Spring Boot

Kim Hye Kyung

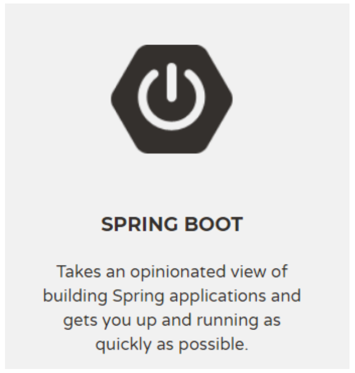
topickim@naver.com

Spring Boot 개요 및 개발 환경 구축

Spring Boot 개요

- Spring Framework의 Sub Project
- Spring과 Boot의 합성어
  - Spring : 오픈 소스 프레임워크
  - Boot : 컴퓨터를 부팅한다 즉 시스템에서 사용 가능한 상태로 만든다는 의미

Spring Framework을 사용 가능한 상태로 만들어 주는 도구


SPRING BOOT

Takes an opinionated view of building Spring applications and gets you up and running as quickly as possible.

Spring Boot 개요

Spring Framework

+

Embedded HTTP Servers (Tomcat)

-

XML <bean> configuration or @Configuration

=

Spring Boot

Spring Boot의 장점

라이브러리 관리 자동화

설정의 자동화

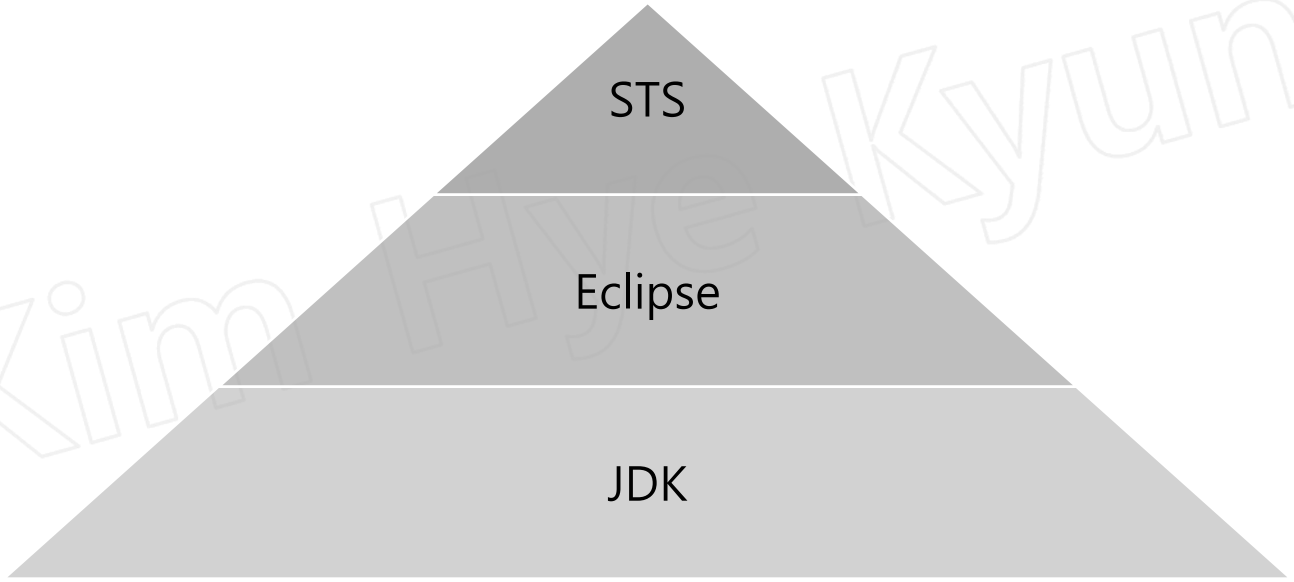
라이브러리 버전 자동 관리

테스트 환경과 내장 톰캣

독립적으로 실행 가능한 JAR

Spring Boot 개발 환경 구축

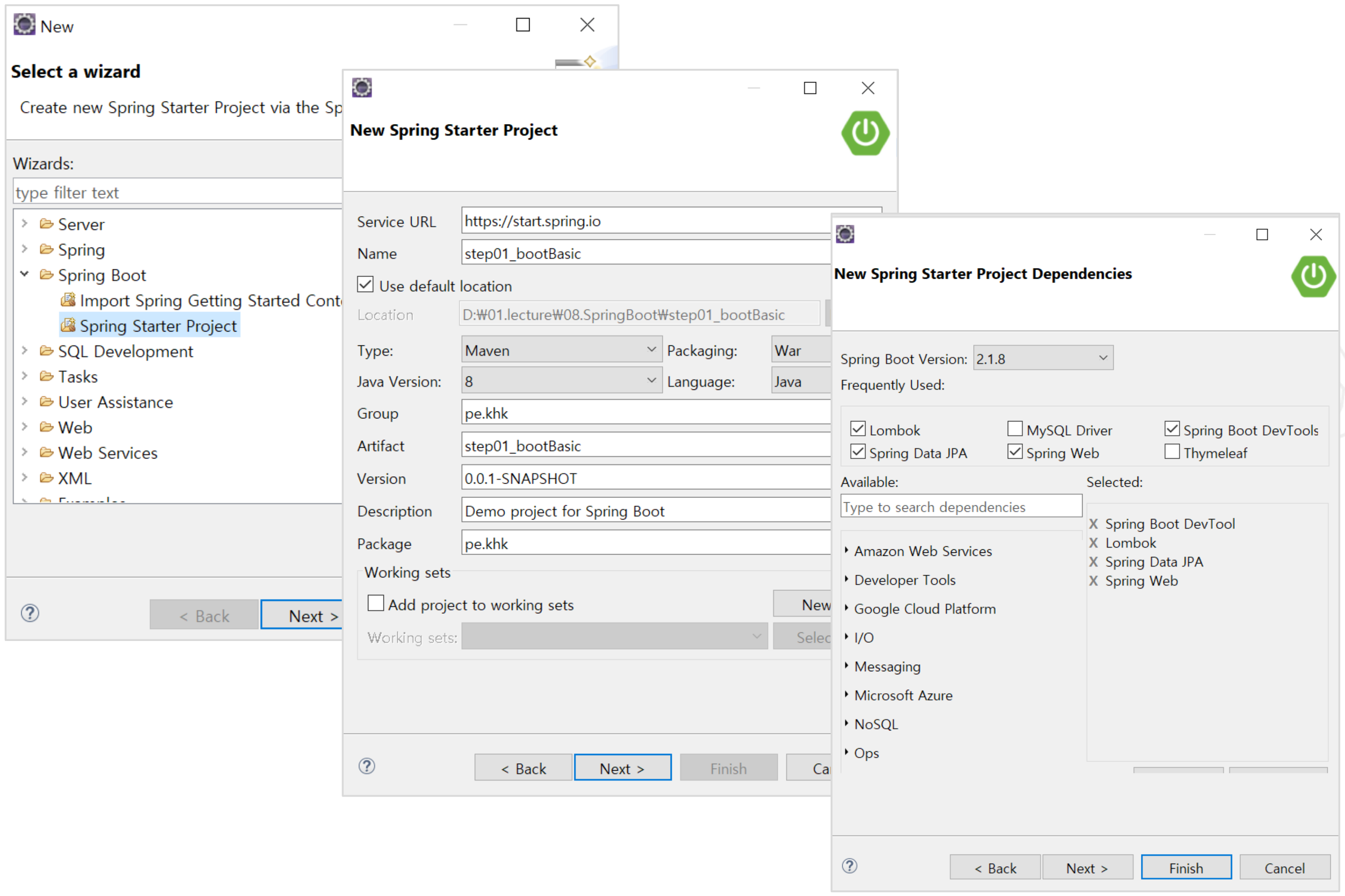
- Spring Boot 2.0 이상인 경우 반드시 JDK 8 이상 설치



Spring Boot 퀵스타트

- Spring Boot로 Project 생성시
    - Spring library 등의 librar들을 개발자가 신경쓸 필요 없음
        - 모든 library들을 자동으로 다운로드 및 관리
    - XML 환경 설정 파일 역시 작성하지 않음
        - Bean 설정을 위한 XML이 아닌 어노테이션 기반으로 처리

Spring Boot Project 생성 단계


New

Select a wizard

Create new Spring Starter Project via the Sp

Wizards:

type filter text

- Server
- Spring
- Spring Boot
  - Import Spring Getting Started Cont
  - Spring Starter Project
- SQL Development
- Tasks
- User Assistance
- Web
- Web Services
- XML

< Back | Next >

New Spring Starter Project

Service URL	https://start.spring.io
Name	step01_bootBasic
☑ Use default location	
Location	D:₩01.lecture₩08.SpringBoot₩step01_bootBasic
Type:	Maven
Packaging:	War
Java Version:	8
Language:	Java
Group	pe.khk
Artifact	step01_bootBasic
Version	0.0.1-SNAPSHOT
Description	Demo project for Spring Boot
Package	pe.khk

Working sets

☐ Add project to working sets

Working sets:

< Back | Next > | Finish | Ca

New Spring Starter Project Dependencies

Spring Boot Version: 2.1.8

Frequently Used:

☑ Lombok	☐ MySQL Driver	☑ Spring Boot DevTools
☑ Spring Data JPA	☑ Spring Web	☐ Thymeleaf

Available:

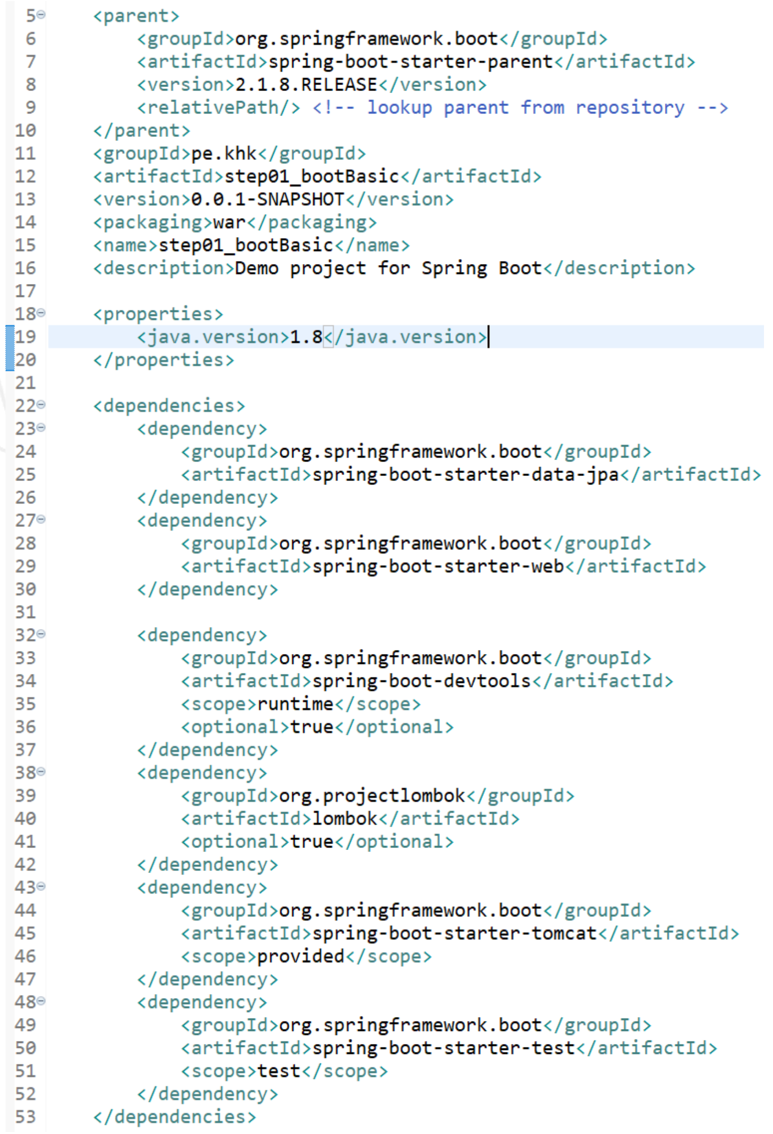
Type to search dependencies

- Amazon Web Services
- Developer Tools
- Google Cloud Platform
- I/O
- Messaging
- Microsoft Azure
- NoSQL
- Ops

Selected:

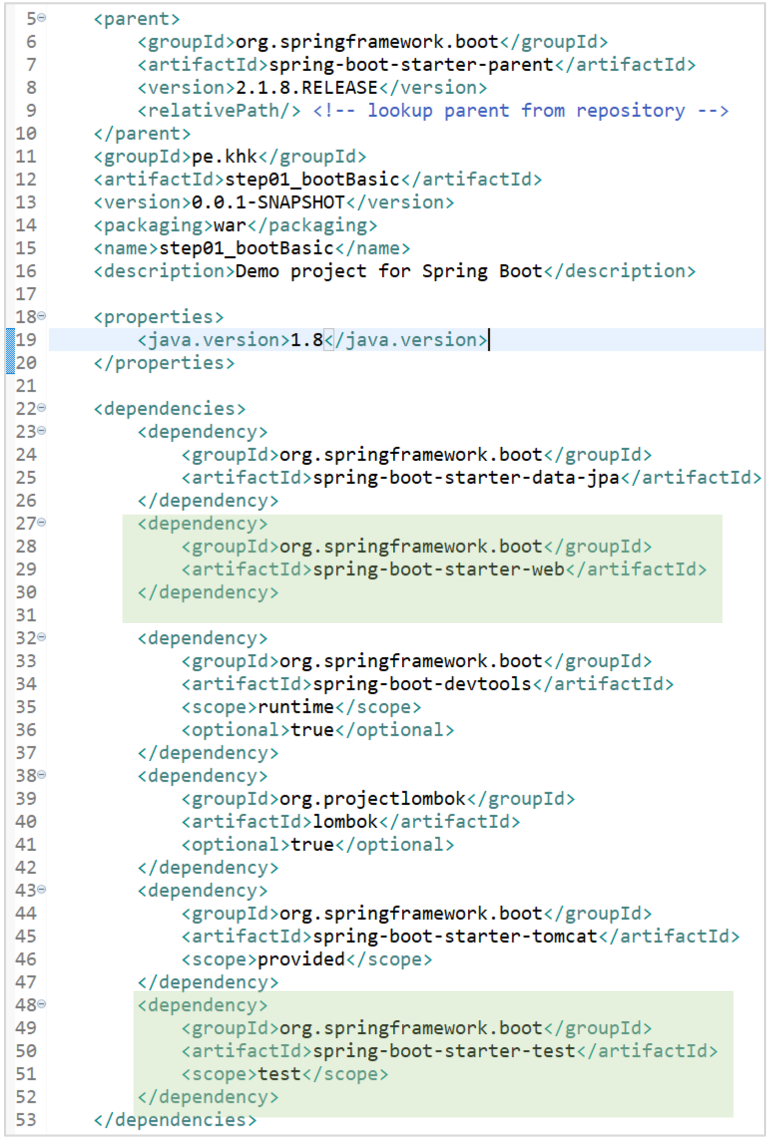
X Spring Boot DevTool
X Lombok
X Spring Data JPA
X Spring Web

< Back | Next > | Finish | Cancel



```xml
<parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.1.8.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>pe.khk</groupId>
<artifactId>step01_bootBasic</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>war</packaging>
<name>step01_bootBasic</name>
<description>Demo project for Spring Boot</description>

<properties>
    <java.version>1.8</java.version>
</properties>

<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-devtools</artifactId>
        <scope>runtime</scope>
        <optional>true</optional>
    </dependency>
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <optional>true</optional>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-tomcat</artifactId>
        <scope>provided</scope>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>
```

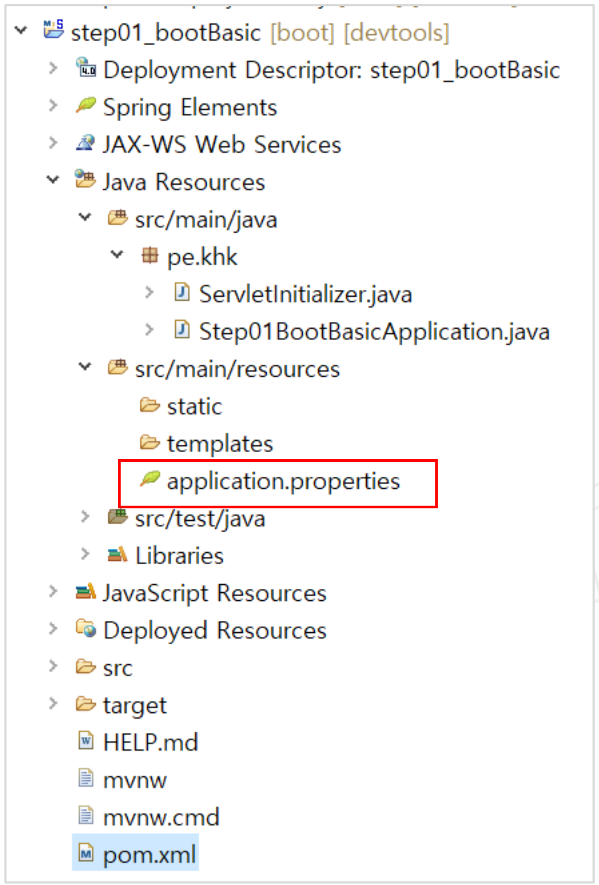
Spring Boot 기본 library



```xml
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.1.8.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>pe.khk</groupId>
    <artifactId>step01_bootBasic</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>war</packaging>
    <name>step01_bootBasic</name>
    <description>Demo project for Spring Boot</description>

    <properties>
        <java.version>1.8</java.version>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-devtools</artifactId>
            <scope>runtime</scope>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-tomcat</artifactId>
            <scope>provided</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>
```

library	설명
spring-boot-starter-web	웹 애플리케이션 개발에 필요한 Spring MVC 관련 library Spring Boot가 웹 프로젝트 환경에 최적화된 library들 등록
spring-boot-starter-test	Junit을 비롯한 test 관련 libtary

Spring Boot Project 구조

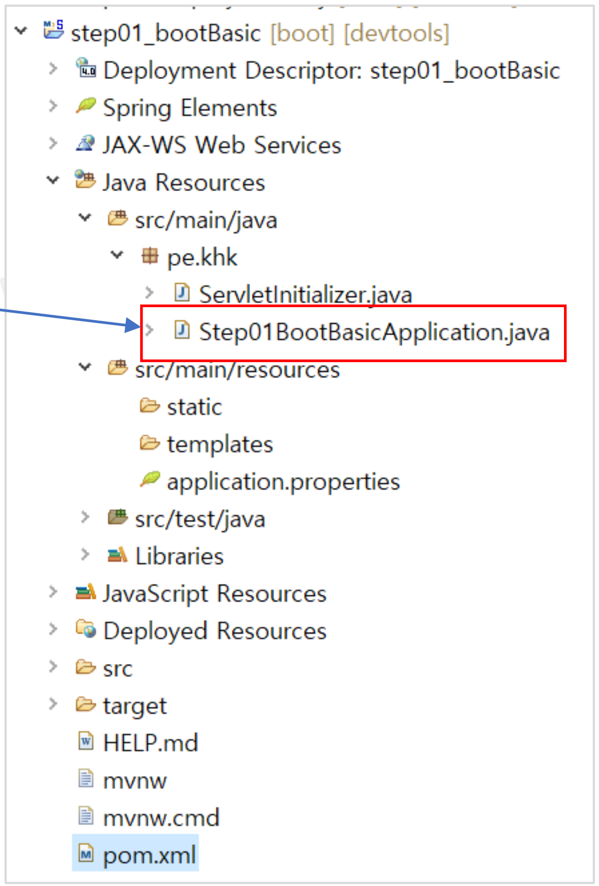


```
step01_bootBasic [boot] [devtools]
  Deployment Descriptor: step01_bootBasic
  Spring Elements
  JAX-WS Web Services
  Java Resources
    src/main/java
      pe.khk
        ServletInitializer.java
        Step01BootBasicApplication.java
    src/main/resources
      static
      templates
      application.properties
    src/test/java
    Libraries
  JavaScript Resources
  Deployed Resources
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

구조	설명
src/main/java	자바 소스
src/main/resources	자바소스가 아닌 xml이나
src/main/resources/static	HTML과 같은 정적인 웹리소스 저장
src/main/resources/templates	타임리프와 같은 템플릿 기반의 웹소스
application.properties	프로젝트 전체에서 사용할 프로퍼티 정보들 저장

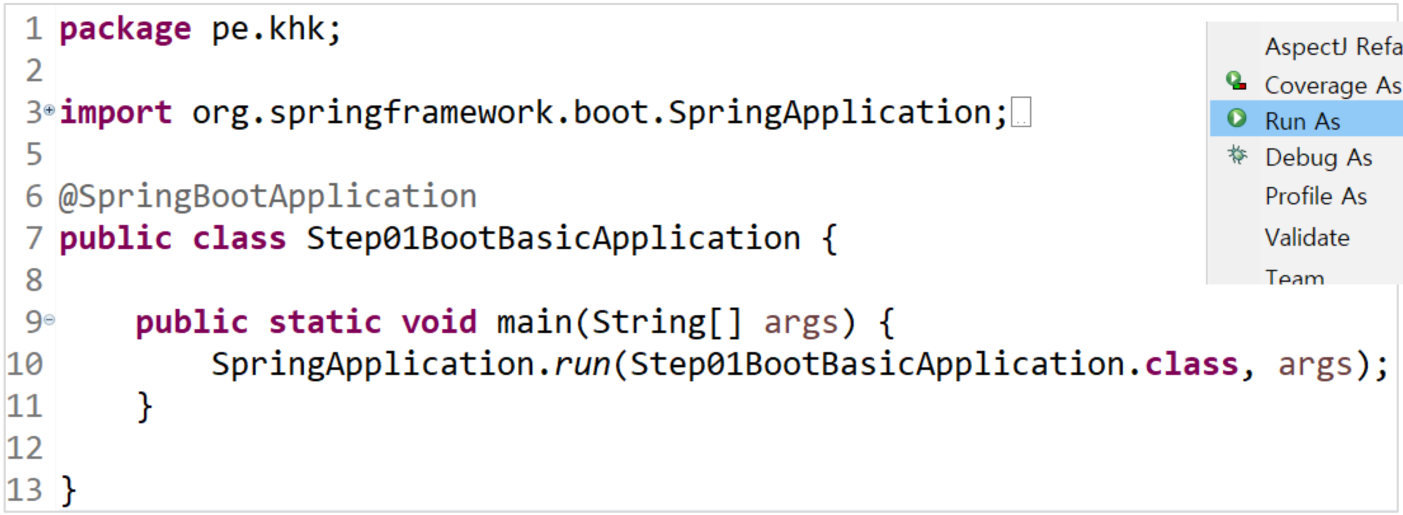
Spring Boot 실행

- main 클래스
  - 자동 생성되는 기본 클래스
    - Project명+Application.java
      - 웹 애플리케이션으로 실행
  - src/main/java 경로에 자동 생성
  - 실행시 내장 tomcat 자동 구동되면서 실행
- 실행 방식
  - 방법 1 – 웹 애플리케이션 방식으로 실행
  - 방법 2 – 일반 자바 애플리케이션 방식으로 실행

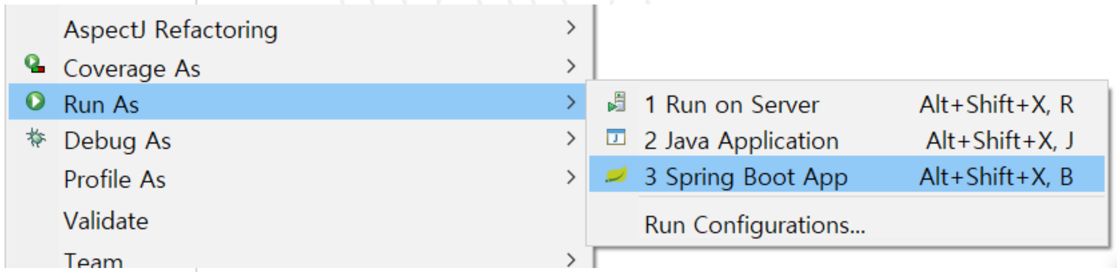

```
step01_bootBasic [boot] [devtools]
  Deployment Descriptor: step01_bootBasic
  Spring Elements
  JAX-WS Web Services
  Java Resources
    src/main/java
      pe.khk
        ServletInitializer.java
        Step01BootBasicApplication.java
    src/main/resources
      static
      templates
      application.properties
    src/test/java
    Libraries
  JavaScript Resources
  Deployed Resources
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

Spring Boot 실행

- 웹 애플리케이션 방식으로 실행
- @SpringBootApplication스프링 부트로 만든 애플리케이션의 시작 클래스 의미



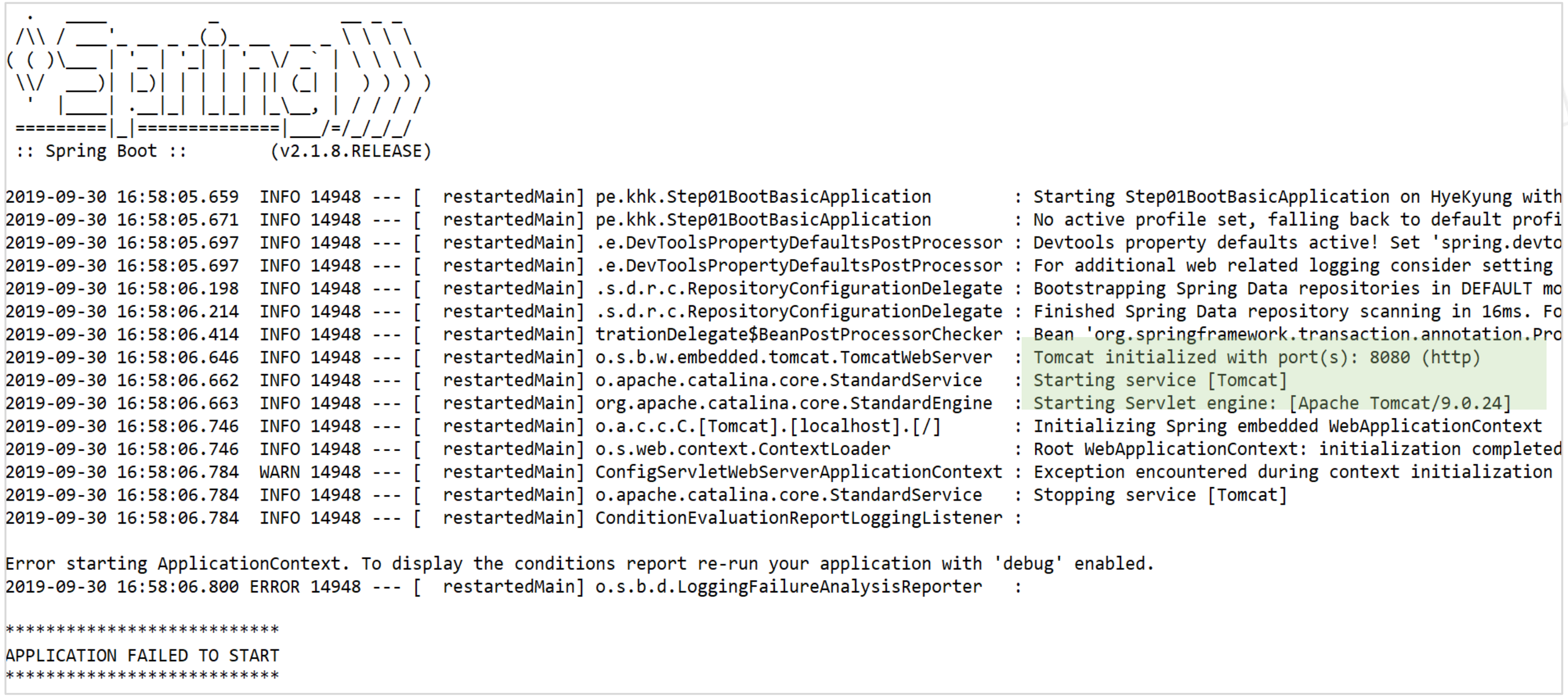
```java
package pe.khk;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class Step01BootBasicApplication {

    public static void main(String[] args) {
        SpringApplication.run(Step01BootBasicApplication.class, args);
    }

}
```


AspectJ Refactoring
Coverage As
Run As
Debug As
Profile As
Validate
Team
1 Run on Server Alt+Shift+X, R
2 Java Application Alt+Shift+X, J
3 Spring Boot App Alt+Shift+X, B
Run Configurations...

```java
//실행 방식 3 : WebApplicationType.SERVLET 설정으로 웹으로 실행
SpringApplication application = new SpringApplication(Step01BootBasicApplication.class);
application.setWebApplicationType(WebApplicationType.SERVLET);
application.run(args);
```

Spring Boot 실행

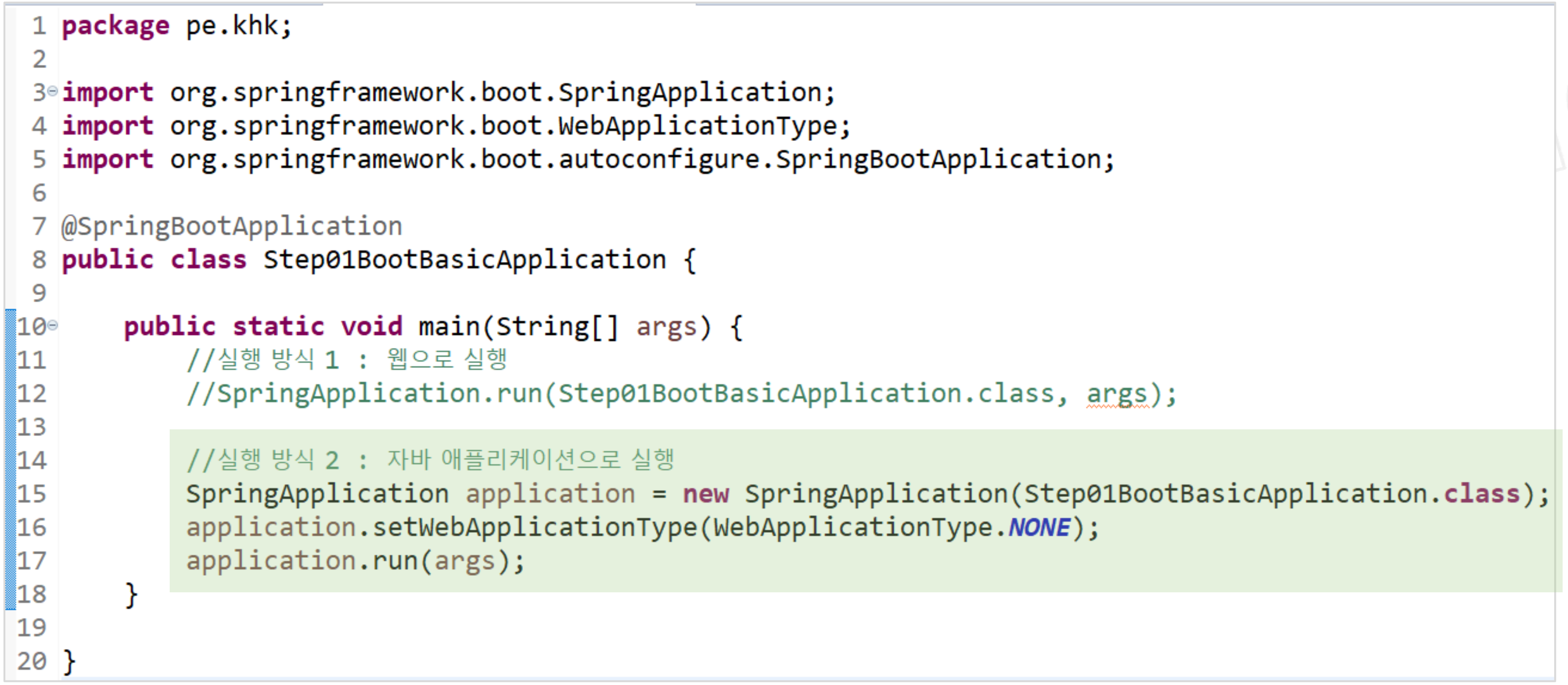
- 웹 애플리케이션 방식으로 실행



```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.1.8.RELEASE)

2019-09-30 16:58:05.659  INFO 14948 --- [  restartedMain] pe.khk.Step01BootBasicApplication        : Starting Step01BootBasicApplication on HyeKyung with
2019-09-30 16:58:05.671  INFO 14948 --- [  restartedMain] pe.khk.Step01BootBasicApplication        : No active profile set, falling back to default profi
2019-09-30 16:58:05.697  INFO 14948 --- [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devto
2019-09-30 16:58:05.697  INFO 14948 --- [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting
2019-09-30 16:58:06.198  INFO 14948 --- [  restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data repositories in DEFAULT mo
2019-09-30 16:58:06.214  INFO 14948 --- [  restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 16ms. Fo
2019-09-30 16:58:06.414  INFO 14948 --- [  restartedMain] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.annotation.Pro
2019-09-30 16:58:06.646  INFO 14948 --- [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
2019-09-30 16:58:06.662  INFO 14948 --- [  restartedMain] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2019-09-30 16:58:06.663  INFO 14948 --- [  restartedMain] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.24]
2019-09-30 16:58:06.746  INFO 14948 --- [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2019-09-30 16:58:06.746  INFO 14948 --- [  restartedMain] o.s.web.context.ContextLoader            : Root WebApplicationContext: initialization completed
2019-09-30 16:58:06.784  WARN 14948 --- [  restartedMain] ConfigServletWebServerApplicationContext : Exception encountered during context initialization
2019-09-30 16:58:06.784  INFO 14948 --- [  restartedMain] o.apache.catalina.core.StandardService   : Stopping service [Tomcat]
2019-09-30 16:58:06.784  INFO 14948 --- [  restartedMain] ConditionEvaluationReportLoggingListener :

Error starting ApplicationContext. To display the conditions report re-run your application with 'debug' enabled.
2019-09-30 16:58:06.800 ERROR 14948 --- [  restartedMain] o.s.b.d.LoggingFailureAnalysisReporter   :

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

Spring Boot 실행

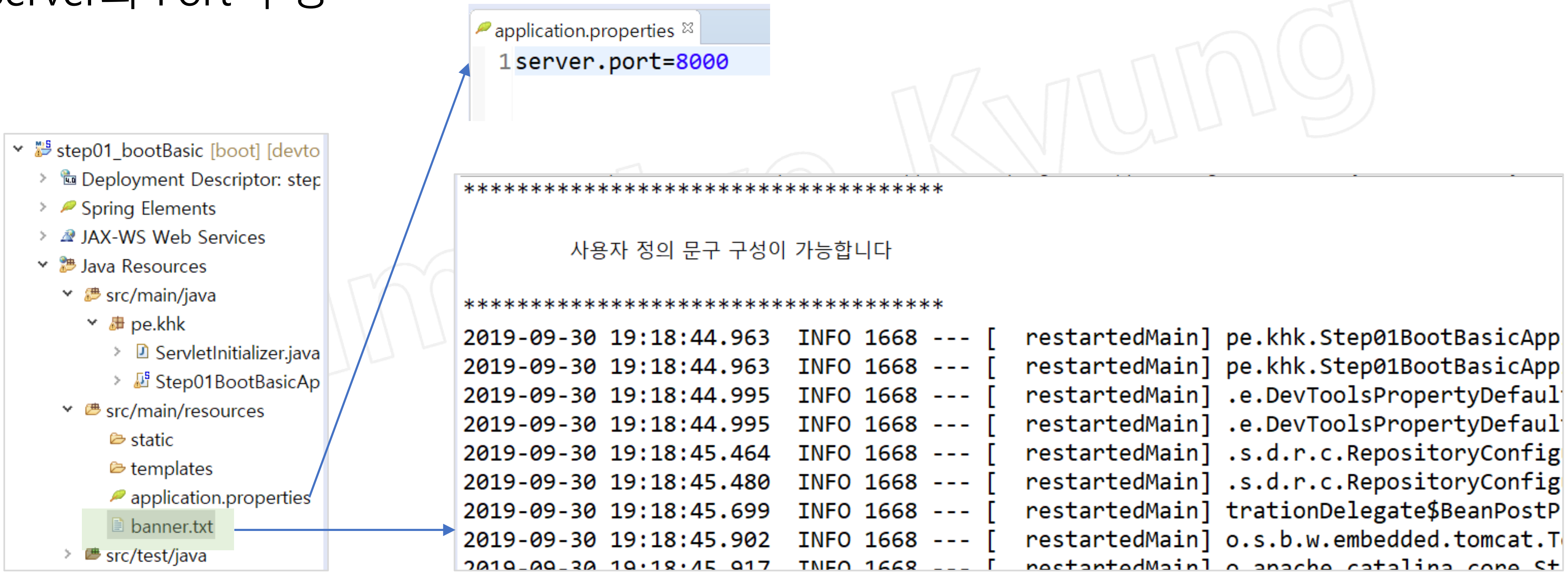
- 일반 자바 애플리케이션 방식으로 실행



```java
package pe.khk;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.WebApplicationType;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class Step01BootBasicApplication {

    public static void main(String[] args) {
        //실행 방식 1 : 웹으로 실행
        //SpringApplication.run(Step01BootBasicApplication.class, args);

        //실행 방식 2 : 자바 애플리케이션으로 실행
        SpringApplication application = new SpringApplication(Step01BootBasicApplication.class);
        application.setWebApplicationType(WebApplicationType.NONE);
        application.run(args);
    }

}
```

Spring Boot 실행

- 사용자 정의 logo 메세지 출력을 위한 설정
- Server의 Port 수정



application.properties

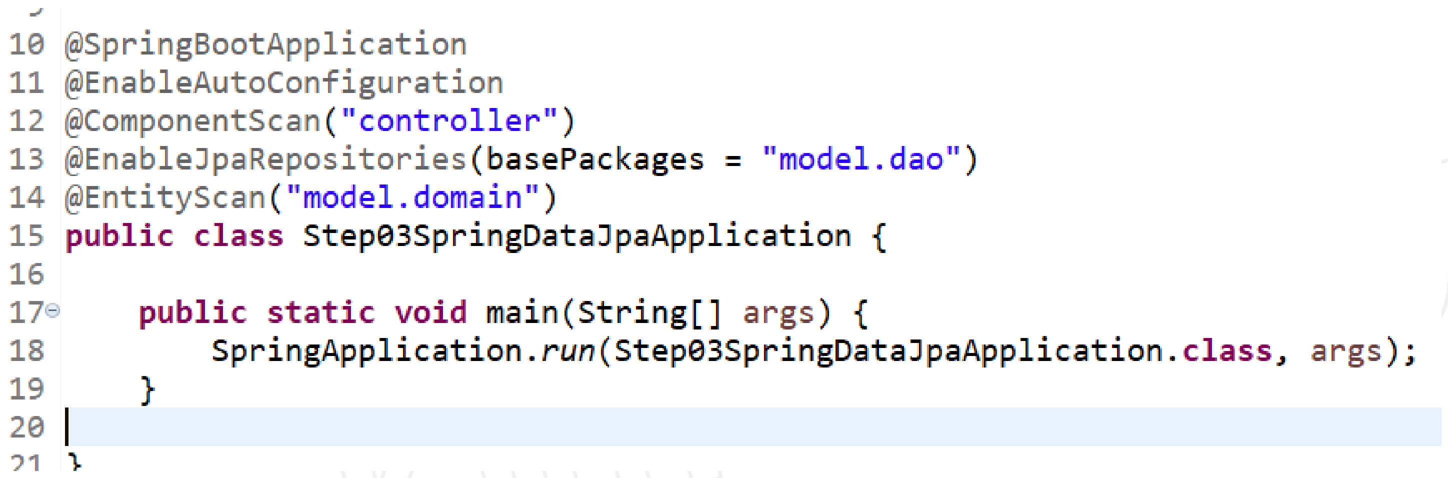
```
server.port=8000
```

```
step01_bootBasic [boot] [devto
  > Deployment Descriptor: step
  > Spring Elements
  > JAX-WS Web Services
  v Java Resources
    v src/main/java
      v pe.khk
        > ServletInitializer.java
        > Step01BootBasicAp
    v src/main/resources
        static
        templates
        application.properties
        banner.txt
    > src/test/java
```

```
***********************************

        사용자 정의 문구 구성이 가능합니다

***********************************
2019-09-30 19:18:44.963  INFO 1668 --- [  restartedMain] pe.khk.Step01BootBasicApp
2019-09-30 19:18:44.963  INFO 1668 --- [  restartedMain] pe.khk.Step01BootBasicApp
2019-09-30 19:18:44.995  INFO 1668 --- [  restartedMain] .e.DevToolsPropertyDefaul
2019-09-30 19:18:44.995  INFO 1668 --- [  restartedMain] .e.DevToolsPropertyDefaul
2019-09-30 19:18:45.464  INFO 1668 --- [  restartedMain] .s.d.r.c.RepositoryConfig
2019-09-30 19:18:45.480  INFO 1668 --- [  restartedMain] .s.d.r.c.RepositoryConfig
2019-09-30 19:18:45.699  INFO 1668 --- [  restartedMain] trationDelegate$BeanPostP
2019-09-30 19:18:45.902  INFO 1668 --- [  restartedMain] o.s.b.w.embedded.tomcat.T
2019-09-30 19:18:45.917  INFO 1668 --- [  restartedMain] o.apache.catalina.core.St
```

Spring Boot 주요 Annotation

Spring Boot 실행을 위한 Annotation



```
@SpringBootApplication
@EnableAutoConfiguration
@ComponentScan("controller")
@EnableJpaRepositories(basePackages = "model.dao")
@EntityScan("model.domain")
public class Step03SpringDataJpaApplication {

    public static void main(String[] args) {
        SpringApplication.run(Step03SpringDataJpaApplication.class, args);
    }

}
```

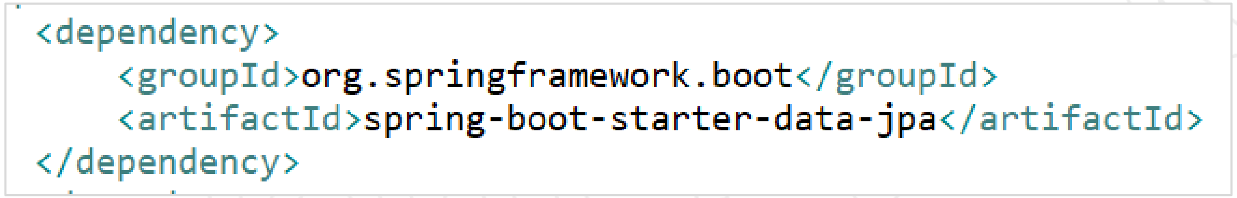
주요 Annotation

- @ComponentScan
  - @CompoentScan은 스프링을 하면 알다시피 @Compoent라는 어노테이이션이 붙어있는 class를 빈으로 등록한다.
  - (@Compoent, @Configuration, @Repository, @Service, @Controller, @RestController ..)
- @EnableAutoConfiguration
  - 중요한 것은 @EnableAutoConfiguration인데 @EnableAutoConfiguration은 스프링부트의 meta 파일을 읽어서, 미리 정의되어 있는 자바 설정 파일(@Configuration)들을 빈으로 등록하는 역할을 수행한다.
  - spring.factories 라는 스프링부트의 meta파일을 읽어들인다.

Spring Data JPA

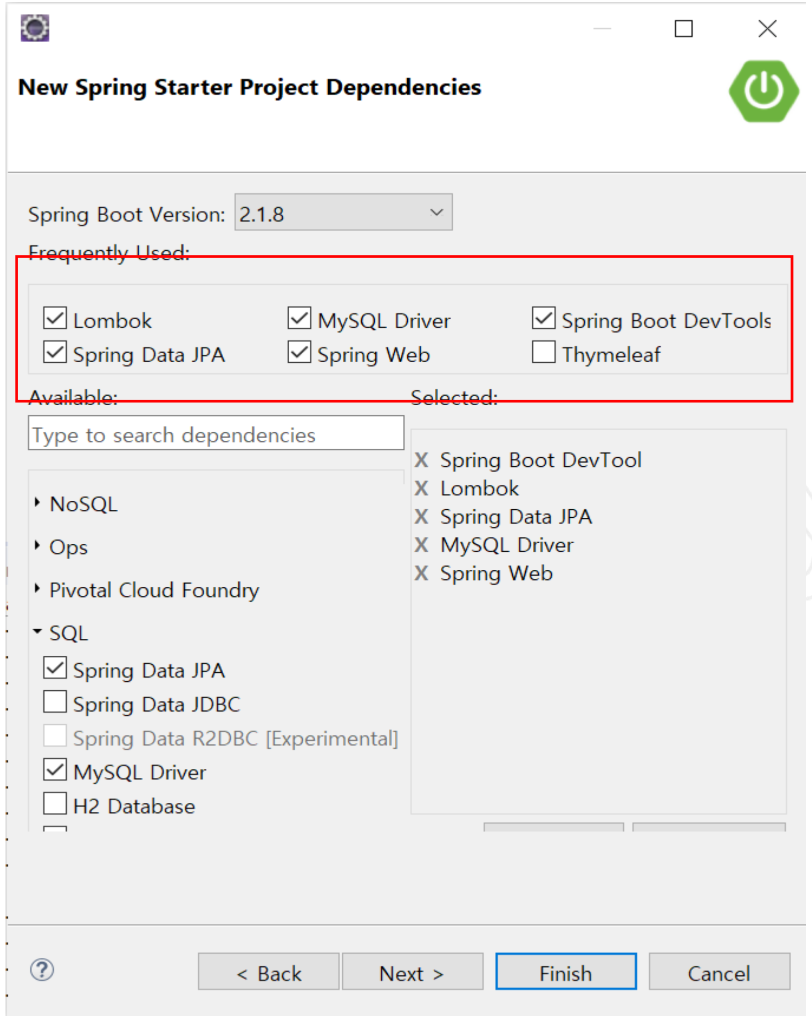
Spring Boot & JPA

- Spring Boot의 JSP start
  - JPA 연동에 필요할 library들과 복잡한 XML설정을 자동으로 처리
  - 장점 : 어려움 없이 JPA 관련 의존성과 XML 설정 처리가 가능



```xml
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

Spring Data JPA Project 생성 및 설정


New Spring Starter Project Dependencies

Spring Boot Version: 2.1.8

Frequently Used:

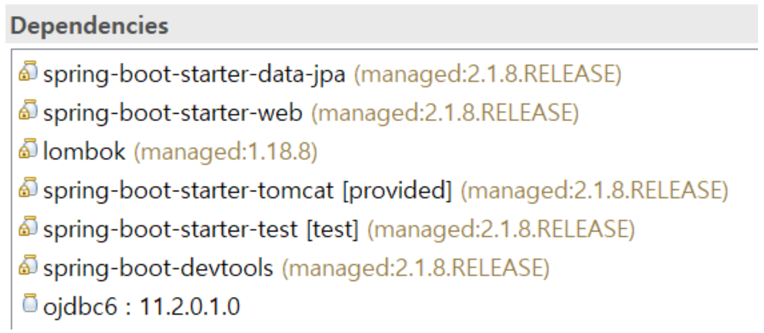
- [✓] Lombok
- [✓] MySQL Driver
- [✓] Spring Boot DevTools
- [✓] Spring Data JPA
- [✓] Spring Web
- [ ] Thymeleaf

Available: Type to search dependencies

- NoSQL
- Ops
- Pivotal Cloud Foundry
- SQL
  - [✓] Spring Data JPA
  - [ ] Spring Data JDBC
  - [ ] Spring Data R2DBC [Experimental]
  - [✓] MySQL Driver
  - [ ] H2 Database

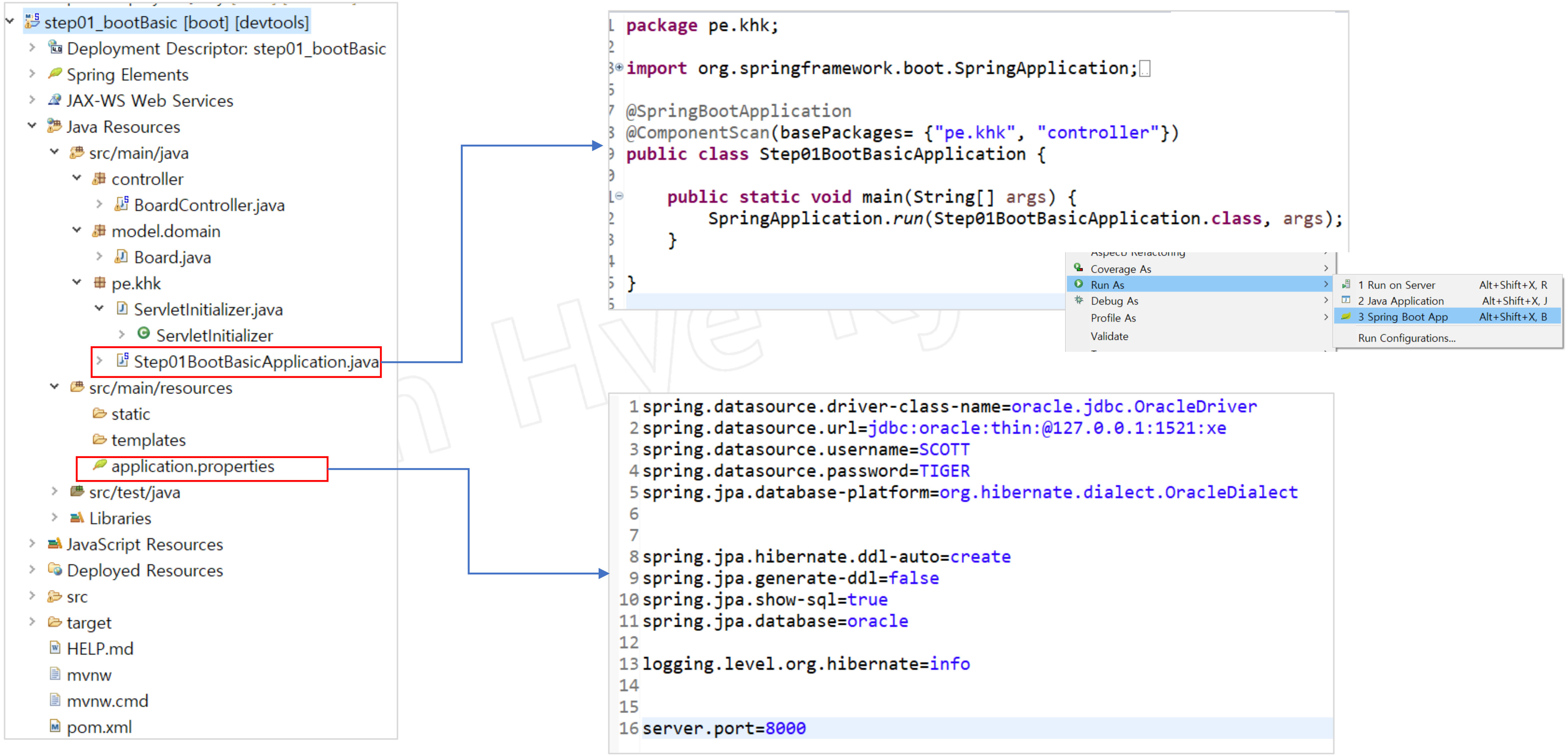
Selected:

- X Spring Boot DevTool
- X Lombok
- X Spring Data JPA
- X MySQL Driver
- X Spring Web


Dependencies

- spring-boot-starter-data-jpa (managed:2.1.8.RELEASE)
- spring-boot-starter-web (managed:2.1.8.RELEASE)
- lombok (managed:1.18.8)
- spring-boot-starter-tomcat [provided] (managed:2.1.8.RELEASE)
- spring-boot-starter-test [test] (managed:2.1.8.RELEASE)
- spring-boot-devtools (managed:2.1.8.RELEASE)
- ojdbc6 : 11.2.0.1.0

Spring Boot 기본 예제 1



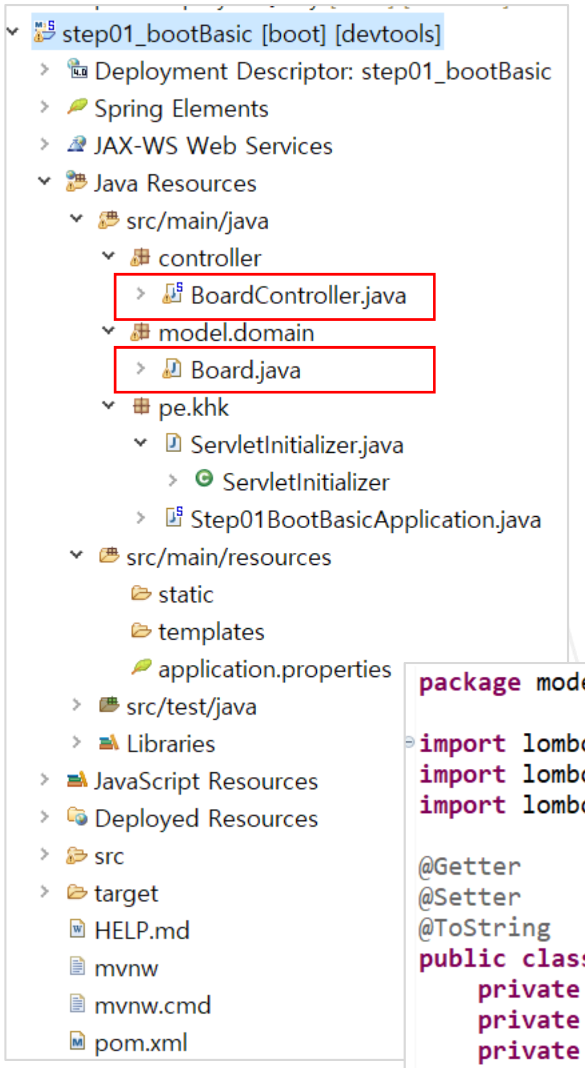
```
step01_bootBasic [boot] [devtools]
  Deployment Descriptor: step01_bootBasic
  Spring Elements
  JAX-WS Web Services
  Java Resources
    src/main/java
      controller
        BoardController.java
      model.domain
        Board.java
      pe.khk
        ServletInitializer.java
          ServletInitializer
        Step01BootBasicApplication.java
    src/main/resources
      static
      templates
      application.properties
    src/test/java
    Libraries
  JavaScript Resources
  Deployed Resources
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

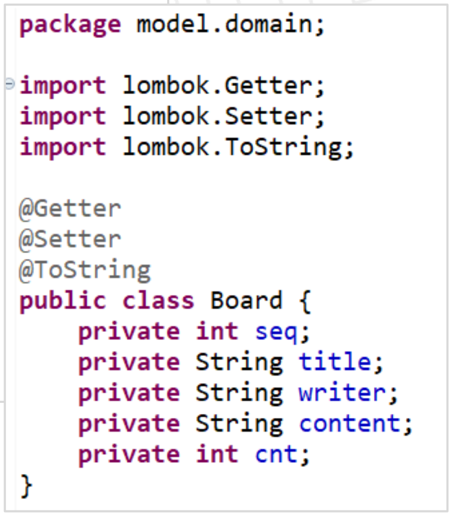
```java
package pe.khk;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
@ComponentScan(basePackages= {"pe.khk", "controller"})
public class Step01BootBasicApplication {

    public static void main(String[] args) {
        SpringApplication.run(Step01BootBasicApplication.class, args);
    }

}
```

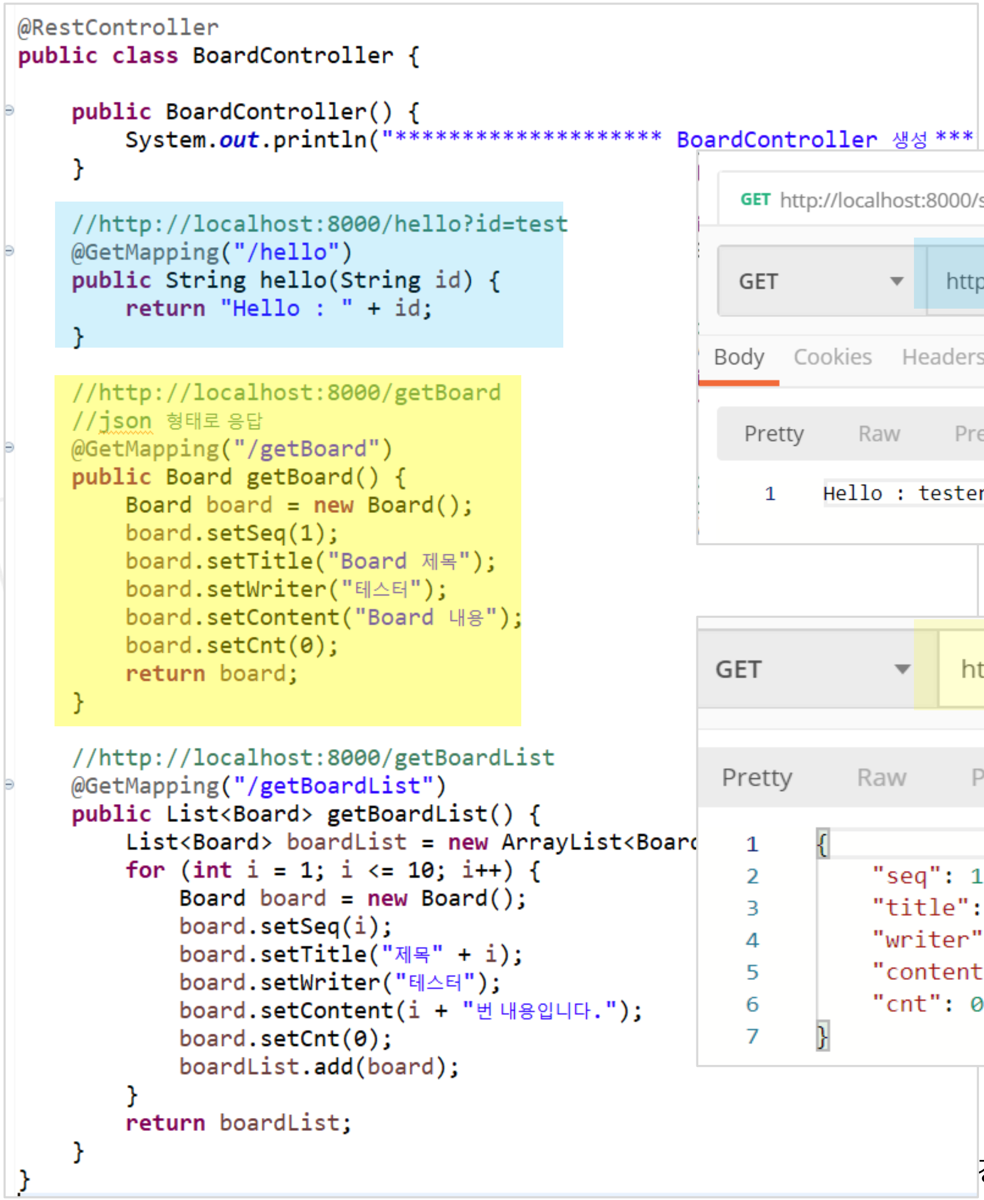
Coverage As
Run As
Debug As
Profile As
Validate
1 Run on Server Alt+Shift+X, R
2 Java Application Alt+Shift+X, J
3 Spring Boot App Alt+Shift+X, B
Run Configurations...

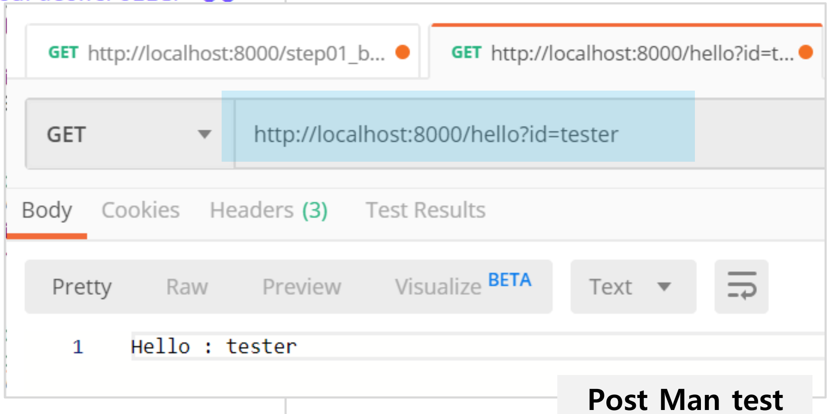
```
spring.datasource.driver-class-name=oracle.jdbc.OracleDriver
spring.datasource.url=jdbc:oracle:thin:@127.0.0.1:1521:xe
spring.datasource.username=SCOTT
spring.datasource.password=TIGER
spring.jpa.database-platform=org.hibernate.dialect.OracleDialect


spring.jpa.hibernate.ddl-auto=create
spring.jpa.generate-ddl=false
spring.jpa.show-sql=true
spring.jpa.database=oracle

logging.level.org.hibernate=info


server.port=8000
```

Spring Boot 기본 예제 1


```
step01_bootBasic [boot] [devtools]
  Deployment Descriptor: step01_bootBasic
  Spring Elements
  JAX-WS Web Services
  Java Resources
    src/main/java
      controller
        BoardController.java
      model.domain
        Board.java
      pe.khk
        ServletInitializer.java
          ServletInitializer
        Step01BootBasicApplication.java
    src/main/resources
      static
      templates
      application.properties
    src/test/java
    Libraries
  JavaScript Resources
  Deployed Resources
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

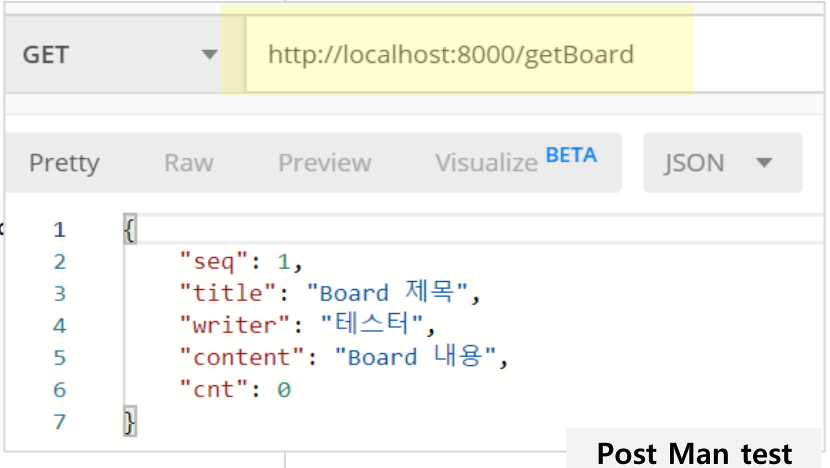

```java
package model.domain;

import lombok.Getter;
import lombok.Setter;
import lombok.ToString;

@Getter
@Setter
@ToString
public class Board {
    private int seq;
    private String title;
    private String writer;
    private String content;
    private int cnt;
}
```


```java
@RestController
public class BoardController {

    public BoardController() {
        System.out.println("******************** BoardController 생성 ***
    }

    //http://localhost:8000/hello?id=test
    @GetMapping("/hello")
    public String hello(String id) {
        return "Hello : " + id;
    }

    //http://localhost:8000/getBoard
    //json 형태로 응답
    @GetMapping("/getBoard")
    public Board getBoard() {
        Board board = new Board();
        board.setSeq(1);
        board.setTitle("Board 제목");
        board.setWriter("테스터");
        board.setContent("Board 내용");
        board.setCnt(0);
        return board;
    }

    //http://localhost:8000/getBoardList
    @GetMapping("/getBoardList")
    public List<Board> getBoardList() {
        List<Board> boardList = new ArrayList<Board
        for (int i = 1; i <= 10; i++) {
            Board board = new Board();
            board.setSeq(i);
            board.setTitle("제목" + i);
            board.setWriter("테스터");
            board.setContent(i + "번 내용입니다.");
            board.setCnt(0);
            boardList.add(board);
        }
        return boardList;
    }
}
```


GET http://localhost:8000/step01_b...
GET http://localhost:8000/hello?id=t...
GET
http://localhost:8000/hello?id=tester
Body
Cookies
Headers (3)
Test Results
Pretty
Raw
Preview
Visualize BETA
Text

```
Hello : tester
```

Post Man test

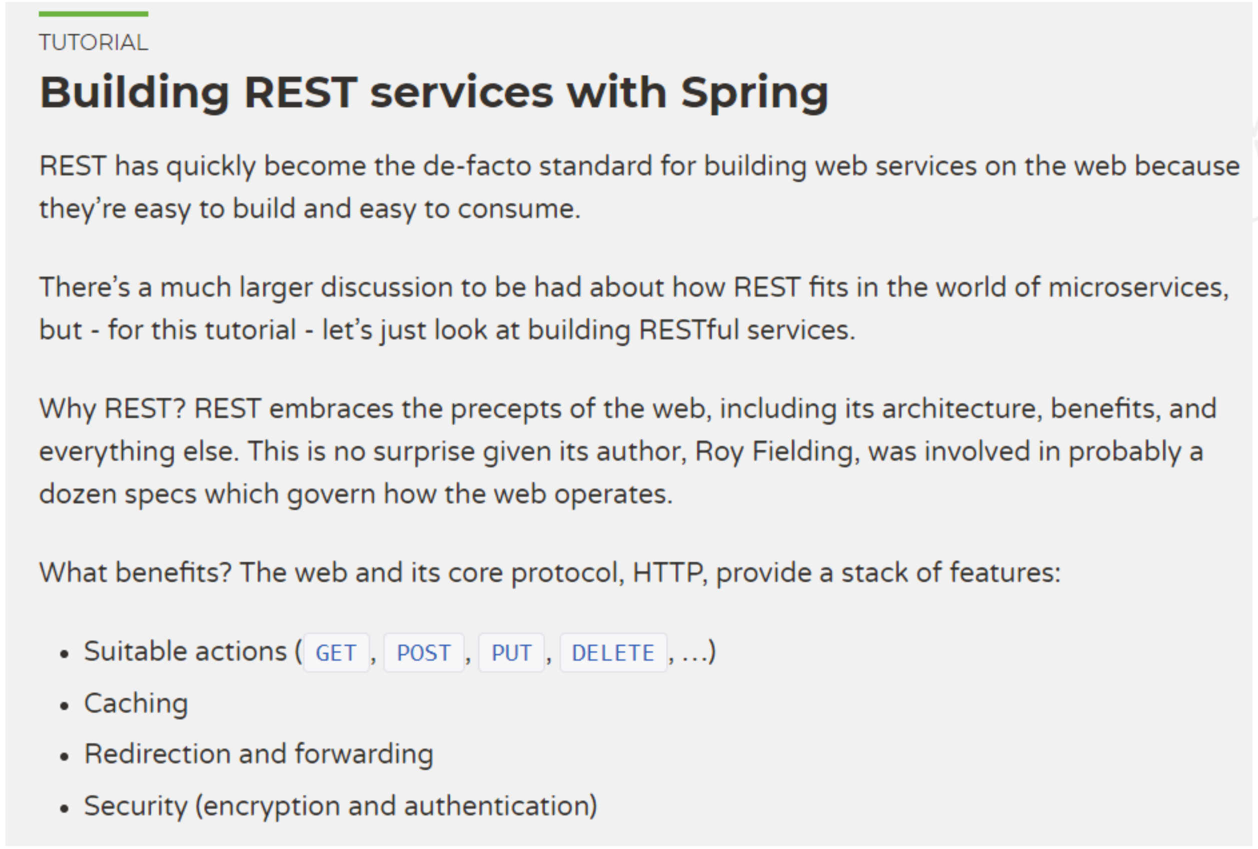


```
{
    "seq": 1,
    "title": "Board 제목",
    "writer": "테스터",
    "content": "Board 내용",
    "cnt": 0
}
```

Post Man test

Spring Boot 기본 예제 2

- Building REST services with Spring
  - https://spring.io/guides/tutorials/rest/



TUTORIAL

Building REST services with Spring

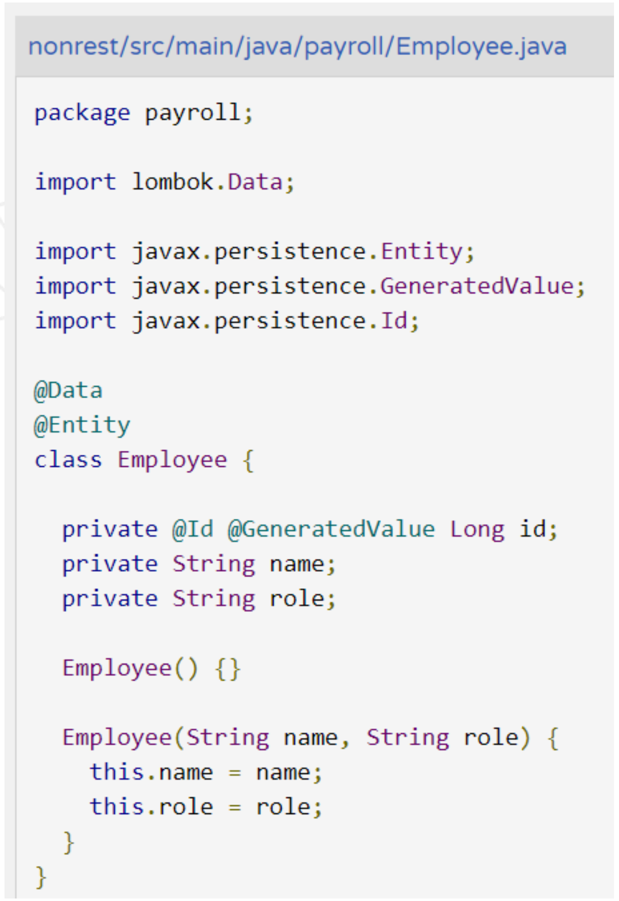
REST has quickly become the de-facto standard for building web services on the web because they're easy to build and easy to consume.

There's a much larger discussion to be had about how REST fits in the world of microservices, but - for this tutorial - let's just look at building RESTful services.

Why REST? REST embraces the precepts of the web, including its architecture, benefits, and everything else. This is no surprise given its author, Roy Fielding, was involved in probably a dozen specs which govern how the web operates.

What benefits? The web and its core protocol, HTTP, provide a stack of features:

- Suitable actions (`GET`, `POST`, `PUT`, `DELETE`, …)
- Caching
- Redirection and forwarding
- Security (encryption and authentication)



nonrest/src/main/java/payroll/Employee.java

```java
package payroll;

import lombok.Data;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;

@Data
@Entity
class Employee {

  private @Id @GeneratedValue Long id;
  private String name;
  private String role;

  Employee() {}

  Employee(String name, String role) {
    this.name = name;
    this.role = role;
  }
}
```