

my sql를 활용하는 마지막

2025년 상반기 K-디지털 트레이닝

오늘은 어제 배운 이론을 바탕으로 어플리케이션을 만든는

JDBC 프로그래밍 - Travel

[KB] IT's Your Life

초기 데이터를 구축하는 방법

하나씩 축적해나가는 것.

오픈소스를 이용API. 데이터를 추출.

사이트의 데이털를 컬링(긁어오기)

특정 기관에서 데이터를 파일로 제공함. 고것을 활용. 특히 공공기관예) 한국관광공사 100대 여행지 정보 - travle.csv, travel-image.zip 파일이 필요.

오늘 실습 2단계 파일 임포트, 초기화단계 어플리케이션 개발



🗸 프로젝트 개요

엑셀로 열면 단순 텍스트파일로 표현한다. 그래서 vscode로 열어주자

○ 관광지 정보를 담고 있는 travel.csv 파일을 데이터베이스로 임포트

	Α	В	С	D	Е	F	G	Н
1	no	district	title	description	address	phone		
2	1	수도권	인천 개항장역사문	1883년 인천 개항	인천광역시 중구	인천 중구 문화유신	팀 032-760)-6448
3	2	수도권	인천 월미문화의거	인천의 대표적인	인천광역시 중구	032-764-0842		
4	3	수도권	인천 연안부두해양	서해 도서로 운항	인천광역시 중구	032-777-1330		
5	4	수도권	인천 하나개해수욕	하나개해수욕장은	인천광역시 중구 -	032-751-8833		
6	5	수도권	양평 세미원	세미원 연꽃박물관	경기도 양평군 양	031-775-1835		
7	6	수도권	시흥 갯골생태공원	시흥갯골생태공원	경기도 시흥시 동	031-488-6900		

첫줄에 헤더정보. == 테이블 컬럼으로 매핑.

각각의 칼럼마다 어떤 자료형이 적절한지 생각해보며 데이터들을 살펴보자

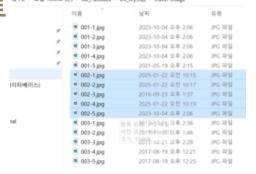
o <u>각 관광지의 이미지 파일 정보를 데이터베이스로 임포트</u>

■ travel-image 폴더에 준비

- 파일명: 관광지no-순번.jpg

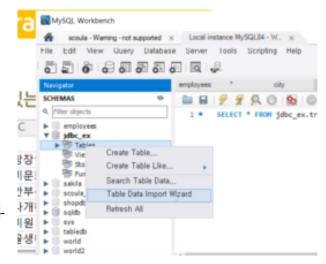
- 예: 007-4.jpg

이미지 파일명 규칙 파악



워크벤치에는 DB에 csv파일을 임포트하는 기능이 있다 원하는 DB>테이블>우클릭>tabledata importwizard 하지만 이 기능은 조건이 까다롭고 데이터가 짤릴수있다 권장하지 않는다. 다른 방법으로 ㄱㄱ

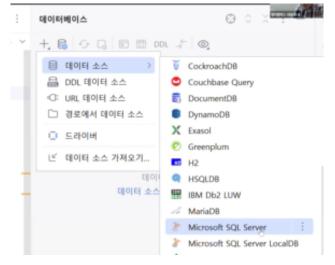
보기 사진 도구



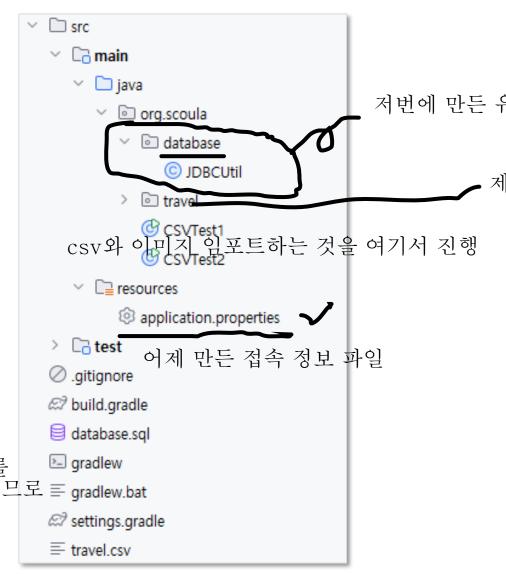
💟 프로젝트 만들기

- 프로젝트: travelapp
- o **빌드 툴: gradle** groovy. gradle DSL은

groupid :: org.scoula artifact:travelapp(프로젝트명)



프로젝트를 새로 생성했으니 데이터소스를 □ gradlew 새로 등록해야한다. 이는 프로젝트 단위이므로 ≡ gradlew.bat



저번에 만든 유틸파일 database패키지에

제작할 앱이 들어갈 패키지



진단

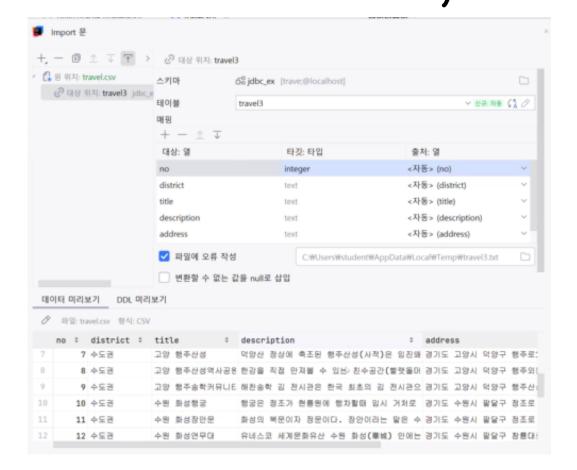
후 완료

JDBC 프로그래밍 - Travel

csv를 자바에서 처리하게 하는 라이브러리. '많은 것들중 하나 **build.gradle** dependencies { implementation 'com.opencsv:opencsv:5.5 implementation 'com.mysql:mysql-connector-j:8.3.0' 아까 워크벤치에서 csv를 임포트하는 기능있엇지만 권장하지 않았다 compileOnly 'org.projectlombok:lombok:1.18.30' annotationProcessor 'org.projectlombok:lombok:1.18.30' 인텔리제이에도 임포트하는 기능이 있다 testCompileOnly 'org.projectlombok:lombok:1.18.30' testAnnotationProcessor 'org.projectlombok:lombok:1.18.30' ☐ travel.sql → testImplementation platform('org.junit:junit-bom:5.10.0' testImplementation 'org.junit.jupiter:junit-jupiter' no idbc ex ∨ trave:@localhos sal파일 travelapp C:\\KB_ 十 새로 작성 생성하고 gradle. > 🕞 서버 3 이터소스 > idea opencsv gradle 자바로 CSV 파일을 다루는 라이브러리 Src src gitignore. @ build.gradle gradlew SQL 스크립트 ≡ gradlew.bat 도구 @ settings.gradle travel.sql 없습니다

resources::/application.properties

driver=com.mysql.cj.jdbc.Driver
url=jdbc:mysql://127.0.0.1:3306/jdbc_ex
id=scoula
password=1234



인텔리제이에서 임포트 이어서 쿼리를 날려서 제대로 됐는가 확인해보자 누락된거 타입 안맞는거 원본에서부터 없는거 확인

▽ CSV 파일 다루기[√]

o CSVReader 클래스

utf-8로 인코딩된 텍스트파일이므로 reader도 됨

- 생성자에 CSV 파일 리더 지정
- 예: new CSVReader(new FileReader("travel.csv"))
- o String[] readNext() 1줄을 읽는 메서드 '
 - ','을 기준으로 끊어서 String 배열로 리턴

데이터로써 콤마가있다면???

이럴때는 라이브러리를 사용하는게 더 안전하고 편리하다. 단순 스플릿으로는 안된다

✓ org.scoula.CSVTest1.java ✓

파일 경로가 상대경로. workingdirectory를 기준으로 찾겠다,, csv파일을 프로젝트 루트에다가 놓자 .csv")):

org.scouladatabase

travel

© CSVTest1

문제는 가독성이 떨어져요. line[2]: 뭘의미함. 첫 칼럼 값은 int여야하는데 string임 또 변환해양해 또한 데이터 처리하는데 배열로 묶여있지만 불편해

=> 캡슐화!! =>어 제배웠던 VO객체로 매핑=> 가독성 높 타입 일치, 편함'. 변환 기능들은 구현해도 되지만 라이브러리 이용해라

1,수도권,인천 개항장역사문화공간,1883년 인천 개항장 126년 세월의 흔적이 쌓인 개항기 역사와 문화의 만남의 장소이다. 이곳 2,수도권,인천 월미문화의거리,인천의 대표적인 관광지 중의 하나인 월미도에는 월미도 문화의 거리, 월미공원, 월미테마파크 등 3,수도권,인천 연안부두해양광장,서해 도서로 운항하는 여객선의 출발지이다. 싱싱한 생선과 젓갈 등을 저렴한 가격으로 구입할 4,수도권,인천 하나개해수욕장,하나개해수욕장은 샘꾸미 선착장에서 차로 10분이면 도착할 수 있다. 무의도에서 가장 큰 갯벌이 5,수도권,양평 세미원,세미원 연꽃박물관은 연꽃이라는 단일한 테마 아래 연꽃관련 생활 용품, 고서, 음식 등의 유물이 전시된 6,수도권,시흥 갯골생태공원,시흥갯골생태공원은 경기도 유일의 내만갯벌과 옛염전의 정취를 느낄 수 있는 아름다운 공원이다. 0 7,수도권,고양 행주산성,덕양산 정상에 축조된 행주산성(사적)은 임진왜란 당시 의병과 승명을 포함한 2천3백명으로 왜군 3만 0

🧿 CSV 파일 다루기

- o CsvToBeanBuilder<<u>T></u> 클래스
 - CSV 파일의 한 라인을 읽어 <u>지정한 T 타</u>입의 인스턴스로 변환해주는 <u>클래</u>스
 - csv 파일의 헤더(첫 번째 줄) 명과 같은 T 클래스의 필드로 매핑
- 클래스로 변환하기

T 는 VO객체 타입이죠

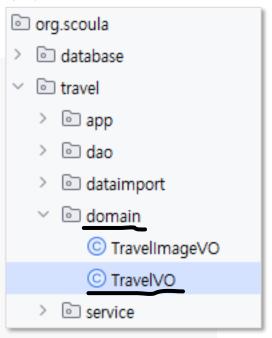
```
List<T> t = new CsvToBeanBuilder<T>(new FileReader("csv 파일 경로"))
.withType(T.class) 해당 타입으로 반환해달라
.build()
.parse();

텍스트정보를 객체정보로 변환
```

VO타입은 도메인이라는 패키지 아래에

org.scoula.travel.domain.TravelVO.java

```
@Data
@Builder
@NoArgsConstructor
@AllArgsConstructor (access=AccessLevel.PUBLIC)
public class TravelV0 {
    private Long no;
    private String district;
    private String title;
    private String description;
    private String address;
    private String phone;
}
```





O csv 파일의 첫 번째 줄에 있는 각 헤더명과 동일하게 필드명 정의

no, district, title, description, address, phone

대부분의 프레임워크들이 이러한 매커니즘을 사용한다

org.scoula

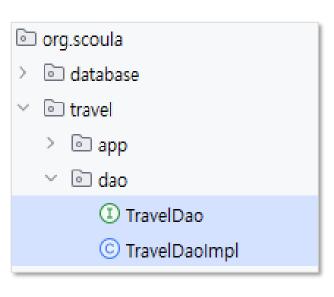
org.scoula.<u>CSVTest</u>2.java

```
 database
public class <u>CSVTest2</u> { 제네릭은 타입만 정한것임
                                                                                            travel
   public static void main(String[] args) throws Exception {
                                                                                            CSVTest1
       List<u>≺</u>TravelVO> travels = <u>new CsvToBeanBuilder<TravelVO>(</u>new FileReader("travel.csv"))
                                                                                            © CSVTest2
              .withType(TravelV0.class) =
              .build()
                                                          해당 타입의 객체를 만들기위한
                                                          정보를 제공.
              .parse();
                                                          class에는 newInstance라는 메소드가 있따. 디폴트생성자로 해당 클래스 객체를 만들어줌. new T();. <-리플렉션 기능
       travels.forEach(travel -> {
           System.out.println(travel);
                                                               람다식인데 넘겨지는 인자가
       });
                                                               사용되는 메서드 하나의 인자로 바로 넘겨질때
                                                               메소드 참조를 쓸 수도 있죠?
                                                               =>System.out::println
```

TravelVO(no=1, district=수도권, title=인천 개항장역사문화공간, description=1883년 인천 개항장 126년 세월의 흔적이 쌓인 개항기 역사와 문화의 만남의 장소이다. 0 TravelVO(no=2, district=수도권, title=인천 월미문화의거리, description=인천의 대표적인 관광지 중의 하나인 월미도에는 월미도 문화의 거리, 월미공원, 월미테마파크 TravelVO(no=3, district=수도권, title=인천 연안부두해양광장, description=서해 도서로 운항하는 여객선의 출발지이다. 심심한 생선과 젓갈 등을 저렴한 가격으로 구입할 TravelVO(no=4, district=수도권, title=인천 하나개해수욕장, description=하나개해수욕장은 샘꾸미 선착장에서 차로 10분이면 도착할 수 있다. 무의도에서 가장 큰 갯벌C TravelVO(no=5, district=수도권, title=양평 세미원, description=세미원 연꽃박물관은 연꽃이라는 단일한 테마 아래 연꽃관련 생활 용품, 고서, 음식 등의 유물이 전시된 TravelVO(no=6, district=수도권, title=시흥 갯골생태공원, description=시흥갯골생태공원은 경기도 유일의 내만갯벌과 옛염전의 정취를 느낄 수 있는 아름다운 공원이다. TravelVO(no=7, district=수도권, title=고양 행주산성, description=덕양산 정상에 축조된 행주산성(사적)은 임진왜란 당시 의병과 승명을 포함한 2천3백명으로 왜군 3만 TravelVO(no=8, district=수도권, title=고양 행주산성역사공원, description=한강을 직접 만져볼 수 있는 친수공간(빨랫돌머리) 군초소 전망대(행호정), 철책 포토존, 바

TravelDao 준비하기

먼저 14페이지부터. 먼저 테이블부터 만들어야한다.



```
package org.scoula.travel.dao;
import org.scoula.travel.domain.TravelV0;

public interface TravelDao {
    void insert(TravelV0 travel);    어제와 다르게 인터페이스에 예외 던지기가 없다
}
```

```
package org.scoula.travel.dao;
public class TravelDaoImpl implements TravelDao {
    Connection conn = JDBCUtil.getConnection();
    @Override
    public void insert(TravelVO travel) {
        String sql = "insert into tbl_travel(no, district, title, description, address, phone) values(?,?,?,?,?)";
        try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
                                                                       jdbc에서 에러가 나며은
            pstmt.setLong(1, travel.getNo());
                                                                       SQLException이 발생한다
exception의 자식이다.
반드시 예외처리를 해야하는 애다.
            pstmt.setString(2, travel.getDistrict());
            pstmt.setString(3, travel.getTitle());
            pstmt.setString(4, travel.getDescription());
                                                                       하지만 나는 반드시 예외처리를 하기 싫은데
exception과 다른 계열인 runtimeException은
예외처리가 강제화되지 않는다. 에러내용을 바꾸지 않고
            pstmt.setString(5, travel.getAddress());
            pstmt.setString(6, travel.getPhone());
                                                                       고대로 RuntimeException으로 넘겨 변환 생성한다.
            int count = pstmt.executeUpdate();
        } catch (SQLException e) {
                                                                       고럼 강제가 에러처리가 아닌 옵션 에러처리가 된다.
그러면 호출하는 쪽에서 예외처리를 생략하거나
추가적인 동작을 할 수 있다.
            throw new RuntimeException(e); //
                                                                       지금은 별 추가적인 동작이 없지만 spring에서는
유용하게 활용할수있다'
            DAO는 준비됐으니
            실제 insert를 해보자
```

☑ travel.csv 파일을 데이터베이스로 임포트 하기

o 테이블 준비 travel.sql

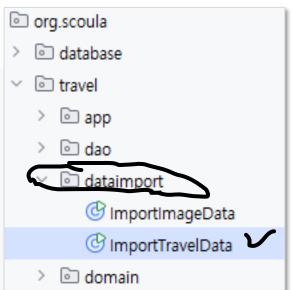
```
DROP TABLE IF EXISTS tbl_travel;
CREATE TABLE tbl_travel
                INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
    no
    district
                VARCHAR(50)
                                    NOT NULL,
    title
                VARCHAR(512)
                                    NOT NULL,
    description TEXT,
    address
                VARCHAR(512),
                VARCHAR(256)
    phone
                                                     우리는 travel.sql
```

src src ✓ □ main java database © JDBCUtil >
in travel CSVTest1 CSVTest2 ✓ □ resources application.properties > 🛅 test .gitignore @ build.gradle database.sql gradlew ≡ gradlew.bat = travel.csv

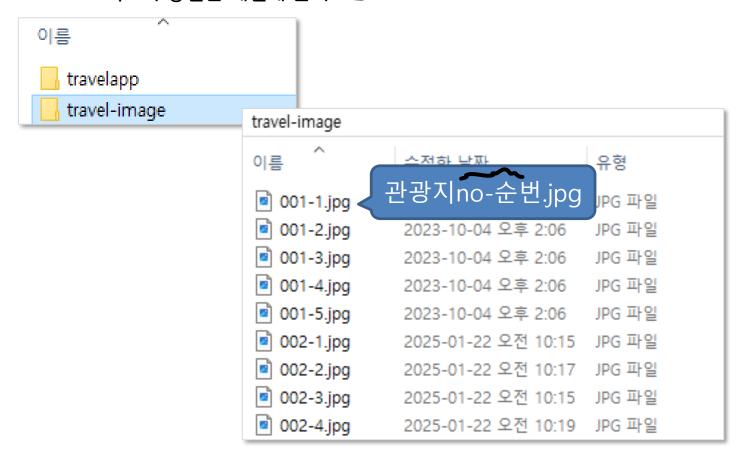
org.scoula.travel.dataimport.ImportTravelData.java

```
public class ImportTravelData {
    public static void main(String[] args) throws CsvValidationException, IOException {
     TravelDao dao = new TravelDaoImpl();
       List<TravelVO> travels = new CsvToBeanBuilder<TravelVO>(new FileReader("travel.csv"))
                .withType(TravelV0.class)
                .build()
                .parse();
        travels.forEach(travel -> {
           System.out.println(travel);

    dao.insert(travel);
        JDBCUtil.close();
```



- ☑ 각 관광지의 이미지 파일 정보를 데이터베이스로 임포트
 - o travel-image 폴더 준비
 - 프로젝트와 동일한 레벨에 준비



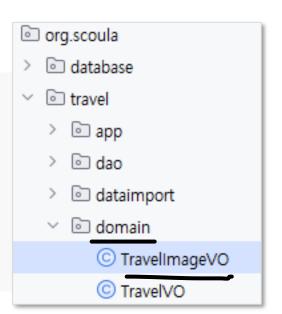
tbl_travel_image 테이블

travel정보가 삭제될때 이미지 정보도 삭제되게끔

위와 같은 추가 조건들은 아까했던 intelij gui로 임포트할때 지정할 수 없다

org.scoula.travel.dao.TravellmageVO.java

```
@Data
@Builder
@NoArgsConstructor
@AllArgsConstructor
public class TravelImageVO {
    private Long no;
    private String filename;
    private Long travelNo;
}
```



```
package org.scoula.travel.dao;
import org.scoula.travel.domain.TravelV0;
public interface TravelDao {
    void insert(TravelV0 travel);
    void insertImage(TravelImageV0 image);
}
```

```
package org.scoula.travel.dao;
public class TravelDaoImpl implements TravelDao {
    Connection conn = JDBCUtil.getConnection();
    @Override
    public void insertImage(TravelImageV0 image) {
        String sql = "insert into tbl_travel_image(filename, travel_no) values(?,?)";
        try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
            pstmt.setString(1, image.getFilename());
            pstmt.setLong(2, image.getTravelNo());
            int count = pstmt.executeUpdate();
        } catch (SQLException e) {
            throw new RuntimeException(e);
```

org.scoula.travel.dataimport.lmportImageData.java

```
public class ImportImageData {
    public static void main(String[] args) {
        TravelDao dao = new TravelDaoImpl();
                                                             travelapp
        File dir = new File("../travel-image");
                                                              travel-image
        File[] files = dir.listFiles(); 

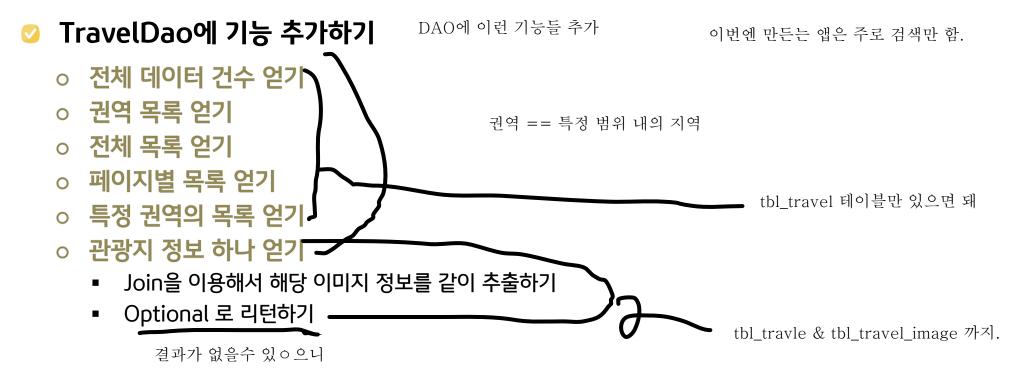
                                                              chapter0와에 날짜:
        for (File file : files) {
             String filename = file.getName()화일의 이름만. . 확장자X
             long travelNo = Long.parseLong(filename.split("-")[0]);
                                                                            // 관광지 no 얻기
                                                                          001-2.jpg => 001
            TravelImageV0 image = TravelImageV0.builder()
                      .filename(filename)
                      .travelNo(travelNo)
                                                 번외 패키지 관례
                     -.build();
                                                groupid.artifactid (여기까지 base package)
.성격별로 나뉘어짐
             System.out.println(image);
                                                .domain - vo객체처럼 DB와매핑되는애들
.dao(repository) DB와 연결을 담당하는
.service - 비즈니스로직이 있는 곳
.controller - 흐름제어. 아직 진도 안나감
             dao.insertImage(image);
                                                 .utility
        JDBCUtil.close();
                                                 .lib
                                                 작업의 성격에 따라서그루핑함
```

org.scoula

org.sc

이미지는 왜 csv처럼 같은 배포 패키지 안에 안두고 따로 같은 레벨에 다른데에 둘냐,,

덩치가 너무 커서 뺀것이다! 굳이 안에 안넣어도 돼



package org.scoula.travel.dao;

org.scoula.travel.dao.TravelDao.java

```
import org.scoula.travel.domain.TravelImageVO;
import org.scoula.travel.domain.TravelVO;
import java.util.List;
import java.util.Optional;
public interface <u>TravelDao</u> {
   void insert(TravelVO travel);
   void insertImage(TravelImageV0 image);
   int getTotalCount();
   List<String> getDistricts(); // 권역 목록 얻기
   List<TravelV0> getTravels(); // 목록 얻기 전체
   List<TravelVO> getTravels(int page);
                                      // 페이지별 목록 얻기
   List<TravelVO> getTravels(String district); // 해당 권역의 목록 얻기
   Optional<TravelV0> getTravel(Long no); // 특정 관광지 정보 얻기 상세보기
```

```
org.scoula

org.sc
```

resultset을 travelVO로 변환(매핑)해주는 메소드가 있으면 편하겠다

org.scoula.travel.dao.TravelDaoImpl.java

```
public class TravelDaoImpl implements TravelDao {
    Connection conn = JDBCUtil.getConnection();
    @Override
    public int getTotalCount() {
         String sql = "select count(*) from tbl_travel";
        try (PreparedStatement pstmt = conn.prepareStatement(sql);
    ResultSet rs = pstmt.executeQuery()) {
             rs.next();
             return rs.getInt(1);
         } catch (SQLException e) {
             throw new RuntimeException(e);
```

org.scoula

odatabase

travel
odao
TravelDao
TravelDaolmpl
odataimport
odamain
oservice

```
// 권역 목록 얻기
@Override
public List<String> getDistricts() {
    List<String> districts = new ArrayList<>();
    String sql = "SELECT DISTINCT(district) FROM tbl_travel ORDER BY district";
    try (PreparedStatement pstmt = conn.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery()) {
       while (rs.next()) {
           } catch (SQLException e) {
       throw new RuntimeException(e);
    return districts; //
```

```
private TravelVO map(ResultSet rs)_throws SQLException {
    return TravelVO.builder()
             .no(rs.getLong("no"))
             .district(rs.getString("district"))
             .title(rs.getString("title"))
             .description(rs.getString("description"))
             .address(rs.getString("address"))
             .phone(rs.getString("phone"))
             .build();
@Override
public List<TravelVO> getTravels() {
    List<TravelVO> travels = new ArrayList<>();
     String sql = "select * from tbl_travel order by district, title";
     try (PreparedStatement pstmt = conn.prepareStatement(sql);
          ResultSet rs = pstmt.executeQuery()) {
        while (rs.next()) {
            TravelV0 travel = map(rs);
            travels.add(travel);
     } catch (SQLException e) {
         throw new RuntimeException(e);
    return travels;
```

```
몇번째 페이지 얻을거냐
@Override
public List<TravelVO> getTravels(int page) {
    List<TravelVO> travels = new ArrayList<>();
    String sql = "select * from tbl_travel order by district, title limit?,?";
   try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
int count = 10; 페이지당 건수
        int start = (page - 1) * count;
        pstmt.setInt(1, start);
                                    시작번호, 사이즈
        pstmt.setInt(2, count);
        try (ResultSet rs = pstmt.executeQuery()) {
            while (rs.next()) {
                TravelVO travel = map(rs);
                travels.add(travel);
    } catch (SQLException e) {
        throw new RuntimeException(e);
    return travels;
```

```
특정 권역 목록얻기
@Override
public List<TravelVO> getTravels(String district) {
   List<TravelVO> travels = new ArrayList<TravelVO>();
   String sql = "select * from tbl_travel where district = ?";
   try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
       pstmt.setString(1, district);
                                                                    보니까 쿼리만 달라지고 있던대
       try (ResultSet rs = pstmt.executeQuery()) {
           while (rs.next()) {
               TravelVO travel = map(rs);
               travels.add(travel);
   } catch (SQLException e) {
       throw new RuntimeException(e);
   return travels;
```

- 마지막 상세보기 관광지 정보 하나 얻기
 - Join을 이용해서 해당 이미지 정보를 같이 추출하기

on과 where의 차이 on-join전에 필터링 whre-join후에 필터링

select t.*, ti.no as tino, ti.filename, ti.travel_no from tbl_travel t

> left outer join tbl_travel_image ti on t.no = ti.travel no

해당 쿼리가 기니까 일부분을 view로 만들어 사용할 수도 있다

tbl_travle(기준) 아우터조인 tbl_travel.image

사진이 없는 관광지 가 있을 수 이있으니까

where t.no = 1;

1번관광지에 대한 정보만

,]	$\dot{\Box}$	الح	י ד	1
レ	1	众	, 디	

□ no ▽ ÷	\square district $ abla$ ÷		\square description $ abla$			□ phone `	
1	1 수도권		1883년 인천 개항장 126년				
2	1 수도권	인천 개항장역사문화	1883년 인천 개항장 126년	! 세 인천광역시 중구	제물량로232	인천 중구 등	문화유산
3	1 수도권	인천 개항장역사문화	: 1883 년 인천 개항장 126 년	! 세 인천광역시 중 <mark>구</mark>	제물량로232	인천 중구 등	문화유산
4	1 수도권	인천 개항장역사문화	1883년 인천 개항장 126년	! 세 인천광역시 중구	제물량로 <mark>232</mark>	인천 중구 등	문화유산
5	1 수도권		1883년 인천 개항장 126년				

‡	□ tino 🎖 🗼 ‡	\square filename $ abla$ ÷	<pre>□ travel_no ♡</pre>	
하유산	1	001-1.jpg		1
마유산	2	001-2.jpg		1
하유산	3	001-3.jpg		1
하유산	4	001-4.jpg		1
하유산	5	001-5.jpg		1

TravelVO

자 까다로운 부분! 조인의 결과를 어떻게 지 정보에 다수의 연관 사진들에 대한 것이 연결되어있어야 하는데,,, 들은 필드멤버가 리스트를 갖게 하면 된다

List < Travellmage VO >



1				
	□ tino 🎖 🗼 ‡	\square filename $ abla$ ÷	\square travel_no \triangledown	\$
1	1	001-1.jpg		1
1	2	001-2.jpg		1
ŀ	3	001-3.jpg		1
1	4	001-4.jpg		1
1	5	001-5.jpg		1
-				

TravelVO

객체 하나

객체들의 모음인 리스트

List < Travellmage VO >

```
TravelVO(
no=1,
district=수도권,
title=인천 개항장역사문화공간,
description=1883년 인천 개항장 126년 세월의 ...,
address=인천광역시 중구 제물량로232번안길 27,
phone=인천 중구 문화유산팀 032-760-6448,
images=[
    TravelImageVO(no=1, filename=001-1.jpg, travelNo=1),
    TravelImageVO(no=3, filename=001-3.jpg, travelNo=1),
    TravelImageVO(no=4, filename=001-4.jpg, travelNo=1),
    TravelImageVO(no=5, filename=001-5.jpg, travelNo=1)
]
```

org.scoula.travel.domain.TravelVO.java

```
@Data
@Builder
@NoArgsConstructor
@AllArgsConstructor
public class TravelVO {
    private Long no;
    private String district;
    private String description;
    private String address;
    private String phone;

    private List<TravelImageVO> images;
}

1:n 관계를 나타내는
```

```
private IravelImageV0 mapImage(ResultSet rs) throws SQLException {
   return TravelImageVO.builder()
            .no(rs.getLong("tino"))
            .filename(rs.getString("filename"))
            .travelNo(rs.getLong("travel_no"))
            .build();
@Override
public Optional<TravelVO> getTravel(Long no) {
    TravelVO travel = null;
   String sql = """
               select t.*, ti.no as tino, ti.filename, ti.travel_no \ view로 등록하여 사용해도돼
               from tbl_travel t
                        left outer join tbl_travel_image ti
                                        on t.no = ti.travel_no
               where t.no = ?;
```

org.scoula.travel.dao.TravelDaoImpl.java

```
try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
   pstmt.setLong(1, no); V
   try (ResultSet rs = pstmt.executeQuery()) {
       if (rs.next()) {
           travel = map(rs);
           List<TravelImageV0> images = new ArrayList<>();
           try {
               do {
                   TravelImageV0 image = mapImage(rs);
                   images.add(image);
               } while (rs.next());
           } catch (SQLException e) {
               // 이미지가 없는 경우 발생
           travel.setImages(images);
           return Optional.of(travel); 데이터가 있는 경우
       } else {
           return Optional.empty(); 데이터가 없는 경우
} catch (SQLException e) {
   throw new RuntimeException(e);
```

번외 optional 존재할수도 잇지만 안할수도 잇는객체를 null이 될수도있는 객체를 감싸는 일종의 래퍼 클래스.

optional의

of메소드는 인자가 null일경우 예외 던짐.

empty메소드는 빈 optional객체 얻어옴

데이터 추출만 담당

🧵 다음 기능들을 실행하는 서비스 객체 정의하기

- 전체 관광지 목록 출력하기 ♡
 - no, district, title만 출력
- 특정 권역의 관광지 목록 출력하기 ∨
 - 권역목록을 출력
 - 출력할 권역을 입력 받음
 - no, district, title만 출력
- 특정 페이지의 관광지 목록 출력하기 \
 - 전체 페이지를 출력
 - 출력할 페이지 번호를 입력 받음
 - no, district, title만, 출력
- 관광지 상세 보기 `
 - 출력할 관광지 번호를 입력 받음
 - 이미지 정보를 포함한 모든 정보 출력하기

비즈니스 로직을 담당하는 객체 == 서비스 객체라고 불림. 유저레벨에서 바라보는 전체 기능들이 뭐가 있을가?

ee 전체 관광지 목록 출력하기 ee

○ 권역명, 제목 순으로 정렬

district title 더 자세한 정보는 상세보기해야함 101 강원권 강릉 경포해변 105 강원권 강릉 솔향수목원 100 강원권 강릉 연곡해변캠핑장 106 강원권 강릉 정동진 모래시계공원 103 강원권 강릉 커피거리 104 강원권 강릉 통일공원 102 강원권 강릉 허균?허난설헌 기념공원 107 강원권 동해 망상해수욕장 98 강원권 속초 아바이마을 강원권 속초 해수욕장관광지 108 강원권 정선 삼탄아트마인 95 강원권 춘천 남이섬

♥ 특정 권역의 관광지 목록 출력하기

- 1] 강원권
- 2] 경상권
- 3] 수도권
- 4] 전라권
- 5] 제주권
- 6] 충청권

선택: 5

제주권 뽑끼

- 109 제주권 제주 사려니 숲길삼나무숲
- 110 제주권 제주 서귀포 치유의숲
- 111 제주권 제주 붉은 오름 자연휴양림
- 112 제주권 제주 서귀포 천지연폭포

특정 페이지의 관광지 목록 출력하기

```
총 12 페이지
```

페이지 선택(1~12): 3

- 78 경상권 경주 보문관광단지
- 79 경상권 고령 대가야 역사테마파크
- 87 경상권 고성 당항포관광지
- 85 경상권 김해 가야테마파크
- 83 경상권 김해 낙동강 레일파크
- 84 경상권 김해 봉하마을
- 86 경상권 김해한옥체험관
- 73 경상권 대구 근대골목
- 75 경상권 대구 비슬산군립공원
- 74 경상권 대구 사문진주막촌

3페이지 목록 출력

💟 관광지 상세 보기

V관광지 No: 1

권역: 수도권

제목: 인천 개항장역사문화공간

설명: 1883년 인천 개항장 126년 세월의 흔적이 쌓인 개항기 역사와 문화의 만남의 장소이다. 이곳은 옛 일본영사관인 중구청과 일본 제1은행 인천지점, 일본 제18은행 인천지점, 일본 제58은행 인천지점 등 근대 건축을 그대로 볼 수 있는 장소이다. 이뿐아니라 일본우선주식회사나 인천의 아트플렛폼, 제물포구락부 등 역사의 흔적들을 직접 눈으로 볼 수 있고 귀로 들을 수있는 곳이다. 이곳은 인천문화 관광해설사가 배치되어 예약을 받아 해설을 하고 있다. 보다 깊은 해설을 들으며 도보관광을 즐길 수 있어 가족 단위나 여행단체의 큰 호응을 얻고 있다.

주소: 인천광역시 중구 제물량로232번안길 27

전화번호: 인천 중구 문화유산팀 032-760-6448

사진들

[1] 001-1.jpg

[2] 001-2.jpg

[3] 001-3.jpg

[4] 001-4.jpg

[5] 001-5.jpg

org.scoula.travel.service.TravelService.java

```
public interface TravelService { 서비스 클래스 인터페이스 // 전체 관광지 목록 출력하기 void printTravels(); // 특정 권역의 관광지 목록 출력하기 void printTravelsByDistrict(); // 특정 페이지의 관광지 목록 출력하기 void printTravelsByPage(); // 광광지 상세보기 void printTravel(); }
```

org.scoula

odatabase

travel
odao
odataimport
odomain
oservice
TravelServiceImpl

```
org.scoula.travel.service.TravelServiceImpl.java

론복 필수 요소에대해서만 생성자를 만들어줌. final이붙은 필드

@RequiredArgsConstructor 스프링에서 사용하니까 기억
                                                                                                      org.scoula
                                                                                                      database

    travel

 public class TravelServiceImpl implements TravelService {
     final TravelDao dao;
                                                                                                         app
                                                                                                         > 🖭 dao
     private int getNumber(String prompt) {
         System.out.print(prompt);
                                                                                                            dataimport
         Scanner sc = new Scanner(System.in);
                                                                                                           domain
         int num = Integer.parseInt(sc.nextLine());
         return num;
                                                                                                            service
                                                                                                               TravelService
    private void printDistricts(List<String> districts) {
                                                                                                               © TravelServiceImpl
         for (int i = 0; i < districts.size(); i++) {</pre>
              System.out.printf("%d] %s\n", i + 1, districts.get(i));
     'private void printTravels(List<TravelVO> travels) {
         for (TravelVO travel : travels) {
             System.out.printf("%3d %6s %s\n", travel.getNo(), travel.getDistrict(), travel.getTitle());
```

org.scoula.travel.service.TravelServiceImpl.java

```
인터페이스 구현
№0verride
public void printTravels() {
    List<TravelVO> travels = dao.getTravels();
    printTravels(travels);
@Override
public void printTravelsByDistrict() {
    List<String> districts = dao.getDistricts();
    printDistricts(districts);
    int ix = getNumber("선택: ");
String district = districts.get(ix - 1);
    List<TravelVO> travels = dao.getTravels(district);
    printTravels(travels);
                                                                   페이지당 건수
@Override
public void printTravelsByPage() {
    int totalPage = (int) Math.ceil(dao.getTotalCount() / 10.0);
    System.out.printf("총 %d 페이지\n", totalPage);
    int page = getNumber(String.format("페이지 선택(1~%d): ", totalPage));
    List<TravelVO> travels = dao.getTravels(page);
    printTravels(travels);
```

org.scoula.travel.service.TravelServiceImpl.java

```
반환된Optional객체
@Override
                                                           있으면 해당 타입 객체로. 없으면 Nosuch예외 발생 runtime예외이므로 예외처리 필수 아님
public void printTravel() {
   long no = getNumber("관광지 No: "); 🖍
   TravelVO travel = dao.getTravel(no).orElseThrow(NoSuchElementException::new); 애초에 디폴트가 nosuchele..예외라
   System.out.println("권역: " + travel.getDistrict());
                                                                            생략해도돼
   System.out.println("제목: " + travel.getTitle());
   System.out.println("설명: " + travel.getDescription());
   System.out.println("주소: " + travel.getAddress());
   System.out.println("전화번호: " + travel.getPhone());
   System.out.println("사진들");
   for (TravelImageV0 image : travel.getImages()) {
       System.out.printf(" [%3d] %s\n", image.getNo(), image.getFilename());    사진 목록 출력
```

지금까지 서비스 레이어.

다음 페이지 부터는 어플리케이션 레벨에서 사용자에게 어떻게 보여줄지.

💟 어플리케이션 만들기

이 메뉴 운영

- 선택 메뉴 실행
 - 해당 메뉴에 해당하는 서비스 기능 호출
- 메뉴와 실행할 서비스 기능 간의 관계 관리 필요 Menultem 클래스로 캡슐화 ✓

☑ 1) 전체목록

```
1) 전체목록 2) 페이지목록 3) 권역별목록 4) 상세보기 5) 종료
선택: 1 🗸
101 강원권 강릉 경포해변
105 강원권 강릉 솔향수목원
100 강원권 강릉 연곡해변캠핑장
106 강원권 강릉 정동진 모래시계공원
103 강원권 강릉 커피거리
104 강원권 강릉 통일공원
102 강원권 강릉 허균?허난설헌 기념공원
107 강원권 동해 망상해수욕장
98 강원권 속초 아바이마을
99 강원권 속초 해수욕장관광지
108 강원권 정선 삼탄아트마인
95 강원권 춘천 남이섬
```

☑ 2) 페이지목록

- 87 경상권 고성 당항포관광지
- 85 경상권 김해 가야테마파크
- 83 경상권 김해 낙동강 레일파크
- 84 경상권 김해 봉하마을
- 86 경상권 김해한옥체험관
- 73 경상권 대구 근대골목
- 75 경상권 대구 비슬산군립공원
- 74 경상권 대구 사문진주막촌

◎ 3) 권역별목록

1) 전체목록 2) 페이지목록 3) 권역별목록 4) 상세보기 5) 종료 선택: 3 🗸 1] 강원권 2] 경상권 3] 수도권 4] 전라권 5] 제주권 6] 충청권 선택: 5 🗸 109 제주권 제주 사려니 숲길삼나무숲 110 제주권 제주 서귀포 치유의숲 111 제주권 제주 붉은 오름 자연휴양림 112 제주권 제주 서귀포 천지연폭포

🗸 4) 상세보기

1) 전체목록 2) 페이지목록 3) 권역별목록 4) 상세보기 5) 종료

선택: 4 🗸

관광지 No: 1 ✔

권역: 수도권

제목: 인천 개항장역사문화공간

설명: 1883년 인천 개항장 126년 세월의 흔적이 쌓인 개항기 역사와 문화의 만남의 장소이다. 이곳은 옛 일본영사관인 중구청과 일본 제1은행 인천지점, 일본 제18은행 인천지점, 일본 제58은행 인천지점 등 근대 건축을 그대로 볼 수 있는 장소이다. 이뿐아니라 일본우선주식회사나 인천의 아트플렛폼, 제물포구락부 등 역사의 흔적들을 직접 눈으로 볼 수 있고 귀로 들을 수있는 곳이다. 이곳은 인천문화 관광해설사가 배치되어 예약을 받아 해설을 하고 있다. 보다 깊은 해설을 들으며 도보관광을 즐길 수 있어 가족 단위나 여행단체의 큰 호응을 얻고 있다.

주소: 인천광역시 중구 제물량로232번안길 27

전화번호: 인천 중구 문화유산팀 032-760-6448

사진들

[1] 001-1.jpg

[2] 001-2.jpg

[3] 001-3.jpg

[4] 001-4.jpg

[5] 001-5.jpg

org.scoula.travel.app.MenuItem.java

```
package org.scoula.travel.app;
import lombok.AllArgsConstructor;
import lombok.Data;

@Data
@AllArgsConstructor COMPOSITE패턴
public class MenuItem {
    String title;
    Runnable command;
}

runnable을 활용함. 매개타입과 리턴타입 없으므로 적절하다
```

org.scoula
colorises org.scoula
database
databas

org.scoula

JDBC 프로그래밍 - Travel

org.scoula.travel.app.TravelApp.java

```
database
package org.scoula.travel.app;
                                                                                        import org.scoula.database.JDBCUtil;
import org.scoula.travel.dao.TravelDao;
                                                                                               © Menultem
import org.scoula.travel.dao.TravelDaoImpl;

☑ TravelApp

import org.scoula.travel.service.TravelService;
import org.scoula.travel.service.TravelServiceImpl;
                                                                                          > 🕞 dao
                                                                                            dataimport
import java.util.Scanner;
                                                                                            domain
public class TravelApp {
                                                                                          > 
in service
   TravelService service;
   MenuItem[] menu; ✓
   Scanner sc = new Scanner(System.in);
                                        생성자에서 DAO 만들고 서비스에 만든 DAO 삽입
   public TravelApp() {
       TravelDao dao = new TravelDaoImpl();
       service = new TravelServiceImpl(dao); 2번째 인자는 반환과 매개변수 없는 함수이므로 runnable로 가리킬수이씀
      _menu = new MenuItem[]{
              new MenuItem("전체목록", service::printTravels),
              new MenuItem("페이지목록", service::printTravelsByPage),
              new MenuItem("권역별목록", service::printTravelsByDistrict),
              new MenuItem("상세보기", service::printTravel),
              new MenuItem("종료", this::exit),
      L};
                                아까 만든 캡슐화한 메뉴 아이템들. 이름과 실제 로직 이어주기
```

org.scoula.travel.app.TravelApp.java

```
public void exit() {
                     종료하는 것은 앱에다가 구현
   sc.close();
   JDBCUtil.close();
   System.exit(0);
public void printMenu() {
   System.out.println("=============");
   for (int i = 0; i < menu.length; i++) {
      MenuItem menuItem = menu[i];
      System.out.printf("%d) %s ", i + 1, menuItem.getTitle());
   System.out.println();
   public int getMenuIndex() {
   System.out.print("선택: ");
   String line = sc.nextLine(); \vee
   int ix = Integer.parseInt(line) - 1;
   return ix;
```

org.scoula.travel.app.TravelApp.java

```
public void run() {
    while (true) {
        printMenu();
       int ix = getMenuIndex();
        Runnable command = menu[ix].getCommand();
        command.run();
public static void main(String[] args) {
    TravelApp app = new TravelApp();
    app.run();
```

JDBC프로그래밍 특 패턴이 보인다. 많은 코드 구성이 반복된다. 서비스 로직 제외지 당연.

패턴이 보인다=>일반화 가능 =>라이브러리화 할 수 있음 =>mybatis가 그 lib!! and Hibernate.

얘네가 JDBC레이러를 다숨겨줘서 조금 편하게 작업가능.

아예 다른 어플리케이션 게시판 앱을 만든다 할때 위에 구조가 바뀔까? 아니! 종료 메뉴출력 메뉴선택 실행되는 구조 안바뀜.

앱마다의 메뉴 이름과 실제 로직 코드는 바뀌겠지.

어떤 패턴이 떠오르냐? 어떤 패턴이 코드를 더 깔끔하게 하냐?

=> Templatemethod 패턴 =>변화 없는 부분은 다 템플릿. 변화는 부분은 메뉴를 수정할때. 해당 메뉴를 만드는 파트만 추상메소드로 구성하고 부모 클래스를 정의=> 코드 재사용성 높아짐