Spring – 자바의 프레임 워크

Intellij를 기반으로 작성.

Start.spring.io 에서 기본적인 프로젝트를 다운받는다.(spring boot)

스프링기반의 프로젝트를 만들어주는 사이트

Maven, gradle – 필요한 라이브러리를 가져와주고, 빌드 라이프? 사이클을 관리해주는 툴

과거에는 메이븐이었지만 요즘에는 그레이들을 선호

그룹명은 보통 기업 도메인. 우리는 아무거나 적어도 된다. 예제는 hello

Artifact는 빌드된 결과물(프로젝트명 같은것). Hello-spring

Dependencies는 어떠한 라이브러리를 사용할 것인지를 묻는 것.

우리는 web프로젝트 이므로 spring web을 선택한다.

또, thymeleaf라는 템플릿 엔진을 사용한다.

템플릿 엔진 - HTML 코드에서 고정적으로 사용되는 부분은 템플릿으로 만들어두고 동적으로 생성되는 부분만 템플릿 특정 장소에 끼워넣는 방식으로 동작할 수 있도록 해준다.

이후 generate하면 프로젝트를 다운받는다.

Build.grdadle을 프로젝트로 열기로 연다.

Main->java->hello-spring의 hellospringApplication.java의 main메소드를 빌드하면 8080포트로 서버가 열린다.

접속에 성공하면 기본적인 설정이 끝이난다.

Spring은 내장 tomcat이 있어서 자동으로 서버를 열게 된다.

intellij설정에서 빌드를 gradle이 아닌 intellij를 통해 빌드되도록 수정한다.

기본설정 끝

System.print.out 보다는 log출력을 할 것.

Src- main-java-hello-spring밑에 controller라는 패키지를 만든다.

패키지 안에 HelloController.java파일을 만들고

@Controller  
public class HelloController {  
 @GetMapping("hello")// /hello라는 호출이 들어오면 이하의 메소드 실행  
 public String hello(Model model){  
 model.addAttribute("data","hello");  
 return "hello";  
 }  
}

이렇게 수정한다.

여기서의 controller와 model은 MVC모델의 그 controller와 model이다.

GetMapping의 Get은 Get과 Post방식이다.

이 코드는 Get방식으로 /hello가 넘어오면 Model객체에 Data키에 hello라는 value를 생성하고

hello라는 템플릿 페이지를 return 한다. 해당 페이지는 아래에서 설명한다.

src – main – resources – templates에 hello.html파일을 만든다.

<!DOCTYPE HTML>  
<html xmlns:th="http://www.thymeleaf.org">  
<head>  
 <title>Hello</title>  
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
</head>  
<body>  
<p th:text="'안녕하세요. ' + ${data}" >안녕하세요. 손님</p>  
</body>  
</html>

여기서 th는 thymeleaf이며

<html xmlns:th="http://www.thymeleaf.org">

는 타임리프 엔진이 템플릿 엔진이 선언되어 있다.

따라서 타임리프 문법을 사용할 수 있다.