

명세_GhiVideo_데이터폴더_입력명세

데이터 폴더 입력 명세 — 스테이션기반 플레이어

웹플레이어가 **동영상이 있는 폴더**를 전달받아 재생할 때, 폴더 안에 어떤 파일이 어떤 형식으로 있어야 하는지를 정의한다. 근거: `client/src/utils/geoData.ts` (폴더 파서) 기준. 실데이터(v3.0, 대전조차장-회덕)로 검증.

1. 동작 개요

1. 사용자가 브라우저에서 **폴더를 선택**하면(`<input type="file" webkitdirectory>`), 폴더 내 전체 파일 목록이 플레이어에 전달된다.
2. 플레이어는 파일명 규칙으로 각 파일을 **자동 식별·파싱**한다(별도 설정 불필요).
3. 기준 이름 **base** = 영상 파일명에서 확장자를 뺀 것 (예: `상행)대전조차장-회덕.MP4` → base는 `상행)대전조차장-회덕`). 나머지 파일은 base 일치를 우선 매칭하고, 없으면 폴백 규칙으로 찾는다.

2. 폴더 구성 요약

#	파일	필수	역할
1	<code><base>.MP4</code>	필수	드론 촬영 항공영상 (mp4/webm)
2	<code><base>.csv</code>	필수	드론 비행로그 — 프레임별 위치·자세 각·초점거리
3	<code>01)측점.csv</code> 또는 <code>측점.csv</code>	필수	측점 목록(이정·좌표·실측 표고) → 중심 선·스테이션바·POI 높이 보완의 근간
4	<code><base>.kmz</code> (또는 <code>.kml</code>)	필수(정책)	POI·구조물의 유일한 소스 (지장물·교 량·터널·출입문·구교·철도역)
5	<code><base>.route.json</code> 또는 <code>route.json</code>	선택	노선 기본 정보 + 구조물 위치 보정
6	<code><base>_poi_overrides.json</code>	선택	POI 드래그 보정 저장값 (역투영 결과)

#	파일	필수	역할
—	.srt, .qgz, .qgs 등 그 외 파일	무시	플레이어가 읽지 않음 (작업용 부산물)

예시 폴더 (실데이터 v3.0):

상행)대전조차장-회덕 데이터(v3.0)/			
├	상행)대전조차장-회덕.MP4	← 영상 (base 기준)	
├	상행)대전조차장-회덕.csv	← 드론 비행로그	
├	01)측점.csv	← 측점 (루트 또는 building/ 하위 모두 허용)	
├	상행)대전조차장-회덕.kmz	← POI·구조물 원본	
├	상행)대전조차장-회덕.route.json	← 노선 정보 (선택)	
├	상행)대전조차장-회덕.srt	← (무시됨)	
└	상행)대전조차장-회덕.qgz	← (무시됨, QGIS 작업파일)	

3. 파일별 상세 명세

3.1 영상 — <base>.MP4

- 확장자: .mp4 또는 .webm (대소문자 무관). building/ 하위가 아닌 루트의 첫 번째 영상 파일을 사용한다.
- 코덱: 브라우저 직접 재생이므로 H.264 권장.
- 용량: 2GB 이상 가능 (URL.createObjectURL() 참조 방식 — 메모리에 통째로 올리지 않음).
- 이 파일명이 폴더 전체의 base 를 결정한다.

3.2 드론 비행로그 — <base>.csv

드론이 촬영과 동시에 기록한 프레임별 데이터. 헤더 이름으로 컬럼을 찾으며(순서 무관), 헤더가 깨졌을 때만 위치 인덱스로 폴백한다.

컬럼(헤더명)	의미	단위/규약
frame_cnt	프레임 번호 (0부터)	정수
latitude / longitude	드론 위치 위도/경도	도(WGS84)
altitude	드론 고도	m (타원체고)
yaw	기수 방위각	도 — 북=0, 시계방향
pitch	카메라 틸트	도 — 음수=아래

컬럼(헤더명)	의미	단위/규약
roll	롤	도
focal_len	초점거리	mm (35mm 환산)

- 추가 컬럼(rel_alt , 5186x , 5186y , X좌표 , Y좌표 등)이 있어도 무시되고 오류 없음.
- latitude 가 숫자가 아닌 행은 건너뛴.
- 파일 선택 규칙: 루트의 .csv 중 ① <base>.csv 정확 일치 우선, ② 없으면 파일명에 POI 가 없는 첫 .csv (영상명 비의존 폴백).
- fps 자동 산출: 최대 frame_cnt ÷ 영상 길이(초) → 표준 fps(24/25/29.97/30/50/60...)에 스넵. 따라서 비행로그의 프레임 수는 영상의 실제 프레임 수와 일치해야 한다.

3.3 측정 — 01)측점.csv (또는 측정.csv)

철도 노선 측량 성과. 시간↔측점 대응표(스테이션바)와 선로 중심선, POI 높이 보완의 근간 데이터.

- 파일 찾기 규칙 (3단 폴백): ① building/ 하위에서 파일명에 01)측점 포함 → ② 루트 포함 아무 위치에서 01)측점 포함 → ③ 파일명에 측정 포함.
- 컬럼 (헤더 이름 기준):

컬럼	필수	의미
측점	필수	이정 표기 (예 157K900) — 스테이션 정렬·표시 기준
lat / lon	필수	측점 좌표 (도, WGS84) — 숫자 아니면 해당 행 제외
Z좌표_한국	권장	실측 정표고(EL, m) — 있으면 최우선 사용
Z좌표	폴백	Z좌표_한국 이 없을 때 사용
비고	선택	방향전환점 (상행->하행, 02:05) 패턴이 있으면 방향전환점으로 추출

- 중심선 자동 생성: v2.0 이후 center.csv 가 없으므로, 측점을 일정 순으로 이어 만든 폴리라인이 선로 중심선이 된다. → 측정 정확도가 곧 측정 투영·POI 높이 보완의 정확도.

△ 주의 (실데이터에서 발견): 상행 v3.0의 01)측점.csv 는 실측 표고가 z 컬럼에 있고 Z좌표 컬럼은 0.0 인데, 파서는 Z좌표_한국 → Z좌표 순서로만 읽으므로 표고가 0으로 로드 된다. 하행 파일처럼 표고 컬럼명을 Z좌표_한국 으로 통일하거나, 파서에 z 컬럼 폴백 추가가 필요하다.

3.4 POI-구조물 — <base>.kmz (또는 .kml)

POI-구조물의 유일한 소스다 (building/ 하위 02)~06) CSV는 더 이상 읽지 않음). 루트의 bare .kml 이 있으면 우선, 없으면 .kmz (zip)를 해제해 doc.kml 을 읽는다.

• KML 폴더 구조 → 분류 매핑:

KML 폴더명 (포함 키워드)	분류	이름 필드 (description 표)
03) 교량	구조물 bridge	구분 (+ 연장(m), 시설종별)
04) 터널	구조물 tunnel	구분 (+ 연장(m), 시설종별)
06) 구교	구조물 bridge(구교)	시설물명 (+ 연장(m))
05) 출입문번호	POI (출입문)	출입문번호
02) 지장물 등 그 외	POI (지장물 등)	구분 → title → placemark name 순
— source=KAKAO_RAIL 또는 category_clean=철도역/역사	POI + 역사 구조물 (스테이션바에도 표출)	위와 동일

- **Placemark 좌표:** <coordinates> 의 lon,lat[,alt] 우선. 없으면 description 속성 lat / lon. 한국 범위 검증(위도 33~39, 경도 124~132) 밖이면 버림.
- **속성(description):** HTML <table> 의 (키, 값) 행으로 기재 — 구분, 연장(m), 시설종별 (1종/2종/3종/기타), 시설물명, 출입문번호, source, category_clean, title, 표고 (z / Z좌표 / 절대고도) 등. 구글어스 재저장본의 표 변형(<tbody>, 엔티티 등)도 허용.
- **표고:** POI 속성에 z / 절대고도 가 있으면 사용, 없으면 0 → 재생 시 **최근접 측점 실측 표고** 로 보완된다.
- **누락 시:** 경고 로그 후 POI-구조물 없이 동작(측점·드론·영상은 정상). CSV 풀백은 없다.

3.5 노선 기본 정보 — <base>.route.json (선택)

루트의 <base>.route.json 우선, 없으면 route.json. UTF-8 JSON.

```
{
  "routeInfo": {
    "name": "대전조차장-회덕", // 노선명 (표시용)
    "direction": "상행", // 방향 라벨
    "lengthKm": 4.25, // 구간 연장 (km)
    "durationSec": 568, // 영상 길이 (초)
```

```

    "startStationName": "대전",          // 시점역
    "endStationName": "신탄진",          // 종점역
    "stationTolerance": 20,              // 측점 인정 반경(m) - 기본 40
    "endStationGapMeters": 0             // 종점 미도착 거리(m) - 0이면 도착
  },
  "structures": [                       // (선택) 구조물 보정 - name 매칭으로
    override
    {
      "name": "회덕천교",                // 괄호 이하 제거 후 부분일치 매칭
      "station": "158k400",              // 최우선: 이 측점값 위치에 배치 (숫자 m 또는
        "158k400")
      "offset": -10,                    // 진행방향 ±N m 미세조정
      "startMileage": 158400,           // 매칭 실패 시 폴백 지정
      "endMileage": 158500,
      "lat": 36.4031, "lon": 127.4216   // 좌표 직접 지정도 가능
    }
  ]
}

```

- `structures` 항목은 KMZ 유래 구조물에 **명시된 필드만 덮어쓰고**, KMZ에 없는 항목은 새 구조물로 추가된다.

3.6 POI 위치 보정 — `<base>_poi_overrides.json` (선택)

영상에서 POI 라벨을 **드래그(역투영)** 해 보정한 좌표의 저장/배포 파일. 폴더에 있으면 로드 시 자동 적용된다.

```

{
  "baseName": "상행)대전조차장-회덕",
  "overrides": {
    "회덕터널": { "lat": 36.40312, "lon": 127.42169, "z": 66.1 }
  }
}

```

- `overrides` 없이 평면 맵({ "회덕터널": {...} })도 허용. `z` 는 정표고(m). POI `title` 정확 일치로 매칭.

4. 공통 규칙 (인코딩·파싱)

항목	규칙
CSV 인코딩	자동 감지 — 앞 3바이트가 UTF-8 BOM(EF BB BF)이면 UTF-8, 아니면 EUC-KR. 파일 별 하드코딩 없음
CSV 컬럼	헤더 이름으로 탐색 (순서 무관, 여분 컬럼 무시). 헤더 훼손 시 위치 인덱스 폴백
CSV 형식	큰따옴표 필드 지원, 빈 줄 무시, <code>\r\n</code> / <code>\n</code> 모두 허용
KML	UTF-8. <code>.kml</code> > <code>.kmz</code> (내부 <code>doc.kml</code>) 우선순위
JSON	UTF-8. 파싱 실패 시 조용히 무시(선택 파일이므로)
폴더 구조	<code>building/</code> 하위는 측정 CSV 탐색 외에는 사용하지 않음. 하위 폴더 깊이는 1단계(<code>building/</code>)만 인식

5. 파일 누락 시 동작

누락 파일	동작
영상	재생 불가 (폴더 인식 실패)
드론 비행로그	측점 투영·POI 오버레이·fps 산출 불가 → 사실상 일반 플레이어로 전락
측점 CSV	중심선·스테이션바·POI 높이 보완 불가 (시간축 재생만 가능)
KMZ/KML	경고 로그 후 POI·구조물 미표시 (측점·드론·영상은 정상 동작)
route.json	기본값으로 동작 (노선명 등 미표시, <code>stationTolerance=40</code>)
poi_overrides.json	보정 없이 원본 좌표 사용

부록 — 구현 근거 (소스 위치)

항목	위치
폴더 통합 로더	<code>client/src/utils/geoData.ts</code> — <code>loadFolderGeoData()</code>
영상/base 식별	같은 파일 — <code>findVideoFile()</code> , <code>deriveBaseName()</code>
드론 CSV 파서	같은 파일 — <code>parseDroneFrames()</code>

항목	위치
측점 CSV 파서 (+방향 전환점)	같은 파일 — <code>parseStations()</code>
KMZ/KML 파서	같은 파일 — <code>parseKmlz()</code>
route.json / 구조물 병합	같은 파일 — <code>parseRouteMeta()</code> , <code>mergeStructures()</code>
POI 보정 로드/적용	같은 파일 — <code>parsePoi0verrides()</code> , <code>applyPoi0verrides()</code>
인코딩 자동 감지	같은 파일 — <code>decodeBytes()</code>
중심선 생성	같은 파일 — <code>buildCenterlineFromStations()</code>
(참고) 서버 검색 API 용 데이터	<code>server/src/routes/geo.ts</code> — 환경변수 <code>GEO_DATA_DIR</code> 폴더에 동일 형상 필요