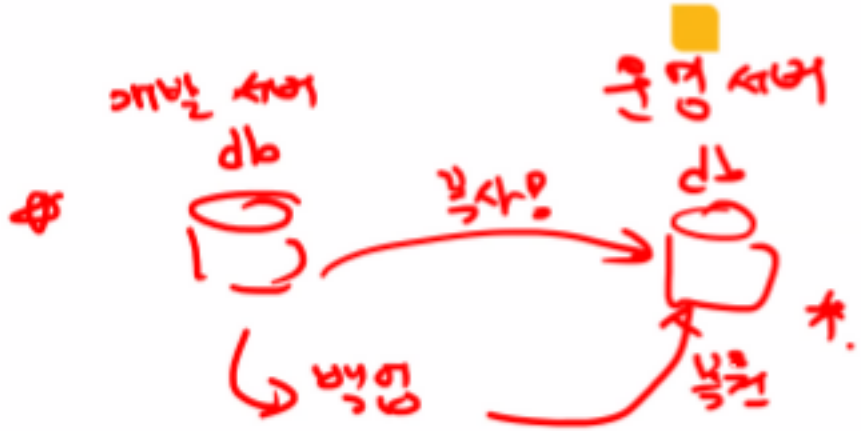
KB금융그룹 | 국민의 평생 금융파트너

개발한 데이터 베이스들을
운영서버로 옮길때
과정들.

jdbc_ex scoula_db

2025년 상반기 K-디지털 트레이닝

Travel - 백엔드



백업할 때 mysql명령 : dump

[KB] IT's Your Life

백업파일은 sql파일이다.
필요할때 실행시키면 된다.

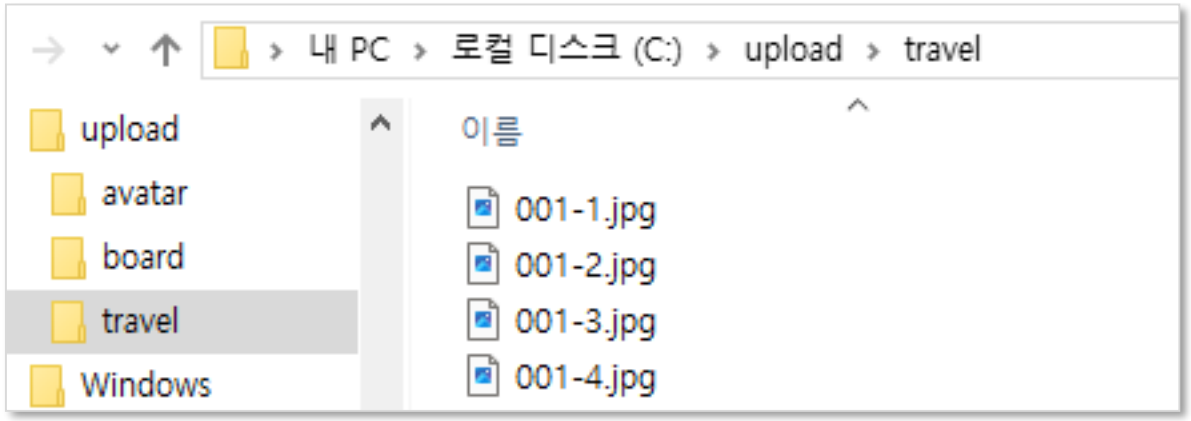
cmd명령하기 powershell말고
mysqldump -u root -p jdbc_ex tbl_travel tbl_travel_image --default-character-set utf8 > travel.sql
jdbc_ex라는 데이터베이스의 tbl_travel, tbl_travel_image 두 개의 테이블만
utf8 문자셋으로 travel.sql이라는 파일에 백업하는 작업을 수행합니다.
참고: 이 파일(travel.sql)은 나중에 mysql 명령으로 다시 불러와 복원할 수 있습니다.

user>사용자>user디렉에 travle.sql파일로 추출돼있을거다. 운영서버의 프로젝트 루트밑으로 옮기면 된다.
프로젝트 명 폴더에.
해당 파일을 필요할때 실행시키면 된다.

KB국민은행

Travel - 백엔드

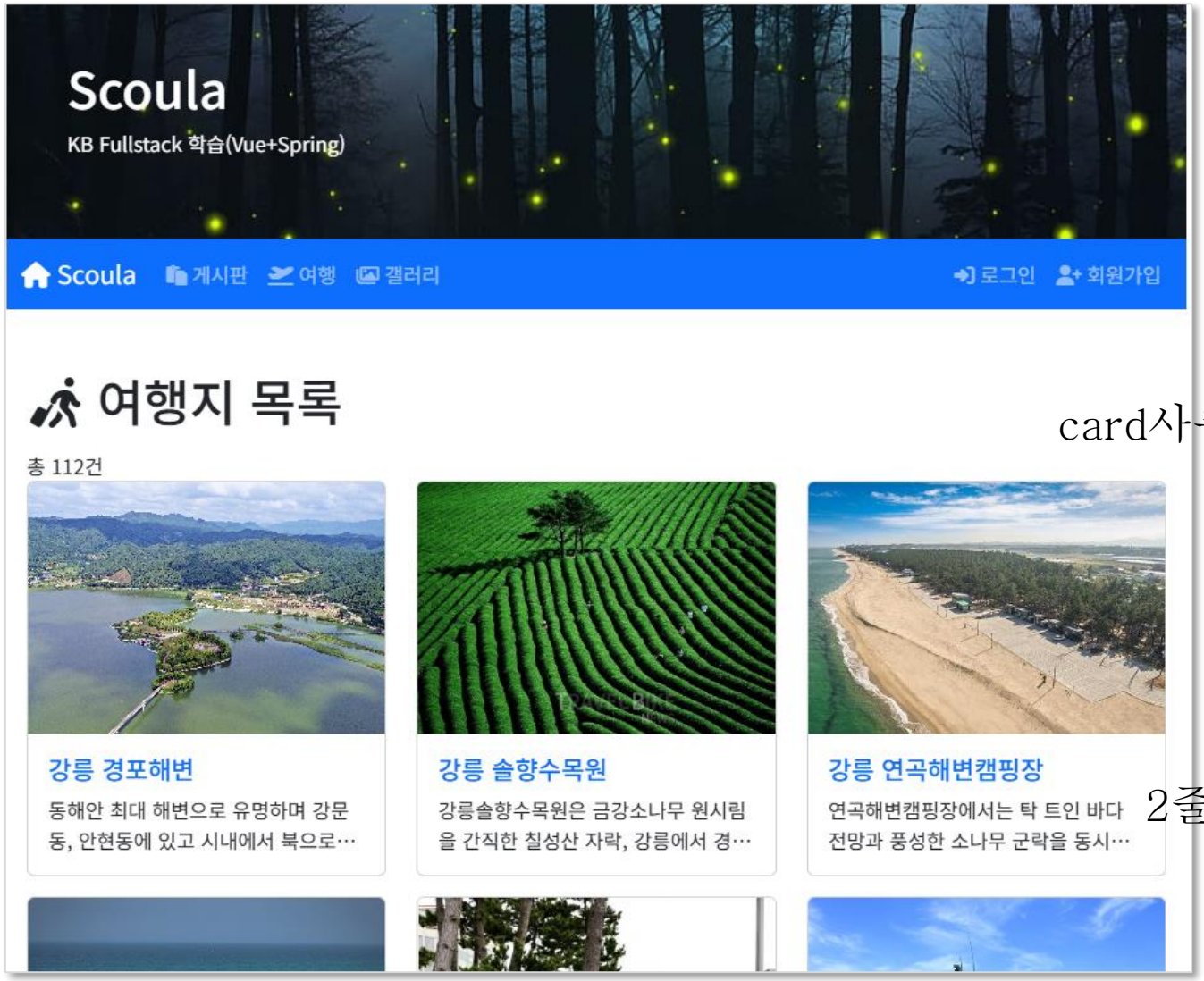
- **travel 데이터베이스 준비**
  - travelapp를 이용해서 scoula_db에 tbl_table, tbl_table_image 테이블에 데이터 임포트
- **travel 이미지 준비**
  - c:/upload/travel에 이미지 복사



이미지는 JDBC 프로그래밍 할때.

Travel - 백엔드

Travel 소개



card사용.

그리드 사용.

2줄로 축약해보자.

Travel - 백엔드

Travel 소개

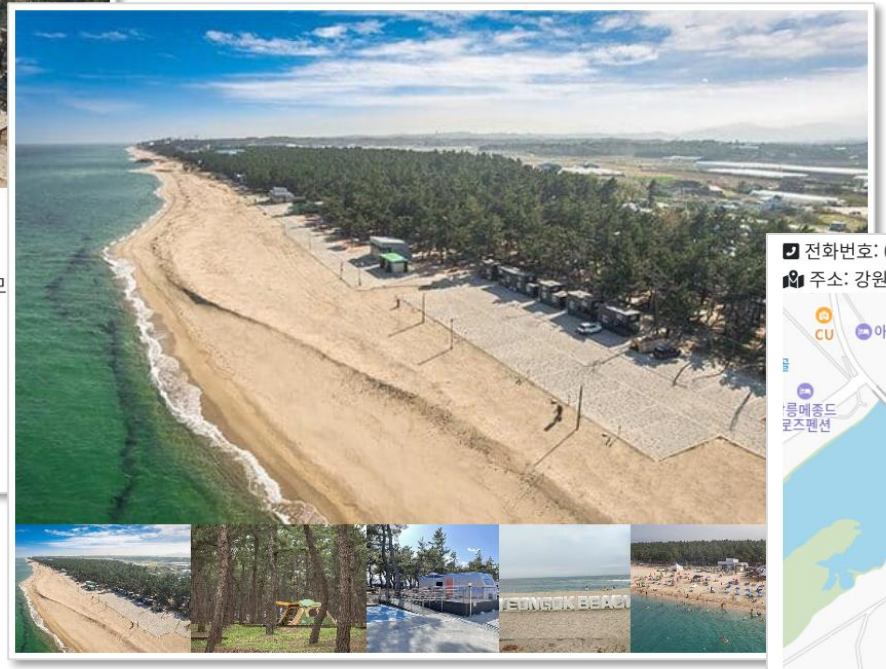
연곡해변캠핑장에서는 탁 트인 바다 전망과 풍성한 소나무 군락을 동시에 누릴 수 있다.

최근 해변에 자리해 있던 군사시설물을 말끔히 정리해 시야에는 오로지 넓은 바다와 흰 모래사장만 있다.

연곡해변캠핑장의 가장 큰 묘미는 울창한 소나무 숲속에서 캠핑을 할 수 있다는 것이다.

소나무가 내뿜는 피톤치드 덕분에 산림욕을 즐기기에도 좋다.

캠핑 자리마다 거리가 확보되어 있어 독립적인 캠핑이 가능하다는 것도 장점이다.



카카오맵 api 사용
위도와 경도만 알면 된다.

하지만 우리DB에서는
경도 위도가 없다.
이는 GEO코드를 이용해서
주소=>경도 위도를 추출할
수있다.

페이지별 목록 보기 데이터


```
amount: 12
list: Array(12)
  0:
    address: "강원도 강릉시 창해로 514 (안현동)"
    description: "동해안 최대 해변으로 유명하며 강문동, 안현동에 있고 시내에서 북으로 6km, 경포대
    district: "강원권"
    images: Array(5)
      0: {no: 501, filename: '101-1.jpg', travelNo: 101, url: '/api/travel/image/501'}
      1: {no: 502, filename: '101-2.jpg', travelNo: 101, url: '/api/travel/image/502'}
      2: {no: 503, filename: '101-3.jpg', travelNo: 101, url: '/api/travel/image/503'}
      3: {no: 504, filename: '101-4.jpg', travelNo: 101, url: '/api/travel/image/504'}
      4: {no: 505, filename: '101-5.jpg', travelNo: 101, url: '/api/travel/image/505'}
      length: 5
      [[Prototype]]: Array(0)
    no: 101
    phone: "033-640-4901"
    title: "강릉 경포해변"
    [[Prototype]]: Object
  1: {imageFiles: null, no: 105, district: '강원권', title: '강릉 솔향수목원', description: '강릉솔
  2: {imageFiles: null, no: 100, district: '강원권', title: '강릉 연곡해변캠핑장', description: '연
  3: {imageFiles: null, no: 106, district: '강원권', title: '강릉 정동진 모래시계공원', description
  4: {imageFiles: null, no: 103, district: '강원권', title: '강릉 커피거리', description: '강원도 강
  5: {imageFiles: null, no: 104, district: '강원권', title: '강릉 통일공원', description: '세계유일
  6: {imageFiles: null, no: 102, district: '강원권', title: '강릉 허균?허난설헌 기념공원', descript
  7: {imageFiles: null, no: 107, district: '강원권', title: '동해 망상해수욕장', description: '망상
  8: {imageFiles: null, no: 98, district: '강원권', title: '속초 아바이마을', description: '청호동아
  9: {imageFiles: null, no: 99, district: '강원권', title: '속초 해수욕장관광지', description: '197
  10: {imageFiles: null, no: 108, district: '강원권', title: '정선 삼탄아트마인', description: '삼탄
  11: {imageFiles: null, no: 95, district: '강원권', title: '춘천 남이섬', description: '남이섬은 [
  length: 12
  [[Prototype]]: Array(0)
pageNum: 1
totalCount: 112
totalPage: 10
```


디비에는 주소가 나와있다.

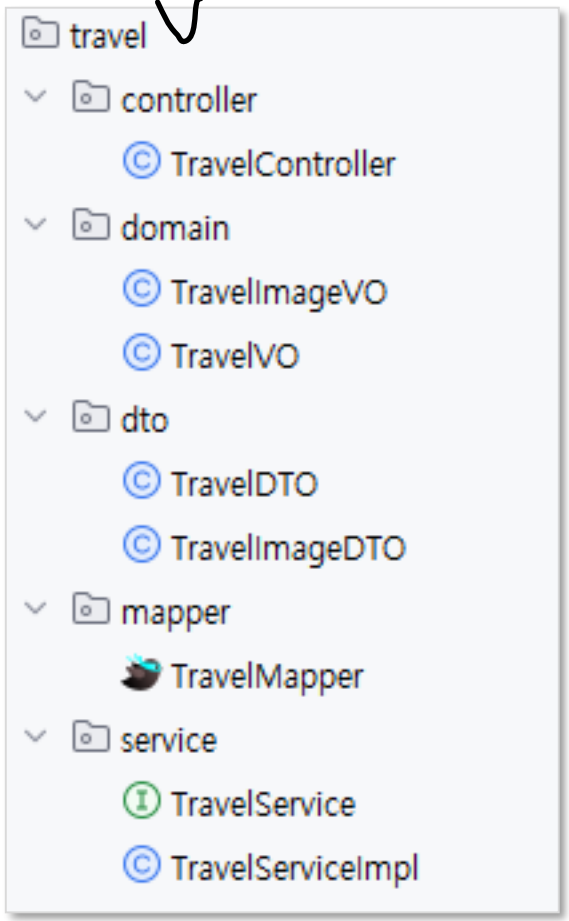
Travel - 백엔드

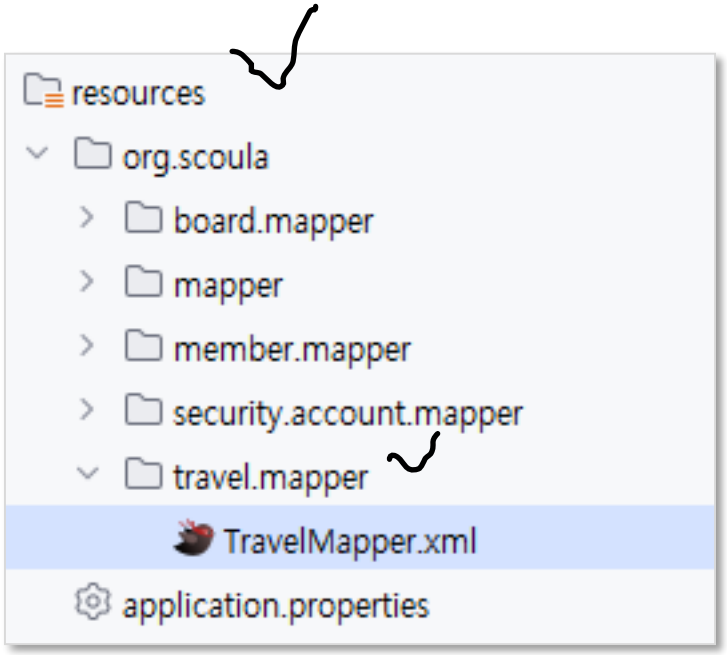
단일 항목 데이터


```
address: "강원도 강릉시 창해로 514 (안현동)"
description: "<p>동해안 최대 해변으로 유명하며 강문동, 안현동에 있고 시내에서 북으로 6km, 경포대아
district: "강원권"
▼ images: Array(5)
  ▶ 0: {no: 501, filename: '101-1.jpg', travelNo: 101, url: '/api/travel/image/501'}
  ▶ 1: {no: 502, filename: '101-2.jpg', travelNo: 101, url: '/api/travel/image/502'}
  ▶ 2: {no: 503, filename: '101-3.jpg', travelNo: 101, url: '/api/travel/image/503'}
  ▶ 3: {no: 504, filename: '101-4.jpg', travelNo: 101, url: '/api/travel/image/504'}
  ▶ 4: {no: 505, filename: '101-5.jpg', travelNo: 101, url: '/api/travel/image/505'}
    length: 5
  ▶ [[Prototype]]: Array(0)
no: 101
phone: "033-640-4901"
title: "강릉 경포해변"
```

Travel - 백엔드

프로젝트 구성

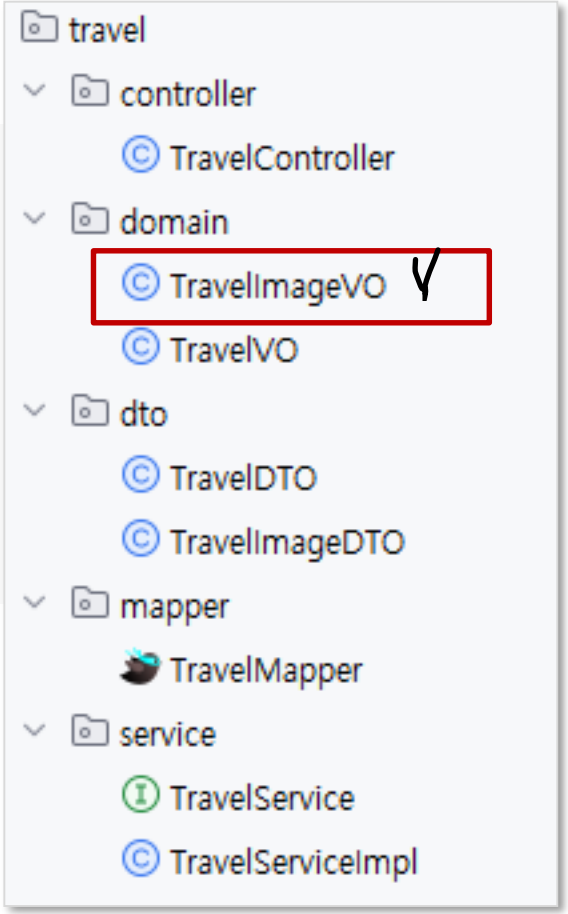




Travel - 백엔드

TravelImageVO.java

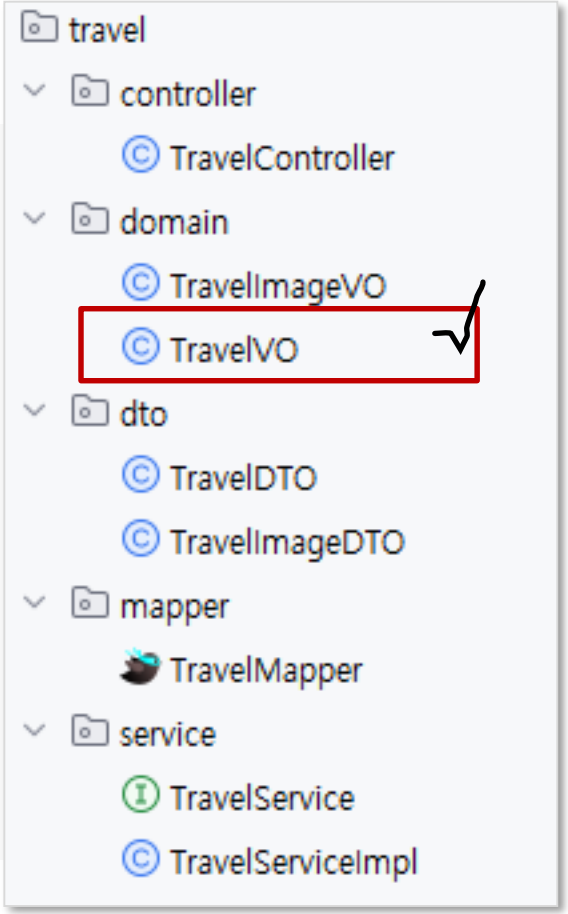
```
@Data
@Builder
@NoArgsConstructor
@AllArgsConstructor
public class TravelImageVO {
    private Long no;
    private String filename;
    private Long travelNo;
}
```



Travel - 백엔드

TravelVO.java

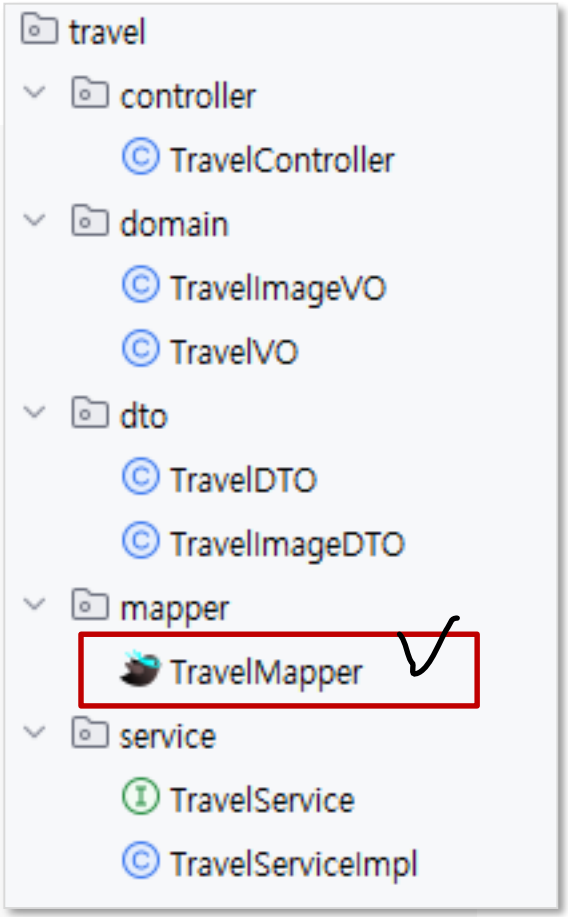
```
@Data
@Builder
@NoArgsConstructor
@AllArgsConstructor
public class TravelVO {
    private Long no;
    private String district;
    private String title;
    private String description;
    private String address;
    private String phone;

    private List<TravelImageVO> images;
}
```



Travel - 백엔드

TravelMapper.~~xml~~ java

```java
@Mapper
public interface TravelMapper {
    int getTotalCount();

    List<String> getDdistricts();    // 권역 목록 얻기

    List<TravelVO> getTravels();    // 목록 얻기

    List<TravelVO> getPage(PageRequest pageRequest);    // 페이지별 목록 얻기

    List<TravelVO> getTravelsByDistrict(String district);  // 해당 권역의 목록 얻기

    TravelVO getTravel(Long no);    // 특정 관광지 정보 얻기


    List<TravelImageVO> getImages(Long travelNo);  // 해당 관광지 이미지 목록 얻기

    TravelImageVO getImage(Long no);  // 이미지 정보 얻기
}
```



전에 만들던 거랑 코드가 거의 유사할텐데 => 코드 중복문제

Travel - 백엔드

- **이미지 정보를 같이 얻어야 하므로 Join 필요**
  - ResultMap 구성
  - 비슷한 sql 문이 사용됨 (공통 부분이 존재)
    - List<TravelVO> getTravels();
    - List<TravelVO> getTravelsByDistrict(String district);
    - TravelVO getTravel(Long no); // 특정 관광지 정보 얻기
  - getPage(PageRequest pageRequest)에서는 Join을 사용할 수 없음
    - Join을 하면 추출할 개 수를 지정 할 수 없음
    - TravelVO 목록만 추출
    - Service에서 목록에 대해 TravelImageVO 추출

MyBatis 공통 SQL문 사용하기

- <sql id="">로 공통 SQL 문장 정의

```
<sql id="travel-select">
        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
</sql>
```

- 필요한 곳에서 <include refid=""/>로 참조

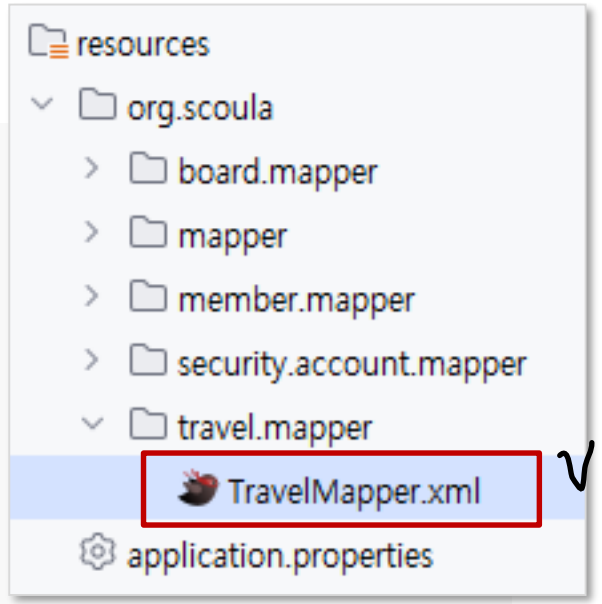
```
<select id="getTravels" resultMap="travelMap">
        <include refid="travel-select"/>
        ORDER BY district, title
</select>
```

오!

태그가
내용으로 치환되는구나!

TravelMapper.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
        PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
        "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="org.scoula.travel.mapper.TravelMapper">
    <select id="getTotalCount" resultType="java.lang.Integer">
        SELECT count(*)
        FROM tbl_travel
    </select>

    <sql id="travel-select">
        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
    </sql>
```



TravelMapper.xml

```
<select id="getDistricts" resultType="java.lang.String">
    SELECT DISTINCT(district)
    FROM tbl_travel
    ORDER BY district
</select>

<resultMap id="imagesMap" type="org.scoula.travel.domain.TravelImageVO">
    <id column="tino" property="no"/>
    <result column="filename" property="filename"/>
    <result column="travel_no" property="travelNo"/>
</resultMap>

<resultMap id="travelMap" type="org.scoula.travel.domain.TravelVO">
    <id column="no" property="no"/>
    <result column="district" property="district"/>
    <result column="title" property="title"/>
    <result column="description" property="description"/>
    <result column="address" property="address"/>
    <result column="phone" property="phone"/>
    <collection property="images" resultMap="imagesMap"/>
</resultMap>
```

TravelMapper.xml

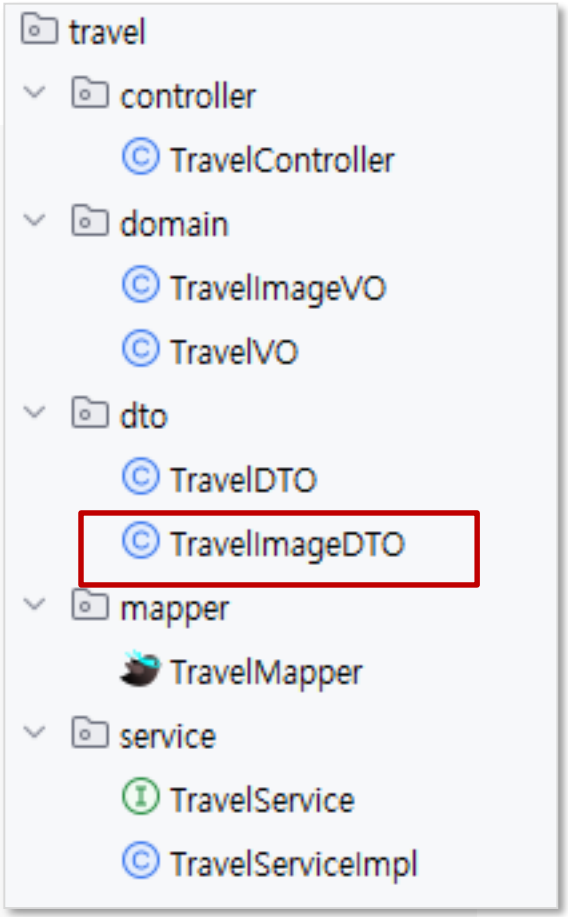
```
<select id="getTravels" resultMap="travelMap">
    <include refid="travel-select"/>
    ORDER BY district, title
</select>

<select id="getPage" resultMap="travelMap">
    SELECT *
    FROM tbl_travel
    ORDER BY district, title LIMIT #{offset}, #{amount}
</select>

<select id="getTravelsByDistrict" resultMap="travelMap">
    <include refid="travel-select"/>
    WHERE district = #{district}
    ORDER BY district
</select>

<select id="getTravel" resultMap="travelMap">
    <include refid="travel-select"/>
    WHERE t.no = #{no}
</select>
```

TravelMapper.xml

```
    <select id="getImages" resultType="org.scoula.travel.domain.TravelImageVO">
        SELECT *
        FROM tbl_travel_image
        WHERE travel_no = #{travelNo}
    </select>

    <select id="getImage" resultType="org.scoula.travel.domain.TravelImageVO">
        SELECT *
        FROM tbl_travel_image
        WHERE no = #{no}
    </select>
</mapper>
```

TravelImageDTO.java

```
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class TravelImageDTO {
    private static final String BASE = "c:/upload/travel/";
    private long no;
    private String filename;
    private long travelNo;

    public static TravelImageDTO of(TravelImageVO vo) {
        return TravelImageDTO.builder()
                .no(vo.getNo())
                .filename(vo.getFilename())
                .travelNo(vo.getTravelNo())
                .build();
    }
```

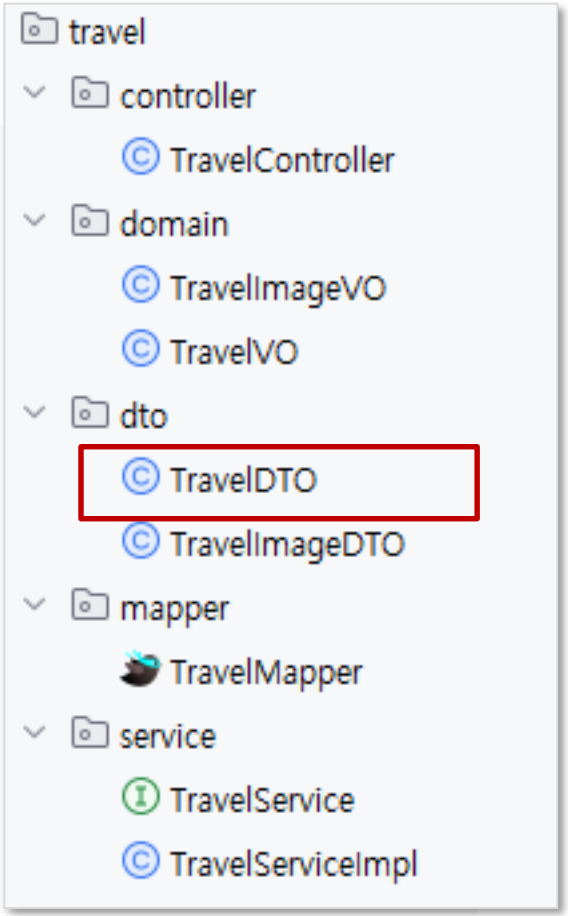


TravelImageDTO.java

```
    @JsonIgnore  // JSON 변환에서 제외
    public String getPath() {
        return BASE + filename;
    }

    // 프론트엔드에서 사용할 url 프로퍼티
    public String getUrl() {
        return "/api/travel/image/" + no;
    }
}
```

TravelDTO.java

```
@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class TravelDTO {
    List<MultipartFile> imageFiles;

    private long no;
    private String district;
    private String title;
    private String description;
    private String address;
    private String phone;
    private List<TravelImageDTO> images;
```



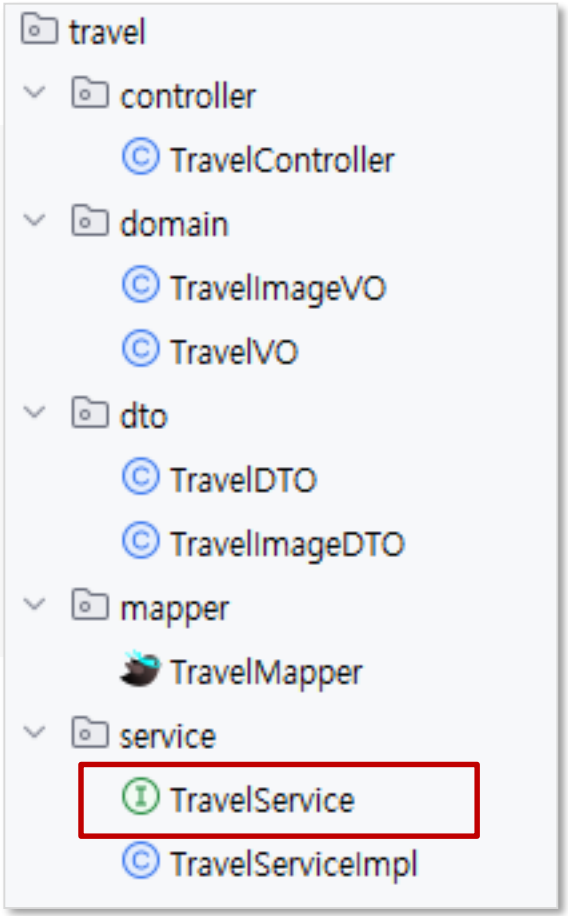
TravelDTO.java

```
    public static TravelDTO of(TravelVO vo) {
        TravelDTO travel = TravelDTO.builder()
                .no(vo.getNo())
                .district(vo.getDistrict())
                .title(vo.getTitle())
                .description(vo.getDescription())
                .address(vo.getAddress())
                .phone(vo.getPhone())
                .build();

        if (vo.getImages() != null) {
            travel.setImages(vo.getImages().stream().map(TravelImageDTO::of).toList());
        }

        return travel;
    }
}
```

Travel - 백엔드

TravelService.java

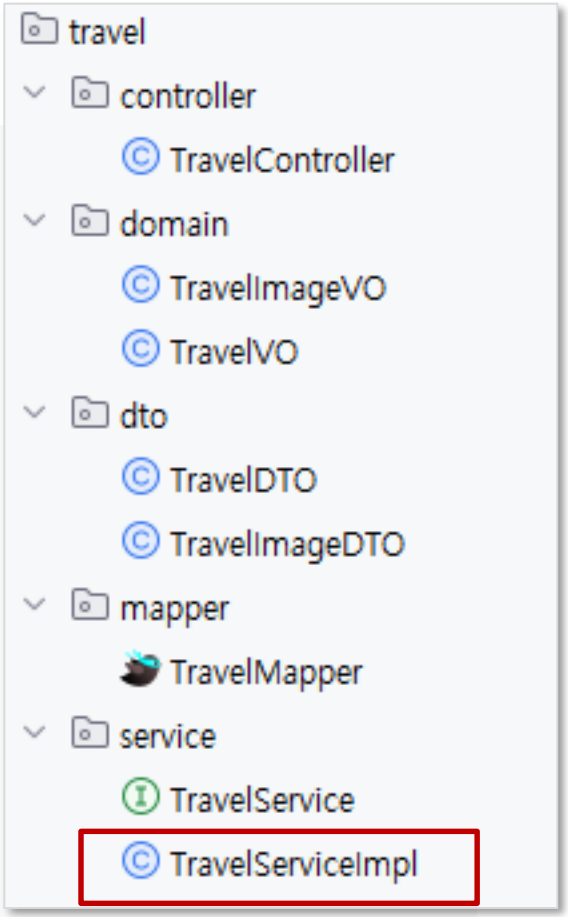
```
public interface TravelService {
    Page<TravelDTO> getPage(PageRequest pageRequest);

    List<TravelDTO> getList();

    TravelDTO get(Long no);

    TravelImageDTO getImage(Long no);

}
```



Travel - 백엔드

TravelService.java

```
@Log4j2
@Service
@RequiredArgsConstructor
public class TravelServiceImpl implements TravelService {
    final TravelMapper mapper;

    @Override
    public Page<TravelDTO> getPage(PageRequest pageRequest) {

        List<TravelDTO> travels = mapper.getPage(pageRequest)
                .stream().map(TravelDTO::of).toList();

        travels.forEach(travel -> {
            List<TravelImageDTO> images = mapper.getImages(
                    travel.getNo()).stream().map(TravelImageDTO::of).toList();
            travel.setImages(images);
        });

        int totalCount = mapper.getTotalCount();

        return Page.of(pageRequest, totalCount, travels);
    }
```



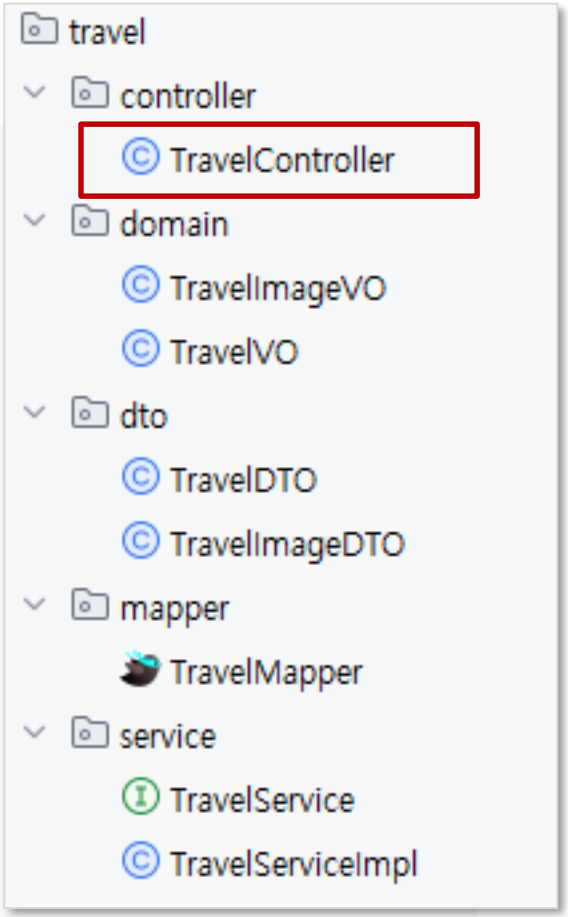
TravelService.java

```
    @Override
    public List<TravelDTO> getList() {
        List<TravelVO> travels = mapper.getTravels();
        return travels.stream().map(TravelDTO::of).toList();
    }

    @Override
    public TravelDTO get(Long no) {
        TravelVO travel = mapper.getTravel(no);
        if (travel == null) {
            throw new NoSuchElementException();
        }
        return TravelDTO.of(travel);
    }

    @Override
    public TravelImageDTO getImage(Long no) {
        TravelImageVO image = mapper.getImage(no);
        return TravelImageDTO.of(image);
    }
}
```

Travel - 백엔드

TravelController.java

```
@RestController
@RequiredArgsConstructor
@Log4j2
@RequestMapping("/api/travel")
public class TravelController {
    final TravelService service;

    @GetMapping("")
    public ResponseEntity<Page> getTravels(PageRequest pageRequest) {
        return ResponseEntity.ok(service.getPage(pageRequest));
    }

    @GetMapping("/{no}")
    public ResponseEntity<TravelDTO> getTravels(@PathVariable("no") Long no) {
        return ResponseEntity.ok(service.get(no));
    }
```



TravelController.java

```
    @GetMapping("/image/{no}")
    public void viewImage(@PathVariable Long no, HttpServletResponse response) {
        TravelImageDTO image = service.getImage(no);
        File file = new File(image.getPath());
        UploadFiles.downloadImage(response, file);
    }
}
```