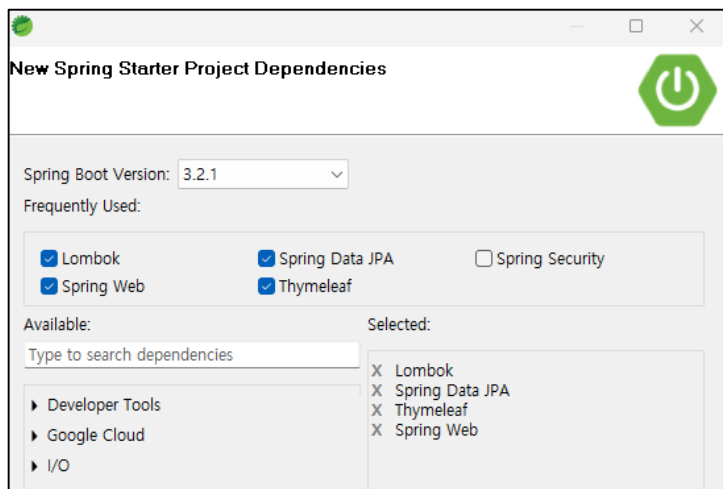


시험준비

시험 진행을 위해 새로운 프로젝트를 생성하세요.

라이브러리는 'Lombok, Spring Data JPA, Thymeleaf, Spring Web'를 선택하세요.

데이터베이스를 연결하기 위해 마리아DB 드라이버와 접속 정보를 추가하세요.



```
build.gradle
20 }
21
22 dependencies {
23     implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
24     implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
25     implementation 'org.springframework.boot:spring-boot-starter-web'
26     compileOnly 'org.projectlombok:lombok'
27     annotationProcessor 'org.projectlombok:lombok'
28     providedRuntime 'org.springframework.boot:spring-boot-starter-tomcat'
29     testImplementation 'org.springframework.boot:spring-boot-starter-test'
30     implementation group: 'org.mariadb.jdbc', name: 'mariadb-java-client', version: '3.1.2'
31 }
```

'org.mariadb.jdbc', name: 'mariadb-java-client',

```
application.properties
1 spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
2 spring.datasource.url=jdbc:mariadb://localhost:3306/bootex
3 spring.datasource.username=bootuser
4 spring.datasource.password=bootuser
5
6 spring.jpa.hibernate.ddl-auto=update
7 spring.jpa.properties.hibernate.format_sql=true
8 spring.jpa.show-sql=true
9
10 spring.thymeleaf.cache=false
```

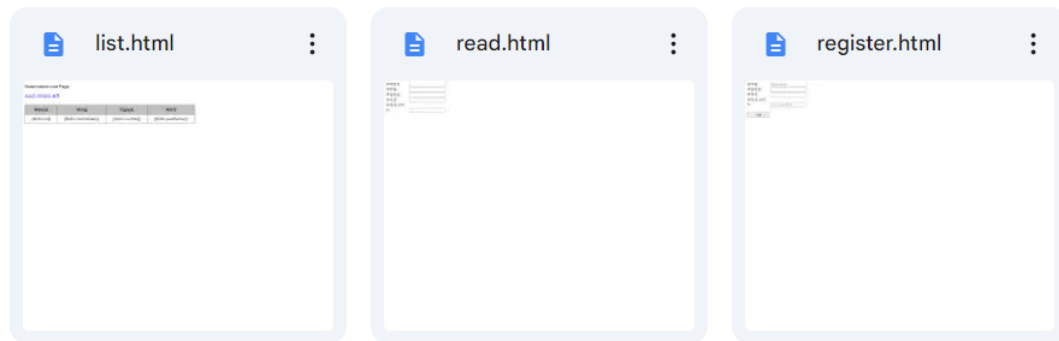
시험준비

미리 준비된 정적 페이지(등록, 목록조회, 상세 페이지)를 다운받으세요.

3개의 HTML 파일 를 활용하여 시험을 진행하세요.

그리고 동적으로 변경해야 하는 부분은 타임리프를 사용하여 처리하세요.






파일








Reservation 엔티티 클래스를 만들고, 해당 엔티티에 대한 테이블을 생성하세요. (20점)

ReservationRepository 인터페이스를 만들고, 이 리파지토리를 사용하여 데이터 조회, 추가, 수정, 삭제 기능을 테스트하세요.

tbl_reservation 테이블

#	컬럼명	Data Type	Not Null	Auto Increment	Key
1	 no	int(11)	[v]	[v]	PRI
2	 check_in_date	date	[v]	[]	
3	 guest_name	varchar(20)	[v]	[]	
4	 guest_phone	varchar(20)	[]	[]	
5	 room_no	int(11)	[v]	[]	

tbl_reservation 테이블에 저장된 데이터

	 no	 check_in_date	 guest_name	 guest_phone	 room_no
1	1	2024-09-05	둘리	010-1111-2222	201
2	2	2024-09-05	또치	010-3333-4444	202
3	3	2024-09-10	도우너	[NULL]	201

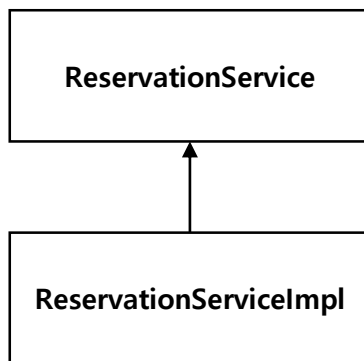
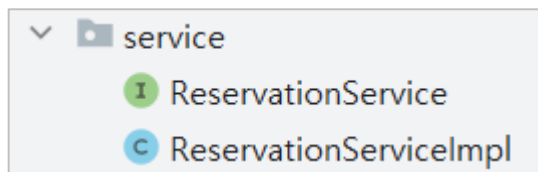
문항2

ReservationService 인터페이스와 ReservationServiceImpl 클래스를 생성하세요. (20점)

해당 서비스는 DTO와 Entity간의 변환, 등록, 목록 조회, 상세 조회 기능을 가지고 있으며, 수정 및 삭제 기능은 없습니다.

이 서비스를 사용하여 조회 및 추가 기능을 테스트하세요.

서비스 계층 구조



단위 테스트

```
@Test
public void 예약정보등록() {

}

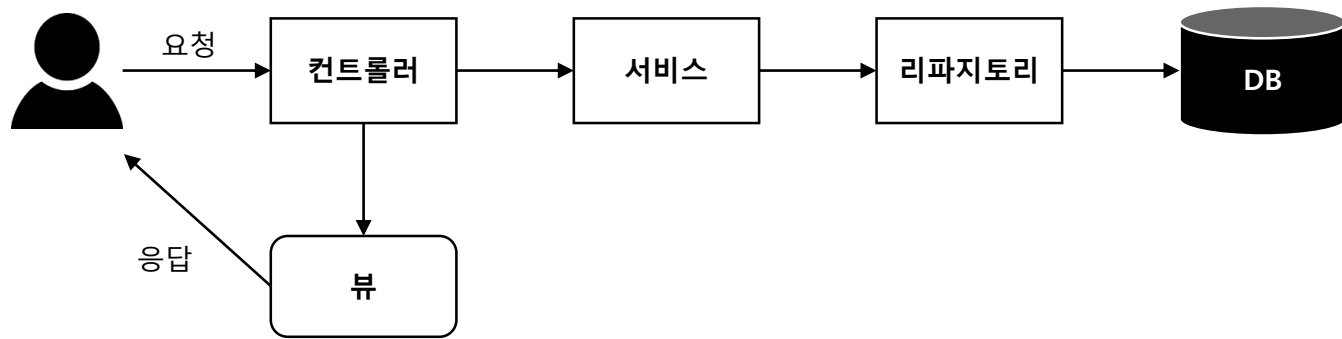
@Test
public void 예약정보목록조회() {

}

@Test
public void 예약정보단건조회() {

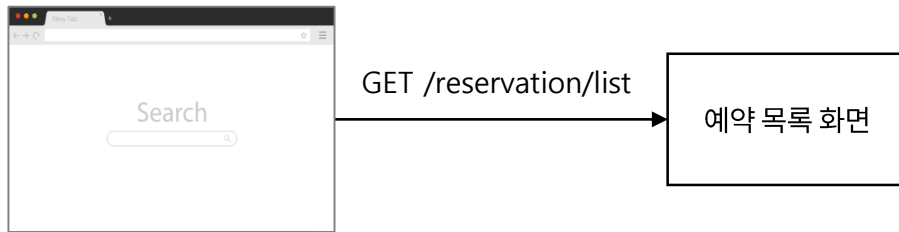
}
```

Spring MVC 패턴을 사용하여 예약정보의 등록, 목록조회, 상세조회 기능을 구현하세요.



예약 목록 조회 기능을 구현하세요. (20점)

목록 조회 흐름



목록 조회 화면

Reservation List Page

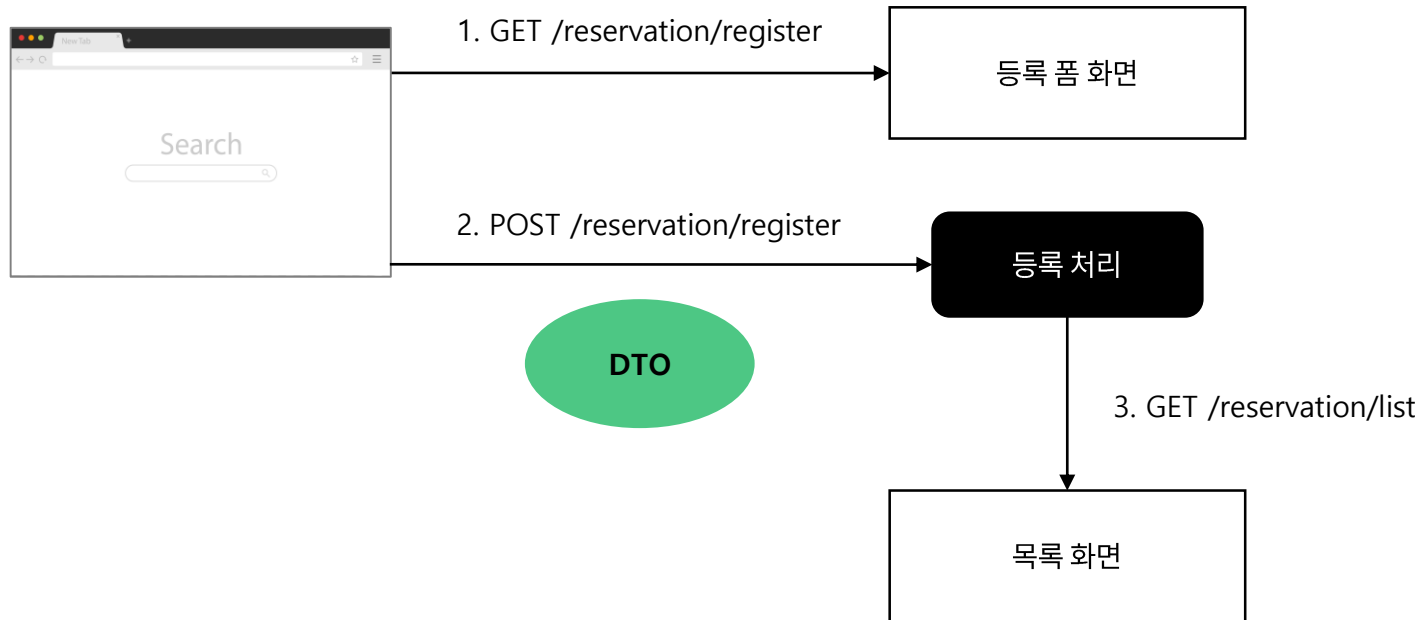
[새로운 예약정보 등록](#)

예약번호	예약일	객실번호	예약자
1	2024-09-05	201	둘리
2	2024-09-05	202	또치
3	2024-09-10	201	도우너

문항4

예약 정보 등록 기능을 구현하세요. (20점)
등록 처리가 끝나면 목록화면으로 이동합니다.

등록 처리 흐름



등록 화면

클릭

Reservation List Page

[새로운 예약정보 등록](#)

예약번호	예약일	객실번호	예약자
1	2024-09-05	201	둘리
2	2024-09-05	202	또치
3	2024-09-10	201	도우너



클릭

Reservation Register Page

예약일

객실번호

예약자

예약자 연락처

[전송](#)



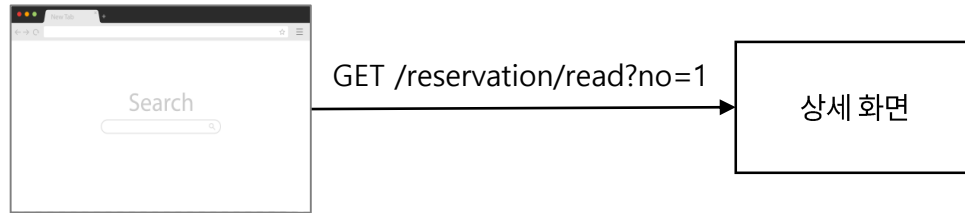
Reservation List Page

[새로운 예약정보 등록](#)

예약번호	예약일	객실번호	예약자
1	2024-09-05	201	둘리
2	2024-09-05	202	또치
3	2024-09-10	201	도우너
4	2024-09-29	204	고길동

예약 상세 조회 기능을 구현하세요. (20점)

상세 조회 흐름



상세 조회 화면

Reservation List Page

[새로운 예약정보 등록](#)

예약번호	예약일	객실번호	예약자
1	2024-09-05	201	둘리
2	2024-09-05	202	또치
3	2024-09-10	201	도우너
4	2024-09-29	204	고길동

클릭



Reservation Read Page

예약번호

예약일

객실번호

예약자

예약자 연락처

결과 제출

프로젝트 구현이 완료되면, GitHub에 저장소를 생성하여 코드를 업로드하세요.
그리고 GitHub 주소를 시험지에 작성하여 제출하세요.

Name
..
gradle/wrapper
src
.gitignore
build.gradle
gradlew
gradlew.bat
settings.gradle



https://github.com/imjiveon/exam.git	
답안 :	