

2025년 상반기 K-디지털 트레이닝

# JDBC 프로그래밍 - Travel

[KB] IT's Your Life



#### 🗸 프로젝트 개요

○ 관광지 정보를 담고 있는 travel.csv 파일을 데이터베이스로 임포트

	Α	В	С	D	E	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		

#### ○ 각 관광지의 이미지 파일 정보를 데이터베이스로 임포트

■ travel-image 폴더에 준비

- 파일명: 관광지no-순번.jpg

- 예: 007-4.jpg

#### 💟 다음과 같이 프로젝트를 만드세요.

○ 프로젝트: travelapp

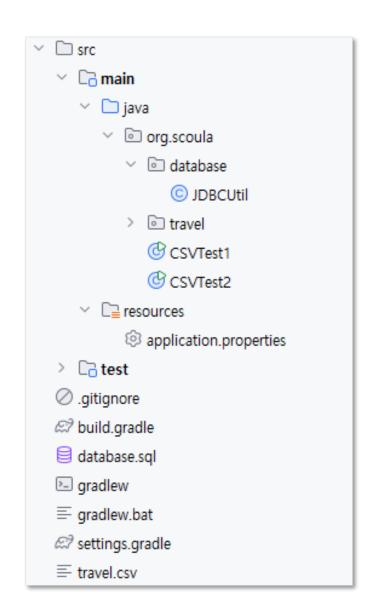
o 빌드 툴: gradle

#### ○ 기존 프로젝트에서 파일 복사

- org.scoula.database.JDBCUtil.java
- resources/application.properties 복사

#### ○ 의존성 설정

- opencsv
- mysql-connector-j
- lombok



## **build.gradle**

```
dependencies {
    implementation 'com.opencsv:opencsv:5.5'

    implementation 'com.mysql:mysql-connector-j:8.3.0'

    compileOnly 'org.projectlombok:lombok:1.18.30'
    annotationProcessor 'org.projectlombok:lombok:1.18.30'

    testCompileOnly 'org.projectlombok:lombok:1.18.30'
    testAnnotationProcessor 'org.projectlombok:lombok:1.18.30'

    testImplementation platform('org.junit:junit-bom:5.10.0')
    testImplementation 'org.junit.jupiter:junit-jupiter'
}
```

#### O opencsv

■ 자바로 CSV 파일을 다루는 라이브러리

#### ☑ travel.csv 파일을 읽어 다음과 같이 출력하는 클래스를 정의하세요.

- 정의 클래스
  - org.scoula.CSVTest1.java
- 출력

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

## org.scoula.CSVTest1.java

```
public class CSVTest1 {
  public static void main(String[] args) throws Exception {
     CSVReader csvReader = new CSVReader(new FileReader("travel.csv"));
     String[] line;
     while ((line = csvReader.readNext()) != null) {
          System.out.println(String.join(",", line));
     }
  }
}
```

```
org.scoula
database
travel
CSVTest1
CSVTest2
```

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

#### travel.csv를 읽어 TravelVO 객체로 변환하여 다음과 같이 출력하세요.

#### ○ 정의 클래스

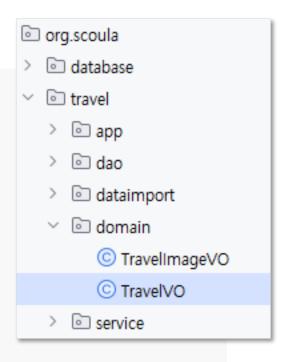
- org.scoula.travel.domain.TravelVO.java
- org.scoula.CSVTest2.java

#### ○ 출력

```
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. 싱싱한 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 갯벌(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
```

## org.scoula.travel.domain.TravelVO.java

```
@Data
@Builder
@NoArgsConstructor
@AllArgsConstructor
public class TravelVO {
   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

## org.scoula.CSVTest2.java

```
org.scoula

database

travel
CSVTest1
```

```
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 갯벌인 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
```

- 🗸 travel.csv 파일을 데이터베이스로 임포트하기위한 tbl\_travel 테이블을 정의하세요.
- ☑ 관광지 이미지 파일을 데이터베이스로 임포트하기위한 tbl\_travel\_image 테이블을 정의하세요.

### tbl\_travel 테이블

```
DROP TABLE IF EXISTS tbl_travel;

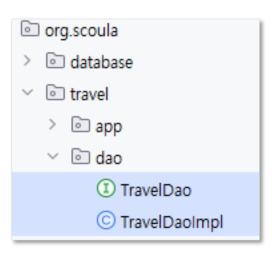
CREATE TABLE tbl_travel

(

no INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
 district VARCHAR(50) NOT NULL,
 title VARCHAR(512) NOT NULL,
 description TEXT,
 address VARCHAR(512),
 phone VARCHAR(256)
);
```

#### tbl\_travel\_image 테이블

- 🗸 TravelDao 인터페이스와 구현체 TravelDaoImpl을 정의하세요.
  - void insert(TravelVO travel);
    - TravelVO를 테이블에 저장한다
  - void insertImage(TravelImageVO image);
    - TravelImageVO를 테이블에 저장한다.



## org.scoula.travel.domain.TravelImageVO.java

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

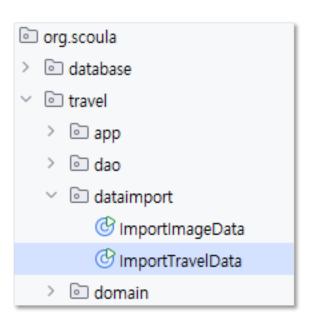
```
package org.scoula.travel.dao;
import org.scoula.travel.domain.TravelVO;
public interface TravelDao {
   void insert(TravelVO travel);
   void insertImage(TravelImageVO image);
}
```

```
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)) {
        pstmt.setLong(1, travel.getNo());
        pstmt.setString(2, travel.getDistrict());
        pstmt.setString(3, travel.getTitle());
        pstmt.setString(4, travel.getDescription());
        pstmt.setString(5, travel.getAddress());
        pstmt.setString(6, travel.getPhone());
        int count = pstmt.executeUpdate();
     } catch (SQLException e) {
        throw new RuntimeException(e);
```

```
@Override
public void insertImage(TravelImageVO 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);
   }
}
```

- 🗸 TravelDao를 이용하여 travel.csv에 있는 데이터를 tbl\_travel에 저장하세요.
  - o 정의 클래스: org.scoula.travel.dataimport.ImportTravelData.java
- ☑ TravelDao를 이용하여 관광지 이미지 정보를 tbl\_travel\_image에 저장하세요.
  - o 정의 클래스: org.scoula.travel.dataimport.ImportImageData.java



## 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(TravelVO.class)
           .build()
           .parse();
     travels.forEach(travel -> {
        System.out.println(travel);
        dao.insert(travel);
     });
     JDBCUtil.close();
```

org.scoula

odatabase

travel
odao
odataimport
of ImportImageData
ImportTravelData
odao
odao
odao

## org.scoula.travel.dao.TravellmageVO.java

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

org.scoula

database

travel

app

app

dao

dataimport

domain

TravellmageVO

TravelVO

## org.scoula.travel.dataimport.ImportImageData.java

```
public class ImportImageData {
  public static void main(String[] args) {
     TravelDao dao = new TravelDaoImpl();
     File dir = new File("../travel-image");
     File[] files = dir.listFiles();
     for (File file: files) {
        String filename = file.getName();
        long travelNo = Long.parseLong(filename.split("-")[0]); // 관광지 no 얻기
        TravellmageVO image = TravellmageVO.builder()
              .filename(filename)
              .travelNo(travelNo)
              .build();
        System.out.println(image);
        dao.insertImage(image);
     JDBCUtil.close();
```

org.scoula

odatabase

travel
odao
odao
odataimport

lmportImageData
odomain
odomain

- 🗸 TravelDao에 다음 기능을 추가하기, 구현하세요.
  - o 전체 데이터 건수 얻기
  - 권역 목록 얻기
  - 전체 목록 얻기
  - 페이지별 목록 얻기
  - 특정 권역의 목록 얻기
  - 관광지 정보 하나 얻기
    - Join을 이용해서 해당 이미지 정보를 같이 추출하기
    - Optional 로 리턴하기

## org.scoula.travel.domain.TravelVO.java

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

private List<TravelImageVO> images;
}
```

## org.scoula.travel.dao.TravelDao.java

```
package org.scoula.travel.dao;
import org.scoula.travel.domain.TravellmageVO;
import org.scoula.travel.domain.TravelVO;
import java.util.List;
import java.util.Optional;
public interface TravelDao {
void insert(TraveIVO traveI);
  void insertImage(TraveIImageVO image);
  int getTotalCount();
  List<String> getDistricts(); // 권역 목록 얻기
  List<TravelVO> getTravels();
                                 // 목록 얻기
  List<TravelVO> getTravels(int page); // 페이지별 목록 얻기
  List<TravelVO> getTravels(String district); // 해당 권역의 목록 얻기
  Optional<TravelVO> getTravel(Long no); // 특정 관광지 정보 얻기
```

org.scoula
database
travel
dao
TravelDao
TravelDaolmpl
dataimport
domain
service

## 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
odomain
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()) {
        districts.add(rs.getString("district"));
  } 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<TraveIVO> getTraveIs() {
   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()) {
         TravelVO travel = map(rs);
         travels.add(travel);
   } catch (SQLException e) {
      throw new RuntimeException(e);
   return travels;
```

```
@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;
     int start = (page - 1) * count;
     pstmt.setInt(1, start);
     pstmt.setInt(2, count);
     try (ResultSet rs = pstmt.executeQuery()) {
        while (rs.next()) {
           TraveIVO travel = map(rs);
           travels.add(travel);
   } catch (SQLException e) {
     throw new RuntimeException(e);
  return travels;
```

```
@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);
     try (ResultSet rs = pstmt.executeQuery()) {
        while (rs.next()) {
           TraveIVO travel = map(rs);
           travels.add(travel);
   } catch (SQLException e) {
     throw new RuntimeException(e);
  return travels;
```

```
private TravellmageVO mapImage(ResultSet rs) throws SQLException {
   return TravellmageVO.builder()
        .no(rs.getLong("tino"))
         .filename(rs.getString("filename"))
         .travelNo(rs.getLong("travel_no"))
         .build();
@Override
public Optional<TravelVO> getTravel(Long no) {
   TraveIVO travel = null;
   String sql = """
           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
           where t.no = ?;
         111111.
```

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