Spring

-Gradle 이나 Maven 같은 툴들은 의존관계를 관리 해준다.

한 개의 라이브러리를 Dependencies 하면 모든 것을 필요한 라이브러리를 전부다 가져온다.

-실무에서는 System.out.println 말고 Log를 사용한다.

-스프링 부트 라이브러리

- spring-boot-starter-web

spring-boot-starter-tomcat: 톰캣 (웹서버)

spring-webmvc: 스프링 웹 MVC

-spring-boot-starter-thymeleaf: 타임리프 템플릿 엔진(View)

-spring-boot-starter(공통): 스프링 부트 + 스프링 코어 + 로깅

spring-boot

spring-core

spring-boot-starter-logging

logback, slf4j

-테스트 라이브러리

spring-boot-starter-test

junit: 테스트 프레임워크

mockito: 목 라이브러리

assertj: 테스트 코드를 좀 더 편하게 작성하게 도와주는 라이브러리

spring-test: 스프링 통합 테스트 지원

-스프링 웹 개발 기초

-정적 컨텐츠 : 서버에서 하는 일 없이 파일을 그대로 웹 브라우저에 내려 주는 것 .

예) static 폴더에 hello-static.html 을 생성 해주고

<http://localhost:8080/hello-static.html> 을 들어가게 되면 그대로 반환이 됩니다.

컨트롤러에 먼저 매핑된 것 이 있는 것이 찾은 후 없을 시 static 폴더에서 찾는 것 입니다.

-MVC와 템플릿 엔진 : 서버에서 변형에서 HTML을 바꿔서 내려 주는 것

MVC : Model , View , Controller

viewResolver가 temlpaters/hello-template.html 을 처리를 해준다.

-API JSON 데이터 포멧으로 또는 XML 포멧으로 내려준다 .

객체로 주면 Default 로 Json 방식으로 준다.

@ResponseBody 가 붙어 있지않으면 viewResolver에게

@ResponseBody이면 HttpMessageConverter ( JsonCOnverter) :객체 (StringConverter) :단순문자

컨트롤러: 웹 MVC의 컨트롤러 역할

서비스: 핵심 비즈니스 로직 구현

리포지토리: 데이터베이스에 접근, 도메인 객체를 DB에 저장하고 관리

도메인: 비즈니스 도메인 객체, 예) 회원, 주문, 쿠폰 등등 주로 데이터베이스에 저장하고 관리됨

Optional<> = null 일수도 있기 때문에 감싸서 반환 자바8에 있는 기능

TestCase를 작성할때는 순서와 상관없이 작동 할 수 있도록 해야한다 .

검증 할 수 있는 틀을 먼저 만들고 구현클래스를 작성 테스트 주도 개발 (TDD)

Spring이 관리를 하게 되면 스프링 컨테이너에 등록하게 된다 .

만약에 new 생성자로 하게 되면 다른컨트롤러에서 사용하기위해서는 스프링 컨테이너에게 등록을 하게되면 한 개만 등록되기 떄문에 사용한다.

@Autowired 를 하게되면

생성자에 붙어있으면 스프링컨테이너에 있는것과 연결을 시켜준다.

스프링 빈을 등록하는 2가지방법

컴포넌트 스캔과 자동 의존관계 설정 @Component … 등등 @레파지토리

자바코드로 직접 스프링빈 등록하기

\*싱글톤 으로 등록한다.(유일하게 하나만 등록해서 공유한다.) 특별한 경우는 제외한다 .

@Component 에노테이션이 있으면 스프링 빈으로 자동 등록된다.

@Controller: 컨트롤러가 스프링 빈으로 자동 등록된 이유도 컴포넌트 스캔 떄문이다.

하위 패키지만 컴포넌트 스캐닝 대상이 된다 .

-