# 강의 링크

<https://www.inflearn.com/course/%EC%8A%A4%ED%94%84%EB%A7%81-%EC%9E%85%EB%AC%B8-%EC%8A%A4%ED%94%84%EB%A7%81%EB%B6%80%ED%8A%B8>

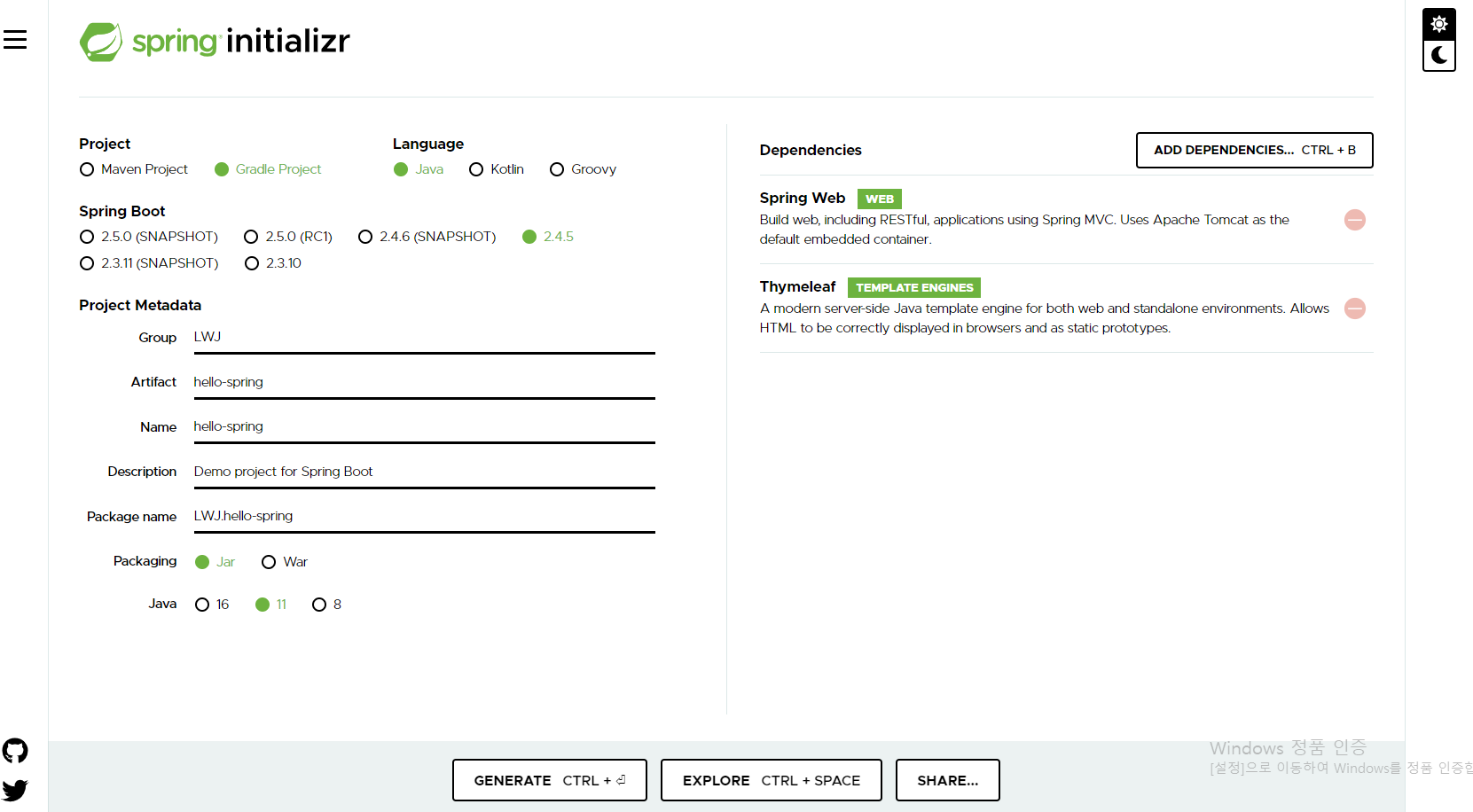
(김영한님)

# 환경 설정

1. java 11 설치 (oracle 가입 필요)
2. IntelliJ 설치
   1. 프로젝트 열고 빌드 시 - 환경설정에서 gradle 검색해보면 build Tools 아래에 Gradle 나올 것, 여기서 Build and run using, Run tests using 모두 IntelliJ IDEA 로 변경해야 빌드 빨리될 것.

# 프로젝트 생성

1. start.spring.io



1. project : Gradle 선택
2. Spring Boot

SNAPSHOT 은 아직 만들고있는 버전

다른것도 아직 정식 릴리즈된 버전이 아님

아무것도 안적혀있는 최신으로.

1. Project Metadata

Group 에 보통 기업 명

Artifact 는 빌드되어 나오는 결과물

1. Dependencies

ADD DEPENDENCIES 클릭해서

WEB - web 검색 후 SPRING WEB (tomcat 내장)

TEMPLETE ENGINES - Thymeleaf (html 관련)

1. GENERATE 클릭하면 압축파일 받아지고, 이걸 압축 풀면됨
2. IntelliJ

open 눌러서 다운받은 폴더 가서 build.gradle 찾아서 이걸 오픈 (Open as Project)

처음 오픈하면 이것저것 다운로드 받을거 (네트워크 연결 필요)

1. 실행

src/main/java 아래에 코드들 있을거.

src/test 는 테스트코드

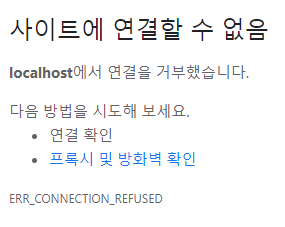
src/main/java 아래에 HelloSpringApplication 에 main 있을거고, 실행하면

~embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)

이런거 나올거고, 인터넷으로 localhost:8080 들어가보면



이런거 나올 것. 아무것도 없으므로 이렇게 나오는게 성공.



이게 실패

1. gradle

build.gradle 에 보면 dependencies 나옴.

첫 화면 생성

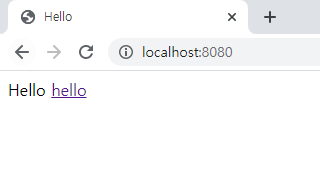
1. Welcome Page

resources/static/index.html 생성

<index.html>

|  |
| --- |
| <!DOCTYPE HTML>  <html>  <head>  <title>Hello</title>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  </head>  <body>  Hello  <a href="/hello">hello</a>  </body>  </html> |

후 실행하면 localhost:8080 에서 아래와 같이 뜰 것.



스프링 부트가 제공하는 welcome page 기능.

resource/static/index.html 을 만들면 이걸 참조해서 뿌려줌.

이러한 기능들 찾아서 할 수 있음. (spring.io 들어가서 project/spring boot/learn 의 reference doc 들어가면 spring boot feature 에서 welcome 또는 index.heml 검색)

1. 템플릿 엔진 (thymeleaf) / controller

controller package 하나 만들고, 아래에 HelloController 생성

|  |
| --- |
| package LWJ.hellospring.controller;  import org.springframework.stereotype.Controller;  import org.springframework.ui.Model;  import org.springframework.web.bind.annotation.GetMapping;  @Controller  public class HelloController {  @GetMapping("hello") // localhost:8080/hello 에서의 hello 를 의미함.  public String hello(Model model) {  model.addAttribute("data", "hello!!");  return "hello"; // templates/hello.html 를 찾아감.  }  } |

@Controller -> 이거 annotation 적어줘야함

@GetMapping -> 이게 사이트 주소 매핑해주는 역할

@GetMapping(“asdf”) 로 하면 localhost:8080/asdf 와 연결됨

return “hello” 는 templetes/hello.html 을 찾을 수 있도록 함.

<참고 그림>

