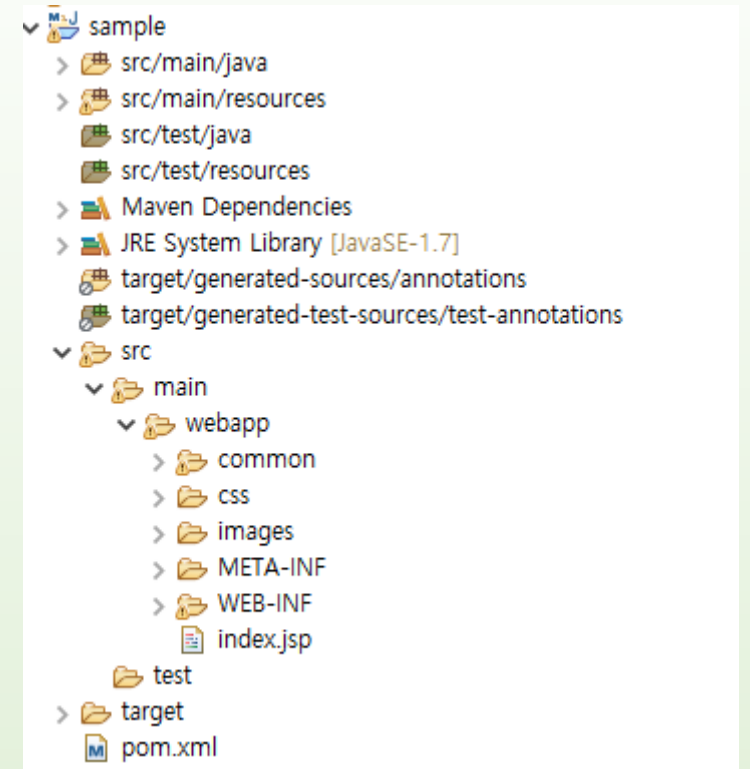
스프링 프레임워크 구조

- 그래도 막상 해보면 별거 아니다



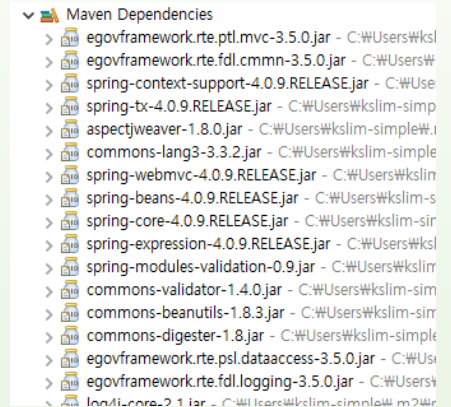
스프링 프로젝트 구조(by maven)

- **src/main/java** : java
- **src/main/resource** : 설정파일
- **src/test/java** : 테스트 코드
- **src/test/resource** : 테스트 시 사용될 설정파일
- **Maven Dependencies** : 메이븐에 의해 활용되는 라이브러리
- **JRE System Library** : 기본 java라이브러리
- **target** : class파일이 들어가있는 경로
- **src**
- **src/main/webapp** : view pom



스프링의 동작원리

- 라이브러리 : 어플리케이션의 실행을 위해 특정 기능들을 미리 구현해 놓은 부분
- 설정 파일 : 라이브러리를 해당 프로젝트에 맞게 활용하기 위하여 설정되는 값 정보
- 어노테이션 : 스프링



```
@Controller
public class EgovSampleController {

    /** EgovSampleService */
    @Resource(name = "sampleService")
    private EgovSampleService sampleService;

    /** EgovPropertyService */
    @Resource(name = "propertiesService")
    protected EgovPropertyService propertiesService;

    /** Validator */
    @Resource(name = "beanValidator")
    protected DefaultBeanValidator beanValidator;
}
```

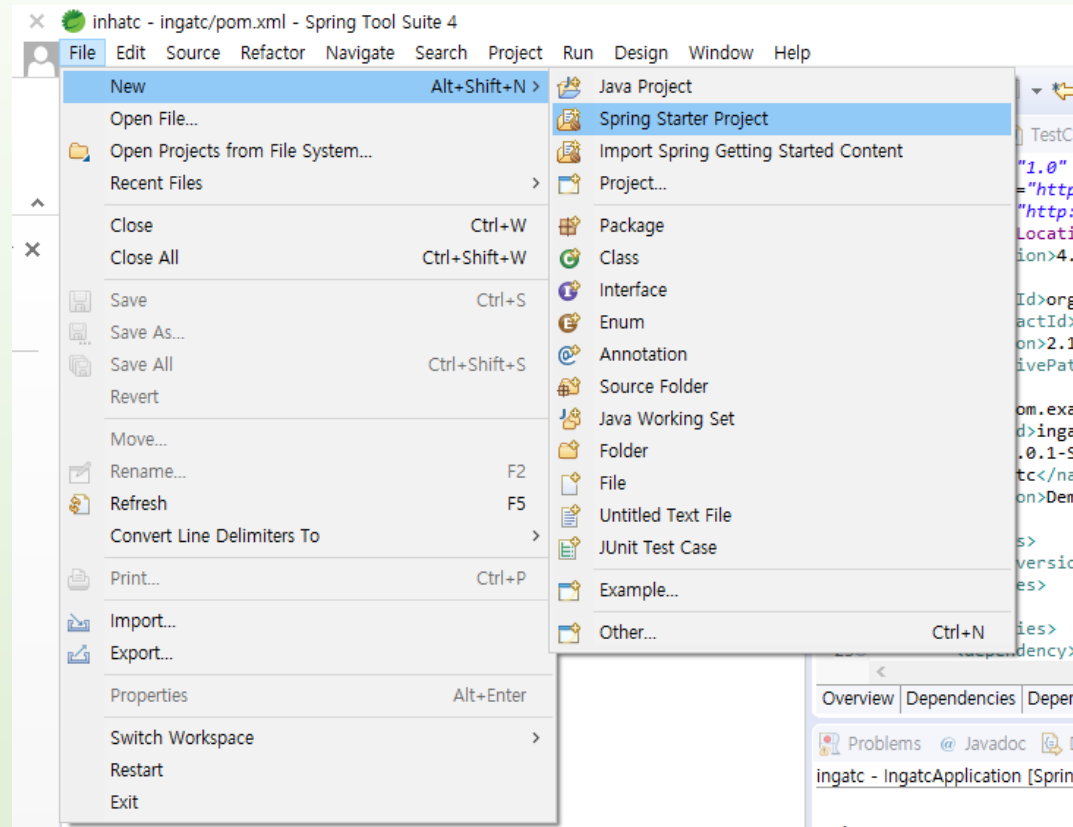
```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
        http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.0.xsd">

    <context:component-scan base-package="egovframework">
        <context:exclude-filter type="annotation" expression="org.springframework.stereotype.Controller" />
    </context:component-scan>

    <bean id="messageSource" class="org.springframework.context.support.ReloadableResourceBundleMessageSource">
        <property name="basenames">
            <list>
                <value>classpath:/egovframework/message/message-common</value>
                <value>classpath:/egovframework/rte/fdl/idgnr/messages/idgnr</value>
                <value>classpath:/egovframework/rte/fdl/property/messages/properties</value>
            </list>
        </property>
    </bean>
</beans>
```

Hello World!!

- 프로젝트 생성 방법



Hello World!!

- 프로젝트 생성 방법

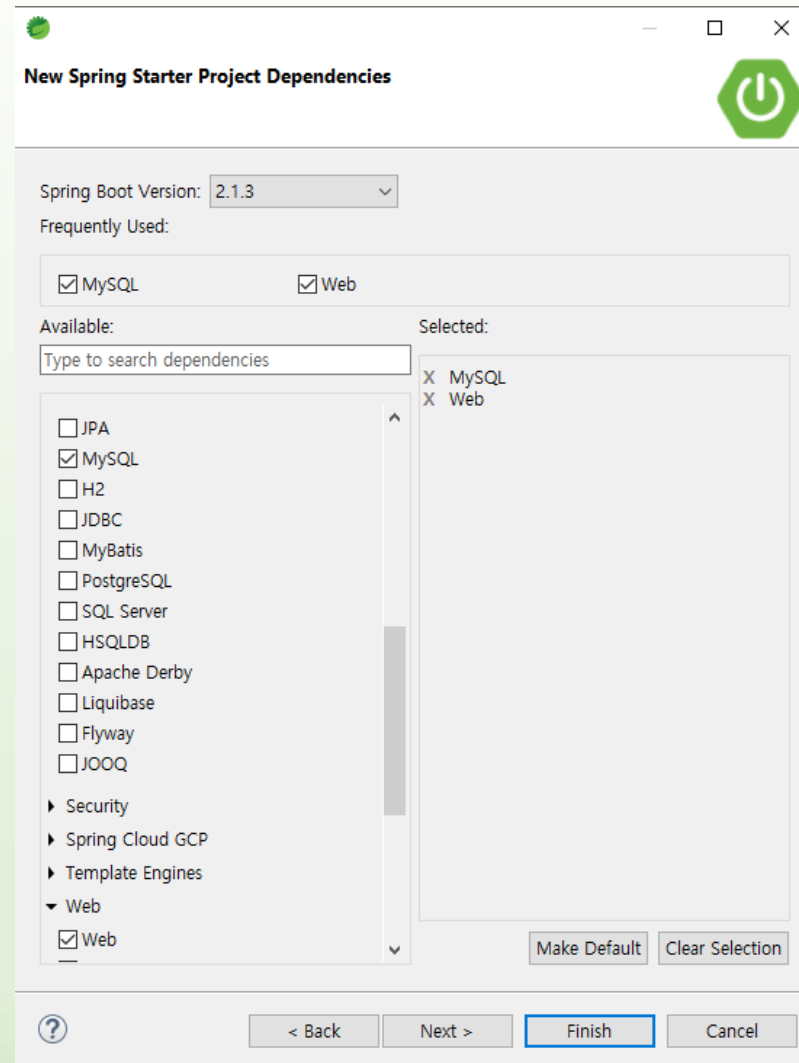
The screenshot shows the 'New Spring Starter Project' dialog box. The fields are filled with the following values:

- Service URL: `https://start.spring.io`
- Name: `hello_spring`
- Use default location: ☒
- Location: `C:\winhac\hello_spring`
- Type: `Maven`
- Packaging: `Jar`
- Java Version: `8`
- Language: `Java`
- Group: `kr.ac.inhac`
- Artifact: `hello_spring`
- Version: `0.0.1-SNAPSHOT`
- Description: `Demo project for Spring Boot`
- Package: `kr.ac.inhac.mvc`

At the bottom, there is a 'Working sets' section with an unchecked checkbox 'Add project to working sets' and a 'New...' button. Below that is a 'Working sets:' dropdown menu and a 'Select...' button. The 'Next >' button is highlighted in blue.

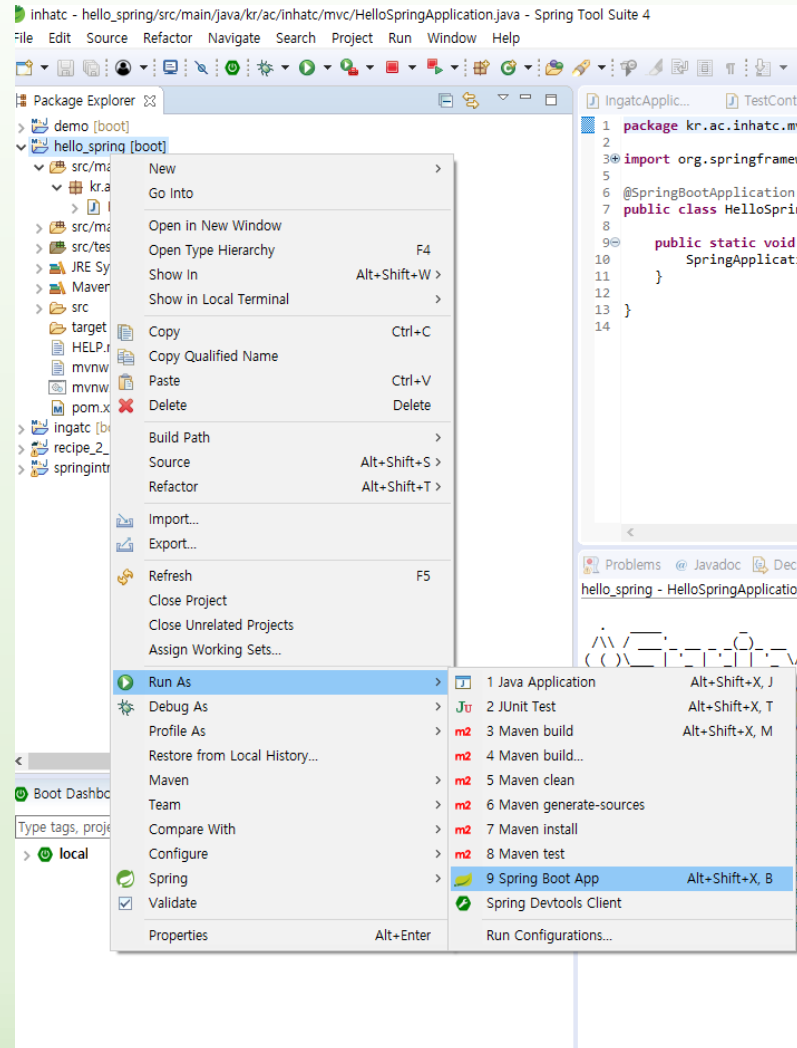
Hello World!!

- 프로젝트 생성 방법



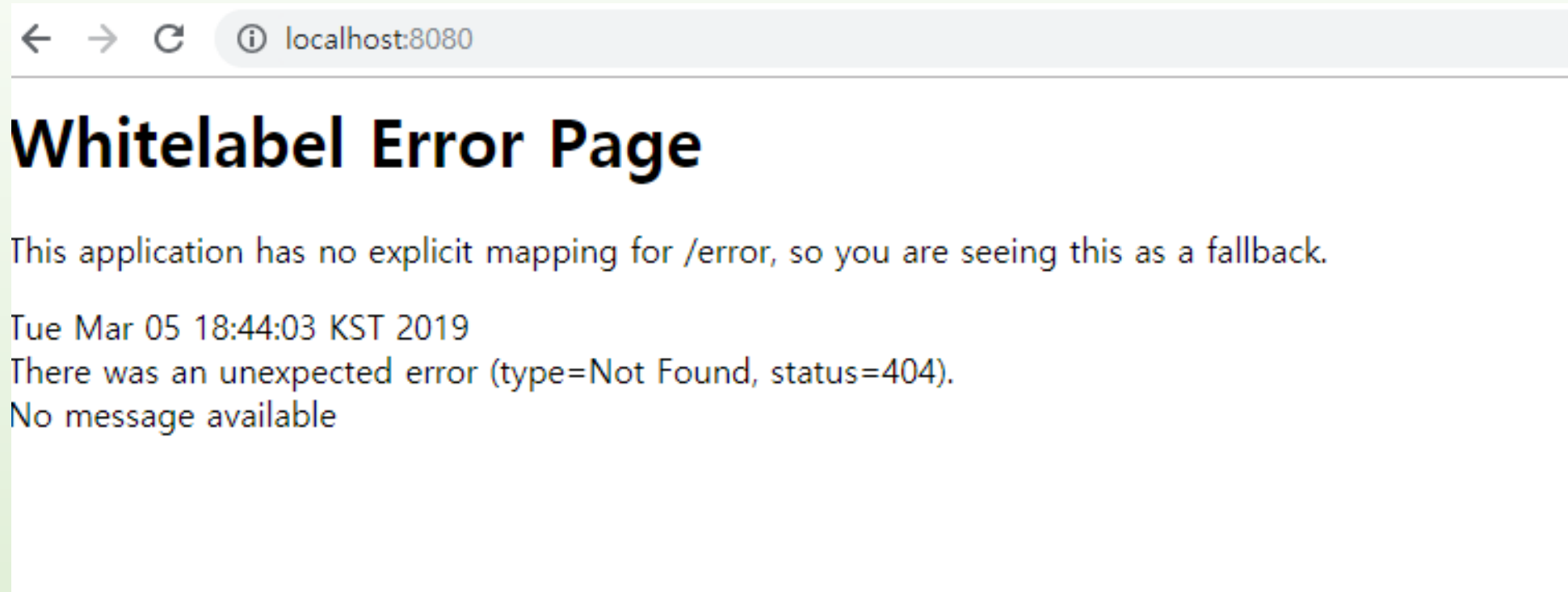
Hello World

• 스프링 프로젝트 실행하기



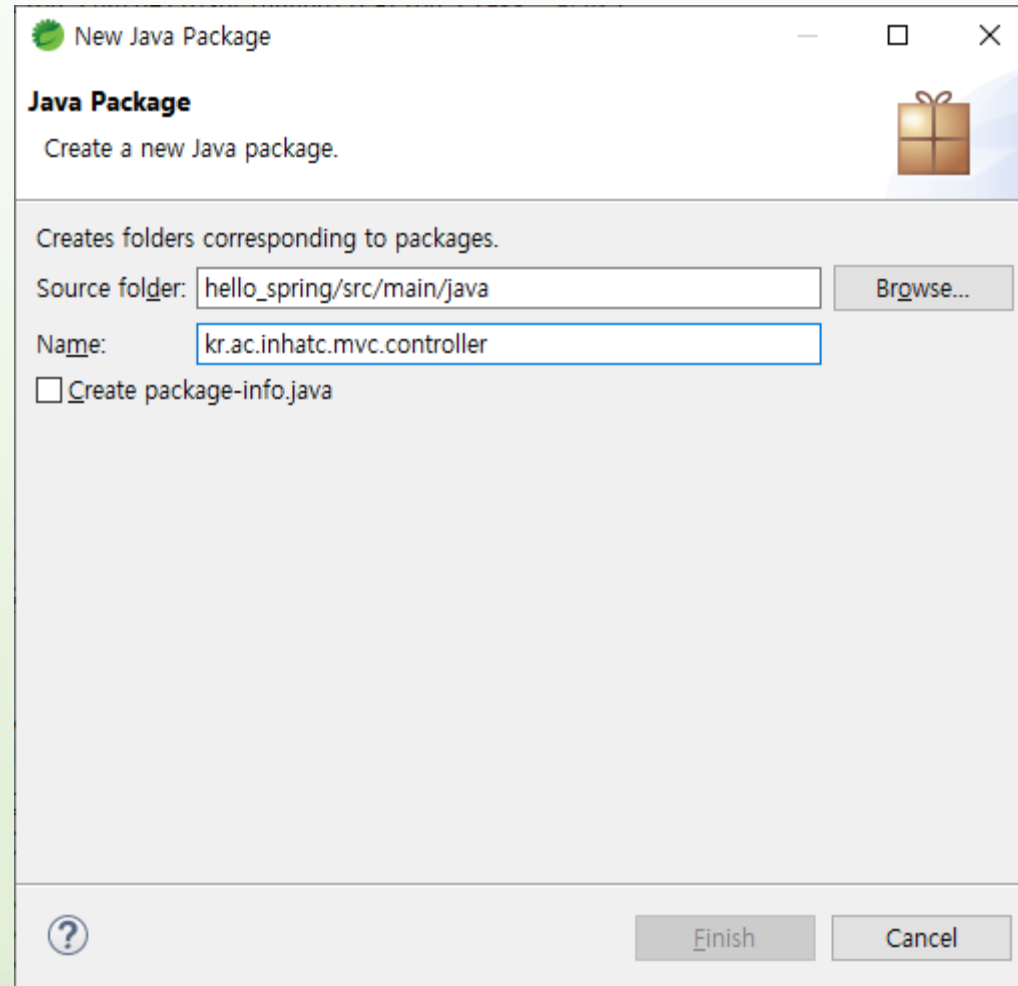
Hello World!!

- 화면 접속해보기



Hello World!!

- package 생성



Hello World

- Class생성

New Java Class

Create a new Java class.

Source folder: Browse...

Package: Browse...

☐ Enclosing type: Browse...

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

? Finish Cancel

Hello World!!

- MvcController.java

```
package kr.ac.inhatc.mvc.controller;

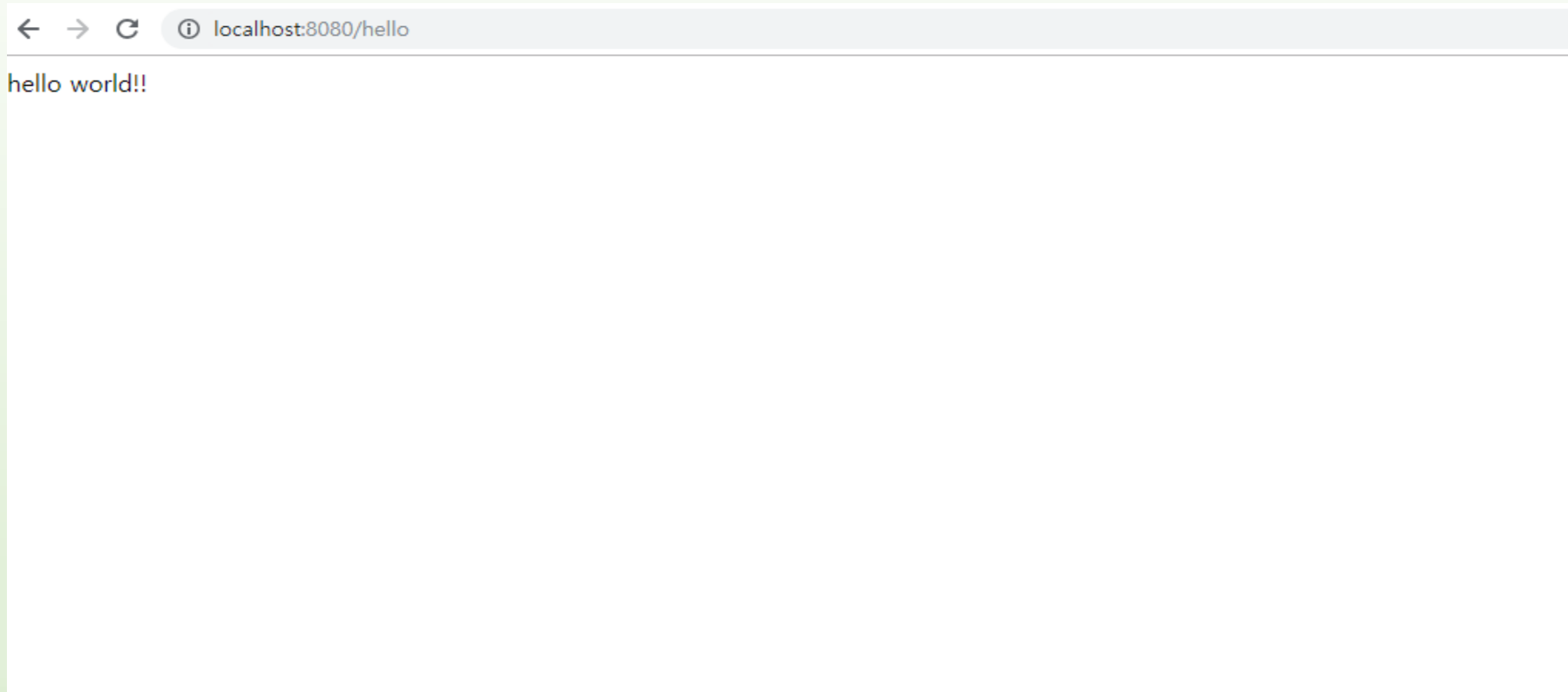
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class MvcController {

    @RequestMapping("hello")
    public String hello() {
        return "hello world!!";
    }
}
```

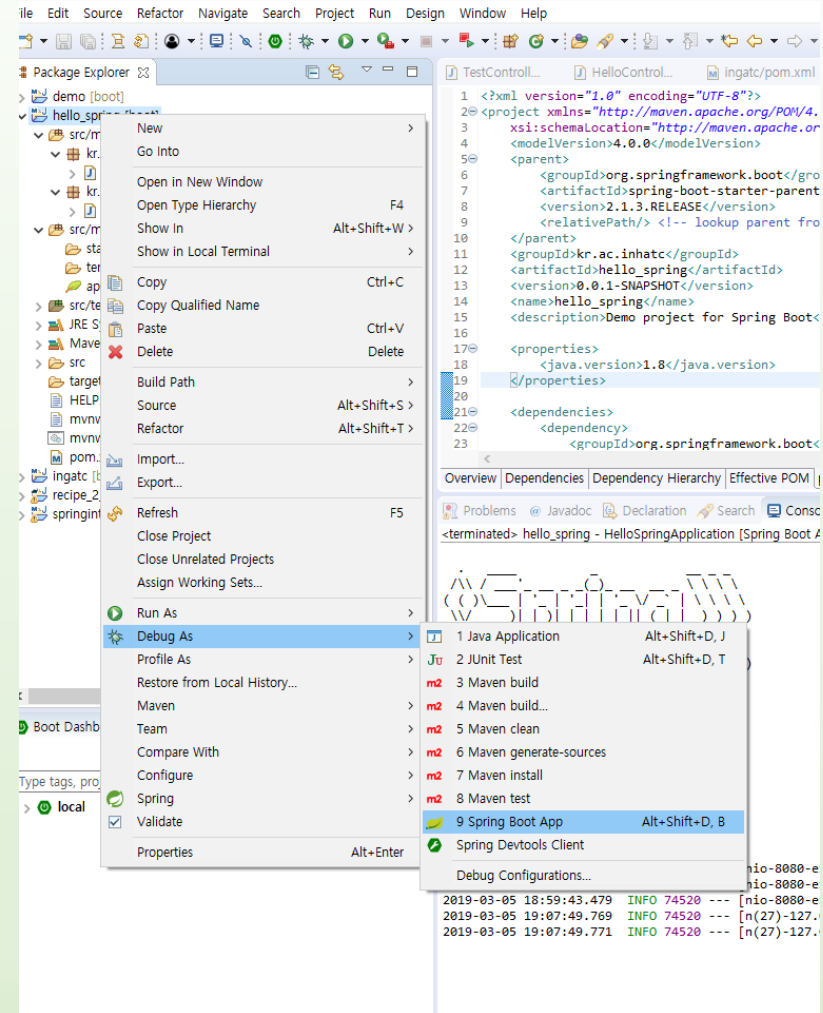
Hello World!!

- 화면출력



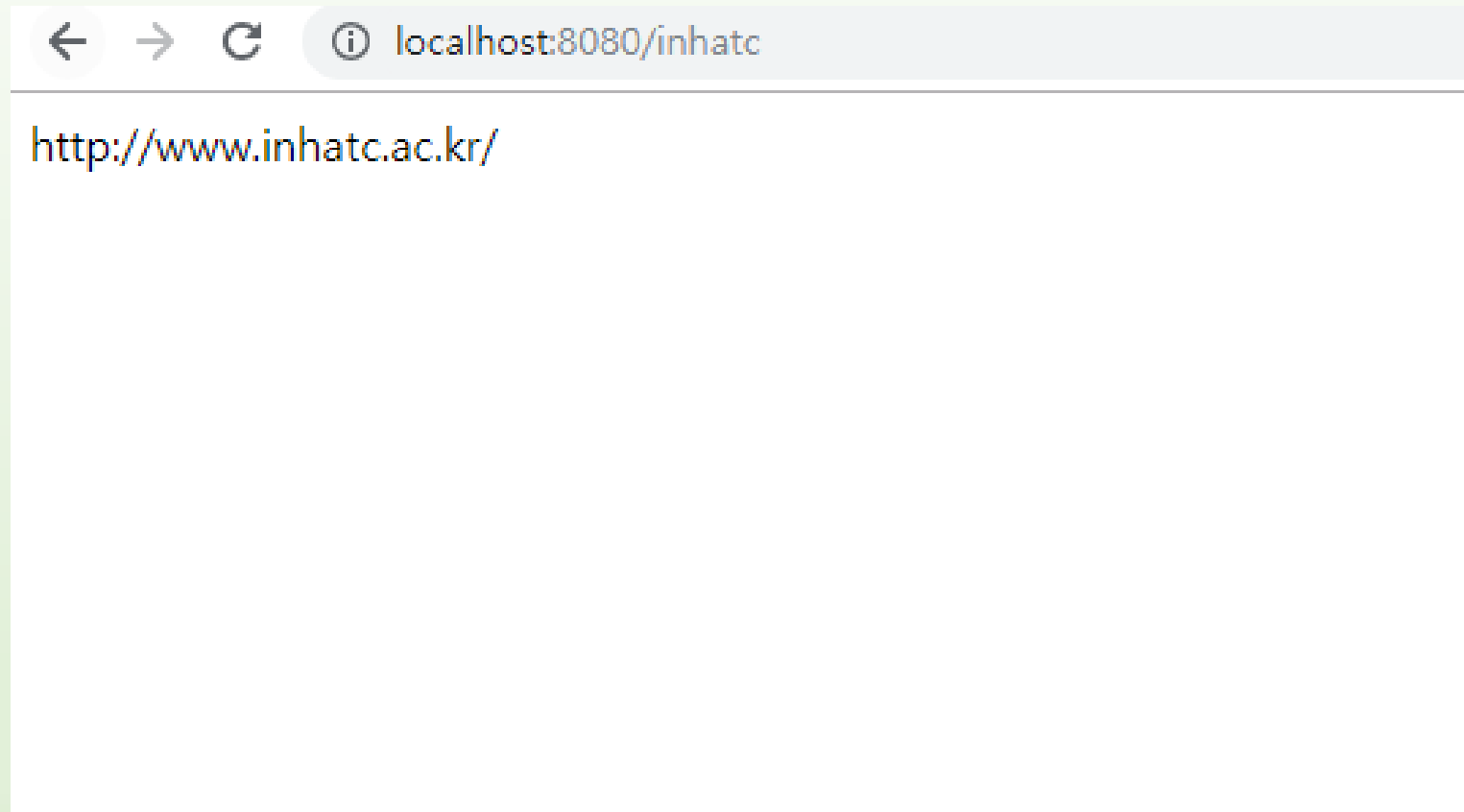
디버깅 해보기

- 디버깅 실행방법
- 디버깅 창 보는 방법
- 디버깅 시 값을 보는 방법



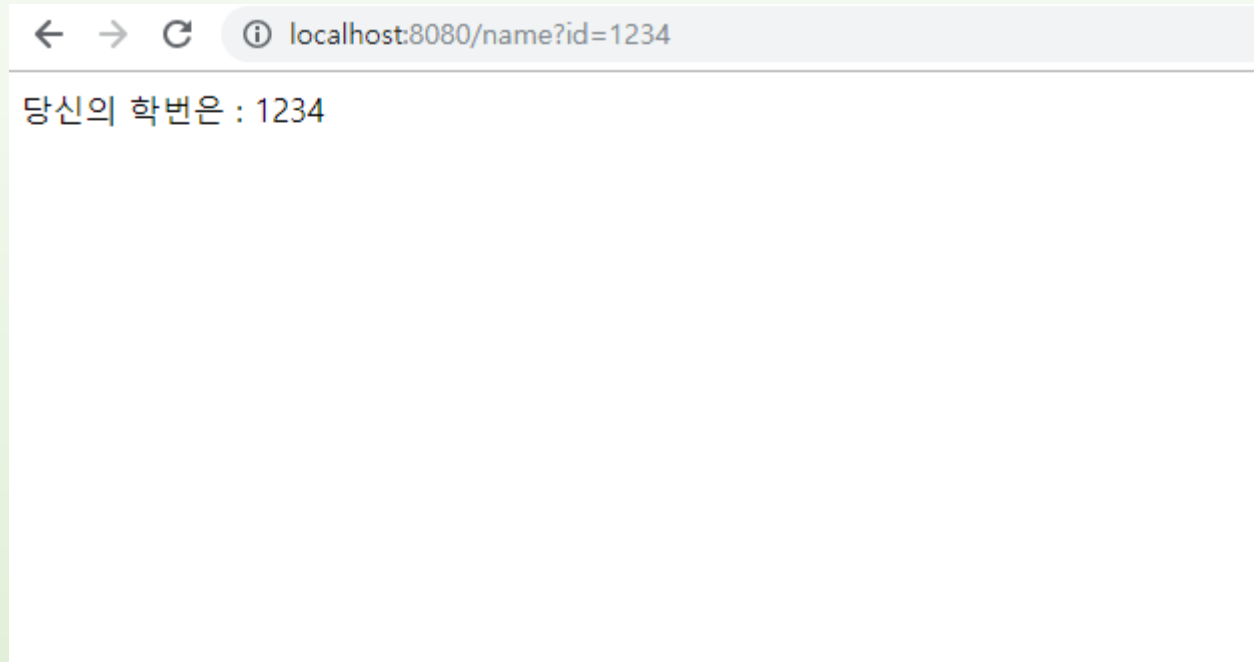
연습문제1

- Inhatc라는 주소로 접근 시 학교 홈페이지 주소를 보여주는 프로그램을 작성하시오



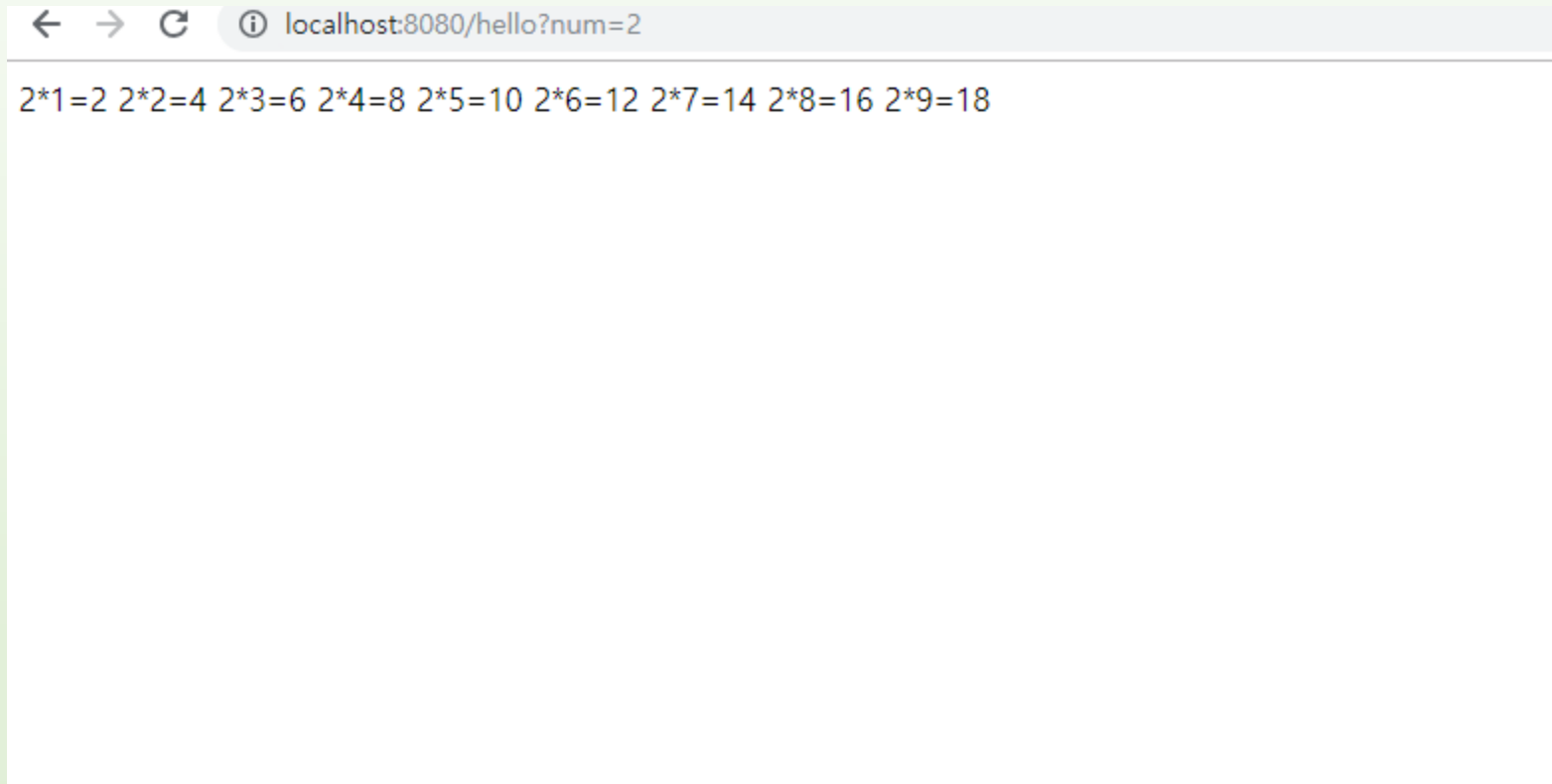
연습문제2

- name 이라는 주소로 접근하고 id라는 파라미터로 학번을 입력받아 입력한 학번을 출력해주는 프로그램을 작성하시오



연습문제3

- Spring 프로젝트를 활용하여 구구단의 숫자를 입력 받고 출력해주는 프로그램 작성
- 예 : `http://localhost:8080/hello?num=2`



Q & A