**SpringBoot란?**

**Spring Boot는 Java를 기반으로 한 웹 어플리케이션 프레임워크**이다.  
Spring Boot가 등장하기 전 Spring 프레임워크가 먼저 등장했는데 Spring의 초기 환경 설정시 시간이 많이 할애되는 문제를 해결하고자 등장한 프레임워크가 Spring Boot이다.

스프링부트는 메이븐이나 그레이들의 dependency에 starter 라이브러리만 작성한다면 초기 셋팅에 필요한 라이브러리들을 모두 세팅해주게 된다. 이를 통해 복잡했던 초기 환경 설정 과정을 간소화시킬 수 있게 되었다.  
또 starter 라이브러리는 dependency에 존재하는 라이브러리들의 버전 관리를 자동적으로 해줘 버전이 맞지 않아 발생하는 오류를 방지할 수 있다.

**Maven과 Gradle은 무엇일까?**

초기 개발 환경을 설정할 때 메이븐의 pom.xml이나 그레이들의 build.gradle 파일을 통해 스프링과 스프링부트의 프로젝트를 관리할 수 있다.

그럼 둘의 차이는 무엇일까?

**Maven**은 자바용 프로젝트 관리 툴이기 때문에 자바에서만 사용 가능한다.  
하지만**Gradle** 은 Java 이외에도 C++, Python 등 다양한 언어를 지원한다. 현재는 메이븐의 비중이 더 크긴 하지만 그레이들이 메이븐에 비해 가독성이 더 좋고, 성능면에서도 더 빠르게 작동해기 때문에 대형 프로젝트의 경우 그레이들을 사용하는 경우가 많아지고 있다.

**내장서버**

스프링 부트의 두 번째 특징은 **톰캣과 같은 내장 서버가 존재**한다는 것이다.  
스프링 프레임워크의 경우 톰캣을 직접 설치해 프로젝트 내에서 서버를 설정해주고, 버전 관리도 함께 해 주어야 했다.  
하지만 스프링 부트는 그럴 필요 없이 톰캣 내장 서버가 존재하여 설치와 버전 관리를 신경 쓸 필요가 없어졌다.

**자동의존성관리**

마지막으로 스프링의 경우 servlet-context, root-context와 같은 xml 파일을 직접 작성해서 웹과 관련된 설정이나 스프링 프로젝트 내 객체 의존성을 관리해줘야 했지만 스프링부트에서는 이럴 필요 없이 자동으로 의존성 관리가 가능해졌다.

**개발환경세팅**



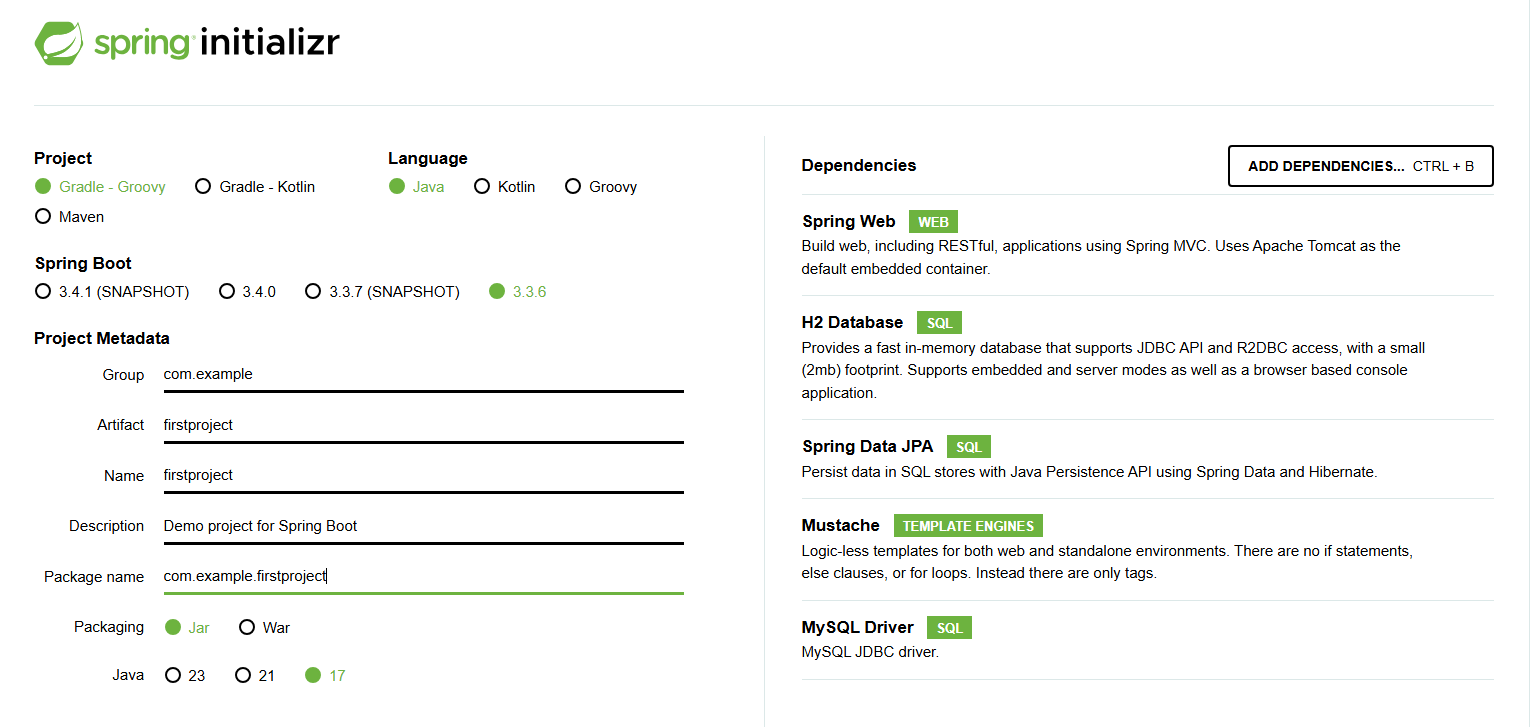
**JDK**

java version "1.8.0\_351"

**IDE 인텔리제이**

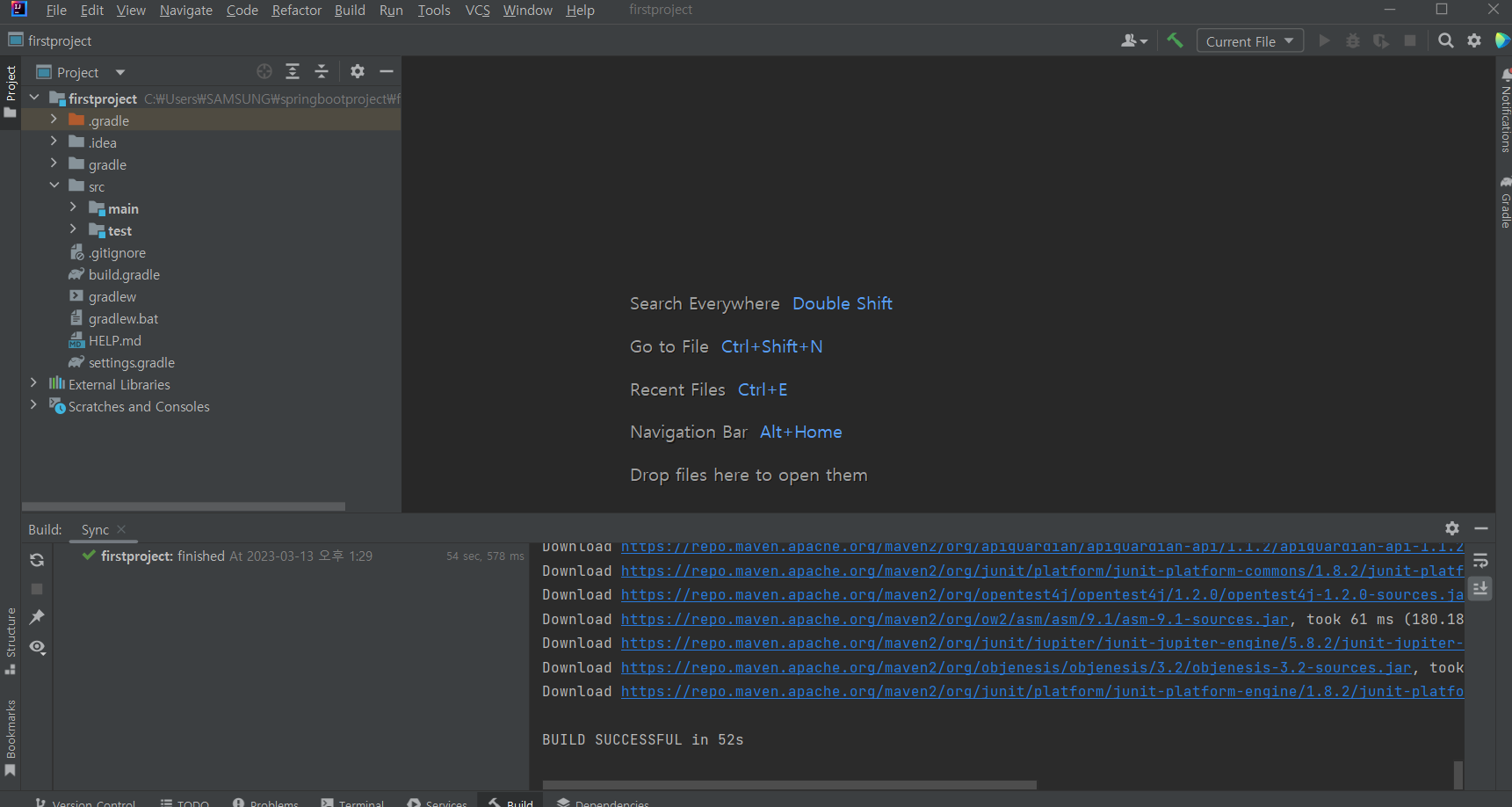
[인텔리제이다운로드](https://www.jetbrains.com/ko-kr/idea/) Community 버전 설치  


**springboot 프로젝트 생성**

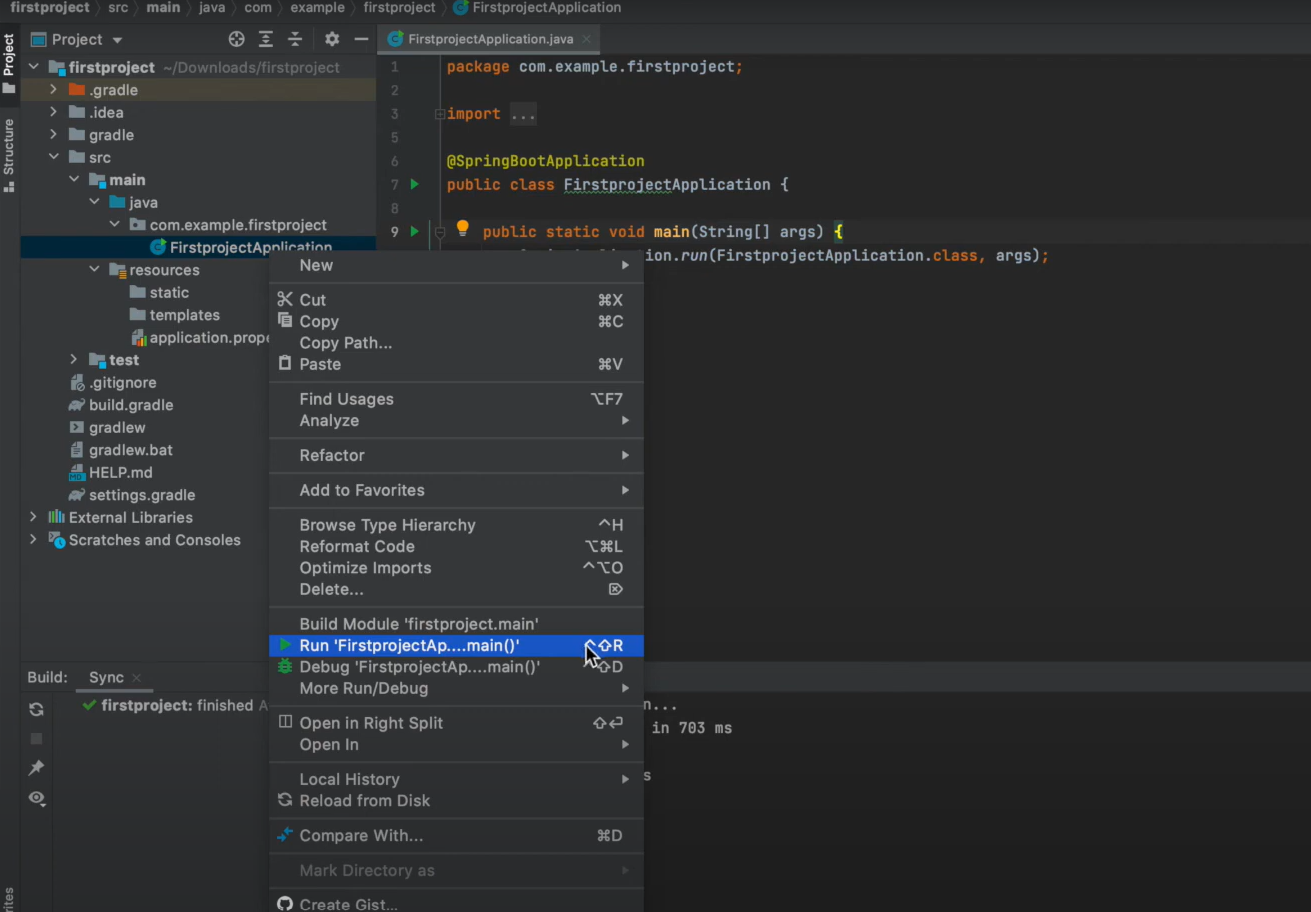
[스프링프로젝트생성](https://start.spring.io/)  


Spring Initializr에서 다운받은 파일  
인텔리제이에서 열기

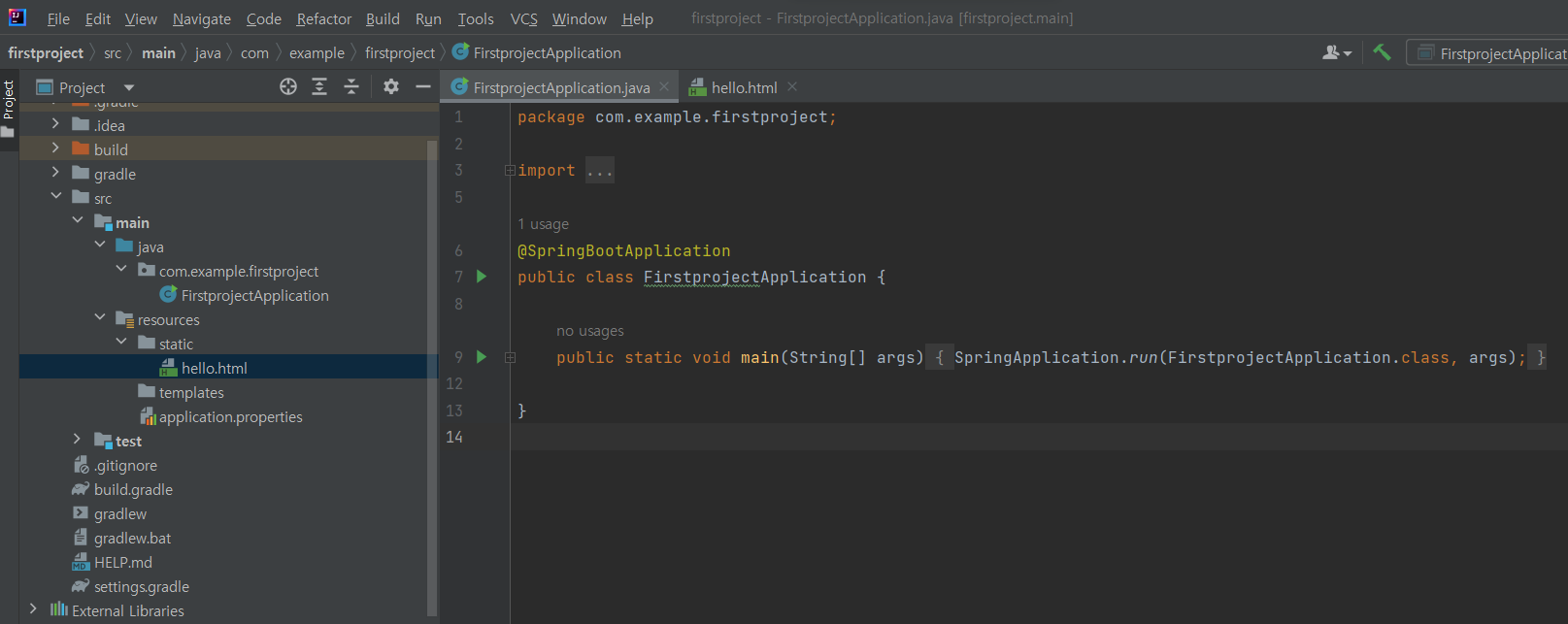
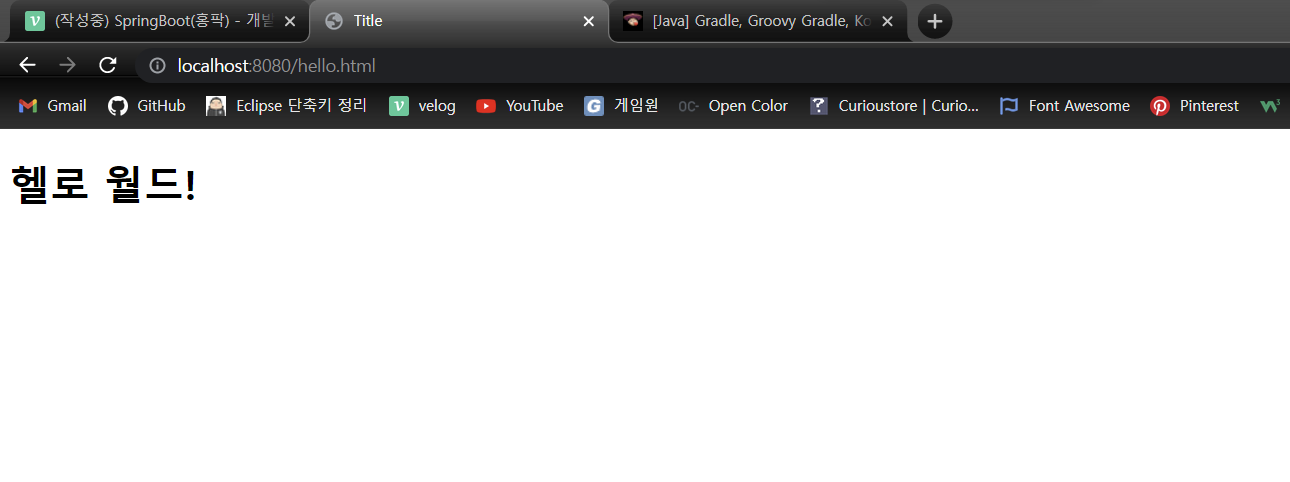
✨ Spring Initializr 기본 설정인  
Spring boot 버전이 3.0 버전은 JDK 17이상이라 오류 발생!!  
2.7버전으로 다시 받았다. 👌

빌드 성공

**헬로월드 띄워보기**

main 메서드 실행  


**html 파일 만들기**

main-resources-static에 hello.html 파일 생성  
  
<http://localhost:8080/hello.html> 로 접속하면  
헬로월드 띄우기 성공👍  


**build.gradle**

Gradle로 작성된 스프링 부트 설정은 build.gradle 에 작성된다. 여기서 스프링 부트 및 JDK 버전이 지정된다. 자신의 개발환경에 맞게 값을 지정하면 된다.

plugins {

id 'org.springframework.boot' version '3.3.6' // 스프링 부트 버전(변경 가능)

id 'io.spring.dependency-management' version '1.0.11.RELEASE'

id 'java'

}

group = 'com.example'

version = '0.0.1-SNAPSHOT'

sourceCompatibility = '17' // JDK 버전(각자 설치된 자바 버전에 맞게 지정. Java 8은 1.8)

repositories {

mavenCentral()

}

dependencies {

implementation 'org.springframework.boot:spring-boot-starter-data-jpa'

implementation 'org.springframework.boot:spring-boot-starter-mustache'

implementation 'org.springframework.boot:spring-boot-starter-web'

runtimeOnly 'com.h2database:h2'

testImplementation 'org.springframework.boot:spring-boot-starter-test'

}

test {

useJUnitPlatform()

}

✨ 내 build.gradle

plugins **{** id 'java'  
 id 'org.springframework.boot' version '3.3.6'  
 id 'io.spring.dependency-management' version '1.1.6'  
**}**group = 'com.example'  
version = '0.0.1-SNAPSHOT'  
  
java **{** toolchain **{** languageVersion = JavaLanguageVersion.*of*(17)  
 **}  
}**repositories **{** mavenCentral()  
**}**dependencies **{** implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
 implementation 'org.springframework.boot:spring-boot-starter-mustache'  
 implementation 'org.springframework.boot:spring-boot-starter-web'  
 //runtimeOnly 'com.h2database:h2'  
 runtimeOnly 'com.mysql:mysql-connector-j'  
 testImplementation 'org.springframework.boot:spring-boot-starter-test'  
 testRuntimeOnly 'org.junit.platform:junit-platform-launcher'  
**}**tasks.named('test') **{** useJUnitPlatform()  
**}**